

# Programmation en IA



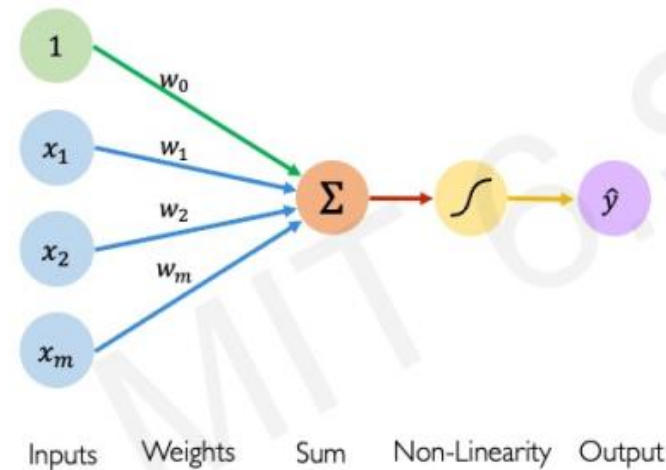
- Dispensé par **MWAMBA KASONGO Dahouda**
- Docteur en génie logiciel et systèmes d'information
- Machine and Deep Learning Engineer

- Assisté par Ass. **Daniel MBAYA**

Heure : 8H00 – 12H00

- E-mail : [dahouda37@gmail.com](mailto:dahouda37@gmail.com)
- Tel.: +243 99 66 55 265

## The Perceptron: Forward Propagation

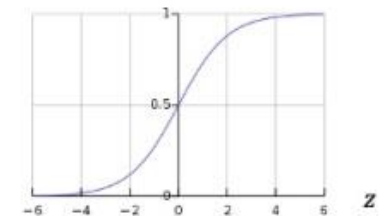


### Activation Functions

$$\hat{y} = g(w_0 + \mathbf{X}^T \mathbf{W})$$

- Example: sigmoid function

$$g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$



## PLAN DU COURS



## CHAPITRE 1 Introduction a l'Intelligence Artificielle

### 1.1 Généralités

- Définition
- Branches de l'Intelligence Artificielle
- Application de l'Intelligence Artificielle

### 1.2. Langages de programmation en Intelligence Artificielle

### 1.3 Python pour l'IA

- Python Fundamentals
- Bibliothèques Python pour l'Intelligence Artificielle

## PLAN DU COURS

### CHAPITRE 2 Machine Learning [Apprentissage Automatique]

#### 1.1 Qu'est-ce que Machine Learning?

#### 1.2 Définition de Machine Learning

#### 1.3 Différences entre l'IA, Machine Learning et Deep Learning

#### 1.4 Exemples concrets du Machine Learning (par exemple, prediction de prix des immobilier)

#### 1.5. Types de Machine Learning

##### 1.5.1 Apprentissage supervise

##### 1.5.2 Apprentissage non supervisé

##### 1.5.3 Apprentissage par renforcement (juste une introduction)



## PLAN DU COURS



## CHAPITRE 3 Apprentissage Profond [Deep Learning]

### 3.1. Introduction aux réseaux de neurones [Neural Networks] Natural Language Processing (NLP)

- Concepts de base
- Function d'activation
- Réseaux de neurons profonds

### 3.2. Réseaux de neurones convolutifs [Convolutional Neural Network (CNN)]

### 3.3. Réseaux de neurones récurrents [Recurrent Neural Network (RNN)]

### 3.4. Variant de RNN : Long Term Short Memory (LSTM)

## CHAPITRE 4 Introduction au Traitement du Langage Naturel [Natural Language Processing (NLP)]

## OBJECTIF DU COURS



Les objectifs généraux du cours de programmation en Intelligence Artificielle visent à fournir aux étudiants une compréhension approfondie des concepts de Machine Learning avec un focus sur le Deep Learning, techniques et applications des réseaux de neurones artificiels [Artificial Neural Netowrk].

Les étudiants devraient être capables de :

- Comprendre les principes fondamentaux des réseaux de neurones : Apprendre comment fonctionnent les réseaux de neurones artificiels et les types d'architectures.
- Maîtriser les techniques d'entraînement de modèles profonds : Connaître les méthodes d'optimisation comme Adam pour éviter le sur apprentissage.
- Construire des réseaux de neurones complexes : Apprendre à concevoir, implémenter et entraîner des architectures de réseaux profonds adaptées à des problèmes spécifiques, tels que la classification d'images.

## OBJECTIF DU COURS



Les objectifs spécifiques sont:

- Comprendre les techniques de transfert d'apprentissage,
- Apprendre à mesurer la performance des modèles de Deep Learning avec des métriques appropriées (Accuracy, précision, rappel, F1-score) et ajuster les hyperparamètres pour améliorer les résultats.
- Maîtriser les Framework de deep learning (TensorFlow et Keras) pour la conception de réseaux de neurones.
- Comprendre les architectures avancées : Apprendre les architectures de réseaux plus complexes comme les réseaux convolutifs (CNN) pour les images, les réseaux récurrents (RNN) et LSTM pour les données séquentielles.
- Utiliser des bibliothèques et Framework avancés : Développer des compétences dans l'utilisation de bibliothèques Python comme Scikit-learn, TensorFlow, et Keras pour concevoir et entraîner des modèles de Machine Learning.

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.1 Généralités

- ❑ Dans le monde d'aujourd'hui, la technologie évolue très rapidement et nous entrons en contact avec différentes nouvelles technologies de jour en jour.

De nos jours, l'une des technologies en plein essor de l'informatique est l'intelligence artificielle qui est prête à créer une nouvelle révolution dans le monde en créant des machines intelligentes.

L'IA est l'un des domaines fascinants et universels de l'informatique qui a un grand avenir. L'IA a tendance à faire fonctionner une machine comme un humain.

#### 1.1.1. Définition

L'intelligence artificielle est composée de deux mots, artificiel et intelligence, où artificiel signifie « fabriqué par l'homme » et intelligence définit « pouvoir de réflexion », d'où le fait que l'IA signifie « un pouvoir de réflexion fabriqué par l'homme ».

Nous pouvons donc définir l'IA comme :

« C'est une branche de l'informatique par laquelle nous pouvons créer des machines intelligentes qui peuvent se comporter comme un humain, penser comme un humain et être capables de prendre des décisions. »

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



- ❑ Le chapitre 1, **Introduction à l'Intelligence Artificielle**, explore les bases de l'IA en commençant par une définition générale, ses principales branches et ses nombreuses applications dans divers domaines.
- ❑ Il aborde ensuite les langages de programmation utilisés en IA, avec un focus particulier sur Python, considéré comme incontournable grâce à sa simplicité et sa puissance.
- ❑ Les concepts fondamentaux de Python y sont présentés, ainsi que ses bibliothèques phares pour l'IA, telles que TensorFlow, NumPy, et Scikit-learn, qui facilitent le développement d'algorithmes d'apprentissage automatique et d'autres projets en IA.



## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.1 Généralités

#### 1.1.2. Why Artificial Intelligence?

- ☐ Avant d'en apprendre davantage sur l'intelligence artificielle, nous devons savoir quelle est l'importance de l'IA et pourquoi nous devrions l'apprendre. Voici quelques-unes des principales raisons d'en apprendre davantage sur l'IA :
- ✓ Avec l'aide de l'IA, vous pouvez créer des logiciels ou des appareils capables de résoudre des problèmes du monde réel très facilement et avec précision, tels que des problèmes de santé, de marketing, de circulation, etc.
- ✓ Avec l'aide de l'IA, vous pouvez créer votre assistant virtuel personnel, tel que Cortana, Google Assistant, Siri, etc.
- ✓ Avec l'aide de l'IA, vous pouvez construire des robots capables de travailler dans un environnement où la survie des humains peut être menacée.
- ☐ L'IA ouvre la voie à d'autres nouvelles technologies, de nouveaux appareils et de nouvelles opportunités.

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE

### 1.1 Généralités

#### 1.1.3. Objectifs de l'intelligence artificielle

Voici les principaux objectifs de l'intelligence artificielle :

- Reproduire l'intelligence humaine
- Résoudre des tâches à forte intensité de connaissances
- Établir une connexion intelligente entre perception et action
- Construire une machine capable d'effectuer des tâches nécessitant l'intelligence humaine, telles que :
  - ✓ Prouver un théorème
  - ✓ Jouer aux échecs
  - ✓ Planifier une opération chirurgicale
  - ✓ Conduire une voiture dans la circulation
- Créer un système capable de présenter un comportement intelligent, d'apprendre de nouvelles choses par lui-même, de démontrer, d'expliquer et de conseiller son utilisateur.



## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.1 Généralités

#### 1.1.4. Histoire de l'IA

- ❑ Tout au long de l'histoire, l'idée de rendre intelligents les objets non vivants a toujours intrigué les gens.
- ❖ Dans l'Antiquité, les récits grecs mentionnaient des dieux créant des machines intelligentes, et en Égypte, des ingénieurs faisaient bouger des statues.
- ❖ Des penseurs comme Aristote et Ramon Llull ont posé les bases de l'IA en décrivant le fonctionnement de la pensée humaine à l'aide de symboles.
- ❖ À la fin des années 1800 et au début des années 1900, l'informatique moderne a commencé à prendre forme.
- ❖ Charles Babbage et Ada Lovelace ont conçu des machines programmables dans les années 1830.
- ❖ Dans les années 1940, John Von Neumann a eu l'idée de stocker des programmes informatiques.
- ❖ Au même moment, Warren McCulloch et Walter Pitts ont commencé à poser les bases des réseaux neuronaux.

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.1 Généralités

#### 1.1.4. Histoire de l'IA

- ☐ Les années 1950 nous ont apporté les ordinateurs modernes, permettant aux scientifiques de se plonger dans l'intelligence artificielle.
- ☐ Le test de Turing d'Alan Turing est devenu un enjeu majeur dans l'intelligence informatique.
- ☐ Le terme « intelligence artificielle » a été utilisé pour la première fois lors d'une réunion au Dartmouth College en 1956, où le premier programme d'IA, Logic Theorist, a été présenté.
- ☐ Les années suivantes ont été marquées par des hauts et des bas pour l'IA, appelés « hivers de l'IA ».
- ☐ Dans les années 1970 et 1980, nous avons atteint les limites de la puissance et de la complexité des ordinateurs.
- ☐ Mais à la fin des années 1990, les choses sont redevenues passionnantes. Les ordinateurs étaient plus rapides et les données étaient plus abondantes.
- ☐ La victoire de Deep Blue d'IBM sur le champion d'échecs Garry Kasparov en 1997 a été un grand moment.

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.1 Généralités

#### 1.1.4. Histoire de l'IA

- ❑ Les années 2000 ont marqué le début d'une nouvelle ère avec l'apprentissage automatique, le traitement du langage et la vision par ordinateur. Cela a donné lieu à de nouveaux produits et services intéressants.
- ❑ Les années 2010 ont vu l'IA décoller avec des choses comme les assistants vocaux et les voitures autonomes.
- ❑ L'IA générative, qui permet de créer des choses créatives, a également commencé à prendre de l'ampleur.
- ❑ Dans les années 2020, l'IA générative comme ChatGPT-3 et Bard de Google a attiré l'attention de tous.

Ces modèles peuvent créer toutes sortes de choses nouvelles lorsque vous leur donnez une consigne, comme des essais ou des œuvres d'art.

Mais n'oubliez pas que cette technologie est encore nouvelle et qu'il y a des choses à corriger, comme s'assurer qu'elle n'invente rien.

