



# MEMOIRE

Présenté par

**ARDOUNE Karima**

Pour l'obtention de diplôme de

**MASTER**

**Filière : Informatique**

**Spécialité : Systèmes Informatiques Intelligents**

**Thème**

**Un approche de Deep learning pour la prédiction des liens dans  
les Bases de Connaissances**

Soutenu le : 22/06 /2023

Devant le Jury composé de :

Qualité	Nom et Prénom	Grade	Université
Président	Mr. BENTRAD Saci	Docteur	Chadli Bendjedid El-Tarf
Examineur	Mme. BOUGARN Imen	Docteur	Chadli Bendjedid El-Tarf

Année Universitaire : 2022/2023

# Remerciements

---

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

*En premier lieu, nous tenons à remercier notre DIEU,*

*Je tiens à remercier mon encadreur Dr. DJEDAI Ala enseignante à Université*

*de Chadli Bendjdid - El Tarf, pour tout le soutien, l'orientation pour accomplir ce travail.*

*J'adresse mes remerciements les plus sincères aux membres du jury qui ont bien voulu examiné ce  
modeste travail.*

*Je tiens à exprimer ma profonde gratitude envers mon mari et mes enfants pour leur soutien tout  
au long de mes études.*

*je tiens à adresser mes remerciement les plus chaleureux vont à l'égard de  
Mon père, Ma mère et mes frères qui m'ont apporté leurs soutiens moral.*

*En vous souhaitant une agréable lecture,*

*Karima ARDJOUNE.*

# Dédicace

---

*Je dédie ce travail A mes parents qui ont partagés avec moi tous*

*Les moments d'émotion lors de la réalisation de ce travail,*

*Mon Marie Farid Qui m'a apporté tout le soutien et m'a soutenu malgré les  
difficultés et les obstacles que j'ai rencontrés.*

*Mes enfants la prunelle de mes yeux Mohamed, A Rahim, Zineb, Amai Malgré leur  
jeune âge, ils m'ont beaucoup aidé et soutenu.*

*A mes frères pour leurs encouragements durant tout mon parcours.*

*A mes proches et particulièrement Hedba, Amel pour leur tendresse, leur Soutien  
moral.*

*. A tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de succès.*

*je vous remercie.*

# Table des matières

---

Remerciements .....	2
Dédicace .....	3
Table des matières .....	4
Liste des figures.....	6
Résumé.....	7
Liste des tableaux .....	11
Liste des acronymes .....	12
Introduction Générale.....	13
1. Contexte du projet et problématique.....	13
2. Motivations.....	13
3. Objectifs .....	14
4. Contenu du mémoire .....	14
Chapitre 1 : Généralités sur l'apprentissage Profond .....	15
1. Introduction .....	15
2. Définitions de l'apprentissage Profond .....	16
3. Types d'apprentissage profond.....	16
4. Différences entre Machine Learning et l'apprentissage profond.....	18
5. Les différents algorithmes de l'apprentissage profond.....	19
5.1. Réseaux neuronaux convolutifs (CNN).....	19
5.2. Réseaux neuronaux récurrents (RNN).....	21
5.3. Le réseau de neurones LSTM.....	22
5.4. Réseaux adversariaux génératifs (GAN) .....	23
6. Les avantages et les inconvénients de l'apprentissage profond.....	24
7. Conclusion.....	25
<b>Chapitre 2 : Les Graphes de Connaissance.....</b>	<b>26</b>
1. Introduction .....	26

2. Définitions .....	26
3. Modélisation des graphes de connaissances .....	27
4. Architecture d'un graphe de connaissance .....	28
5. Prédiction des liens dans les graphes de connaissances .....	29
6. Modèles utilisées pour les graphes de connaissances.....	30
<b>7. Conclusion</b> .....	<b>33</b>
Chapitre 3 : Expérimentation.....	34
<b>1. Introduction</b> .....	<b>34</b>
<b>2. Approche Proposée</b> .....	<b>34</b>
A- Bases des données végétal.....	34
B- Bases des données animal .....	35
<b>3. Architecture générale du système</b> .....	<b>36</b>
<b>4. Implémentation</b> .....	<b>38</b>
<b>A- Environnement de développement</b> .....	<b>39</b>
<b>B- Langage de programmation et bibliothèques</b> .....	<b>40</b>
<b>C- Prétraitement des données</b> .....	<b>41</b>
• <b>Conversion en triplets</b> .....	<b>41</b>
<b>D- Phase d'apprentissage</b> .....	<b>47</b>
<b>5. Résultats et Discussion</b> .....	<b>45</b>
<b>Conclusion et Perspectives</b> .....	<b>51</b>
<b>Références</b> .....	<b>52</b>

# Liste des figures

---

Figure 1. Les sous-branches de l'intelligence artificiel. [3] .....	16
Figure 2. Réseaux neuronaux convolutifs [3].....	19
Figure 3. Réseaux neuronaux récurrents (RNN) [11].....	21
Figure 4. Architecture LSTM (Long Short-Term Memory [11] .....	22
Figure 5. Réseaux adversariaux génératifs [3] .....	23
Figure 6. Architecture d'un graphe de connaissances. [17] .....	28
Figure 7. Architecture CompGCN [20] .....	30
Figure 8. Architecture R-GCN [21].....	32
Figure 9. Bases des données des filières végétales (xlsx).....	35
Figure 10. Bases des données des filières animales (xlsx) .....	35
Figure 11. Architecture générale du système .....	36
Figure 12. Google Drive.....	39
Figure 13. Google Colab .....	40
Figure 14. Le code d'identification de la base de données végétal .....	42
Figure 15. dataset sous forme node- relation-edge .....	44
Figure 16. Transformer dataset sous forme text .....	45
Figure 17. affichage dataset animal .....	46
Figure 18. dataset animal sous forme node- relation-edge .....	46
Figure 19. Commande d'installations.....	48
Figure 20. Résultat de l'apprentissage .....	51

يعد الذكاء الاصطناعي من أهم المجالات الرائدة في عصر العولمة، حيث يستخدم على نطاق واسع في الحياة اليومية في مجالات متنوعة مثل الفلاحة والصناعة والتحول الرقمي والأنظمة الذكية والطب والبيولوجيا. يعتبر التعلم العميق فرعًا مهمًا من فروع الذكاء الاصطناعي، حيث حقق تقدمًا نوعيًا في مجال الذكاء الاصطناعي وأصبح المجال الأكثر أهمية في عالم الحوسبة. يعتمد التعلم العميق على شبكات عصبية معقدة لتمثيل البيانات واكتساب المعرفة، ويحاكي بشكل مشابه طريقة تفكير البشر وتعلمهم. في هذا البحث، نركز على مجال الفلاحة نظرًا للدور الفعال الذي يمكن أن يلعبه الذكاء الاصطناعي في تقدم هذا القطاع. نقدم نهجًا جديدًا يستند إلى شبكات تلافيفية لرسوم بيانية متعددة العلاقات، وتم تدريب هذا النموذج على قاعدين بيانات، الأول يتضمن بيانات الفلاحين والمحاصيل الزراعية مثل القمح والذرة والشعير، والثاني يتضمن بيانات جودة الماشية مثل الأبقار والأغنام والماعز. قمنا

بمعالجة وتعديل هذه البيانات وتحويلها إلى صيغة  
ثلاثية الأطراف، ثم قمنا بتقسيمها إلى مجموعات  
للاختبار والتصحيح والتدريب. نتائج التحليل كانت  
واحدة ومشجعة، ونأمل في استخدام وتطوير هذا  
النموذج في قطاع الفلاحة في الجزائر.

# Résumé

---

L'intelligence artificielle est l'un des domaines les plus importants et en plein essor dans l'ère de la mondialisation. Elle est largement utilisée dans la vie quotidienne dans divers domaines tels que l'agriculture, l'industrie, la transformation numérique, les systèmes intelligents, la médecine et la biologie. L'apprentissage profond est l'une des branches les plus importantes de l'intelligence artificielle, ayant connu une avancée significative dans ce domaine et devenant le domaine le plus crucial de l'informatique. Il repose sur des réseaux neuronaux profonds pour représenter les données et acquérir des connaissances, imitant ainsi la façon dont les humains pensent et apprennent.

Dans cette étude, nous nous concentrons sur le domaine de l'agriculture en raison du rôle essentiel que l'intelligence artificielle peut jouer dans son développement. Nous proposons une nouvelle approche basée sur des graphes de connaissances multiples et des réseaux de propagation dans lesquels nous avons entraîné un modèle. Les deux bases de données utilisées sont la première comprenant des informations sur les agriculteurs et les cultures agricoles telles que le blé, le maïs et l'orge, et la deuxième comprenant des données sur la qualité du bétail telles que les vaches, les moutons et les chèvres. Nous avons traité et modifié ces données en les convertissant en une représentation tripartite (sujet, prédicat, objet), puis les avons divisées en trois ensembles : un pour les tests, un pour la validation et un pour l'entraînement. Les résultats de l'analyse ont été encourageants et prometteurs, ce qui nous encourage à exploiter et à développer ce modèle dans le secteur agricole en Algérie.

**Mots-clés :** Apprentissage profond, Graphe de connaissance, Prédiction des liens,

Graph Neural Network

# Abstract

---

Artificial intelligence is one of the most important and rapidly advancing fields in the era of globalization. It is widely used in various domains of daily life, such as agriculture, industry, digital transformation, smart systems, medicine, and biology. Deep learning, as a major branch of artificial intelligence, has achieved remarkable progress in the field and has become the most crucial area in computer science. It relies on deep neural networks to represent data and acquire knowledge, imitating the way humans think and learn.

In this study, we focus on the agriculture domain due to the significant role that artificial intelligence can play in its development. We propose a novel approach based on multiple knowledge graphs and propagation networks, in which we have trained a model. The two databases used consist of information about farmers and agricultural crops such as wheat, corn, and barley in the first database, and data about livestock quality including cows, sheep, and goats in the second database. We processed and modified this data by converting it into a tripartite representation (subject, predicate, object), and then divided it into three sets: one for testing, one for validation, and one for training. The analysis results have been encouraging and promising, which motivates us to further exploit and develop this model in the agricultural sector in Algeria.

# Liste des tableaux

---

Tableau 1. Comparaison entre l'apprentissage automatique et l'apprentissage en profondeur .....	18
Tableau 2. Affichage résultats des datasets végétal et animal .....	50

# Liste des acronymes

---

<b>ML</b>	Machine Learning
<b>RL</b>	Apprentissage par renforcement
<b>DRL</b>	Deep RL
<b>DNN</b>	Réseaux de neurones profonds
<b>CNN</b>	Réseaux de neurones convolutifs
<b>RNN</b>	réseaux de neurones récurrents
<b>LSTM</b>	les mémoires à long terme
<b>GRU</b>	les unités récurrentes fermées
<b>AE</b>	les encodeurs automatiques
<b>RBM</b>	les machines de Boltzmann restreintes
<b>GAN</b>	Generative Adversarial Networks
<b>NLP</b>	le traitement du langage naturel
<b>KG</b>	Graphes de connaissances
<b>COMPGCN</b>	Réseaux convolutionnels De graphiques multi-relationnels basés Sur composition
<b>RGCN</b>	Modélisation de Données Relationnelles avec des Réseaux Convolutifs de Graphes
<b>ID</b>	Identification fellah
<b>OCR</b>	Reconnaissance optique de caractères
<b>GNN</b>	Graph Neural Network
<b>MRR</b>	Mean Reciprocal Rank

# Introduction Générale

---

En 2012, Google a introduit le graphe de connaissances dans son moteur de recherche, marquant une avancée majeure dans la recherche sémantique et l'amélioration des résultats de recherche. Cette initiative visait à représenter l'information de manière plus efficace et à fournir des résultats de recherche plus pertinents pour les utilisateurs.

## 1. Contexte du projet et problématique

---

Le projet concerne la prédiction des liens dans les bases de connaissances dans le domaine des agriculteurs. Une base de connaissances est une collection structurée d'informations sur un sujet donné, et dans ce cas précis, elle contient des informations spécifiques aux agriculteurs.

L'objectif principal de ce projet est d'améliorer la qualité et la cohérence des bases de connaissances existantes en prédisant les liens manquants entre les différentes entités qui composent la base de connaissances. Par exemple, il peut s'agir de prédire les relations entre les agriculteurs et les types de cultures qu'ils cultivent, En utilisant des techniques d'apprentissage automatique et de traitement du langage naturel, le projet vise à exploiter les informations déjà présentes dans la base de connaissances pour identifier les liens manquants et les relations implicites entre les entités.

Les bases de connaissances existantes peuvent contenir des informations incomplètes ou manquantes. Cela rend la tâche de prédire les liens entre les entités encore plus complexe. Comment identifier les relations manquantes et les liens implicites entre les données existantes ?

## 2. Motivations

---

Pour répondre au problème nous avons cherché une solution basée sur l'exploitation maximale des données par ce que certaines informations peuvent être sous-utilisées ou rester non liées. La prédiction des liens permet de tirer pleinement parti de ces données en identifiant les relations entre les différentes entités et en créant des liens significatifs entre elles.

# Introduction Générale

---

## 3. Objectifs

---

L'objectif d'une approche de deep learning pour la prédiction des liens dans les bases de connaissances agricoles est d'utiliser la puissance des réseaux de neurones profonds pour capturer des relations complexes, prédire avec précision les liens manquants, gérer la variabilité des données, traiter de grandes quantités de données et fournir des prédictions interprétables et utiles dans le secteur agricole.

## 4. Contenu du mémoire

---

Ce projet débute par une introduction générale et est divisé en trois chapitres distincts.