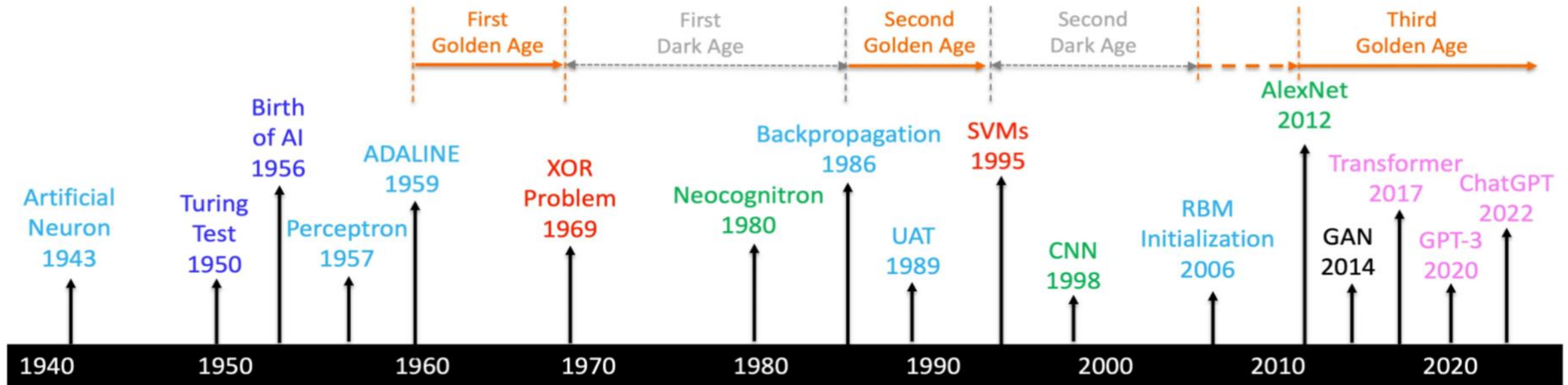
# Programmation en Intelligence Artificielle

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE

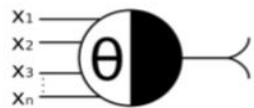


M

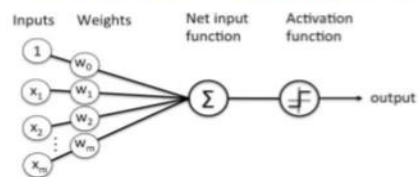
### A Brief History of AI with Deep Learning



McCulloch-Pitts

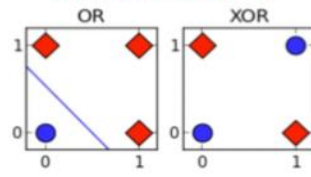


Rosenblatt

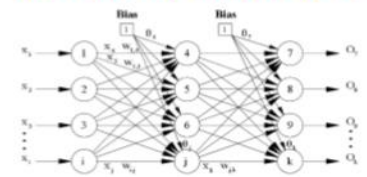


Widrow-Hoff

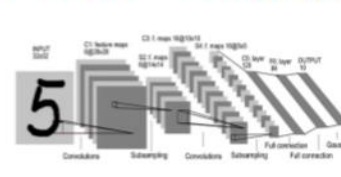
Minsky-Papert



Rumelhart, Hinton et al.



LeCun



Hinton-Ruslan Krizhevsky et al.



Vaswani



# Programmation en Intelligence Artificielle

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



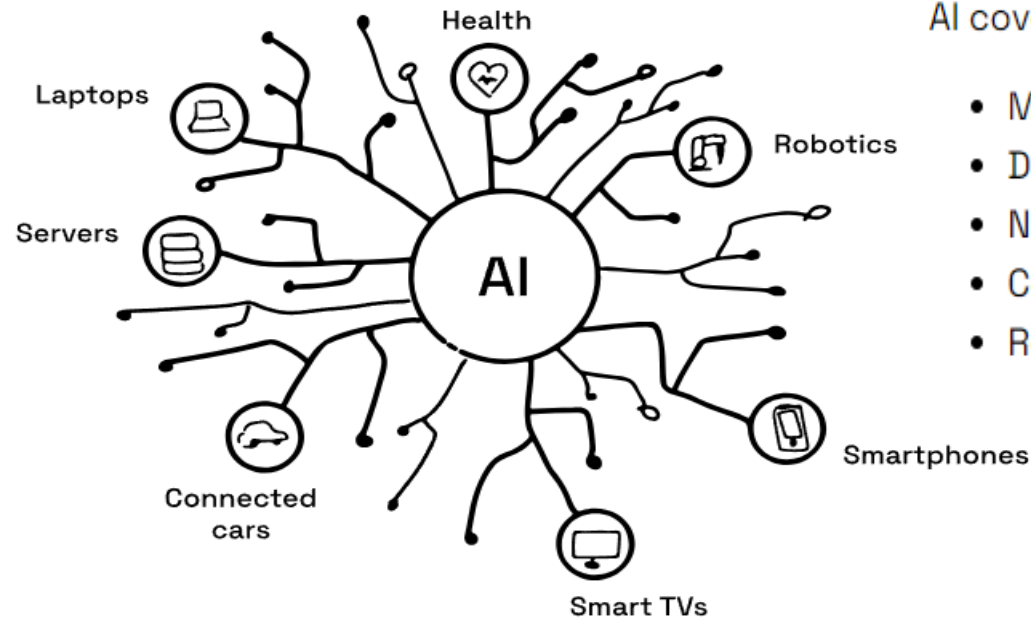
### 1.1 Généralités

#### 1.1.5. Qu'est-ce que l'intelligence artificielle ?

❑ Pour créer l'IA, nous devons d'abord savoir comment l'intelligence est composée, donc l'intelligence est une partie intangible de notre cerveau qui est une combinaison de raisonnement, d'apprentissage, de perception de résolution de problèmes, de compréhension du langage, etc.

❑ Pour atteindre les facteurs ci-dessus pour une machine ou un logiciel, l'intelligence artificielle nécessite la discipline suivante :

- ✓ Mathématiques
- ✓ Biologie
- ✓ Psychologie
- ✓ Sociologie
- ✓ Informatique
- ✓ Étude des neurones
- ✓ Statistiques



AI covers different techniques and approaches, including:

- Machine Learning (ML)
- Deep Learning
- Natural Language Processing (NLP)
- Computer vision
- Robotics

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



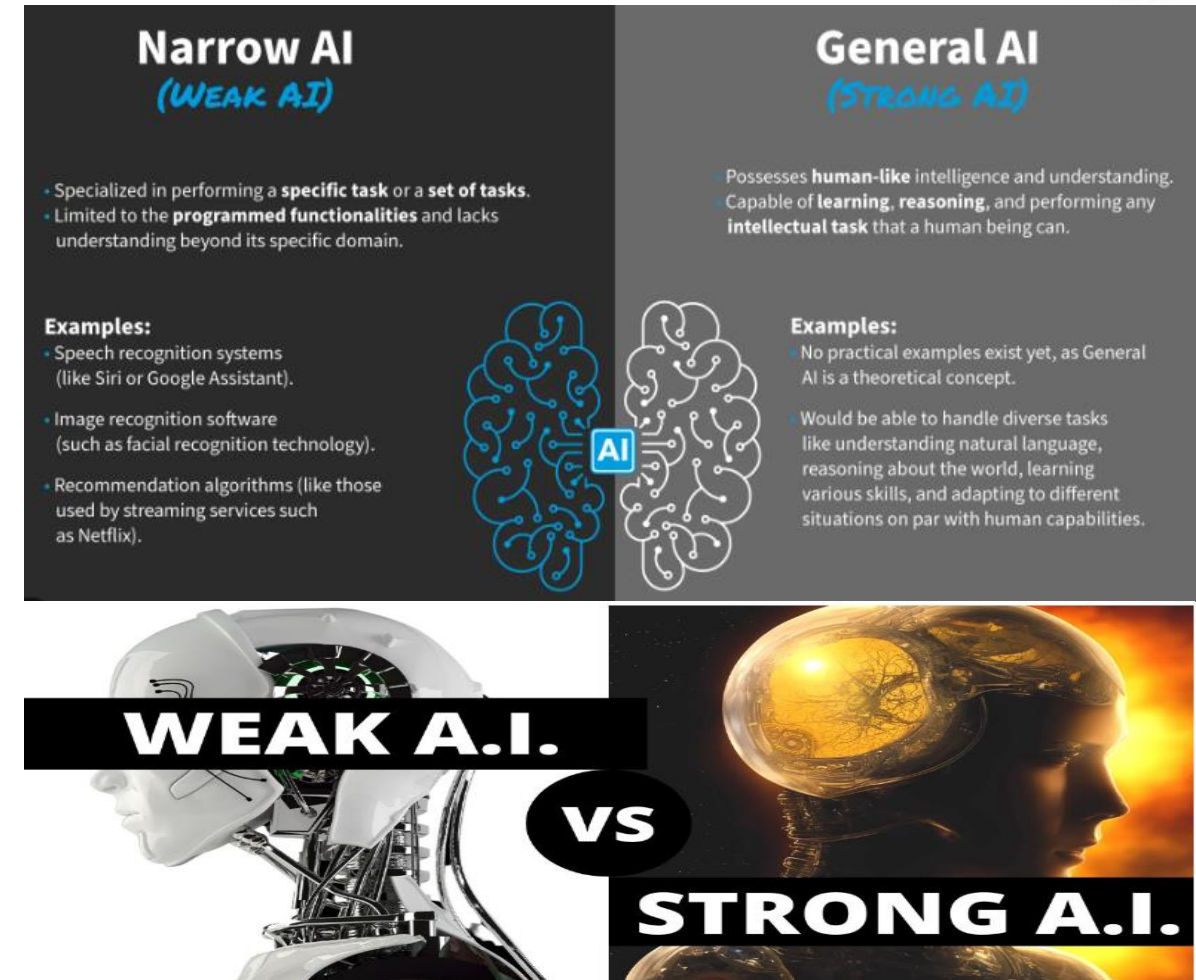
### 1.1 Généralités

#### 1.1.6. Branches de l'IA : Weak AI vs General AI vs Super AI

##### ❖ Weak AI or Narrow AI

L'IA étroite, également connue sous le nom d'IA faible, est comme un spécialiste dans le monde de l'intelligence artificielle.

- Imaginez-la comme un expert virtuel dédié à l'exécution d'une tâche spécifique avec intelligence.
- Par exemple, pensez à Siri d'Apple.
  - Il est assez intelligent lorsqu'il s'agit de commandes vocales et de réponses aux questions, mais il ne comprend pas ou ne fait pas grand-chose au-delà de cela.
- L'IA étroite fonctionne dans des limites strictes, et si vous lui demandez de sortir de sa zone de confort, elle risque de ne pas fonctionner comme prévu.
- Ce type d'IA est omniprésent dans le monde d'aujourd'hui, des voitures autonomes à la reconnaissance d'images sur votre smartphone.





## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



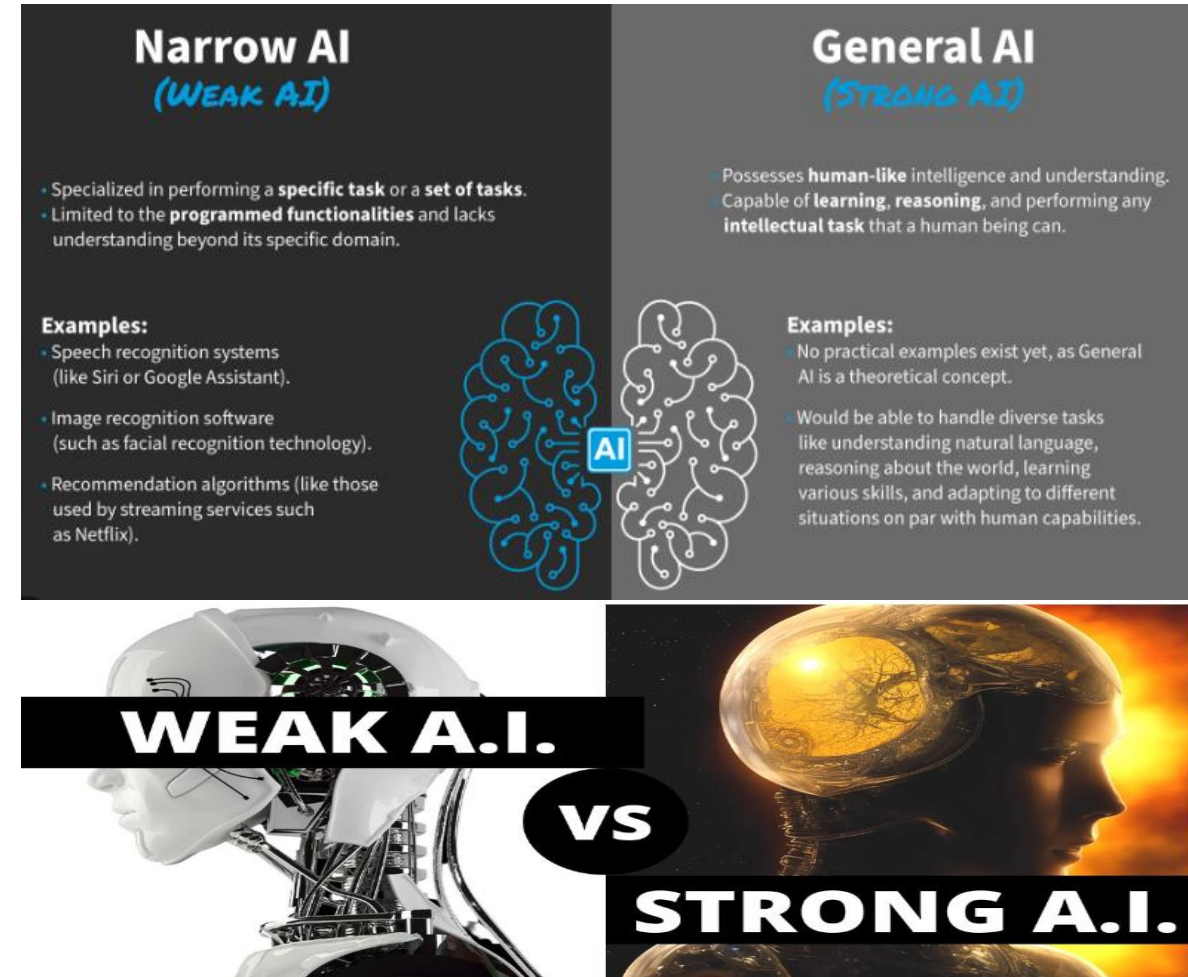
### 1.1 Généralités

#### 1.1.6. Branches de l'IA : Weak AI vs General AI vs Super AI

##### ❖ General AI

L'IA générale : l'IA générale, souvent appelée IA forte, est comme le Saint Graal de l'intelligence artificielle.