**Le premier chapitre** introduit de manière concise le concept d'apprentissage profond, les différents types d'algorithmes utilisés dans l'apprentissage profond. Enfin, nous aborderons les avantages et les inconvénients de cette approche.

**Le deuxième chapitre** propose une discussion sur les graphes de connaissance, la modélisation des graphes et leur architecture. Nous concluons ce chapitre en examinant les prédictions de liens et les modèles d'utilisation associés.

**Le troisième chapitre** est la partie la plus étendue et est divisé en deux parties distinctes. La première partie porte sur l'approche proposée, y compris l'architecture générale du système et les bases de données utilisées. La deuxième partie se concentre sur l'implémentation, en décrivant l'environnement spécifique et en présentant les prétraitements nécessaires pour les données d'entrée du modèle utilisé. Enfin, nous décrirons la création, l'entraînement et le test de ce modèle sur les données choisies, ainsi que les résultats obtenus.

Le mémoire se conclut par une conclusion générale qui récapitule notre approche, ainsi que des perspectives de recherche pour de futures études dans ce domaine.

# Chapitre 1 : Généralités sur l'apprentissage Profond

---

## 1. Introduction

---

L'apprentissage automatique (Machine Learning - ML) est une technologie de l'intelligence artificielle qui permet aux ordinateurs d'apprendre à partir de données sans être explicitement programmés à cet effet. Cependant, pour que les ordinateurs puissent apprendre et se développer, ils ont besoin de données sur lesquelles s'entraîner et les analyser. En ce sens, le ML permet de tirer pleinement parti du Big Data.

L'apprentissage profond, une forme d'apprentissage automatique, imite le fonctionnement du cerveau humain. Il est utilisé pour détecter la fraude et le blanchiment d'argent, traiter les données pour la détection d'objets, reconnaître la parole, traduire les langues et prendre des décisions. C'est une fonction de l'IA qui offre de nombreuses possibilités dans divers domaines grâce à sa capacité à analyser et à comprendre des données complexes

Le Deep Learning permet de traiter des données non structurées telles que des images, des textes, des signaux audio, etc., en exploitant des architectures de réseaux neuronaux profonds. Ces architectures sont capables d'apprendre de manière hiérarchique des caractéristiques complexes à partir des données brutes, permettant ainsi une analyse plus approfondie et une meilleure compréhension des informations contenues dans ces données.[1]

# Chapitre 1 : Généralités sur l'apprentissage Profond

---

## 2. Définitions de l'apprentissage Profond

---

L'apprentissage profond a été introduit par Rina Dechter en 1986 dans le domaine de l'apprentissage automatique, puis par Igor Aizenberg et ses collègues en 2000 dans le contexte des réseaux de neurones artificiels. Il s'agit d'une sous-branche de l'intelligence artificielle qui permet à une machine d'apprendre automatiquement à partir de grandes quantités d'informations. Contrairement aux approches traditionnelles, l'algorithme d'apprentissage profond identifie directement les caractéristiques essentielles du traitement, sans nécessiter de traitement préalable réalisé par un humain.[2]

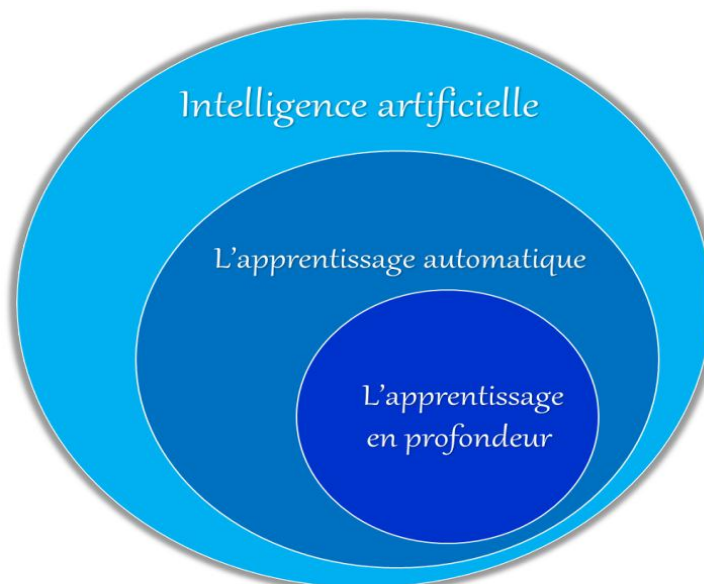


Figure 1. Les sous-branches de l'intelligence artificiel. [3]

## 3. Types d'apprentissage profond

---

L'apprentissage en profondeur peuvent être classées en trois catégories : supervisées, semi-supervisées et non supervisées. En plus de cela, il existe une catégorie distincte appelée apprentissage par renforcement (RL) ou Deep RL (DRL), qui est souvent abordée dans le cadre d'approches semi-supervisées ou parfois non supervisées.

# Chapitre 1 : Généralités sur l'apprentissage Profond

---

- 1- Apprentissage supervisé :** est une technique qui utilise des données étiquetées, où l'environnement fournit des entrées et des sorties correspondantes. Les approches d'apprentissage supervisé en deep learning incluent les réseaux de neurones profonds (DNN), les réseaux de neurones convolutifs (CNN), les réseaux de neurones récurrents (RNN), tels que les mémoires à long terme (LSTM) et les unités récurrentes fermées (GRU). Ces approches itèrent pour ajuster les paramètres du réseau afin d'approximer au mieux les sorties souhaitées. Chaque méthode a ses propres caractéristiques et applications spécifiques dans l'apprentissage supervisé en deep learning. [4]
- 2- Apprentissage semi-supervisé :** est basé sur des ensembles de données partiellement étiquetés, également connus sous le nom d'apprentissage par renforcement. Dans certains cas, les méthodes d'apprentissage semi-supervisé incluent l'utilisation de Deep RL (DRL) et de Réseaux Adversaires Génératifs (GAN). De plus, les réseaux de neurones récurrents (RNN), y compris les LSTM et les GRU, sont également utilisés dans l'apprentissage semi-supervisé. Ces approches permettent d'exploiter à la fois les données étiquetées et non étiquetées pour améliorer les performances et la généralisation des modèles d'apprentissage en profondeur dans des scénarios où les données étiquetées sont limitées. [4]
- 3- Apprentissage non supervisé :** L'apprentissage non supervisé est un processus d'apprentissage qui se fait sans l'utilisation d'étiquettes sur les données. Dans ce cas, le réseau apprend des représentations internes ou des caractéristiques significatives pour découvrir des relations ou une structure inconnue dans les données d'entrée. Les techniques couramment utilisées en apprentissage non supervisé comprennent le regroupement (clustering), la réduction de dimensionnalité et les techniques génératives. Certains membres de la famille du deep learning, tels que les encodeurs automatiques (AE), les machines de Boltzmann restreintes (RBM) et les GAN (Generative Adversarial Networks), sont particulièrement adaptés pour le clustering et la réduction de dimensionnalité non linéaire. De plus, les réseaux de neurones récurrents (RNN), tels que les LSTM (Long Short-Term Memory), ainsi que l'apprentissage par renforcement (RL), sont également utilisés dans l'apprentissage non supervisé dans de nombreux domaines d'application. [5]

# Chapitre 1 : Généralités sur l'apprentissage Profond

- 4- **Apprentissage par transfert** : a permis de réduire les exigences en termes de données. Il consiste à utiliser un modèle pré-entraîné, avec ses poids et ses paramètres déjà optimisés sur une grande base de données, puis à affiner ce modèle avec nos propres données. Le modèle pré-entraîné agit comme un extracteur de caractéristiques, où nous remplaçons la dernière couche par notre propre classificateur. Ensuite, nous figeons les poids des autres couches du réseau et procédons à l'entraînement normal du modèle (gel des couches pour ne pas modifier les poids lors de l'optimisation du gradient). Cette approche permet de bénéficier des connaissances préalables du modèle pré-entraîné tout en adaptant le modèle aux spécificités de notre ensemble de données. [6]

## 4. Différences entre Machine Learning et l'apprentissage profond

On peut résumer les différences majeures entre l'apprentissage profond et machine Learning dans ce tableau suivant :

Spécialité	apprentissage profond	Machine Learning
Mode d'organisation des informations	Données non structuré	Données structurée
Base de données	Supérieur à 1 million de donnée	Limité ou contrôlable
Type d'entraînement	Méthode d'apprentissage autonome	Entraînement par l'humain obligatoire
Algorithme	Réseau de neurone	Algorithme modifiable
Domaine d'application	Tache complexe (cybers sécurité, assistant vocale, ....)	Actions routinière simple, marketing en ligne, charbot,....)

Tableau 1. Comparaison entre l'apprentissage automatique et l'apprentissage en profondeur [7]

# Chapitre 1 : Généralités sur l'apprentissage Profond

## 5. Les différents algorithmes de l'apprentissage profond

Les algorithmes de Machine Learning les plus utilisés dans les applications d'apprentissage profond sont :

### 5.1. Réseaux neuronaux convolutifs (CNN)

Les réseaux de neurones à convolution (CNN) sont une forme spécifique de réseaux neuronaux qui utilisent l'opération mathématique de convolution dans au moins une de leurs couches. Les CNN sont largement considérés comme l'un des meilleurs algorithmes d'apprentissage pour effectuer la convolution, ce qui permet d'extraire des caractéristiques pertinentes à partir de données locales corrélées.

Dans un CNN, les noyaux convolutifs appliquent la convolution sur les entrées, ce qui signifie qu'ils effectuent une opération de multiplication locale entre les poids du noyau et les valeurs des entrées correspondantes. Cette convolution permet de capturer les motifs et les structures présents dans les données d'entrée.

La sortie des noyaux convolutifs est ensuite envoyée à une fonction d'activation non linéaire, également appelée unité de traitement non linéaire. Cette fonction d'activation joue un rôle crucial dans l'apprentissage des abstractions et ajoute de la non-linéarité à l'espace des caractéristiques extraites. Elle permet de générer différents modèles d'activation pour différentes réponses, facilitant ainsi l'apprentissage des différences sémantiques dans les images. [8]

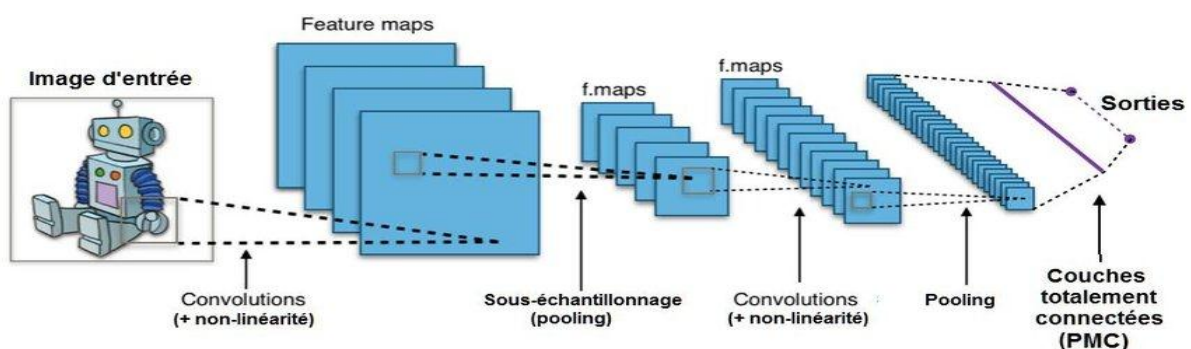


Figure 2. Réseaux neuronaux convolutifs [3]

# Chapitre 1 : Généralités sur l'apprentissage Profond

---

- **L'architecture des réseaux neuronaux convolutifs**

L'architecture des réseaux neuronaux convolutifs (CNN) est composée de :

**1. Couche de convolution :** la couche de convolution est la première couche d'un réseau de neurones à convolution (CNN) et joue un rôle fondamental dans l'extraction des entités d'une image d'entrée. La convolution permet de préserver la relation spatiale entre les pixels de l'image en utilisant de petits carrés de données d'entrée.

La convolution est une opération mathématique qui implique l'application d'un filtre ou d'un noyau à l'image d'entrée. Le filtre est une petite matrice de poids qui est glissée sur l'image en effectuant des multiplications locales entre les valeurs de l'image et les valeurs du filtre. Ces multiplications locales sont ensuite sommées pour obtenir une nouvelle valeur qui représente la caractéristique extraite à cette position spécifique. [9]

**2. La couche de pooling :** est généralement placée entre deux couches de convolution dans un réseau de neurones à convolution (CNN). Le pooling est un processus de réduction de la dimensionnalité basé sur l'échantillonnage. Son objectif est de créer une représentation agrégée d'une région donnée (comme une image ou une carte de caractéristiques) en réduisant sa taille tout en préservant les caractéristiques importantes.

Il existe différents types de pooling, les deux plus couramment utilisés étant :

- **Pooling moyen (average pooling) :** Dans cette méthode, chaque région de pooling est divisée en sous-régions, et la valeur moyenne des pixels de chaque sous-région est calculée pour obtenir une valeur agrégée. Cela permet de réduire la dimensionnalité en conservant une estimation de la tendance centrale des pixels dans chaque région.
- **Pooling maximal (max pooling) :** Dans cette méthode, chaque région de pooling est également divisée en sous-régions, et la valeur maximale des pixels de chaque sous-région est extraite pour obtenir une valeur agrégée. Cela permet de réduire la dimensionnalité en conservant la valeur maximale, qui représente la présence de caractéristiques distinctes dans chaque région.