- Imaginez-la comme un système capable d'effectuer n'importe quelle tâche intellectuelle avec l'efficacité d'un humain.
- L'IA générale vise à créer des machines qui pensent et apprennent comme les humains, mais voici le problème : il n'existe pas encore de système de ce type.
- Les chercheurs du monde entier travaillent avec diligence pour en faire une réalité, mais c'est un voyage complexe qui nécessitera beaucoup de temps et d'efforts



## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.1 Généralités

#### 1.1.6. Branches de l'IA : Weak AI vs General AI vs Super AI

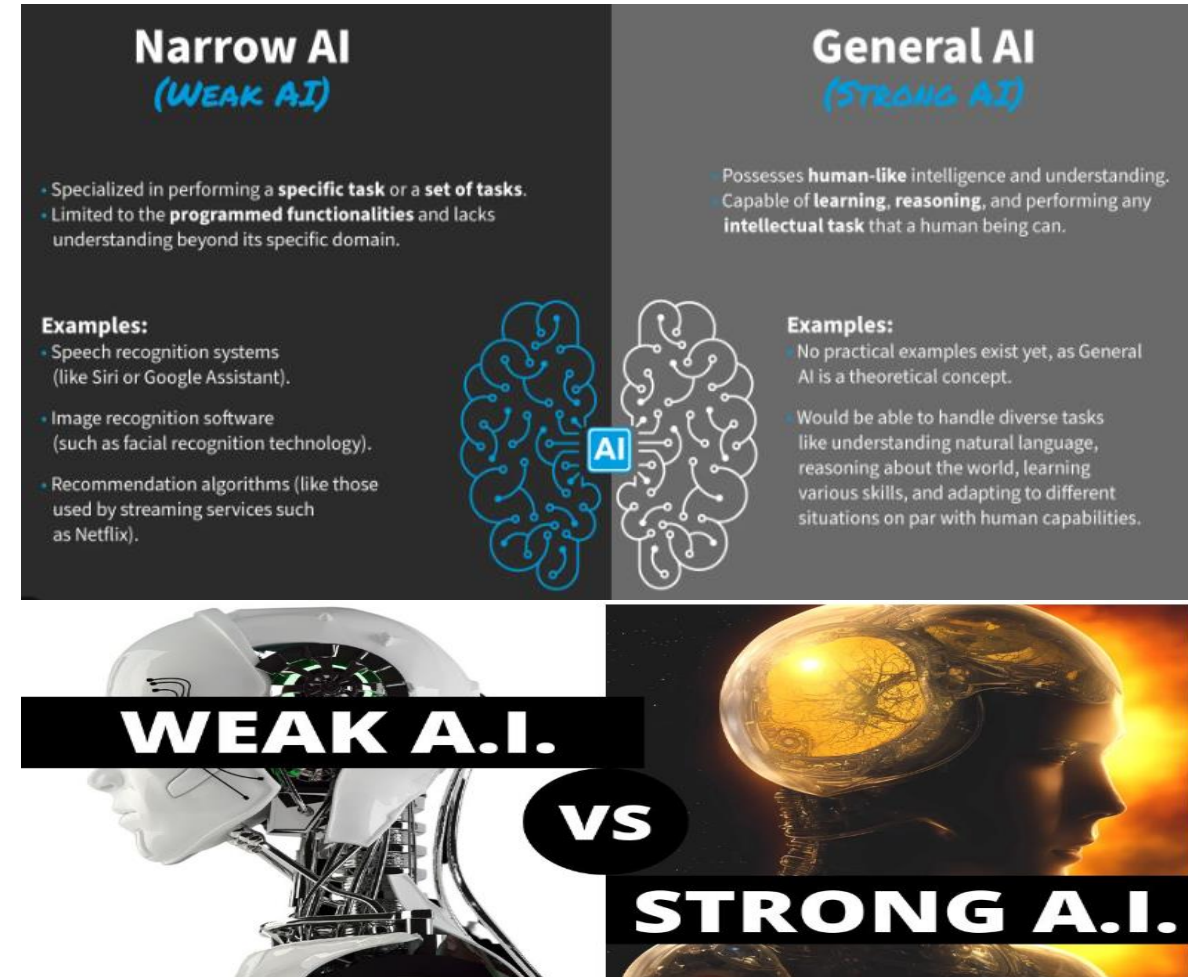
##### ❖ Super AI

**Super IA :** La super IA amène l'IA à un tout autre niveau. C'est le summum de l'intelligence artificielle, où les machines surpassent les capacités humaines dans tous les aspects cognitifs.

➤ Ces machines peuvent penser, raisonner, résoudre des énigmes, émettre des jugements, planifier, apprendre et communiquer de manière indépendante.

➤ Cependant, il est important de noter que la super IA est actuellement un concept hypothétique.

➤ Atteindre un tel niveau d'intelligence artificielle ne serait rien de moins qu'une révolution, et c'est un défi qui se profile encore à l'horizon.



# Programmation en Intelligence Artificielle

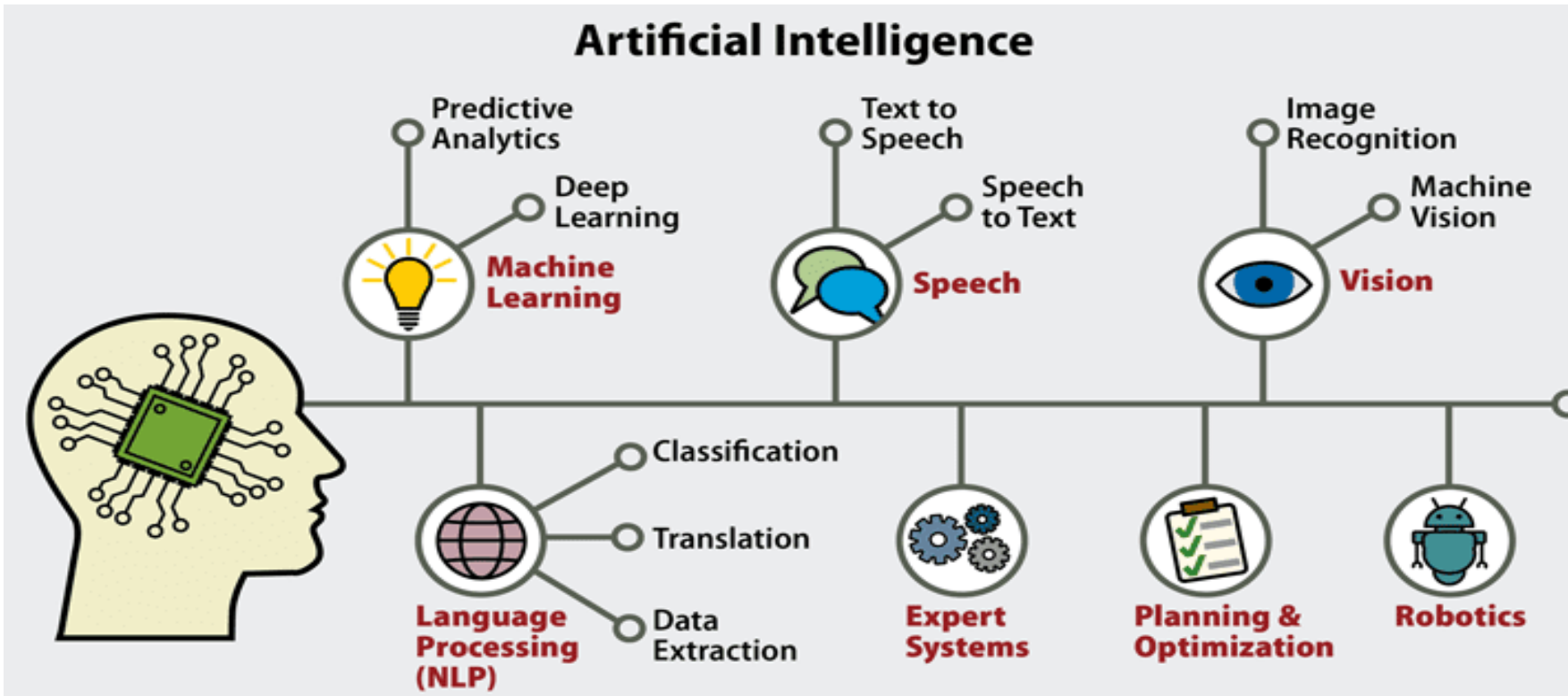
## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.1 Généralités

#### 1.1.7. Application de l'Intelligence Artificielle

- ❑ L'IA étant actuellement utilisée dans presque tous les secteurs, ses applications sont aujourd'hui très variées.
- ❑ Les secteurs dans lesquels l'IA est actuellement mise en œuvre comprennent :



-  Cybersécurité des soins de santé,
-  Commercialisation de la fabrication,
-  Formation bancaire,
-  Ressources humaines du bâtiment,
-  Transports E-commerce,
- ❑ Vision par ordinateur,
- ❑ Traitement du langage naturel (NLP),
- ❑ Robots autonomes,
- ❑ Jeux.

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



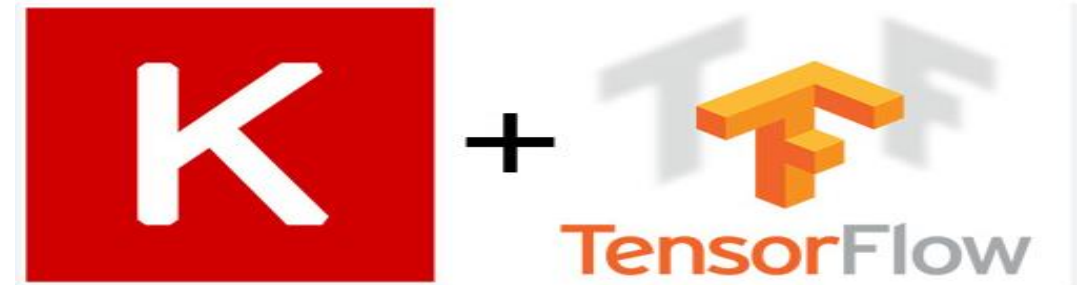
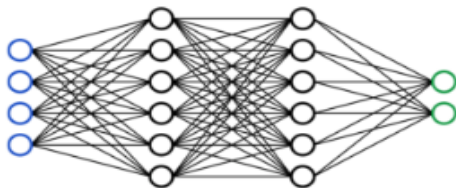
### 1.2 Langages de programmation en Intelligence Artificielle

- ❑ Python est le langage dominant dans l'IA (avec des bibliothèques comme **Keras**, **TensorFlow**, **PyTorch**).

#### 1.2.1. Introduction to Keras

##### ❖ Keras ?

- ❑ Keras est une bibliothèque Python d'apprentissage profond de haut niveau pour le développement de modèles de réseaux neuronaux.
- ❑ Keras est un wrapper d'API de haut niveau. Il peut s'exécuter sur les bibliothèques Tensorflow, CTNK et Theano.
- ❑ Keras est développé pour le développement facile et rapide de modèles de réseaux neuronaux



Keras est une interface simplifiée pour TensorFlow.



# Programmation en Intelligence Artificielle

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.2 Langages de programmation en Intelligence Artificielle

#### 1.2.1. Introduction to Keras

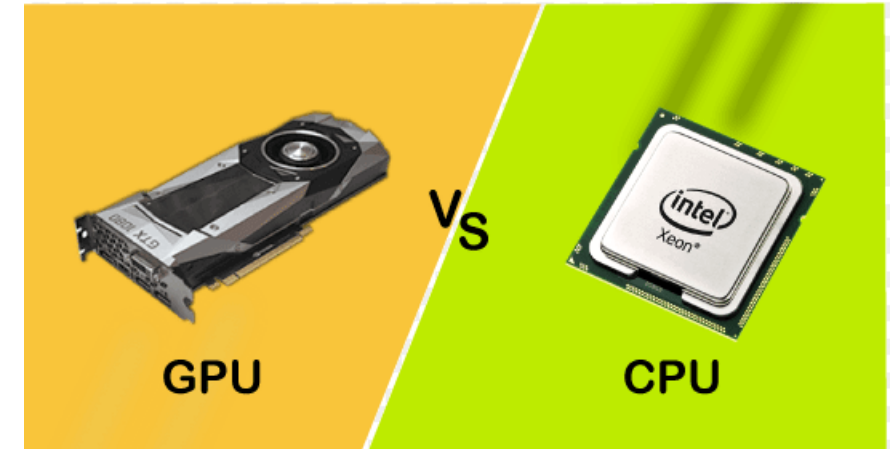
##### ❖ Avantages et limites

##### ❑ Keras offre les avantages suivants :

- ✓ Keras est une bibliothèque Python facile à apprendre et à utiliser.
- ✓ Développement plus rapide
- ✓ Keras peut fonctionner sur CPU et GPU.
- ✓ Keras peut fonctionner avec une variété d'algorithmes d'apprentissage profond tels que CNN, RNN et LSTM.

##### ❑ Keras offre les limitations suivantes :

- ✓ Keras dépend de bibliothèques de niveau inférieur telles que
- ✓ TensorFlow et Theano qui peuvent provoquer des erreurs de bas niveau.
- ✓ Keras prend uniquement en charge le GPU NVIDIA.
- ✓ Parfois, Keras est plus lente que son backend.



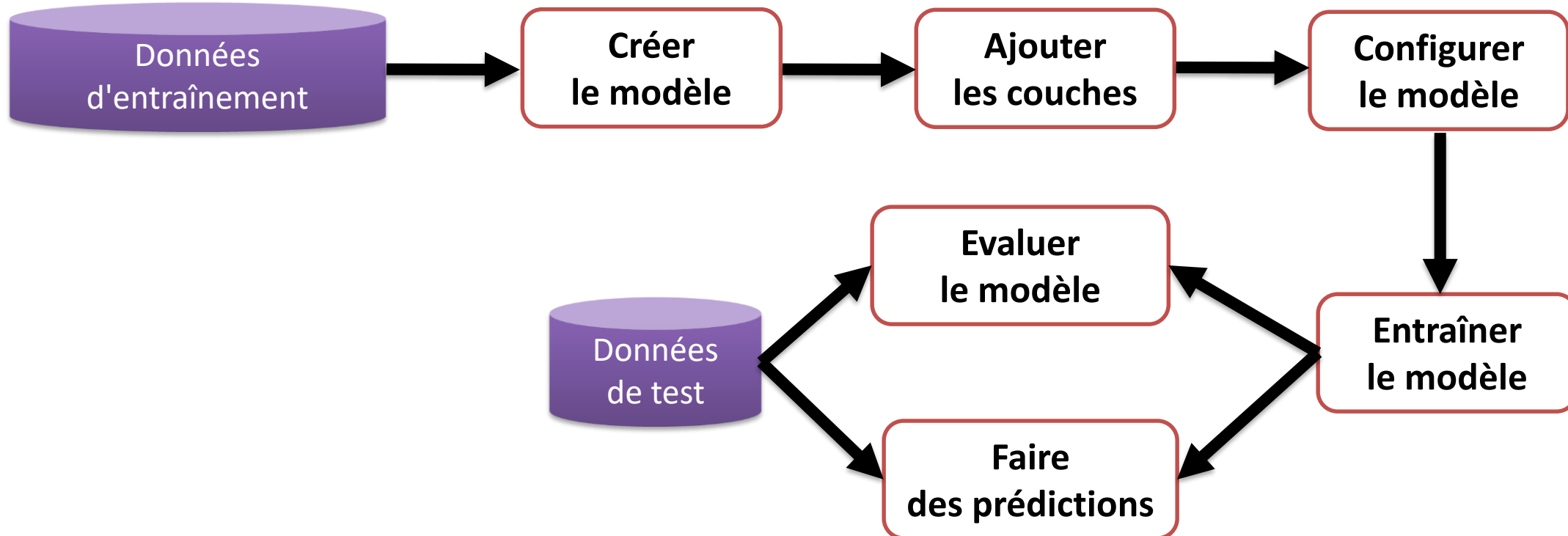
## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.2 Langages de programmation en Intelligence Artificielle

#### 1.2.1. Introduction to Keras

##### ❖ Keras Workflow



## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.2 Langages de programmation en Intelligence Artificielle

#### 1.2.1. Introduction to Keras

##### ❖ Les Composants de Keras

1. **Modèle séquentiel** : Keras fournit un moyen simple de créer une perception multicouche à l'aide du modèle séquentiel.
2. **Ajouter une couche** : la fonction `add()` est utilisée pour ajouter une couche au réseau neuronal.

Nous devons ajouter le type de couche que nous souhaitons ajouter au modèle séquentiel. (Next slide)

3. **Compilation du modèle** : elle est utilisée pour compiler le modèle. Elle prend l'optimiseur et la fonction de perte comme paramètres d'entrée.

- ✓ Les **algorithmes d'optimisation** les plus populaires sont Stochastic Gradient Descent (SGD), ADAM et RMSprop.
- ✓ Pour la **fonction de perte**, nous pouvons utiliser l'erreur quadratique moyenne (pour les problèmes de régression), `binary_crossentropy` (pour le problème de classification binaire) ou `categorical_crossentropy` (pour la classification multi-classe).

# CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



## 1.2.1. Introduction to Keras

### ❖ Les Composants de Keras

**4. Entraînement du modèle** : la fonction `fit()` utilisée pour entraîner le modèle. Il prend les paramètres suivants en entrée : données d'entraînement (X\_train et y\_train), données de validation (X\_test, y\_test) et nombre d'époques (epochs).

**5. Faire des prédictions** : la fonction `predict()` est utilisée pour faire des prédictions sur de nouvelles données d'entrée (X\_test).

**6. Évaluation du modèle** : la fonction `evaluate()` est utilisée pour évaluer les performances du modèle.

Il prend des fonctionnalités de test et des étiquettes en entrée.



## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.2 Langages de programmation en Intelligence Artificielle

#### 1.2.1. Introduction to Keras

##### ❖ Les Couches (Layers) de Keras

- ☐ Dans Keras, un framework d'apprentissage profond, une couche est l'élément de base des réseaux neuronaux.
- ☐ Les couches dans Keras sont responsables de la transformation des entrées en sorties via une série d'opérations, telles que des transformations linéaires, des fonctions d'activation, etc.
- **Couche d'entrée [Input Layer]** : La couche d'entrée est l'endroit où vous spécifiez la forme de vos données d'entrée. Elle est utilisée pour définir la forme des données qui seront introduites dans le modèle.

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.2 Langages de programmation en Intelligence Artificielle

#### 1.2.1. Introduction to Keras

##### ❖ Les Couches (Layers) de Keras

- **Couche Dense [Dense layer]** : La couche dense est une couche entièrement connectée, ce qui signifie que chaque neurone de la couche est connecté à chaque neurone de la couche précédente.  
Il s'agit de la couche la plus courante dans les réseaux neuronaux à propagation directe et elle est généralement utilisée pour des tâches telles que la classification et la régression.
- **Couche d'activation [Activation layer]** : Bien que les fonctions d'activation puissent être appliquées directement dans d'autres couches (par exemple, dans Dense), vous pouvez également ajouter une couche d'activation distincte pour appliquer n'importe quelle fonction d'activation comme ReLU, Sigmoid ou Softmax.

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.2 Langages de programmation en Intelligence Artificielle

#### 1.2.1. Introduction to Keras

##### ❖ Les Couches (Layers) de Keras

- **Couche convolutionnelle [Convolutional layer] (Conv2D, Conv1D, Conv3D)** : Couche de traitement des images, utilisée pour les réseaux neuronaux convolutionnels.
- **Couche récurrente [Reccurent layer] (LSTM, GRU, SimpleRNN)**: Couche de traitement des séquences de données, utilisée pour les réseaux neuronaux récurrents.
- **Couche Pooling [MaxPooling, AveragePooling]** : Couche de sous-échantillonnage des cartes de caractéristiques en prenant la valeur maximale dans des blocs rectangulaires non superposés, utilisée pour préserver les caractéristiques importantes tout en réduisant les risques de sur-ajustement.

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.2 Langages de programmation en Intelligence Artificielle

#### 1.2.1. Introduction to Keras

##### ❖ Les Couches (Layers) de Keras

- **Couche d'aplatissement [Flatten layer]** : Les couches de regroupement sont utilisées pour sous-échantillonner l'entrée, réduisant ainsi les dimensions spatiales (hauteur et largeur) des cartes de caractéristiques.
- **BatchNormalization layer** : La couche BatchNormalization normalise la sortie d'une couche précédente en soustrayant la moyenne du lot et en la divisant par l'écart type du lot. Elle permet d'accélérer la formation, d'améliorer la stabilité et d'utiliser des taux d'apprentissage plus élevés.

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.2 Langages de programmation en Intelligence Artificielle

#### 1.2.1. Introduction to Keras

##### ❖ Les Couches (Layers) de Keras

- **Couche d'abandon [Dropout layer]** : Couche qui définit aléatoirement les unités d'entrée sur 0 (à l'aide d'une fréquence définie) pendant le temps d'apprentissage, utilisée comme technique de régularisation pour éviter le sur-ajustement dans les réseaux neuronaux.
- **Couche d'intégration [Embedding layer]**: Couche qui représente des mots ou des phrases dans un espace vectoriel à haute dimension, utilisée pour mapper des mots ou des phrases à des vecteurs denses à utiliser comme entrée dans un réseau neuronal.

Ce ne sont là que quelques exemples des nombreux types de couches disponibles dans l'API séquentielle Keras.

Chaque couche est conçue pour effectuer un type de calcul spécifique sur les entrées, et elles peuvent être combinées pour créer une architecture de réseau neuronal puissante.

**Labo: Implémentation d'un réseau neuronal convolutif (CNN) en Python avec le framework Keras**

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.2 Langages de programmation en Intelligence Artificielle

#### 1.2.1. Introduction to Keras

##### ❖ Comment créer un modèle dans Keras ?

❑ Keras propose deux méthodes principales pour créer des modèles : **API Séquentielle** et **API Fonctionnelle**.

✓ **API séquentielle** : API séquentielles sont faciles à utiliser avec des modèles dotés d'une seule entrée et sortie et d'une pile linéaire de couches.

##### ▪ Création d'un modèle à l'aide de l'API séquentielle

Voici comment définir un modèle séquentiel :

- Créer un modèle séquentiel.
- Ajouter une couche entièrement connectée (Dense) avec 64 unités et une activation ReLU.
- Ajouter une autre couche Dense avec 10 unités (pour la classification) et une activation softmax.

```
1 from keras.models import Sequential
2 from keras.layers import Dense, Activation
3
4 model = Sequential()
5 model.add(Dense(units=64, input_dim=100))
6 model.add(Activation('relu'))
7 model.add(Dense(units=10))
8 model.add(Activation('softmax'))
9 model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	6,464
activation (Activation)	(None, 64)	0
dense_1 (Dense)	(None, 10)	650
activation_1 (Activation)	(None, 10)	0

Total params: 7,114 (27.79 KB)  
Trainable params: 7,114 (27.79 KB)  
Non-trainable params: 0 (0.00 B)

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.2 Langages de programmation en Intelligence Artificielle

#### 1.2.1. Introduction to Keras

##### ❖ Comment créer un modèle dans Keras ?

❑ Keras propose deux méthodes principales pour créer des modèles : **API Séquentielle** et **API Fonctionnelle**.

✓ **API Fonctionnelle** : API fonctionnelle peut être utilisée pour les modèles qui nécessitent plusieurs entrées et sorties, ou les couches ayant plusieurs entrées ou sorties

##### ▪ Création d'un modèle à l'aide de l'API Fonctionnelle

Voici comment définir un modèle Fonctionnel :

- Définir deux couches d'entrée (input1 et input2).
- Créer des couches cachées distinctes pour chaque entrée.
- Fusionner les couches cachées à l'aide de la fonction concatenate.
- Enfin, ajouter une couche de sortie avec l'activation SoftMax.