# Chapitre 1 : Généralités sur l'apprentissage Profond

**3. La couche fully-connected :** entièrement connectée dans un réseau de neurones à convolution (CNN) est similaire aux réseaux entièrement connectés traditionnels. La sortie de la phase précédente du CNN, qui comprend les couches de convolution et de pooling répétitives, est introduite dans la couche fully-connected. [10]

## 5.2. Réseaux neuronaux récurrents (RNN)

Les réseaux neuronaux récurrents (RNN) sont une famille de réseaux de neurones utilisés pour le traitement de données séquentielles. Ils ont été inventés par l'informaticien américain Michael Irwin Jordan en 1986. Jürgen Schmidhuber, un chercheur allemand, considère que les réseaux neuronaux récurrents sont similaires au cerveau humain, qui est un vaste réseau de neurones rétroactifs connectés. Ces réseaux peuvent apprendre à traduire un flux d'entrée sensorielle en une séquence de sorties motrices utiles tout au long de la vie [11]

La caractéristique la plus importante des réseaux neuronaux récurrents (RNN) réside dans leur capacité à modéliser la dimension temporelle des données. Les RNN sont spécifiquement conçus pour reconnaître les caractéristiques séquentielles et les motifs dans les données, leur permettant ainsi de prédire le scénario suivant le plus probable.

### • L'architecture des réseaux neuronaux récurrents (RNN)

Les réseaux de neurones récurrents sont une extension des réseaux de neurones à flux, où des connexions récurrentes sont ajoutées. Ces connexions s'étendent sur des pas de temps adjacents, ce qui confère au modèle la notion de temps. Les connexions récurrentes peuvent former des cycles, y compris des connexions de rétroaction vers les neurones d'origine dans le futur (voir Figure 3)

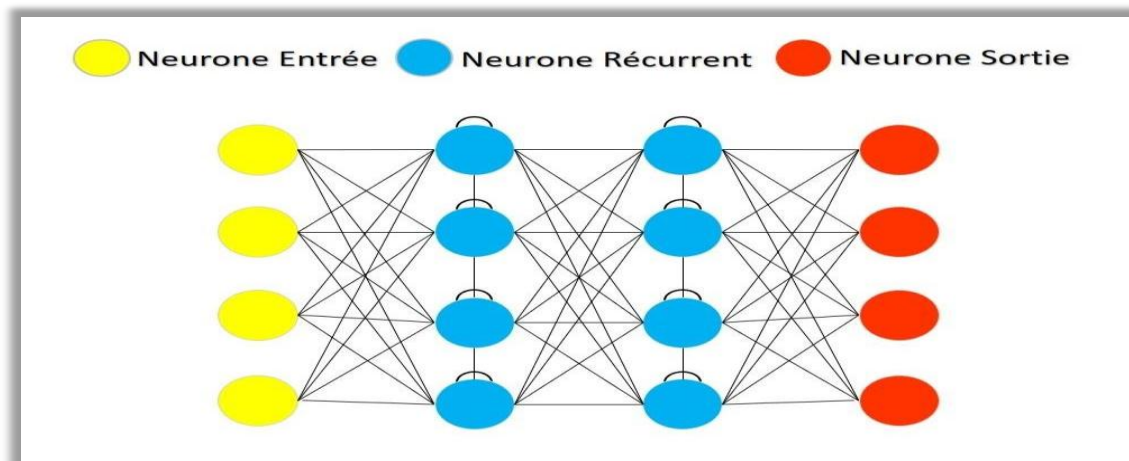


Figure 3. Réseaux neuronaux récurrents (RNN) [11]

# Chapitre 1 : Généralités sur l'apprentissage Profond

## 5.3. Le réseau de neurones LSTM

Les LSTM (Long Short-Term Memory) « Voir Figure 5 » ont été spécifiquement conçus pour résoudre le problème de la dépendance à long terme. Contrairement à d'autres modèles de réseaux de neurones, les LSTM sont naturellement capables de retenir des informations sur de longues périodes sans effort. Au lieu de devoir apprendre à retenir ces informations, c'est leur fonctionnement par défaut.

Les LSTM permettent aux RNN de se souvenir de l'information sur une longue période de temps en stockant les données dans une mémoire similaire à celle d'un ordinateur. Cette mémoire peut être considérée comme une cellule qui décide si elle doit stocker ou supprimer des informations en fonction de leur importance. Les poids des informations sont également appris par l'algorithme, ce qui signifie qu'il apprend progressivement à discerner les informations importantes de celles qui ne le sont pas. En somme, les LSTM sont dotés d'une mémoire qui leur permet de retenir les informations importantes sur une longue période, ce qui les rend idéales pour les tâches de traitement de données séquentielles. [11]

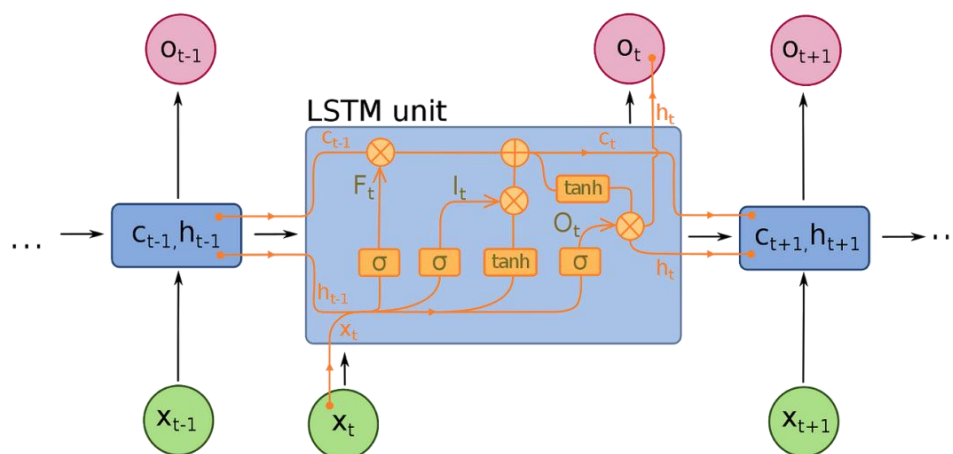


Figure 4 .

Figure 4. Architecture LSTM (Long Short-Term Memory) [11]

# Chapitre 1 : Généralités sur l'apprentissage Profond

## 5.4. Réseaux adversariaux génératifs (GAN)

Les Réseaux antagonistes génératifs (GAN) sont un exemple de réseau qui utilise l'apprentissage non supervisé pour entraîner deux modèles en parallèle. Les GANs sont capables de synthétiser de nouvelles images en apprenant à partir d'autres images. Ils peuvent également être utilisés dans d'autres domaines, tels que :

- La génération de sons.
- La génération de vidéos.
- La génération d'images à partir de descriptions textuelles.

Les GANs sont un outil puissant pour la génération de contenu créatif et ont de nombreuses applications potentielles dans divers domaines.

Un aspect clé des GANs est la façon dont ils utilisent un compte de paramètres ou un ensemble de données d'entrée qui est beaucoup plus petit que la normale en ce qui concerne la quantité de données sur lesquelles nous entraînons le réseau, le réseau est obligé de représenter efficacement les données d'entraînement, ce qui le rend plus efficace pour générer des données semblables aux données d'entraînement . [11]

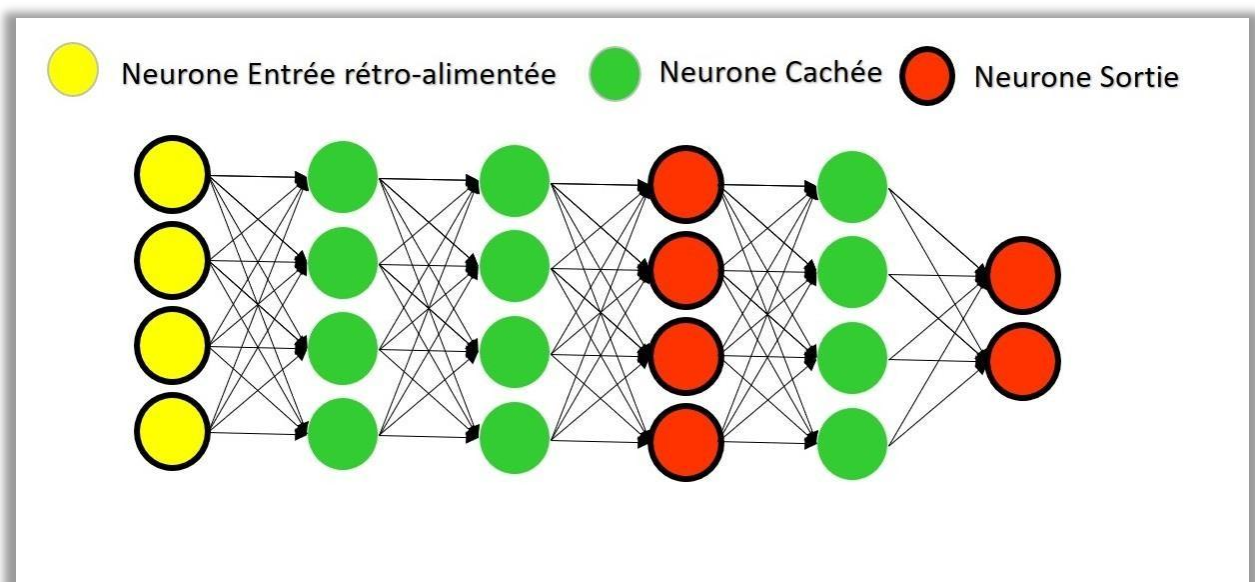


Figure 5. Réseaux adversariaux génératifs [3]

# Chapitre 1 : Généralités sur l'apprentissage Profond

---

## 6. Les avantages et les inconvénients de l'apprentissage profond

---

### A. Les avantages

L'apprentissage profond a de nombreux avantages telle que :

- ✚ un moteur de transformation de l'intelligence artificielle. Il offre des performances de prédiction et de classification exceptionnelles et est capable d'effectuer des tâches complexes.
- ✚ L'une des principales caractéristiques de l'apprentissage profond est sa capacité à apprendre à partir des données et à améliorer constamment ses performances, ce qui permet aux systèmes d'intelligence artificielle d'offrir des prédictions précises et performantes. La capacité d'apprentissage profond à apprendre des données variées et complexes et à prendre en compte plusieurs caractéristiques est un autre avantage.
- ✚ Le développement et le perfectionnement des modèles d'apprentissage profond et de leurs fonctionnalités ne cessent d'augmenter et fournissent des performances de prédiction et de classification optimales. Les modèles basés sur l'apprentissage profond sont incroyablement flexibles et peuvent être appliqués à des domaines variés, tels que la reconnaissance vocale, la vision par ordinateur et la compréhension du langage naturel.
- ✚ L'apprentissage profond peut également s'adapter à des systèmes qui n'ont pas besoin d'être reconfigurés ou réécrits à chaque fois qu'ils sont utilisés. Cela signifie que les systèmes peuvent s'améliorer et s'adapter en fonction des données et des conditions changeantes, ce qui est particulièrement utile pour des tâches à grande échelle, telles que l'analyse des données et le traitement des images. [12]

# Chapitre 1 : Généralités sur l'apprentissage Profond

---

## B. Les inconvénients

- ✚ Besoin de grandes quantités de données et de puissance de calcul : L'apprentissage profond nécessite souvent des ensembles de données volumineux pour être efficace. De plus, l'entraînement de modèles d'apprentissage profond peut nécessiter des ressources informatiques considérables, telles que des processeurs graphiques puissants et des infrastructures de calcul distribuées.
- ✚ Opacité et manque d'interprétabilité : Les modèles d'apprentissage profond sont souvent considérés comme des boîtes noires, car il peut être difficile de comprendre comment ils arrivent à leurs prédictions. Cela soulève des préoccupations en matière d'explicabilité et d'interprétation des décisions prises par ces modèles.
- ✚ Sensibilité aux données d'entraînement biaisées : Si les données d'entraînement sont biaisées ou non représentatives de la population cible, les modèles d'apprentissage profond peuvent reproduire ces biais et prendre des décisions discriminatoires.
- ✚ Coût et complexité de mise en œuvre : La mise en œuvre de l'apprentissage profond peut être coûteuse et complexe, nécessitant des experts en apprentissage automatique et une infrastructure appropriée pour l'entraînement et le déploiement des modèles. [13]

## 7. Conclusion

---

Dans ce chapitre, nous avons défini l'apprentissage profond et exploré ses types, ainsi que les différences avec l'apprentissage automatique. Nous avons examiné les algorithmes et les architectures associés, et nous avons discuté des avantages et des inconvénients de l'apprentissage profond. Dans le prochain chapitre, nous nous pencherons sur l'étude des graphes de connaissances et plus spécifiquement sur la prédiction des liens au sein de ces graphes.

# Chapitre 2 : Les Graphes de Connaissance

---

## 1. Introduction

---

La notion de base de connaissances, issue des sciences informatiques, se réfère à une base de données qui regroupe des connaissances spécifiques à un domaine spécialisé dans un format exploitable par un ordinateur. Elle contient des règles, des faits ou d'autres représentations nécessaires pour soutenir une activité spécifique pour laquelle la base de connaissances a été développée. [14]

## 2. Définitions

---

Pour rappel, un graphe est constitué d'un ensemble d'entités interconnectées, représentées par des nœuds, et de relations entre ces entités, représentées par des arcs ou des arêtes. Les relations ou arcs peuvent être enrichis par des attributs ou des valeurs quantitatives qui représentent le poids ou l'intensité de la relation. [15]

Le terme "Knowledge Graph", introduit par Google, fait référence à une représentation de connaissances liées à un domaine ou à une entreprise, structurée de manière à être facilement exploitée par les machines. Le Knowledge Graph est composé d'entités et de relations, où les entités peuvent représenter des noms de personnes, des concepts ou des objets. [15]

Le graphe de connaissances est une base de connaissances sémantique qui permet de décrire la signification des sources d'information et de rendre leur contenu explicite [14]. En effet, une entité ou un mot seul ne possède pas beaucoup de sens en lui-même. Son sens se révèle lorsque l'entité est prise dans son contexte, défini par ses propriétés et ses relations avec d'autres entités. Chaque entité contribue donc à la compréhension des entités auxquelles elle est liée. On dit que la base de connaissances est sémantique car elle encode le sens des données, ce qui permet aux machines de mieux comprendre les requêtes formulées par les êtres humains. On peut ainsi utiliser le langage naturel pour interroger la base de connaissances et générer une réponse en langage naturel, car les mots sont liés à des concepts. C'est ce qui permet aux assistants vocaux tels que Google Assistant, Siri ou Alexa de comprendre les demandes des utilisateurs et de fournir des réponses précises [14].

# Chapitre 2 : Les Graphes de Connaissances

---

## 3. Modélisation des graphes de connaissances

---

La modélisation des graphes de connaissances est une étape essentielle pour représenter les connaissances de manière structurée et exploitable. Les principaux concepts pour la modélisation des graphes sont :

- **Nœuds et entités** : Les nœuds représentent les entités du monde réel dans un graphe de connaissances. Chaque nœud correspond à une entité spécifique, telle qu'une personne, un lieu, un objet ou un concept. Les nœuds sont souvent identifiés par des identifiants uniques. Par exemple, dans un graphe de connaissances représentant des personnes, chaque nœud peut représenter une personne spécifique avec des attributs tels que le nom, l'âge, l'adresse, etc. [16]
- **Relations** : Les relations décrivent les connexions et les liens sémantiques entre les nœuds d'un graphe de connaissances. Elles représentent les associations ou les interactions entre les entités. Par exemple, dans un graphe de connaissances représentant des relations sociales, une relation peut indiquer que deux personnes sont des amis, qu'un film est réalisé par un réalisateur donné, ou qu'une ville est située dans un pays spécifique. Les relations sont souvent représentées par des arcs ou des arêtes reliant les nœuds correspondants. [16]
- **Propriétés** : Les propriétés sont des attributs ou des caractéristiques associés aux nœuds ou aux relations dans un graphe de connaissances. Elles fournissent des informations supplémentaires sur les entités ou les liens. Par exemple, pour un nœud représentant une personne, des propriétés telles que le nom, l'âge, l'adresse peuvent être associées. Les propriétés peuvent également être attachées aux relations pour exprimer des détails spécifiques sur ces relations. [16]

# Chapitre 2 : Les Graphes de Connaissance

## 4. Architecture d'un graphe de connaissance

En général, l'architecture des graphes de connaissances peut être représentée comme indiqué dans la Figure 6. Du point de vue des graphes de connaissances basés sur l'ontologie, il existe deux approches principales pour la création de graphes de connaissances : l'approche descendante et l'approche ascendante.

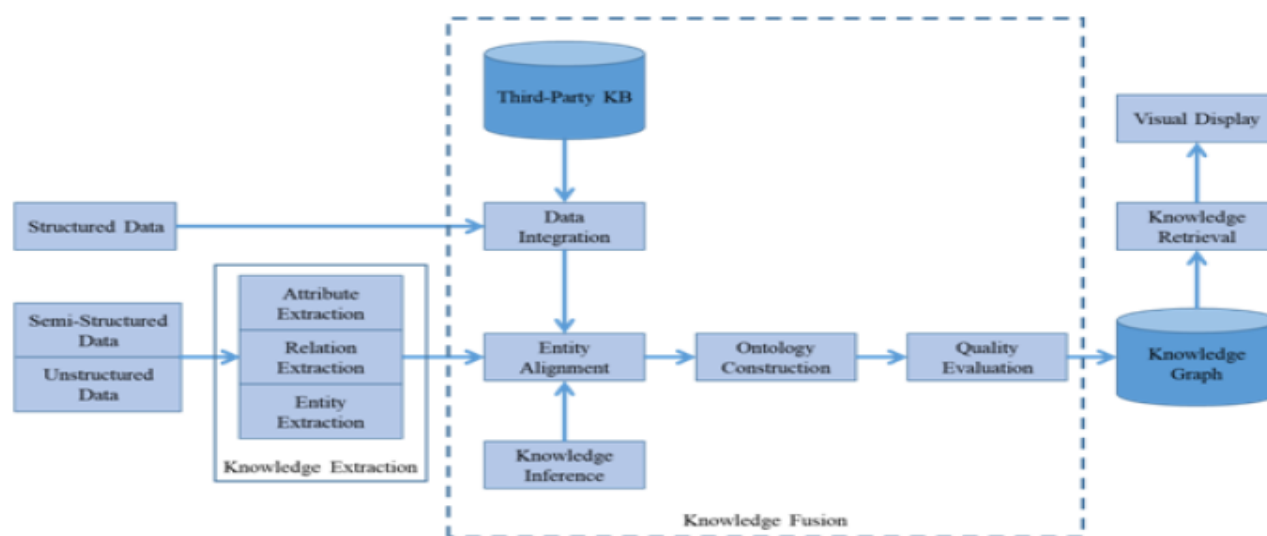


Figure 6. Architecture d'un graphe de connaissances. [17]

- **L'approche descendante** se concentre sur les ontologies de domaine et leur schéma pour construire les graphes de connaissances. Elle définit d'abord les ontologies, puis ajoute les instances de connaissances. En revanche, l'approche ascendante extrait des instances de connaissances à partir de ressources existantes, les fusionne et construit ensuite les ontologies de niveau supérieur pour créer un graphe de connaissances complet. [17]
- **L'approche ascendante** construit des graphes de connaissances en extrayant des instances de connaissances à partir de ressources existantes. Ces instances sont fusionnées pour créer un graphe de connaissances complet. Ensuite, des ontologies de niveau supérieur sont construites en utilisant les instances de connaissances. Cette approche utilise le traitement du langage naturel (NLP) pour analyser les textes et créer des graphes de connaissances polyvalents. [17]

# Chapitre 2 : Les Graphes de Connaissance

---

## 5. Prédiction des liens dans les graphes de connaissances

---

### 5.1. Prédiction de liens

La prédiction de liens est l'une des tâches les plus courantes pour évaluer les performances des systèmes de graphes de connaissances (KGE), Elle se divise en deux sous-tâches :

- la prédiction d'entités, également appelée classement d'entités, pour prédire les entités manquantes, c'est-à-dire prédire  $h$  étant donné  $(?, r, t)$  ou  $t$  étant donné  $(h, r, ?)$ .
- la prédiction de relations pour prédire les relations manquantes, c'est-à-dire prédire  $r$  étant donné  $(h, ?, t)$ , où "?" représente une entité ou une relation manquante pour un triplet  $(h, r, t)$ . Pour les modèles de prédiction de liens, l'objectif est de prédire les relations manquantes entre les entités en utilisant les relations existantes dans un graphe de connaissances. Il remplace la relation de chaque triplet de test par toutes les relations du graphe de connaissances pour obtenir les échantillons négatifs. Cela est également défini comme une tâche de tri d'entités. Ensuite, il détermine si le nouveau triplet, qui n'est pas observé dans le graphe de connaissances, est valide selon la fonction de score. [18]

### 5.2. Les différents objectifs de la prédiction de liens

La prédiction de liens dans les graphes de connaissances peut avoir plusieurs objectifs différents, tels que :

- Prédiction de l'existence de liens : C'est la tâche la plus courante, qui consiste à prédire si une relation ou un lien entre deux nœuds donnés va se former ou non dans le graphe. Cela peut être utile pour découvrir des connexions cachées ou manquantes entre les entités.
- Prédiction du poids des liens : Cette tâche vise à prédire la valeur numérique ou la force d'un lien entre deux nœuds. Elle permet de quantifier l'intensité de la relation entre les entités et peut être utilisée dans des domaines tels que la recommandation de produits ou la prédiction des interactions sociales.
- Prédiction du type de liaison : Dans certains graphes de connaissances, il peut exister différents types de liens entre les nœuds, chacun représentant une relation spécifique. La prédiction du type de lien consiste à déterminer quel type de relation va se former entre deux entités.

# Chapitre 2 : Les Graphes de Connaissance

- Prédiction de la cardinalité des liens : Cette tâche concerne la prédiction du nombre de liens qu'un nœud va avoir dans le futur. Elle peut être utilisée, par exemple, pour prédire le nombre d'amis qu'une personne va avoir sur les réseaux sociaux. [18]

## 6. Modèles utilisés pour les graphes de connaissances

### 1. COMPGCN

(Réseaux convolutionnels De graphiques multi-relationnels basés Sur composition)

#### A. Définition

COMPGCN, un nouveau cadre basé sur les Graph Convolution pour les graphes multi-relationnels qui exploite une variété d'opérateurs de composition issus des techniques d'embedding de Knowledge Graph pour intégrer conjointement les nœuds et les relations dans un graphe. Notre méthode généralise plusieurs méthodes de GCN multi-relationnelles existantes. De plus, notre méthode résout le problème de la sur-para métrisation en partageant les embeddings de relations entre les couches et en utilisant une décomposition de base. À travers des expériences approfondies sur des tâches de prédiction de liens, de classification de nœuds et de graphes de connaissances. [19]

#### B. Architecture CompGCN

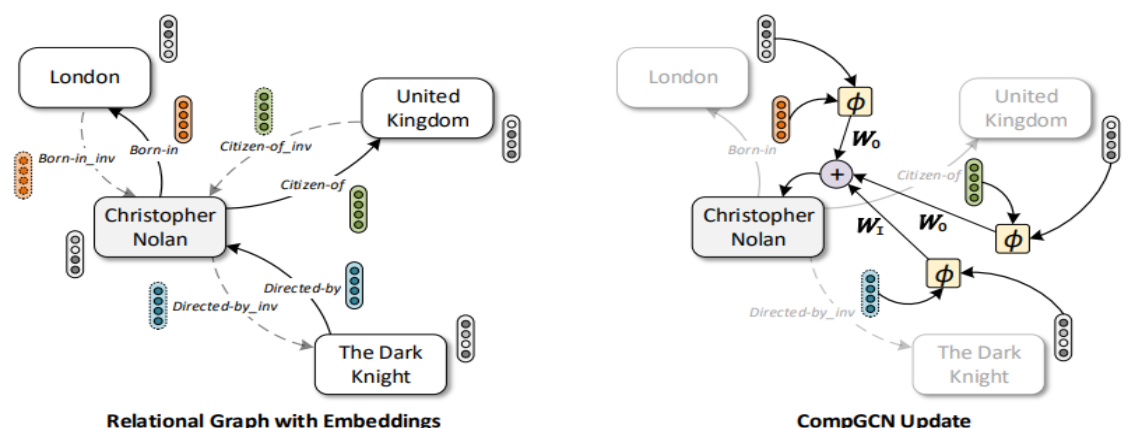


Figure 7. Architecture CompGCN [20]

COMPGCN donné des embeddings de nœuds et de relations, COMPGCN effectue une opération de composition  $\phi(\cdot)$  pour chaque arête dans le voisinage d'un nœud central.

# Chapitre 2 : Les Graphes de Connaissance

---

Les embeddings composés sont ensuite convolués avec des filtres spécifiques  $W_O$  et  $W_I$  pour les relations originales et inverses respectivement. Nous avons omis la boucle de rétroaction dans le diagramme pour plus de clarté. Le message de tous les voisins est ensuite agrégé pour obtenir un embedding mis à jour du nœud central. De plus, les embeddings de relations sont transformés à l'aide d'une matrice de poids séparée. (voir la figure 7). [20]

## C. GCN sur les graphiques non orientés

Étant donné un graphe  $G = (V, E, X)$ , où  $V$  représente l'ensemble des sommets,  $E$  est l'ensemble des arêtes, et  $X \in \mathbb{R}^{|V| \times d_0}$  représente les caractéristiques d'entrée de chaque sommet de dimension  $d_0$ . [21]

## D. GCN sur les graphes multi-relationnels

La formule d'un graphe multi-relationnel est définie comme suit :

$G = (V, E, R, X)$ , où  $R$  représente l'ensemble des relations

- chaque arête  $(u, v, r)$  représente la relation  $r$  existante du nœud  $u$  au  $v$ .
- chaque arête  $(u, v, r)$ , une arête inverse  $(v, u, r^{-1})$  est incluse dans  $G$ . [19]

## E. Fonctionnement de COMPGCN ( les tâches mentionnées) :

- **La prédiction de liens** : la tâche consistant à déduire des faits manquants à partir de faits connus dans les graphes de connaissances. utilisons les ensembles de données FB15k-et WN18RR pour l'évaluation. En suivant Bordes et al. (2013), un paramétrage filtré utilisé pour l'évaluation et rapportons le rang moyen réciproque (MRR), le rang moyen (MR) et Hits@N.
- **La classification de nœuds** : La classification de graphes consiste à apprendre une représentation pour chaque graphe, qui est ensuite utilisée par un classificateur pour prédire son étiquette, à partir d'un ensemble de graphes et de leurs étiquettes correspondantes. Nous évaluons notre modèle sur deux ensembles de données de bio-informatique : MUTAG (graph) et PTC [19].

# Chapitre 2 : Les Graphes de Connaissance

## 2. RGCN

(Modélisation de Données Relationnelles avec des Réseaux Convolutifs de Graphes )

### A. L'objectif:

L'objectif est de prédire les informations manquantes dans les bases de connaissances en utilisant des méthodes d'apprentissage automatique et des approches statistiques pour améliorer la couverture et l'exhaustivité des connaissances stockées.