```
1 from keras.layers import Input, Dense, concatenate
2 from keras.models import Model
3
4 input1 = Input(shape=(100,))
5 input2 = Input(shape=(50,))
6 hidden1 = Dense(64, activation='relu')(input1)
7 hidden2 = Dense(32, activation='relu')(input2)
8 merged = concatenate([hidden1, hidden2])
9 output = Dense(10, activation='softmax')(merged)
10
11 model = Model(inputs=[input1, input2], outputs=output)
12 model.summary()
```

Model: "functional"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 100)	0	-
input_layer_1 (InputLayer)	(None, 50)	0	-
dense (Dense)	(None, 64)	6,464	input_layer[0][0]
dense_1 (Dense)	(None, 32)	1,632	input_layer_1[0][0]
concatenate (Concatenate)	(None, 96)	0	dense[0][0], dense_1[0][0]
dense_2 (Dense)	(None, 10)	970	concatenate[0][0]

Total params: 9,066 (35.41 KB)

Trainable params: 9,066 (35.41 KB)

Non-trainable params: 0 (0.00 B)

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.2 Langages de programmation en Intelligence Artificielle

#### 1.2.1. Introduction to Keras

##### ❖ Comment créer un modèle dans Keras ?

- ❑ Keras propose deux méthodes principales pour créer des modèles : **API séquentielle** et **API fonctionnelle**.
- ✓ **API séquentielle** : API séquentielles sont faciles à utiliser avec des modèles dotés d'une seule entrée et sortie et d'une pile linéaire de couches.
- ✓ **API fonctionnelle** : API fonctionnelle peut être utilisée pour les modèles qui nécessitent plusieurs entrées et sorties, ou les couches ayant plusieurs entrées ou sorties.



## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.1. Interpréteur

Python est un langage de programmation généraliste de haut niveau largement utilisé. Il a été créé par Guido van Rossum en 1991 et développé par la Python Software Foundation.

Il a été conçu en mettant l'accent sur la lisibilité du code et sa syntaxe permet aux programmeurs d'exprimer leurs concepts en moins de lignes de code. Python est un langage de programmation qui vous permet de travailler rapidement et d'intégrer des systèmes plus efficacement. Il existe deux versions majeures de Python : Python 2 et Python 3.

Les deux sont assez différentes.

L'interpréteur Python est un programme qui exécute du code Python. Il lit et interprète le code source Python puis exécute les instructions spécifiées dans ce code. L'interpréteur est chargé de traduire le code Python de haut niveau en code machine de bas niveau que le matériel de l'ordinateur peut exécuter.

Avant de commencer la programmation Python, nous avons besoin d'un interpréteur pour interpréter et exécuter nos programmes.

Sur Windows, des nombreux interpréteurs sont disponibles gratuitement pour exécuter des scripts Python tels que l'IDE (Integrated Development Environment). Sur Linux, Python est préinstallé avec les distributions Linux populaires telles que Ubuntu et Fedora.

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.1. Interpréteur

Vous pouvez également utiliser des environnements de développement intégrés tels que PyCharm, VSCode ou Jupyter Notebooks, qui fournissent un environnement plus riche en fonctionnalités et interactif pour écrire et exécuter du code Python.

- ✓ Pour vérifier quelle version de Python vous utilisez, tapez « python » dans l'émulateur de terminal.

```
Command Prompt - python
Microsoft Windows [Version 10.0.19045.3930]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MWAMBA>python
Python 3.11.4 | packaged by Anaconda, Inc. | (main, Jul 5 2023, 13:38:37) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

L'interprète doit démarrer et imprimer le numéro de version.

Généralement, Python 2.7 est fourni avec le système.

Vous pouvez installer manuellement Python 3 depuis <http://python.org/>.

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.1. Installer Anaconda et exécuter Jupyter Notebook

Dans la communauté scientifique, Anaconda et Jupyter Notebook sont respectivement les distributions et les outils les plus utilisés pour exécuter la programmation Python. C'est pourquoi dans cette section, je couvrirai les instructions étape par étape sur la façon d'installer la distribution Anaconda, de configurer Jupyter Notebook et d'exécuter quelques exemples.

- ❖ **Anaconda** est la plate-forme de distribution de langages de programmation Python la plus utilisée dans la communauté de la science des données et de l'apprentissage automatique, car elle simplifie l'installation de packages tels que pandas, NumPy, SciPy et bien d'autres.
- ❖ **Conda** est le gestionnaire de packages sur lequel repose la distribution Anaconda. Il s'agit d'un gestionnaire de packages à la fois multiplateforme et indépendant du langage. Nous pouvons utiliser conda pour installer des packages tiers.
- ❖ **Jupyter Notebook** est un environnement d'interface utilisateur Web interactif permettant de créer des documents de bloc-notes pour les langages Python.

# Programmation en Intelligence Artificielle

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.1. Installer Anaconda et exécuter Jupyter Notebook

##### ❖ Téléchargez et installez la distribution Anaconda

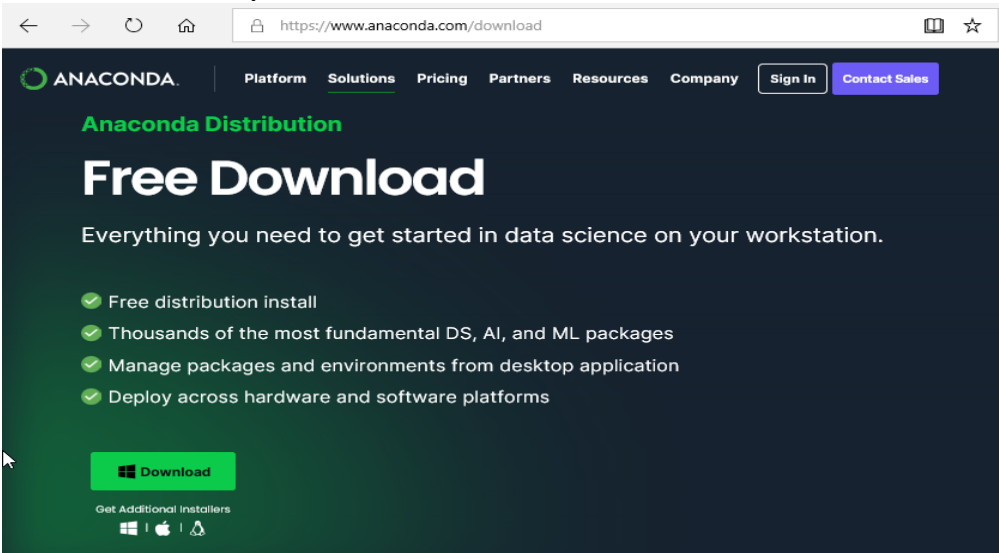
✓ Suivez les instructions étape par étape ci-dessous pour installer la distribution Anaconda.

##### *Télécharger la distribution Anaconda*

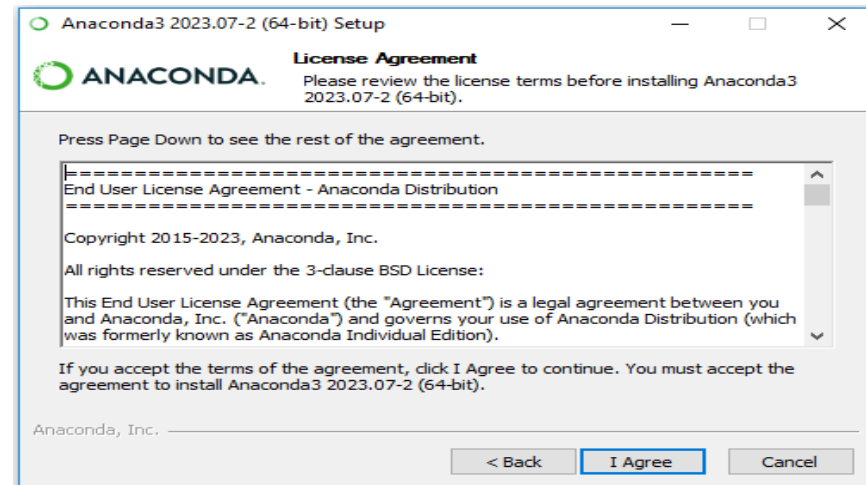
Accédez à <https://anaconda.com/> et cliquer sur le bouton Download pour télécharger la dernière version d'Anaconda.

Cela télécharge le fichier .exe dans le dossier de téléchargement Windows.

En double-cliquant sur le fichier .exe, vous démarrez l'installation d'Anaconda. Suivez les captures d'écran ci-dessous et terminez l'installation



##### *Installer Anaconda*



# Programmation en Intelligence Artificielle

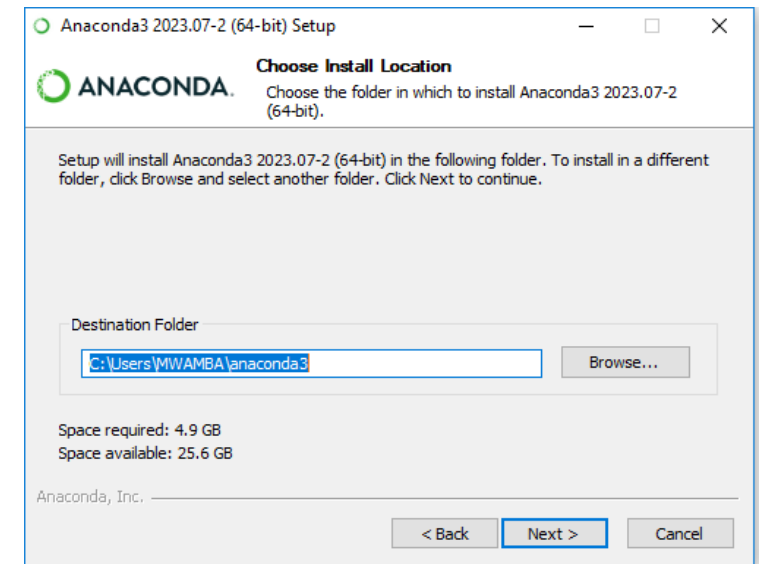
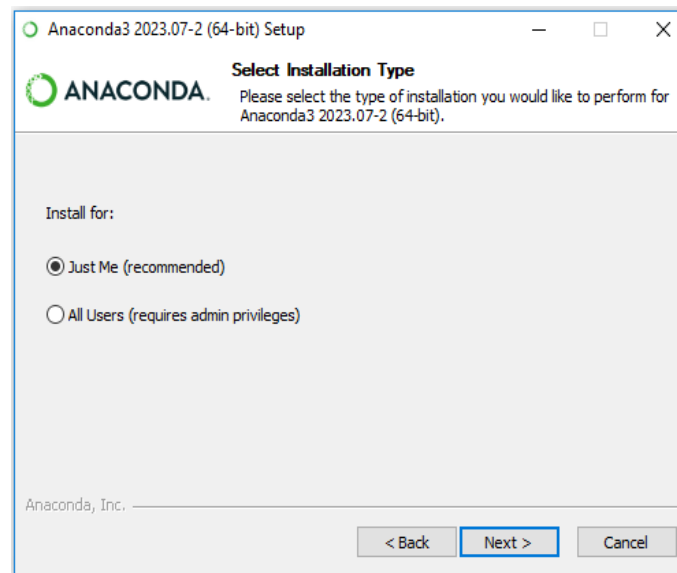
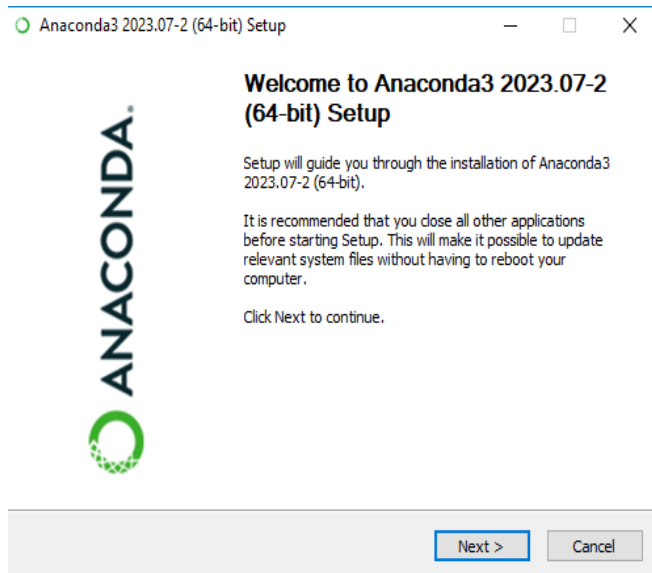
## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.1. Installer Anaconda et exécuter Jupyter Notebook