Le modèle RGCN utilise un réseau de convolution graphique relationnel pour encoder les entités dans le graphe relationnel. Ce modèle est ensuite utilisé pour prédire les labels des entités en utilisant des classifieurs softmax. [21]

### B. Réseaux Convolutifs des graphes relationnels

Ce type de transformation s'est avéré très efficace pour accumuler et coder les caractéristiques des quartiers locaux et structurés, et a conduit à des améliorations significatives dans des domaines tels que la classification des graphes et la semi-représentation basée sur les graphes. -apprentissage supervisé. Motivés par ces architectures, le modèle de propagation simple suivant pour calculer la mise à jour en avant d'une entité ou d'un nœud noté  $v_i$  dans un multi-graphe relationnel (orienté et étiqueté) . [21]

### C. Architecture RGCN

Il y'a dans le modèle R-GCN deux cas :

(a) pour la classification des entités avec une fonction de perte par nœud.

(b) Modèle de prédiction de liens avec un encodeur R-GCN (entremêlé avec des couches entièrement connectées) et un décodeur DistMult qui prend des paires de représentations de nœuds cachés et produit un score pour chaque (potentiel) arête dans le graphe. La perte est évaluée par arête. [21]

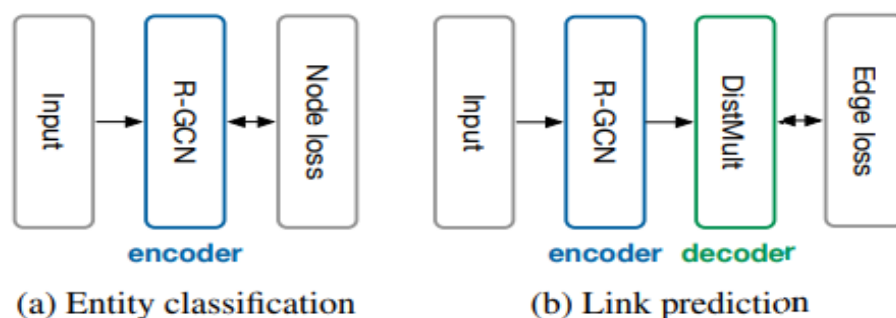


Figure 8. Architecture R-GCN [21]

# Chapitre 2 : Les Graphes de Connaissance

---

Le modèle RGCN inscrit dans le cadre de la prédiction de liens dans une base de connaissances incomplète. Il utilise un modèle d'auto-encodeur de graphe avec un encodeur R-GCN pour représenter les entités du graphe et une fonction de score basée sur la factorisation DistMult. Nous nous distinguons des approches précédentes en utilisant un encodeur pour calculer les représentations des entités. Nous entraînons notre modèle en utilisant l'échantillonnage négatif, où nous échantillonnons plusieurs exemples négatifs pour chaque exemple observé en corrompant aléatoirement le sujet ou l'objet. L'objectif de l'entraînement est de maximiser la différence de score entre les triplets observables et les exemples négatifs, ce qui est réalisé en optimisant la perte de cross-entropy. Nos expériences démontrent que notre modèle améliore les performances de prédiction de liens par rapport aux approches existantes. [21]

## 7. Conclusion

---

Le deuxième chapitre de ce projet donne un aperçu approfondi des graphes de connaissance. Il commence par une introduction générale sur la définition des graphes de connaissance et explore ensuite les différentes modélisations et architectures utilisées pour les représenter. Ensuite, une approche spécifique axée sur la prédiction des liens dans les graphes de connaissance est examinée, en mettant en évidence les méthodes et les modèles tels que CompGCN et RGCN qui ont obtenu de bons résultats dans ce domaine. En conclusion du chapitre, nous soulignons l'importance des modèles spécifiques que nous avons choisis d'utiliser, tels que CompGCN, et leur pertinence par rapport à notre objectif de prédiction des liens dans les graphes de connaissance.

# Chapitre 3 : Expérimentation

---

## 1. Introduction

---

La wilaya d'El Tarf est une région située dans le nord-est de l'Algérie, bordée par la mer Méditerranée à l'est, occupe une superficie agricole estimée à 84 031 hectares, dont 88% sont arables cela équivaut à 74.173 hectares, L'agriculture y joue un rôle important dans l'économie locale et la subsistance des habitants.

Les principales activités agricoles dans la wilaya d'El Tarf sont centrées sur la culture des céréales, des légumes, des olives, des fruits et de l'élevage du bétail. Les conditions climatiques favorables, avec des hivers doux et des étés chauds, ainsi que la proximité de la mer, offrent un environnement propice à l'agriculture.

La culture des céréales, notamment le blé et l'orge, est largement pratiquée dans la région. Les agriculteurs cultivent également des légumes tels que les tomates, les pommes de terre, les carottes, les oignons et les courgettes. Les vergers produisent des fruits tels que les agrumes, les pêches, les abricots et les figues.

## 2. Approche Proposée

---

Dans le cadre de votre projet, vous avez extrait deux bases des données du secteur agricole de l'état d'El-Tarf. La première base de données concerne le secteur agricole végétal, tandis que la deuxième concerne le secteur agricole animal.

### **A. Bases des données végétales :**

La base de données végétal recense les agriculteurs qui cultivent des cultures telles que le blé dur, l'orge, le maïs et les pois chiches. Chaque agriculteur est identifié par un ID unique et les informations additionnelles comprennent son âge, sa commune d'appartenance et le type de filière agricole qu'il cultive. Par exemple, l'ID 1 correspond à l'agriculteur ABADLIA ADEL qui réside à Ain Assel et qui, à l'âge de 46 ans, cultive du blé dur. Cette base de données contient 5 colonnes et 5590 lignes.

# Chapitre 3 : Expérimentation

	A	B	C	D	E
1	ID Fellah	commune	NOM	age	Produire
2	1	Ain El Asse	ABADLIA ADEL	46	bles dur,
3	2	Ain El Asse	ABADLIA DJAMEL	60	bles dur,mais,
4	3	Ain El Asse	ABADLIA KHMIS	44	bles dur,
5	4	Ain El Asse	ABADLIA SEBTI	29	bles dur,mais,
6	5	Ain El Asse	ABDOU WAHEB	46	bles dur,
7	6	Ain El Asse	ABED MOUSSA	48	bles dur,orge,
8	7	Ain El Asse	ABIDET AMMAR	60	bles dur,orge,
9	8	Ain El Asse	ABIDET MOUSSA	64	bles dur,
10	9	Ain El Asse	ABIDET NABIL	46	bles dur,orge,mais,
11	10	Ain El Asse	AFIF KAMEL	60	bles dur,
12	11	Ain El Asse	AMRAOUI CHOKRI	44	bles dur,
13	12	Ain El Asse	AOUAN SALEM	29	orge,mais,pois chiche,
14	13	Ain El Asse	AOUN CHADLI	46	bles dur,
15	14	Ain El Asse	ARDJOUN WAHID	48	bles dur,
16	15	Ain El Asse	ARDJOUNI ZIDANE	60	bles dur,
17	16	Ain El Asse	ATI HAFNAOUI	64	bles dur,
18	17	Ain El Asse	ATI HALIM	46	bles dur,
19	18	Ain El Asse	ATI ZIDANE	60	orge,
20	19	Ain El Asse	BAIRAI LAKHDAR	44	bles dur,orge,mais,pois chiche,
21	20	Ain El Asse	BEDIAF AZIZ	29	bles dur,
22	21	Ain El Asse	BEKAKRA MABROUK	46	bles dur,

Figure 9. Bases des données des filières végétales (xlsx)

## B. Bases des données animales

La deuxième base de données recense les agriculteurs qui s'occupent d'élevage animal, tels que les bovins, les ovins et les caprins. Comme pour la première base de données, chaque agriculteur est identifié par un ID unique et les informations supplémentaires incluent l'âge, la commune d'appartenance, le type de filière agricole et le nombre d'animaux élevés. Par exemple, l'ID 1 correspond à l'agriculteur ABADLIA ADEL qui réside à Ain Assel et qui, à l'âge de 46 ans, élève 07 têtes de bovins. Cette base de données comporte également 5 colonnes et 1311 lignes.

	A	B	C	D	E
1	ID	Commune	Nom fellah	Nbre	produit
2	1	Ain El Asse	ABADLIA ADEL	7	Bovins
3	2	Ain El Asse	ABADLIA DJAMEL	6	Bovins
4	3	Ain El Asse	ABADLIA KHMIS	8	Bovins
5	4	Ain El Asse	ABADLIA SEBTI	6	Bovins
6	5	Ain El Asse	ABDOU WAHEB	12	ovins
7	6	Ain El Asse	ABED MOUSSA	11	ovins
8	7	Ain El Asse	ABIDET AMMAR	8	Bovins
9	8	Ain El Asse	ABIDET MOUSSA	9	caprins
10	9	Ain El Asse	ABIDET NABIL	13	Bovins
11	10	Ain El Asse	AFIF KAMEL	10	Bovins
12	11	Ain El Asse	AMRAOUI CHOKRI	6	caprins
13	12	Ain El Asse	AOUAN SALEM	0	
14	13	Ain El Asse	AOUN CHADLI	8	ovins
15	14	Ain El Asse	ARDJOUN WAHID	7	ovins
16	15	Ain El Asse	ARDJOUNI ZIDANE	9	ovins
17	16	Ain El Asse	ATI HAFNAOUI	8	caprins
18	17	Ain El Asse	ATI HALIM	10	caprins
19	18	Ain El Asse	ATI ZIDANE	0	
20	19	Ain El Asse	BAIRAI LAKHDAR	6	caprins
21	20	Ain El Asse	BEDIAF AZIZ	9	ovins
22	21	Ain El Asse	BEKAKRA MABROUK	10	Bovins

Figure 10. Bases des données des filières animales (xlsx)

# Chapitre 3 : Expérimentation

Ces deux bases de données constituent une source précieuse d'informations sur les agriculteurs du secteur agricole végétal et animal de la wilaya d'El-Tarf. Elles fournissent des détails importants tels que l'identité des agriculteurs, leur localisation géographique, leur âge, le type de filière agricole qu'ils cultivent et les quantités produites. Ces données peuvent servir de base pour des analyses plus approfondies, des prédictions de liens ou des recommandations dans le domaine agricole.

## 3. Architecture générale du système

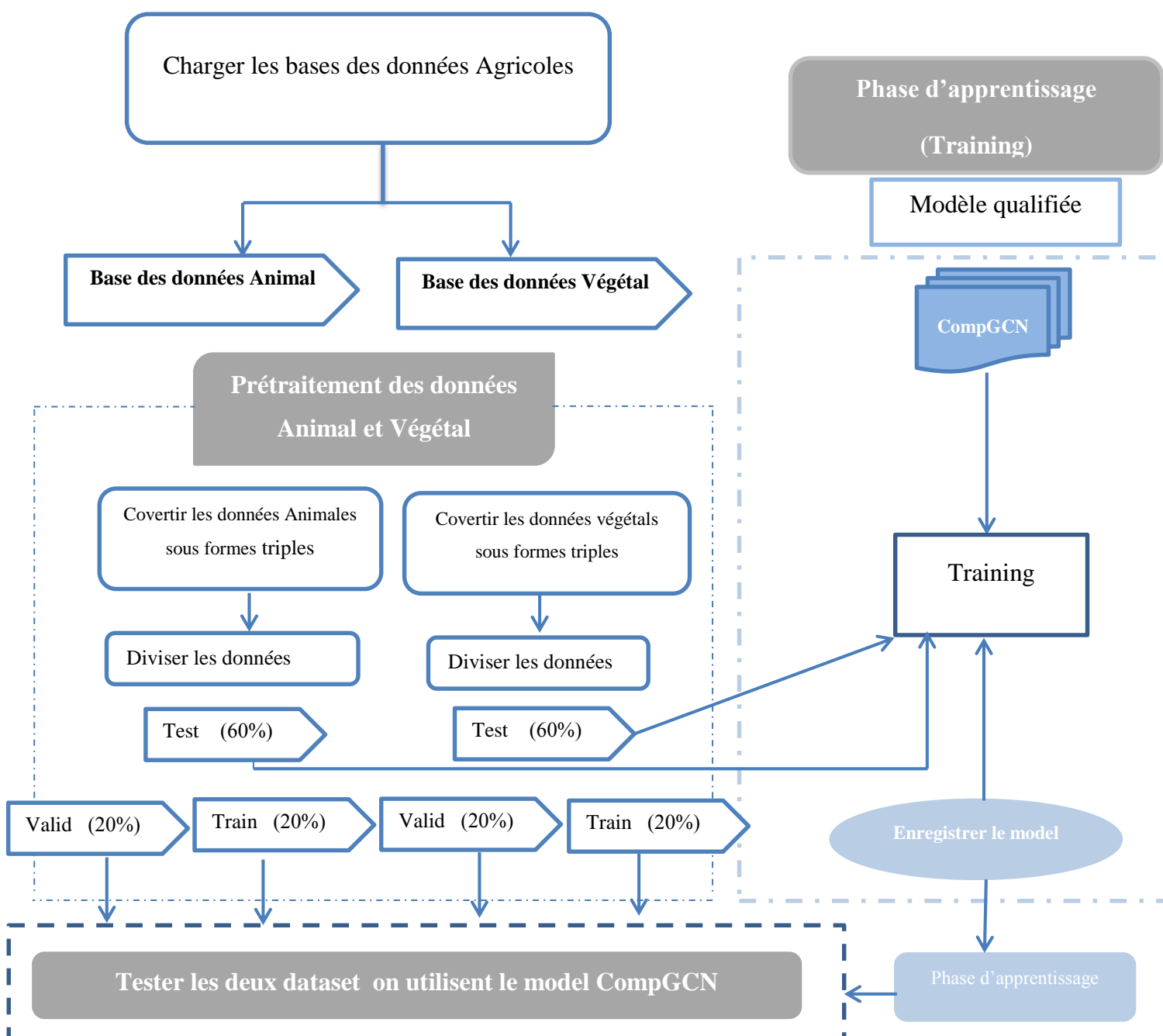


Figure 11. Architecture générale du système

# Chapitre 3 : Expérimentation

---

L'objectif de notre travail est de concevoir un système intelligent capable de prédire de nouvelles relations entre les agriculteurs et les cultures agricoles en utilisant deux bases de données. La première base de données concerne les filières agricoles, tandis que la deuxième base de données concerne les produits animaux.

Notre système suit une architecture qui comprend les étapes suivantes :

- **Prétraitement des données :**

Les données des deux bases de données sont prétraitées en utilisant la méthode de conversion des données en forme de triplets (entité, relation, objet). Cette représentation en triplets permet de capturer les relations entre les agriculteurs, les cultures agricoles et les produits animaux.

- **Division des données :**

Les données prétraitées sont divisées en trois sous-ensembles distincts : un ensemble de test, un ensemble d'entraînement et un ensemble de validation. Cela nous permet d'évaluer les performances de notre système sur des données non vues lors de l'entraînement et de régler les hyperparamètres.

- **Apprentissage des données :**

le modèles utilisé pour effectuer la prédiction est le modèle CompgCN (Composition Graph Convolutional Network), qui exploite les informations des graphes des filières agricoles et des produits animaux pour apprendre les relations entre les entités. les relations complexes entre les entités et d'apprendre des représentations riches des données.

- **Prédiction :** Après l'entraînement du modèle, il peut être utilisé pour prédire de nouvelles relations entre les agriculteurs et les cultures agricoles. Les modèles tirent parti des informations contenues dans les graphes des filières agricoles et des produits animaux pour générer des prédictions en se basant sur les caractéristiques des entités et les relations apprises pendant l'apprentissage.

En résumé, notre système intelligent suit une architecture composée de prétraitement des données, de division des données, d'apprentissage de modèle CompgCN, et de prédiction des relations entre les agriculteurs et les cultures agricoles. Cette approche permet d'exploiter les informations des graphes et d'apprendre des représentations riches pour réaliser des prédictions précises.

# Chapitre 3 : Expérimentation

---

## 4. Implémentation

---

Cette partie est consacrée aux détails de l'environnement de développement et au langage de programmation utilisée pour la réalisation de notre système. Nous avons également présenté la base d'apprentissage et de test utilisée, ainsi que les détails de l'implémentation.

### A. Environnement de développement

#### • Google Drive :

Google Drive est un service de stockage en ligne proposé par Google. Il permet aux utilisateurs de stocker et de synchroniser leurs fichiers sur le Cloud, ce qui leur permet d'y accéder à partir de différents appareils et de les partager avec d'autres personnes.[22]

Principales fonctionnalités de Google Drive :

- **Stockage en ligne** : Google Drive offre un espace de stockage en ligne où les utilisateurs peuvent télécharger, stocker et organiser leurs fichiers, tels que des documents, des photos, des vidéos, des présentations, etc.
- **Synchronisation multiplateforme**: Les fichiers stockés sur Google Drive peuvent être synchronisés automatiquement avec différents appareils tels que des ordinateurs, des smartphones et des tablettes.
- **Partage de fichiers** : Les utilisateurs peuvent partager des fichiers et des dossiers spécifiques avec d'autres personnes en définissant les autorisations appropriées (lecture seule, modification, partage avec un lien, etc.).
- **Collaboration en temps réel**: Google Drive permet à plusieurs utilisateurs de travailler simultanément sur un même document, ce qui permet des modifications en temps réel.
- **Intégration avec d'autres services Google**: Google Drive est intégré à d'autres services Google tels que Google Docs, Google Sheets et Google Slides, ce qui permet de créer, éditer et collaborer sur des documents directement dans le navigateur, sans nécessiter de logiciel supplémentaire.
- **Reconnaissance optique de caractères (OCR)** : Google Drive utilise la technologie OCR pour reconnaître et extraire du texte à partir d'images et de fichiers PDF, ce qui facilite la recherche et l'édition de contenu.

# Chapitre 3 : Expérimentation

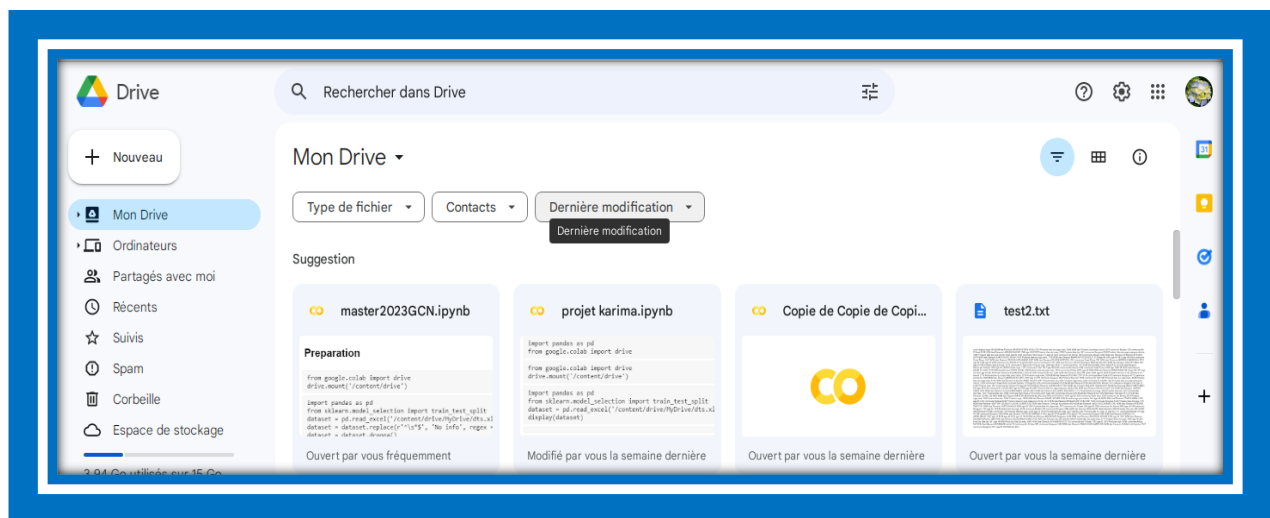


Figure 12. Google Drive

## • Google Colab

Le nom de "Colab", est une plateforme offerte par Google qui permet d'écrire et d'exécuter du code Python via un navigateur web. Basée sur Jupyter Notebook, elle est spécialement conçue pour la formation et la recherche en apprentissage automatique, Cette plateforme présente plusieurs avantages, notamment :

- **Amélioration des compétences en codage Python** : Colab offre un environnement interactif où vous pouvez écrire, exécuter et expérimenter du code Python. Cela permet aux utilisateurs de développer leurs compétences en programmation.
- **Développement d'applications en deep learning** : Colab prend en charge les bibliothèques populaires de deep learning telles que Keras, TensorFlow, PyTorch et OpenCV. Cela facilite le développement d'applications basées sur le deep learning en utilisant ces bibliothèques puissantes.
- **Environnement de développement prêt à l'emploi** : Colab utilise Jupyter Notebook, qui offre une interface conviviale pour écrire et exécuter du code. Il n'est pas nécessaire de configurer un environnement de développement, ce qui permet de gagner du temps et de se concentrer sur la tâche à accomplir.
- **Accès à un GPU gratuit** : L'une des fonctionnalités distinctives de Colab est l'accès gratuit à un processeur graphique (GPU). Cela permet d'accélérer les calculs intensifs et d'entraîner des modèles de machine learning plus rapidement.

# Chapitre 3 : Expérimentation

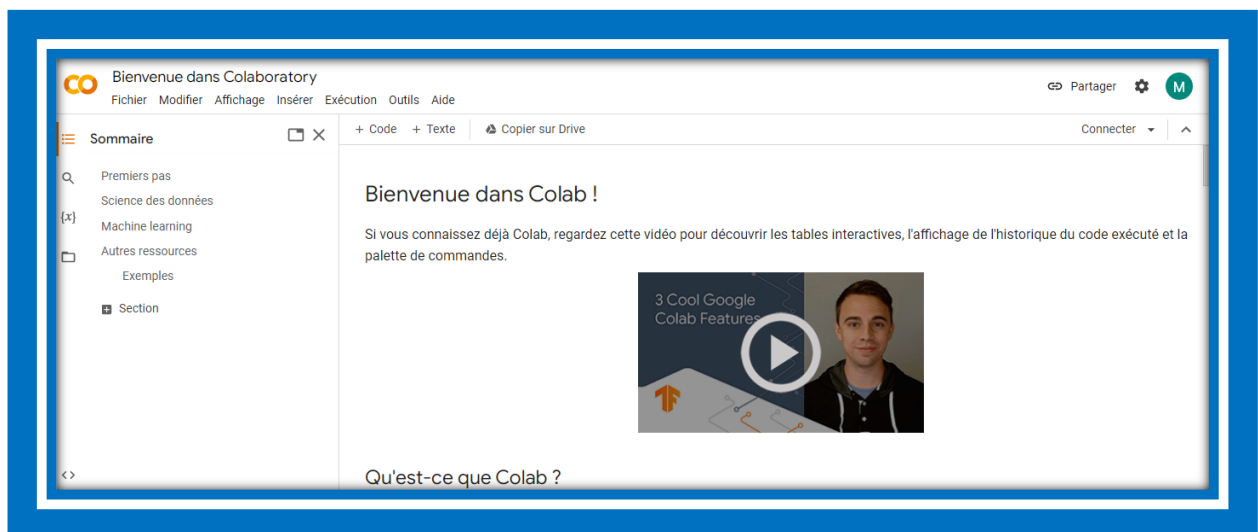


Figure 13. Google Colab

## B. Langage de programmation et bibliothèques

**Python :** Au fil des années, Python est devenu le langage de programmation le plus largement adopté par les informaticiens. Il s'est rapidement positionné en tête dans des domaines tels que la gestion d'infrastructure, l'analyse de données et le développement de logiciels.

Python bénéficie également d'une grande variété de bibliothèques et de modules qui couvrent une gamme étendue de fonctionnalités. Que ce soit pour le traitement des données, le développement web, l'apprentissage automatique ou la visualisation graphique, Python offre des outils puissants et prêts à l'emploi. [22]

### • Bibliothèques utilisées

- **Pandas :** Pandas est une bibliothèque open-source populaire en Python utilisée pour la manipulation et l'analyse des données. Elle fournit des structures de données et des fonctionnalités permettant de manipuler et de traiter des données tabulaires de manière efficace. [23]
- **Sklearn :** est une bibliothèque propose une variété de techniques pour le prétraitement des données, y compris la normalisation, la standardisation, le codage one-hot, le traitement des données manquantes, la réduction de dimension, etc. Ces techniques permettent de préparer les données avant de les utiliser pour la modélisation. [22]

# Chapitre 3 : Expérimentation

---

- **NumPy** : est une bibliothèque essentielle en Python pour le calcul scientifique et numérique. Elle fournit des structures de données puissantes pour les tableaux multidimensionnels et offre une vaste gamme de fonctions mathématiques pour effectuer des opérations sur ces tableaux. [24]

## C. Prétraitement des données

Pour effectuer les prétraitements sur les données avant de les utiliser dans votre modèle, vous pouvez suivre les étapes suivantes :

- **Conversion en triplets :**

- 1- Concernant la base des données végétales :**

Nous avons effectué une conversion de notre base de données initiale, qui comportait 5 colonnes et 5457 lignes, vers un format de triplets contenant 3 colonnes et 21828 lignes. Cette conversion a pour but de représenter chaque ligne de notre base de données sous la forme d'une relation entre deux entités distinctes.

Pour mieux illustrer cette conversion, prenons un exemple concret : supposons que notre base de données initiale contienne une ligne avec les colonnes "agriculteur", "cultive" et "blé dur". Avant la conversion, cette information était stockée dans une ligne unique avec les cinq colonnes correspondantes.

Après la conversion en triplets, cette information est représentée par quatre lignes distinctes, chacune contenant une relation entre deux entités.

# Chapitre 3 : Expérimentation

Ainsi, chaque ligne de notre base de données est maintenant représentée par une relation entre deux entités dans le format de triplets. Cette conversion a généré un total de 21828 lignes de triplets à partir de notre base de données initiale contenant 5457 lignes.

le prétraitement de la base passe par plusieurs étapes. Nous expliquerons ces étapes dans ce qui suit :

- Premièrement charge le fichier Excel à partir du chemin '/content/drive/MyDrive/dts.xlsx' et le stocke dans le DataFrame dataset.
- Ensuite, nous utilisons la méthode replace de pandas pour remplacer les valeurs vides ou contenant uniquement des espaces par la chaîne de caractères 'No info' dans tout le DataFrame. Cela permet de traiter les valeurs manquantes ou incorrectes dans le jeu de données.
- la fonction dataset = dataset.dropna(), supprime les lignes contenant des valeurs manquantes dans le DataFrame dataset.
- print(dataset.iat[3292,4]) affiche la valeur située à la ligne 3292 et à la colonne 4 du DataFrame. Cela peut être utile pour vérifier une valeur spécifique dans les données.
- la fonction display(dataset) affiche le DataFrame dataset dans une représentation tabulaire, permettant d'examiner les données et leurs attributs.

Ces étapes de prétraitement des données permettent de charger les données à partir d'un fichier Excel, de remplacer les valeurs manquantes ou incorrectes, de supprimer les lignes contenant des valeurs manquantes et d'afficher les données pour l'inspection.