##### ❖ Installation d'Anaconda



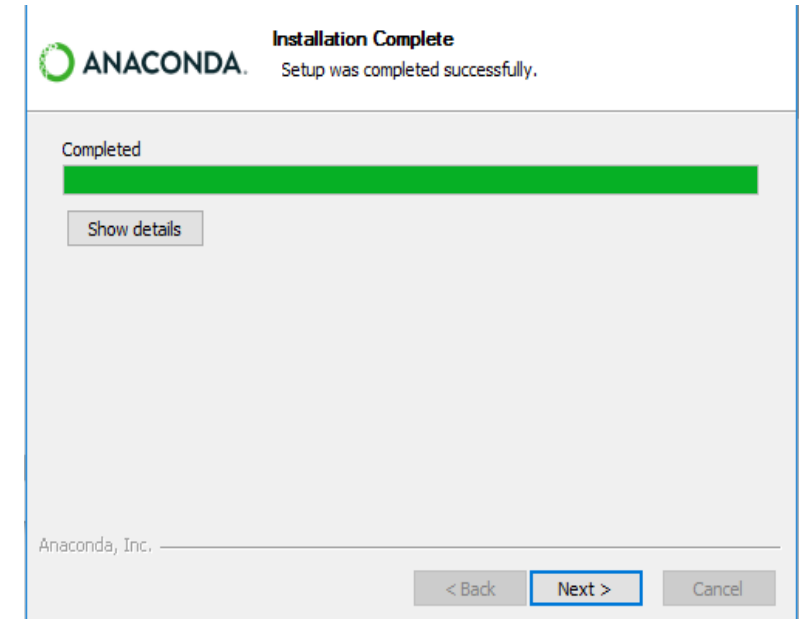
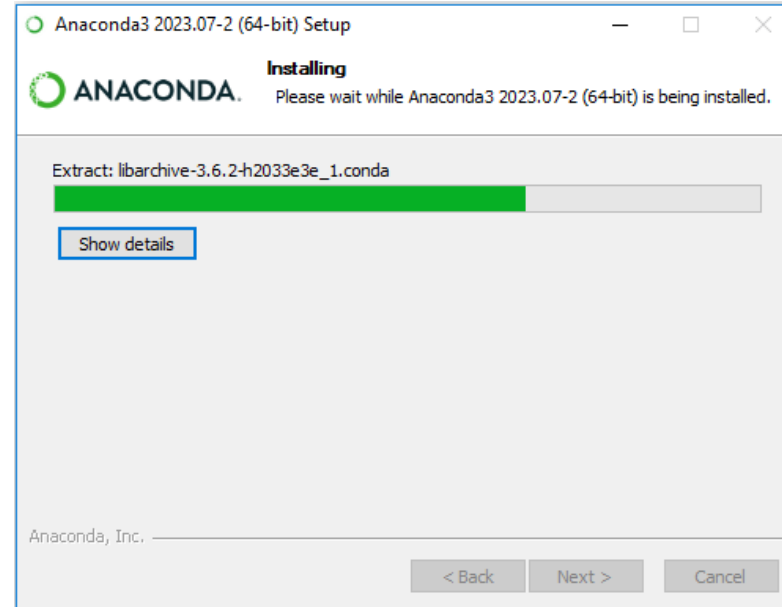
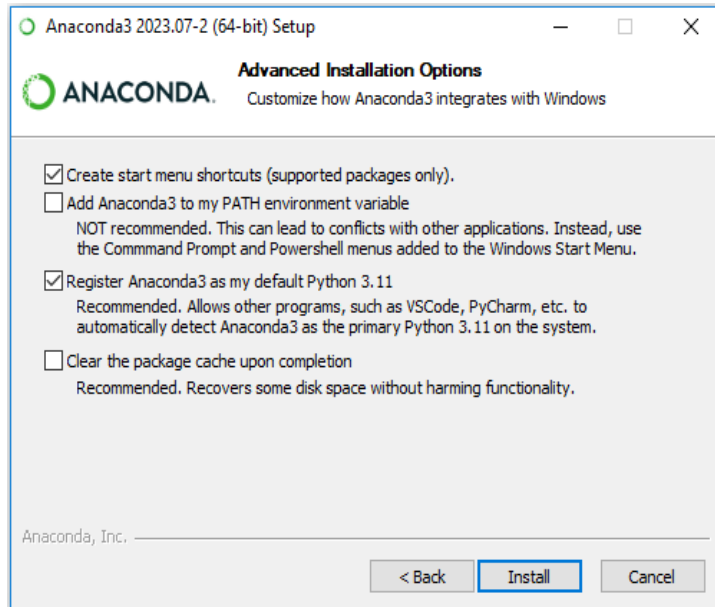
## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.1. Installer Anaconda et exécuter Jupyter Notebook

##### ❖ Installation d'Anaconda



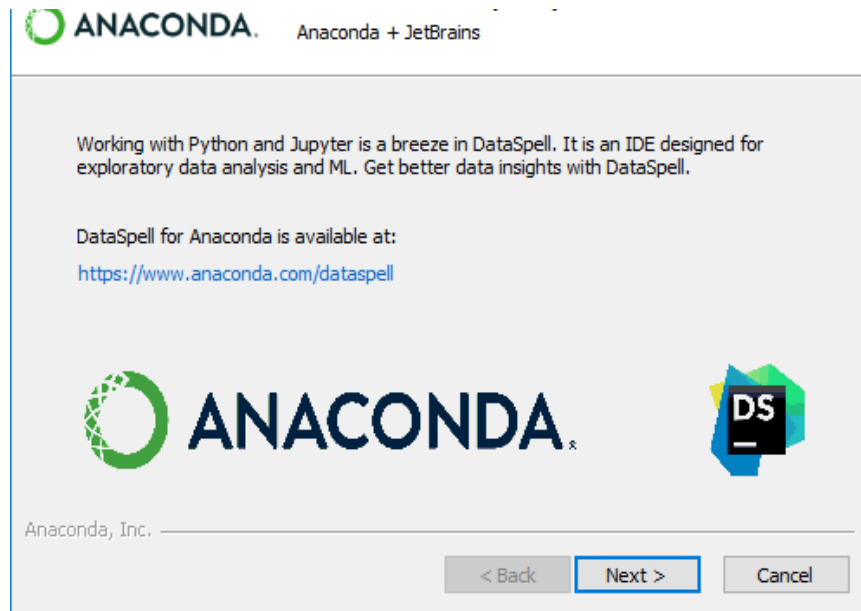
## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.1. Installer Anaconda et exécuter Jupyter Notebook

##### ❖ Installation d'Anaconda

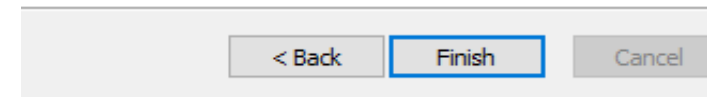


Thank you for installing Anaconda Distribution.

Here are some helpful tips and resources to get you started. We recommend you bookmark these links so you can refer back to them later.

☒ Anaconda Distribution Tutorial

☒ Getting Started with Anaconda



- ✓ Ceci termine l'installation de la distribution Anaconda, voyons maintenant comment créer un environnement et installer Jupyter Notebook.



## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.1. Installer Anaconda et exécuter Jupyter Notebook

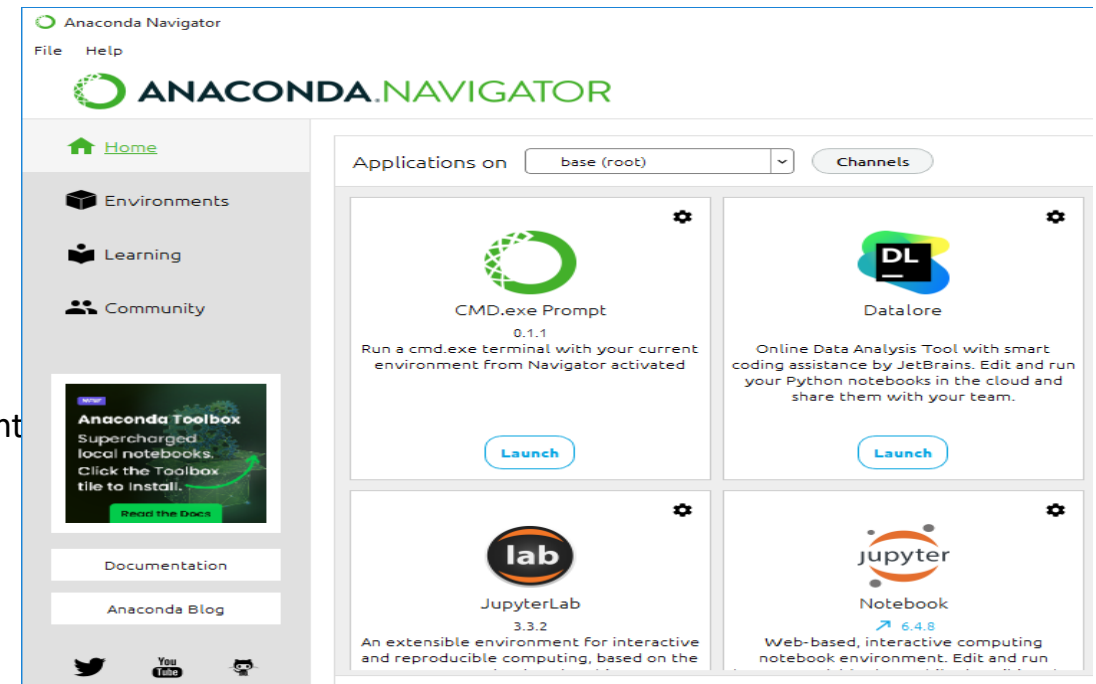
##### ❖ Créer un environnement Anaconda à partir de Navigator

❑ Un environnement conda est un répertoire qui contient une collection spécifique de packages conda que vous avez installés. Par exemple, vous pouvez disposer d'un environnement avec NumPy 1.7 et ses dépendances et d'un autre environnement avec NumPy 1.6 pour les tests existants.

- ❑ L'outil que nous utiliserons principalement est Jupyter Notebook
- ❑ (déjà fourni avec l'installation d'Anaconda).
- ❑ Il s'agit d'un bloc-notes dans lequel vous pouvez saisir et exécuter votre code ainsi qu'ajouter du texte et des notes.

##### ➤ Ouvrir le navigateur Anaconda

- ✓ Ouvrez Anaconda Navigator à partir du démarrage de Windows ou en le recherchant
- ✓ Anaconda Navigator est une application d'interface utilisateur dans laquelle vous pouvez contrôler les packages et les environnements Anaconda.





# Programmation en Intelligence Artificielle

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE

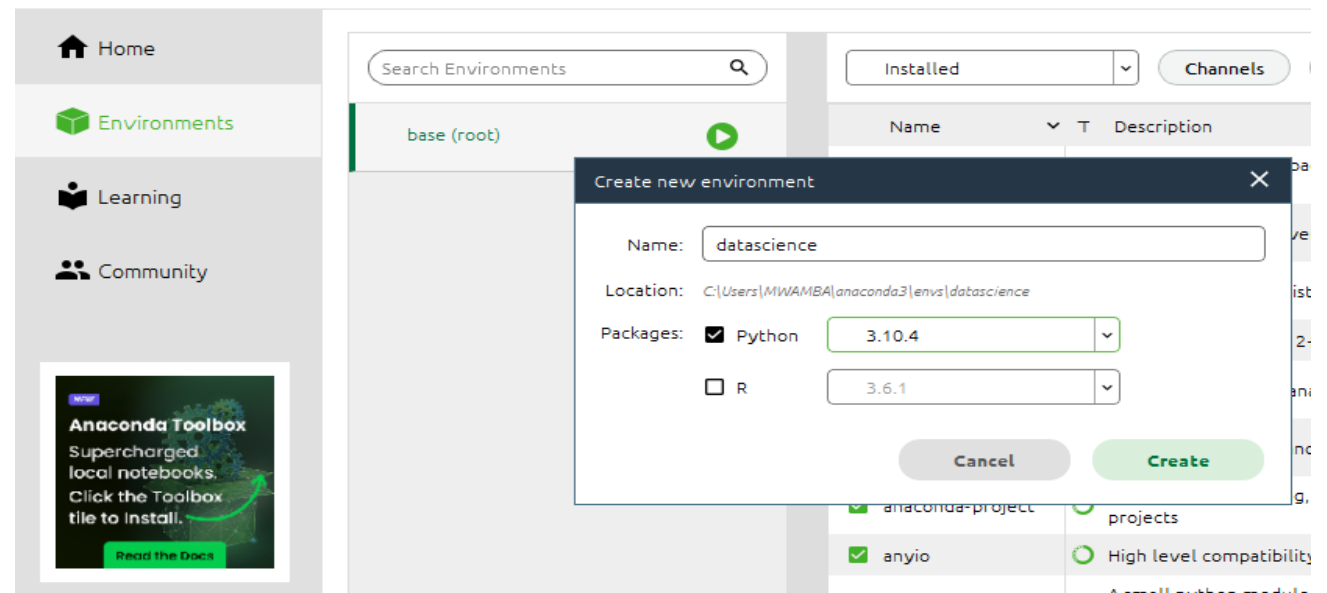
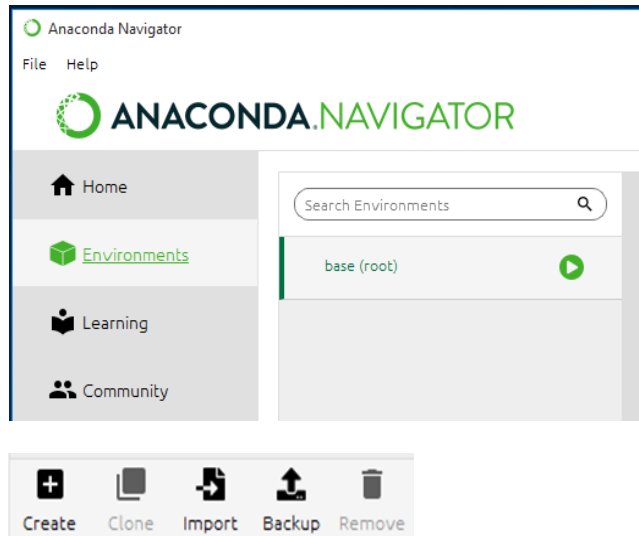


### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.1. Installer Anaconda et exécuter Jupyter Notebook

##### ❖ Créer un environnement pour exécuter Jupyter Notebook

- Ceci est facultatif mais recommandé pour créer un environnement avant de continuer.
- Cela donne une séparation complète des différentes installations de packages pour les différents projets sur lesquels vous aller travailler.
- Si vous disposez déjà d'un environnement, vous pouvez également l'utiliser sinon sélectionnez + **Créer** l'icône en bas de l'écran pour créer un environnement Anaconda (Nom + version de python).



## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



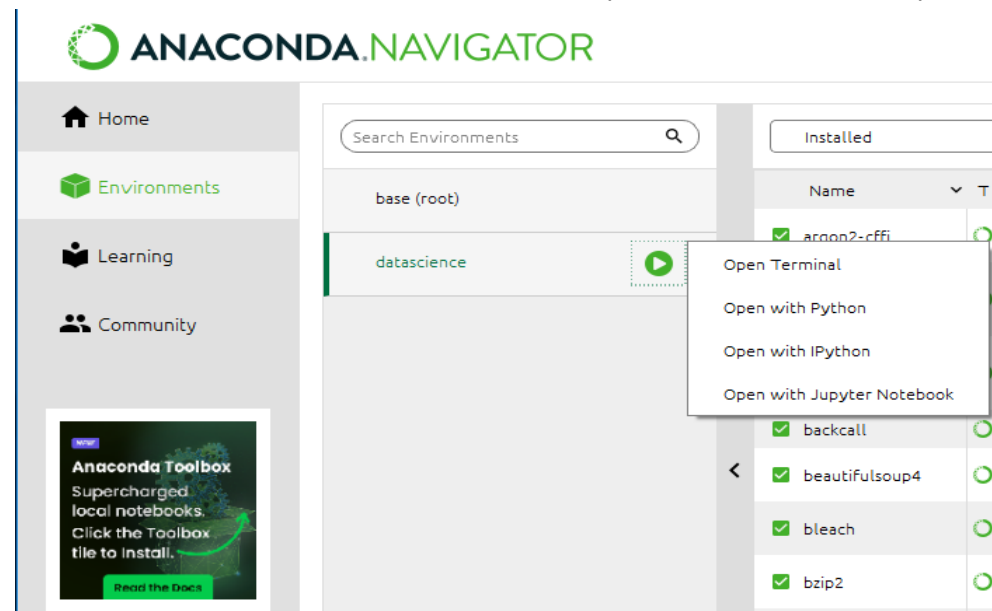
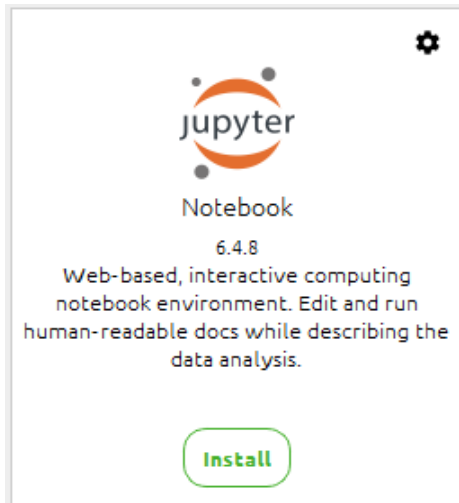
### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.1. Installer Anaconda et exécuter Jupyter Notebook

##### Installer et exécuter Jupyter Notebook

- Une fois que vous avez créé l'environnement Anaconda, revenez à la page d'accueil d'Anaconda Navigator et installez Jupyter Notebook à partir d'une application sur le panneau de droite. L'installation de Jupyter dans votre environnement prendra quelques secondes.

Une fois l'installation terminée, vous pourrez ouvrir Jupyter à partir du même écran ou en accédant à Anaconda Navigator -> Environnements -> votre environnement (le mien **datascience**) -> sélectionnez Ouvrir avec Jupyter Notebook.



## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.1. Installer Anaconda et exécuter Jupyter Notebook

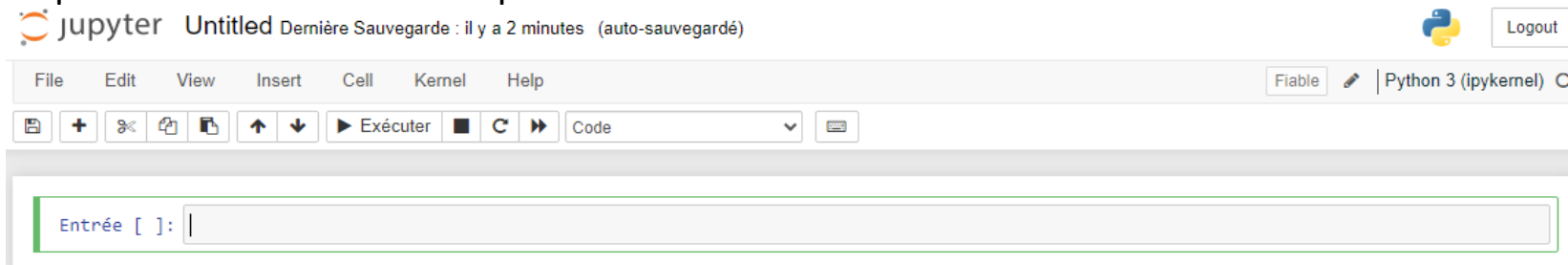
##### Installer et exécuter Jupyter Notebook

- Cela ouvre Jupyter Notebook dans le navigateur par défaut.



Sélectionnez maintenant Nouveau -> Python 3.

- Sur Jupyter, chaque cellule est une instruction, vous pouvez donc exécuter chaque cellule indépendamment lorsqu'il n'y a aucune dépendance avec les cellules précédentes.



- Ceci termine l'installation d'Anaconda et l'exécution de Jupyter Notebook.

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.2. Quelques bases de python

Les bases du langage Python, les structures de données (listes, tuples, dictionnaires), les fonctions, les boucles et les conditions [Prérequis].

#### 1.3.2. Bibliothèques Python pour l'IA

- ✓ NumPy : Manipulation de tableaux et calculs mathématiques.
- ✓ Pandas : Gestion et manipulation des datasets.
- ✓ Matplotlib & Seaborn : Visualisation des données.
- ✓ Scikit-learn : Algorithmes d'apprentissage automatique.
- ✓ Keras, TensorFlow et PyTorch : Outils pour la création et l'entraînement de réseaux de neurones

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.2. Quelques bases de python

Les bases du langage Python, les structures de données (listes, tuples, dictionnaires), les fonctions, les boucles et les conditions [Prérequis].

#### 1.3.2. Bibliothèques Python pour l'IA

##### ❖ NumPy (Numerical Python)

Pour installer: `pip install numpy`

<https://pypi.org/project/numpy/>

- NumPy est un package fondamental pour le calcul numérique en Python.
- ✓ Il prend en charge les tableaux multidimensionnels, les fonctions mathématiques, la génération de nombres aléatoires, l'algèbre linéaire, etc.
- ✓ Les tableaux NumPy peuvent être unidimensionnels (1D), bidimensionnels (2D) ou multidimensionnels (2D ou 3D ou 4D).

##### □ Tableaux unidimensionnels (1D) :

- Les tableaux unidimensionnels sont similaires aux listes Python.
- Ils sont créés à l'aide de la fonction `numpy.array()` en passant une liste ou un tuple Python.

```
<class 'numpy.ndarray'>  
[1 2 3 4]
```

```
[133]: # Tableaux unidimensionnels (1D)  
  
# Importer la library  
import numpy as np  
  
# Créez un tableau NumPy 1D à partir d'une liste Python  
tableau_1D = np.array([1, 2, 3, 4])  
print(type(tableau_1D))  
print(tableau_1D)  
  
<class 'numpy.ndarray'>  
[1 2 3 4]
```

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.2. Quelques bases de python

Les bases du langage Python, les structures de données (listes, tuples, dictionnaires), les fonctions, les boucles et les conditions [Prérequis].

#### 1.3.2. Bibliothèques Python pour l'IA

##### ❖ NumPy (Numerical Python)

Pour installer: `pip install numpy`

<https://pypi.org/project/numpy/>

➤ NumPy est un package fondamental pour le calcul numérique en Python.

##### □ Tableaux bidimensionnels (2D) :

- Les tableaux à deux dimensions sont comme des matrices ou des tableaux avec des lignes et des colonnes.
- Ils sont créés à l'aide de listes Python imbriquées ou en remodelant un tableau 1D.

```
<class 'numpy.ndarray'>
[[ 10  20  30  40  50]
 [ 60  70  80  90 100]]
(2, 5)
```

```
[136]: # Tableaux bidimensionnels (2D)

# Importer la Library
import numpy as np

# Créez un tableau NumPy 2D à partir d'une Liste Python imbriquée
tableau_2D = np.array([[10, 20, 30, 40, 50],[60, 70, 80, 90, 100]])
print(type(tableau_2D))
print(tableau_2D)
print(tableau_2D.shape)

<class 'numpy.ndarray'>
[[ 10  20  30  40  50]
 [ 60  70  80  90 100]]
(2, 5)
```

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.2. Quelques bases de python

Les bases du langage Python, les structures de données (listes, tuples, dictionnaires), les fonctions, les boucles et les conditions [Prérequis].

#### 1.3.2. Bibliothèques Python pour l'IA

##### ❖ NumPy (Numerical Python)

Pour installer: `pip install numpy`

<https://pypi.org/project/numpy/>

- NumPy est un package fondamental pour le calcul numérique en Python.

##### ❑ Tableaux multidimensionnels (3D) :

- Les tableaux multidimensionnels peuvent avoir plus de deux dimensions.

```
<class 'numpy.ndarray'>
[[[10 20 30]
  [40 50 60]
  [70 80 90]]]
(1, 3, 3)
```

```
[137]: # Tableaux multidimensionnels (3D)

# Importer la library
import numpy as np

# Créez un tableau NumPy 3D à partir d'une liste Python imbriquée
tableau_3D = np.array([[
    [10, 20, 30],
    [40, 50, 60],
    [70, 80, 90]
]])
print(type(tableau_3D))
print(tableau_3D)
print(tableau_3D.shape)

<class 'numpy.ndarray'>
[[[10 20 30]
  [40 50 60]
  [70 80 90]]]
(1, 3, 3)
```

- Ils sont créés à l'aide de listes Python imbriquées.



## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.2. Quelques bases de python

Les bases du langage Python, les structures de données (listes, tuples, dictionnaires), les fonctions, les boucles et les conditions [Prérequis].