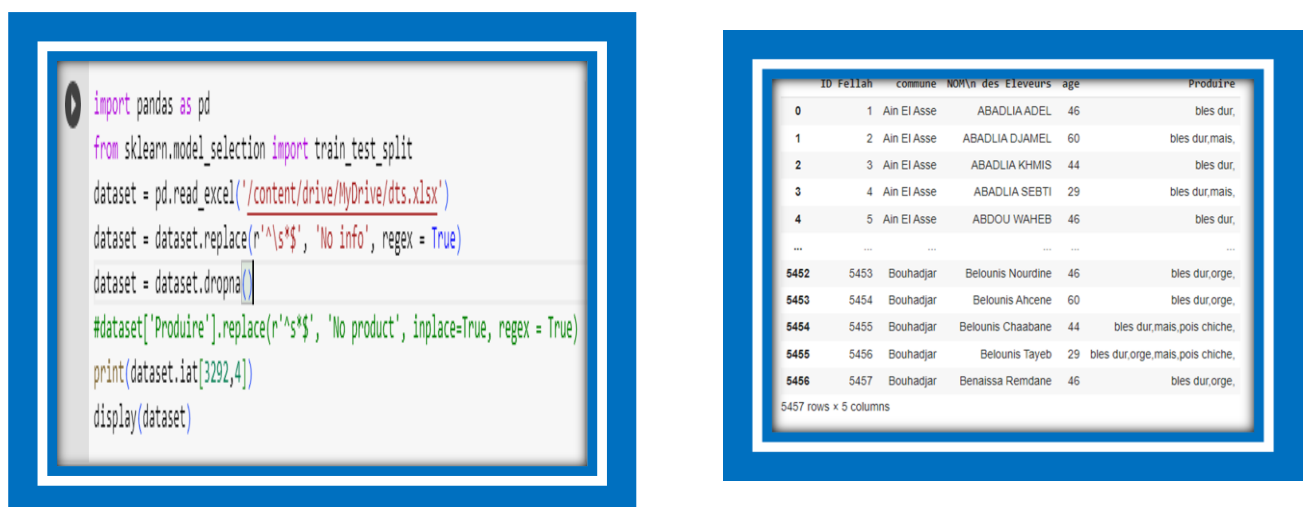


Figure 14. Le code d'identification de la base de données végétal

## Chapitre 3 : Expérimentation

---

- Préparer les données pour l'apprentissage automatique en créant les listes des entités, des relations et des arêtes, en les regroupant dans un DataFrame, en remplissant les valeurs manquantes et en divisant les données en ensembles d'entraînement, de validation et de test :
  - Les listes `entity1`, `entity2`, et `rel` sont initialement vides. Nous allons les remplir avec les valeurs des colonnes du DataFrame `dataset`.

À l'aide d'une boucle `for`, nous itérons sur les colonnes du DataFrame `dataset` en excluant la première colonne (car elle contient les ID des Fellahs). Pour chaque colonne, nous ajoutons les valeurs de la colonne "ID Fella" à la liste `entity1`, les valeurs de la colonne en cours à la liste `entity2`, et le nom de la colonne (remplaçant les sauts de ligne par des espaces) à la liste `rel`.

- Nous créons un dictionnaire `dict1` avec les clés "node" (contenant `entity1`), "relation" (contenant `rel`), et "edge" (contenant `entity2`).

Nous créons un nouveau DataFrame `ds` à partir du dictionnaire `dict1`.

- La fonction `display(ds)` affiche le DataFrame `ds` dans une représentation tabulaire, permettant d'examiner les données nouvellement préparées.
- Nous utilisons la méthode `fillna(0)` pour remplacer les valeurs manquantes (NaN) par des zéros dans le DataFrame `ds`.
- Nous utilisons la fonction `train_test_split` pour diviser le DataFrame `ds` en ensembles d'entraînement (`train`) et de test (`test`). Ici, nous spécifions que 20% des données seront utilisées pour l'ensemble de test.
- Nous utilisons à nouveau la fonction `train_test_split`, cette fois-ci pour diviser l'ensemble d'entraînement (`train`) en ensembles d'entraînement (`train`) et de validation (`val`). Ici, nous spécifions que 15% des données de l'ensemble d'entraînement seront utilisées pour l'ensemble de validation.

# Chapitre 3 : Expérimentation

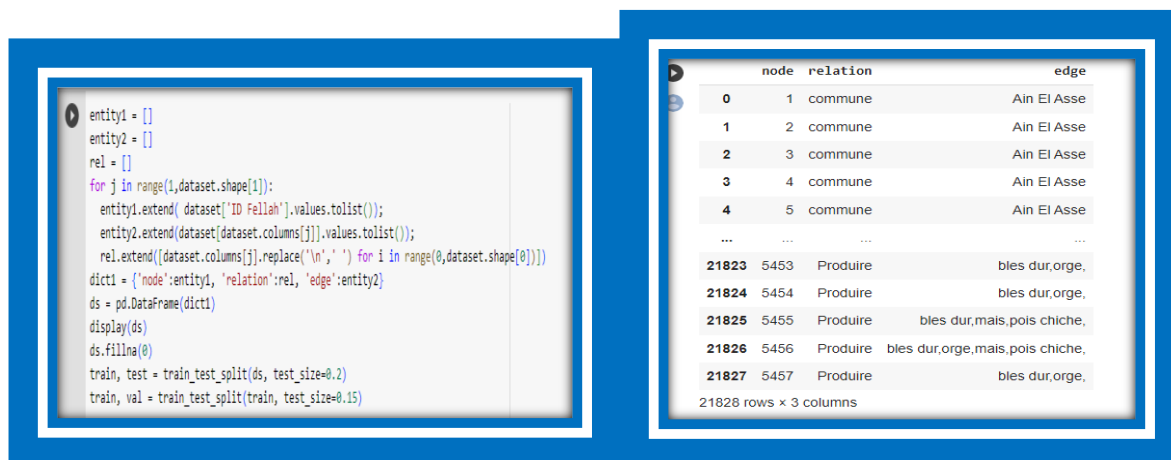


Figure 15. dataset sous forme node- relation-edge

- Dernière étapes de prétraitement des données : Les DataFrames train, val et test sont enregistrés au format CSV en utilisant les méthodes to\_csv. Les fichiers de sortie sont nommés 'train.txt', 'valid.txt' et 'test.txt' respectivement. Les données sont séparées par des tabulations ('\t') et l'index des lignes n'est pas inclus dans les fichiers CSV.
- Une chaîne de caractères resLines est initialisée pour stocker les lignes résultantes.
- Une boucle for est utilisée pour itérer sur les fichiers 'train.txt', 'test.txt' et 'valid.txt'.
- Pour chaque fichier, nous ouvrons le fichier en mode lecture et écriture ('r+'). Nous lisons ensuite toutes les lignes du fichier avec readlines().
- Nous itérons sur chaque ligne du fichier et vérifions si elle est composée de trois éléments séparés par des tabulations. Si c'est le cas, nous ajoutons cette ligne à la chaîne de caractères resLines.

Finalement, nous réécrivons le contenu du fichier avec f.write(resLines). Cela écrase le contenu existant du fichier avec les lignes filtrées que nous avons extraites.

Ces étapes permettent de filtrer les lignes qui contiennent exactement trois éléments séparés par des tabulations dans les fichiers 'train.txt', 'test.txt' et 'valid.txt'.

# Chapitre 3 : Expérimentation

---

```
train.to_csv('train.txt', sep='\t', index=False)
val.to_csv('valid.txt', sep='\t', index=False)
test.to_csv('test.txt', sep='\t', index=False)
resLines = ''
for ff in ['train', 'test', 'valid']:
    with open(ff+'.txt', 'r+') as f:
        tlines = f.readlines()
        #print(tlines)
        for line in tlines:
            #print(line.split('\t'))
            if len(line.split('\t'))==3:
                resLines = resLines+line
            f.write(resLines)
```

Figure 16. Transformer dataset sous forme text

## 2 -Concernant la base des données animales :

Nous avons effectué une conversion de notre base de données initiale, qui comportait 5 colonnes et 1251 lignes, vers un format de triplets contenant 3 colonnes et 5004 lignes. Pour illustrer cette conversion, prenons un exemple concret : l'entité "ID" avec la relation "a une relation" vers l'objet "Ain Assel". Avant la conversion, cette information aurait été stockée dans une ligne de notre base de données avec les cinq colonnes correspondantes. Après la conversion en triplets, cette information est représentée par trois colonnes distinctes dans une ligne dédiée :

Ainsi, nous obtenons une représentation plus compacte et structurée de nos données d'origine, où chaque triplet correspond à une relation entre une entité, une relation et un objet dans notre graphe de connaissance. Dans notre cas, cette conversion a permis de générer un total de 5004 lignes de triplets à partir de notre base de données initiale contenant 1251 lignes.

Les étapes et les fonctions utilisées pour le prétraitement des données pour la dataset animal sont les même.

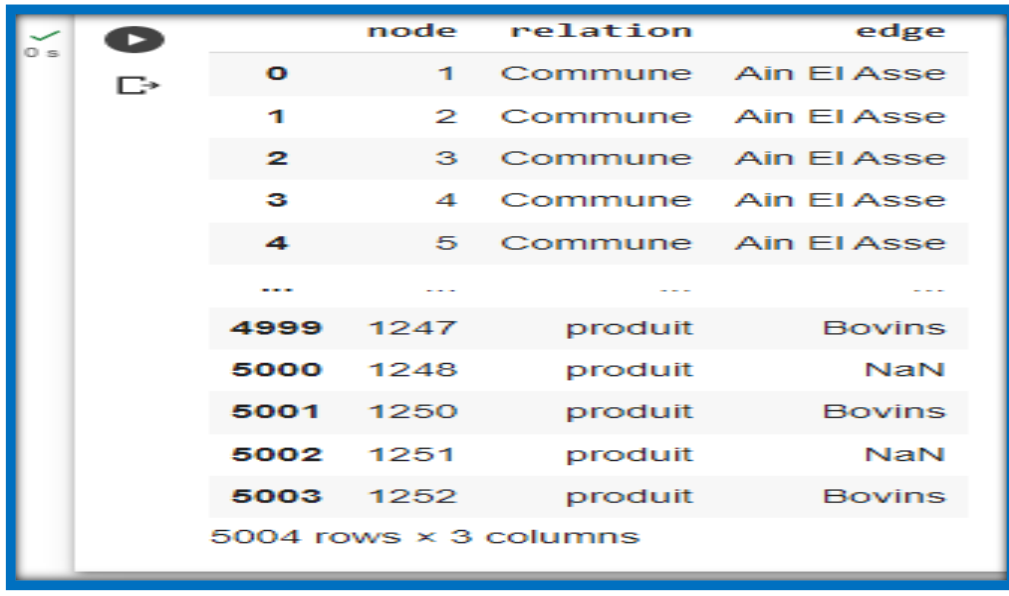
# Chapitre 3 : Expérimentation



	ID	Commune	Nom fellah	Nbre	produit
0	1	Ain El Asse	ABADLIA ADEL	7.0	Bovins
1	2	Ain El Asse	ABADLIA DJAMEL	6.0	Bovins
2	3	Ain El Asse	ABADLIA KHMIS	8.0	Bovins
3	4	Ain El Asse	ABADLIA SEBTI	6.0	Bovins
4	5	Ain El Asse	ABDOU WAHEB	12.0	ovins
...	...	...	...	...	...
1246	1247	Bougous	ZOUINI Boudjemaa	4.0	Bovins
1247	1248	Bougous	ZOUINI EL ARBI	0.0	NaN
1248	1250	El Tarf	ATI Khaled	8.0	Bovins
1249	1251	El Tarf	ABADA AMAR	0.0	NaN
1250	1252	El Tarf	ABASSI Khelifa	6.0	Bovins

1251 rows x 5 columns

Figure 17. affichage dataset animal



	node	relation	edge
0	1	Commune	Ain El Asse
1	2	Commune	Ain El Asse
2	3	Commune	Ain El Asse
3	4	Commune	Ain El Asse
4	5	Commune	Ain El Asse
...	...	...	...
4999	1247	produit	Bovins
5000	1248	produit	NaN
5001	1250	produit	Bovins
5002	1251	produit	NaN
5003	1252	produit	Bovins

5004 rows x 3 columns

Figure 18. dataset animal sous forme node- relation-edge

# Chapitre 3 : Expérimentation

---

Une fois que vous avez effectué ces prétraitements sur vos données, elles sont prêtes à être utilisées dans notre modèle d'apprentissage CompGCN. nous pouvons appliquer notre modèle, tel qu'un Graph Neural Network (GNN), sur les données d'apprentissage, puis l'utiliser pour faire des prédictions sur les données de test ou de validation.

## D. Phase d'apprentissage

### 1- Concernant la base des données végétales :

Après avoir prétraité la base de données, les données sont alimentées dans le modèle **COMPGCN** pour l'apprentissage :

- Les commandes `pip install ordered_set`, `pip install torch_scatter`, `pip install dataloader`, `pip install scikit_learn` semblent être des commandes d'installation de paquets Python à l'aide de l'outil pip. Chacune de ces commandes installe un paquet spécifique (respectivement `ordered_set`, `torch_scatter`, `dataloader`, `scikit_learn`) qui peuvent être des dépendances requises par le code contenu dans le projet CompGCN-master.
- la commande `ordered_set` : cette commande permet d'utiliser des ensembles ordonnés qui maintient une collection d'éléments uniques dans un ordre spécifique .
- `torch_scatter` est une fonction utilisée pour effectuer des opérations de réduction ou d'agrégation sur des données dispersées, où des valeurs sont réparties sur différentes positions dans un tenseur. elle prend trois arguments principaux : `input`, `Index`, `dim`
- Le `DataLoader` est une fonction utilisé pour faciliter le chargement et le traitement des données lors de l'entraînement des modèles d'apprentissage automatique.
- `scikit-learn` : est une fonction utilisé pour l'encodage des variables catégorielles
-

# Chapitre 3 : Expérimentation



```
!cp /content/train.txt /content/drive/MyDrive/CompGCN-master/CompGCN-master/data/FB15k-237/
!cp /content/test.txt /content/drive/MyDrive/CompGCN-master/CompGCN-master/data/FB15k-237/
!cp /content/valid.txt /content/drive/MyDrive/CompGCN-master/CompGCN-master/data/FB15k-237/

!cp '/content/drive/MyDrive/CompGCN-master/run.py' /content/CompGCN-master/CompGCN-master/
#pip install ordered_set
#pip install torch_scatter
#pip install data_loader
#pip install scikit_learn
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.org/downloads/public/simple/
Collecting ordered_set
  Downloading ordered_set-4.1.0-py3-none-any.whl (7.6 kB)
Installing collected packages: ordered_set
Successfully installed ordered_set-4.1.0
Looking in indexes: https://pypi.org/simple, https://us-python.org/downloads/public/simple/
Collecting torch_scatter
  Downloading torch_scatter-2.1.1.tar.gz (187 kB)
----- 187.6/187.6 kB 9.7 MB/s eta 0:00:00
Preparing metadata (setup.py) ... done
Building wheels for collected packages: torch_scatter
Building wheel for torch_scatter (setup.py) ... done
Created wheel for torch_scatter: filename=torch_scatter-2.1.1-cp310-cp310-linux_x86_64.whl size=3512354 sha256=86520177895c6e888f73d0c185495048f17fde488
Stored in directory: /root/.cache/pip/wheels/ef/70/50/656a6b61c8eef2a0c2c324c8056f255508c9f831976f
Successfully built torch_scatter
Installing collected packages: torch_scatter
Successfully installed torch_scatter-2.1.1
Looking in indexes: https://pypi.org/simple, https://us-python.org/downloads/public/simple/
Collecting data_loader
  Downloading data_loader-3.8.tar.gz (9.1 kB)
Preparing metadata (setup.py) ... done
Building wheels for collected packages: data_loader
Building wheel for data_loader (setup.py) ... done
```

Figure 19. Commande d'installations

Pour avoir les meilleurs paramètres qui conduisent aux bons résultats nous avons relancé l'apprentissage plusieurs fois. Les paramètres principaux a fixé pour l'apprentissage sont :

- **Score\_func transe** : La fonction de score dans le modèle TransE est utilisée pour calculer une mesure de similarité entre les entités et les relations dans un modèle de représentation de connaissances basé sur les triplets (sujet, relation, objet). Le modèle TransE vise à apprendre des représentations vectorielles pour les entités et les relations de sorte que la distance entre l'entité sujet et l'entité objet, après avoir été transformée par la relation correspondante, soit minimale dans l'espace vectoriel.

Dans TransE, la fonction de score est définie comme suit :

$$\text{score}(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$$

- **Opn sub** : cette fonction permet de combiner les emédings des entités et des relations dans le modele avec « Sub », les embeddings des entités et des relations soustrait l'un des autres.
- **Gamma** : paramètre de contrôle calculer la similarité entre les embeddings des triples d'entité.
- **Hid drop** : c'est le taux de dropout (réduire le surapprentissage), l'activation mettre à 0.
- **Init\_dim** : c'est la dimension initiale = 200.

# Chapitre 3 : Expérimentation

---

- **batch-size** : Le batch size est le nombre d'échantillons utilisés lors d'une itération dans l'entraînement d'un modèle. Il joue un rôle important dans la stabilité de l'estimation du gradient basée sur l'ensemble de données complet.
- **Epoch** : Le nombre d'epochs est souvent utilisé comme critère d'arrêt lors de l'entraînement d'un modèle. Fixer le nombre d'epochs à **5** signifie que le modèle sera entraîné en effectuant 5 itérations complètes sur l'ensemble de données. Après chaque epoch, les performances du modèle peuvent être évaluées sur un ensemble de validation pour surveiller son apprentissage et sa convergence.

## 5. Résultats et Discussion

---

Après l'apprentissage du modèle, nous avons utilisé les données de test (végétales et animaux) pour obtenir des données prédites, Le Tableau (2) montre les résultats obtenus et la Figure 4.5 montre la perte de notre modèle pendant l'entraînement et la validation. les résultats de chaque époque d'entraînement, l'époque (E:0, E:1, etc.) suivi par le numéro d'itération (0, 100, etc.). Les informations affichées comprennent la perte d'entraînement (Train Loss) et le MRR (Mean Reciprocal Rank) sur l'ensemble de validation (Val MRR).

**MRR (Mean Reciprocal Rank)** est une mesure d'évaluation utilisée pour évaluer l'ordre de pertinence des résultats dans les tâches de classement. Il est calculé en prenant la moyenne des inverses des rangs des éléments pertinents dans les listes des résultats. Un MRR élevé indique une meilleure qualité de classement des résultats pertinents.

Le MRR dans ce modèle est stable : Moy = 0.00503.

Dans ce modèle, le MRR (Reciprocal Rank moyen) est de 0.00503. Cela indique que, en moyenne, la position du premier résultat correct dans la liste des prédictions est de 0.00503. Un MRR plus élevé serait préférable, car cela signifierait que les résultats corrects sont généralement mieux classés.

## Chapitre 3 : Expérimentation

---

Datasets	Epoche	Loss
Dataset filière agricole végétal	E 0	0.012084
	E 1	0.01208
	E 2	0.012081
	E 3	0.012086
	E 4	0.12089
Dataset filière agricole animal	E 0	0.012084
	E 1	0.01208
	E 2	0.012081
	E 3	0.012086
	E 4	0.12089

**Tableau 2.** Affichage résultats des datasets végétal et animal

Remarque : Les valeurs obtenues pour les dataset végétal et animal sont les mêmes, car les données (ID fellah, nom) sont identiques dans les deux ensembles de données.

# Chapitre 3 : Expérimentation



```
'opn': 'sub',
'restore': False,
'score_func': 'transe',
'seed': 41504}

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:560: UserWarning: This DataLoader will create 10 worker processes in total. Our suggested
warnings.warn(_create_warning_msg(
2023-06-11 02:59:07,656 - [INFO] - [E:0] 0]: Train Loss:0.01182, Val MRR:0.0   testrun_11_06_2023_02:58:52
2023-06-11 03:12:41,905 - [INFO] - [E:0] 100]: Train Loss:0.012131, Val MRR:0.0   testrun_11_06_2023_02:58:52
2023-06-11 03:18:29,467 - [INFO] - [Epoch:0]: Training Loss:0.01208

/content/CompGCN-master/CompGCN-master/run.py:319: UserWarning: where received a uint8 condition tensor. This behavior is deprecated and will be removed in a fu
pred
= torch.where(label.byte(), -torch.ones_like(pred) * 10000000, pred)
2023-06-11 03:18:33,909 - [INFO] - [Valid, Tail_Batch Step 0]   testrun_11_06_2023_02:58:52
2023-06-11 03:19:54,138 - [INFO] - [Valid, Head_Batch Step 0]   testrun_11_06_2023_02:58:52
2023-06-11 03:20:50,717 - [INFO] - [Epoch 0 valid]: MRR: Tail : 0.00022, Head : 0.00021, Avg : 0.00022
2023-06-11 03:20:58,781 - [INFO] - [Epoch 0]: Training Loss: 0.012077, Valid MRR: 0.00022

2023-06-11 03:21:11,249 - [INFO] - [E:1] 0]: Train Loss:0.0094291, Val MRR:0.00022   testrun_11_06_2023_02:58:52
2023-06-11 03:24:27,504 - [INFO] - [E:1] 100]: Train Loss:0.01208, Val MRR:0.00022   testrun_11_06_2023_02:58:52
```

Figure 20. Résultat de l'apprentissage

# Conclusion et Perspectives

---

Dans le cadre de ce travail, nous avons abordé l'importance de l'apprentissage profond pour la prédiction des liens dans les Bases de Connaissances, L'utilisation de techniques d'apprentissage profond permet d'améliorer la compréhension des données et d'obtenir de nouvelles connaissances à partir des informations déjà disponibles.

Pour mieux comprendre notre sujet, nous avons structuré notre travail en trois chapitres. Le premier chapitre offre une vue d'ensemble de l'apprentissage profond, en expliquant les concepts et les techniques clés. Le deuxième chapitre se concentre sur les graphes de connaissances et la prédiction des liens dans ces graphes, en mettant en avant le modèle choisi, CompGcn, et en réalisant une étude approfondie sur ce modèle. Ensuite, nous passons à l'implémentation et à l'évaluation des modèles proposés. Nous présentons les bibliothèques Python utilisées, les prétraitements appliqués aux données immobilières, ainsi que la création et l'entraînement des trois modèles proposés, suivis de leur évaluation pour obtenir des résultats significatifs.

Comme perspective de ce travail :

En collectant des données exhaustives sur les agriculteurs et leur région, y compris des informations détaillées sur les pratiques agricoles, les cultures cultivées et les conditions environnementales, il devient possible d'identifier les zones qui ont le potentiel de produire une qualité de produits agricoles jamais atteinte auparavant. En particulier, en obtenant des données précises sur la qualité du sol grâce à des analyses approfondies, il devient possible de déterminer les caractéristiques du sol qui favorisent une production optimale. Ces informations permettent aux agriculteurs de prendre des décisions éclairées sur les techniques de culture, les amendements du sol et la sélection des cultures, ce qui peut conduire à des rendements plus élevés et à des produits de meilleure qualité.

# Références

---

## A. Références Bibliographiques

- [1] Greener, JG, Kandathil, SM, Moffat, L. et Jones, DT (2022). Un guide d'apprentissage automatique pour les biologistes. *Nature Reviews Molecular Cell Biology* , 23 (1), 40-55.
- [2] Jeske, M., Müller, C. W., & Ephrussi, A. (2017). The LOTUS domain is a conserved DEAD-box RNA helicase regulator essential for the recruitment of Vasa to the germ plasm and nuage. *Genes & development*, 31(9), 939-952.
- [3] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- [4] Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., ... & Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*.
- [5] Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6), 26-38.
- [6] Hon, M., & Khan, N. M. (2017, November). Towards Alzheimer's disease classification through transfer learning. In *2017 IEEE International conference on bioinformatics and biomedicine (BIBM)* (pp. 1166-1169). IEEE.
- [7] Galsky, M. D., Saci, A., Szabo, P. M., Han, G. C., Grossfeld, G., Collette, S., ... & Sharma, P. (2020). Nivolumab in Patients with Advanced Platinum-resistant Urothelial Carcinoma: Efficacy, Safety, and Biomarker Analyses with Extended Follow-up from CheckMate 275CheckMate 275 Extended Follow-up and Biomarker Analyses. *Clinical Cancer Research*, 26(19), 5120-5128.
- [8] Khan, S. A. R., Yu, Z., Belhadi, A., & Mardani, A. (2020). Investigating the effects of renewable energy on international trade and environmental quality. *Journal of Environmental management*, 272, 111089.
- [9] Adimalla, N., Li, P., & Qian, H. (2018). Evaluation of groundwater contamination for fluoride and nitrate in semi-arid region of Nirmal Province, South India: a special emphasis on human health risk assessment (HHRA). *Human and ecological risk assessment: an international*
- [10] Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual understanding of convolutional neural network-a deep learning approach. *Procedia computer science*, 132, 679-688.
- [11] Patterson, J., & Gibson, A. (2017). *Deep learning: A practitioner's approach*. " O'Reilly Media, Inc."  
  
LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- [12] Mittal, S. (2019). *Explainable AI: Interpreting, Explaining, and Visualizing Deep Learning*. Apress.
- [13] Alexandre faure, la base de connaissances element central de la gestion des collections électroniques des bibliotheques universitaires, submitted on 12 FEB 2014, P 35-36.

- [14] uksg kbart. kbart: bases de connaissances et outils associés/rapport préparé par le groupe de travail niso/uksg kbart janvier 2010 [en ligne]. [s.l.] : united kingdom serials group, [s.d].
- [15] H. Paulheim, "Knowledge Graphs: Structure, Use, and Dynamics", Synthesis Lectures on the Semantic Web: Theory and Technology, 2017
- [16] Xu, B., Shen, H., Cao, Q., Cen, K., & Cheng, X. (2020). Graph convolutional networks using heat kernel for semi-supervised learning. *arXiv preprint arXiv:2007.16002*.
- [17] zhanfang zhao. architecture of knowledge graph construction techniques international journal of pure and applied mathematics/volume 118 no. 19 2018,1869-1883.
- [18] Malinouski, M., Hasan, N. M., Zhang, Y., Seravalli, J., Lin, J., Avanesov, A., ... & Gladyshev, V. N. (2014). Genome-wide RNAi ionomics screen reveals new genes and regulation of human trace element metabolism. *Nature communications*, 5(1), 3301.
- [19] Bastings, J., Titov, I., Aziz, W., Marcheggiani, D., & Sima'an, K. (2017). Graph convolutional encoders for syntax-aware neural machine translation. *arXiv preprint*
- [20] Nolan, C. J., & Prentki, M. (2019). Insulin resistance and insulin hypersecretion in the metabolic syndrome and type 2 diabetes: Time for a conceptual framework shift. *Diabetes and Vascular Disease Research*, 16(2), 118-127.
- [21] Kipf, T., Fetaya, E., Wang, K. C., Welling, M., & Zemel, R. (2018, July). Neural relational inference for interacting systems. In *International conference on machine learning* (pp. 2688-2697). PMLR.

## B. Références Web (Techniques)

- [22] <https://scikit-learn.org/stable/index.html>
- [23] <https://pandas.pydata.org/>
- [24] <https://numpy.org/>