#### 1.3.2. Bibliothèques Python pour l'IA

##### ❖ NumPy (Numerical Python)

##### ➤ Opérations sur les tableaux Numpy

Soient 2 matrices : a et b

```
[150]: import numpy as np

# Définir 2 matrices
a = np.array([10, 20, 30, 40, 50])
b = np.array([60, 70, 80, 90, 100])
```

##### ■ Addition

```
[151]: # 1. Addition
tableau_Add = a + b
print(tableau_Add)

[ 70  90 110 130 150]
```

##### ■ Multiplication

```
[153]: # 3. Multiplication
tableau_Mul = a * b
print(tableau_Mul)

[ 600 1400 2400 3600 5000]
```

##### ■ Soustraction

```
[152]: # 2. Soustraction
tableau_Soustrac = a - b
print(tableau_Soustrac)

[-50 -50 -50 -50 -50]
```

##### ■ Division

```
[154]: # 4. Division
tableau_Div = b / a
print(tableau_Div)

[6.      3.5      2.66666667 2.25      2.      ]
```

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.2. Quelques bases de python

Les bases du langage Python, les structures de données (listes, tuples, dictionnaires), les fonctions, les boucles et les conditions [Prérequis].

#### 1.3.2. Bibliothèques Python pour l'IA

##### ❖ Pandas

Pour installer: `pip install pandas`

<https://pypi.org/project/pandas/>

- Pandas est une bibliothèque puissante pour la manipulation et l'analyse de données.
- ✓ Il propose des structures de données telles que DataFrame et Series, qui permettent une manipulation et une manipulation faciles des données structurées.
- ✓ Pandas fournit des fonctionnalités pour le nettoyage, le remodelage [reshaping], la fusion [merging], le regroupement et l'agrégation des données.

Vous trouverez ci-dessous un exemple simple d'utilisation de pandas pour travailler avec des données tabulaires:

```
[179]: import pandas as pd
# Créer un dictionnaire de données
my_data = {
    "Nom": ['KABAMBA', 'KASONGO', 'KAZADI', 'ILUNGA', 'KABULO'],
    "Age" : [24, 27, 28, 30, 32],
    "Ville": ['Kinshasa', 'Lubumbashi', 'Goma', 'Kisangani', 'Kolwezi'],
    "Profession": ['Medecin', 'Informaticien', 'Avocat', 'Professeur', 'Ingenieur']
}
```



```
[180]: my_data
[180]: {'Nom': ['KABAMBA', 'KASONGO', 'KAZADI', 'ILUNGA', 'KABULO'],
      'Age': [24, 27, 28, 30, 32],
      'Ville': ['Kinshasa', 'Lubumbashi', 'Goma', 'Kisangani', 'Kolwezi'],
      'Profession': ['Medecin', 'Informaticien', 'Avocat', 'Professeur', 'Ingenieur']}
```

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.2. Quelques bases de python

Les bases du langage Python, les structures de données (listes, tuples, dictionnaires), les fonctions, les boucles et les conditions [Prérequis].

#### 1.3.2. Bibliothèques Python pour l'IA

##### ❖ Pandas

Pour installer: `pip install pandas`

<https://pypi.org/project/pandas/>

##### ➤ Créer un DataFrame à partir du dictionnaire

```
import pandas as pd
# Créer un dictionnaire de données
my_data = {
    "Nom": ['KABAMBA', 'KASONGO', 'KAZADI', 'ILUNGA', 'KABULO'],
    "Age" : [24, 27, 28, 30, 32],
    "Ville": ['Kinshasa', 'Lubumbashi', 'Goma', 'Kisangani', 'Kolwezi'],
    "Profession": ['Medecin', 'Informaticien', 'Avocat', 'Professeur', 'Ingenieur']
}
```

```
# Créer un DataFrame à partir du dictionnaire
dataframe = pd.DataFrame(my_data)

# Afficher le dataframe
dataframe
```

```
[181]: # Créer un DataFrame à partir du dictionnaire
dataframe = pd.DataFrame(my_data)

# Afficher le dataframe
dataframe
```

```
[181]:
```

	Nom	Age	Ville	Profession
0	KABAMBA	24	Kinshasa	Medecin
1	KASONGO	27	Lubumbashi	Informaticien
2	KAZADI	28	Goma	Avocat
3	ILUNGA	30	Kisangani	Professeur
4	KABULO	32	Kolwezi	Ingenieur

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.2. Quelques bases de python

Les bases du langage Python, les structures de données (listes, tuples, dictionnaires), les fonctions, les boucles et les conditions [Prérequis].

#### 1.3.2. Bibliothèques Python pour l'IA

##### ❖ Pandas

Pour installer: `pip install pandas`

<https://pypi.org/project/pandas/>

##### ➤ Opérations de base avec DataFrame

###### ▪ Accéder aux Colonnes

```
[182]: # Opérations de base avec DataFrame
# Accéder aux colonnes
```

```
dataframe['Nom']

[182]: 0    KABAMBA
1    KASONGO
2    KAZADI
3    ILUNGA
4    KABULO
Name: Nom, dtype: object
```

```
[183]: print(dataframe[['Nom', 'Age']])
```

	Nom	Age
0	KABAMBA	24
1	KASONGO	27
2	KAZADI	28
3	ILUNGA	30
4	KABULO	32

##### Accéder aux lignes

```
[184]: # Accéder aux lignes
dataframe.iloc[2]
```

```
[184]: Nom          KAZADI
Age             28
Ville           Goma
Profession      Avocat
Name: 2, dtype: object
```

```
[185]: # Découper Les Lignes
dataframe.iloc[2:]
```

```
[185]:
```

	Nom	Age	Ville	Profession
2	KAZADI	28	Goma	Avocat
3	ILUNGA	30	Kisangani	Professeur
4	KABULO	32	Kolwezi	Ingenieur

# Programmation en Intelligence Artificielle

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.2. Quelques bases de python

Les bases du langage Python, les structures de données (listes, tuples, dictionnaires), les fonctions, les boucles et les conditions [Prérequis].

#### 1.3.2. Bibliothèques Python pour l'IA

##### ❖ Pandas

Pour installer: `pip install pandas`

##### ➤ Opérations de base avec DataFrame

##### ■ Filtrage des données en fonction de la condition

```
[186]: # Filtrage des données en fonction de la condition
```

```
[187]: dataframe[dataframe['Age'] > 29]
```

```
[187]:
```

	Nom	Age	Ville	Profession
3	ILUNGA	30	Kisangani	Professeur
4	KABULO	32	Kolwezi	Ingenieur

##### ■ Ajout d'une nouvelle colonne

```
[189]: dataframe['Genre'] = ['M','F','M','F','M']
```

```
[190]: dataframe
```

```
[190]:
```

	Nom	Age	Ville	Profession	Genre
0	KABAMBA	24	Kinshasa	Medecin	M
1	KASONGO	27	Lubumbashi	Informaticien	F
2	KAZADI	28	Goma	Avocat	M
3	ILUNGA	30	Kisangani	Professeur	F
4	KABULO	32	Kolwezi	Ingenieur	M

##### ■ Supprimer une colonne

```
[222]: # Supprimer une colonne  
dataframe.drop(columns=['Ville'], inplace=True)
```

```
[223]: dataframe
```

```
[223]:
```

	Nom	Age	Profession	Genre
0	KABAMBA	24	Medecin	M
1	KASONGO	27	Informaticien	F
2	KAZADI	28	Avocat	M
3	ILUNGA	30	Professeur	F
4	KABULO	32	Ingenieur	M

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.2. Quelques bases de python

Les bases du langage Python, les structures de données (listes, tuples, dictionnaires), les fonctions, les boucles et les conditions [Prérequis].

#### 1.3.2. Bibliothèques Python pour l'IA

##### ❖ Matplotlib

Pour installer: `pip install matplotlib`

<https://pypi.org/project/matplotlib/>

- Matplotlib est une bibliothèque de traçage largement utilisée en Python pour créer des visualisations statiques, interactives et de bonne qualité.
- ✓ Il fournit une interface de type MATLAB et prend en charge divers types de tracés, notamment les tracés linéaires, les nuages de points, les tracés à barres, les histogrammes.

Exemple : **numpy.linspace()** est une fonction de la bibliothèque **NumPy** qui génère un tableau de nombres régulièrement espacés sur un intervalle spécifié.

- Paramètres:
- ✓ **start** : La valeur de départ de la séquence.
- ✓ **stop** : la valeur de fin de la séquence (incluse par défaut, sauf si le point final est défini sur False).
- ✓ **num** : Le nombre d'échantillons à générer (la valeur par défaut est 50).

```
import numpy as np

# Générez un tableau de 20 nombres régulièrement espacés entre 0 et 1
tableau = np.linspace(0, 1, num=20)
print(type(tableau))
print(tableau)

<class 'numpy.ndarray'>
[0.          0.05263158 0.10526316 0.15789474 0.21052632 0.26315789
 0.31578947 0.36842105 0.42105263 0.47368421 0.52631579 0.57894737
 0.63157895 0.68421053 0.73684211 0.78947368 0.84210526 0.89473684
 0.94736842 1.          ]
```

## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.2. Quelques bases de python

Les bases du langage Python, les structures de données (listes, tuples, dictionnaires), les fonctions, les boucles et les conditions [Prérequis].

#### 1.3.2. Bibliothèques Python pour l'IA

##### ❖ Matplotlib

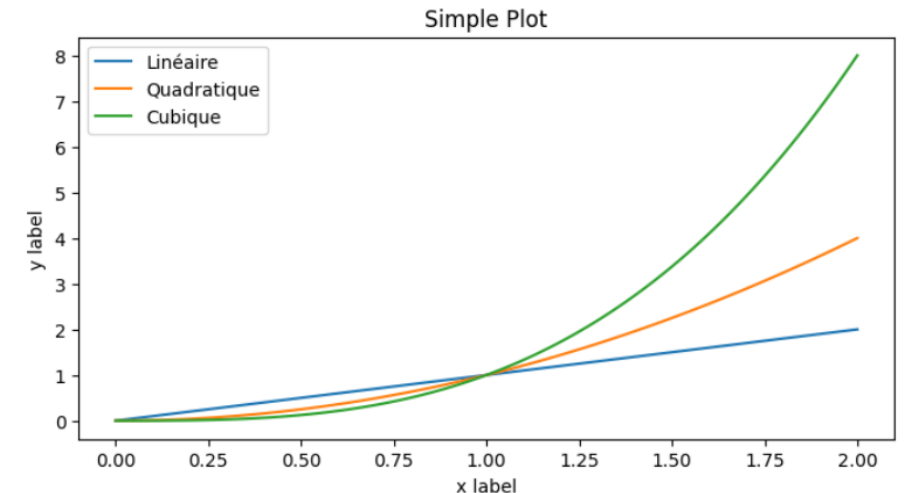
Pour installer: `pip install matplotlib`

<https://pypi.org/project/matplotlib/>

- Exemple : Nous allons tracer trois lignes représentant les fonctions linéaires, quadratiques et cubiques de  $x$ , ajouter des étiquettes aux axes, définir un titre pour le tracé et ajouter une légende pour distinguer les lignes.

[368]: <matplotlib.legend.Legend at 0x1f705478640>

```
[368]: 1 import matplotlib.pyplot as plt
2
3 x = np.linspace(0, 2, 100)
4
5 # Notez que même dans le style OO, nous utilisons `.pyplot.figure` pour créer la figure.
6 fig, ax = plt.subplots(figsize=(8, 4)) # largeur, hauteur
7 ax.plot(x, x, label='Linéaire') # Tracez des données sur les axes.
8 ax.plot(x, x**2, label='Quadratique')
9 ax.plot(x, x**3, label='Cubique')
10 ax.set_xlabel('x label') # Ajoute une étiquette x aux axes.
11 ax.set_ylabel('y label') # Ajoute une étiquette y aux axes.
12 ax.set_title('Simple Plot')
13 ax.legend()
```





## CHAPITRE 1 INTRODUCTION A L'INTELLIGENCE ARTIFICIELLE



### 1.3 Python pour l'IA : Python Fundamentals

#### 1.3.2. Quelques bases de python

Les bases du langage Python, les structures de données (listes, tuples, dictionnaires), les fonctions, les boucles et les conditions [Prérequis].

#### 1.3.2. Bibliothèques Python pour l'IA

##### ❖ Seaborn

Pour installer: `pip install matplotlib`

<https://pypi.org/project/seaborn/>

- Seaborn est une bibliothèque de visualisation de données statistiques basée sur Matplotlib.
- ✓ Il fournit une interface de haut niveau pour créer des graphiques statistiques attrayants et informatifs.
- ✓ Seaborn simplifie la création de visualisations complexes telles que des tracés catégoriels, des tracés de distribution et des tracés de régression.

```
[376]: 1 import seaborn as sns
      2 import matplotlib.pyplot as plt
      3 # Charger un exemple d'ensemble de données depuis Seaborn
      4 iris = sns.load_dataset('iris')
      5 # Créez un nuage de points en utilisant Seaborn
      6 sns.scatterplot(x='sepal_length', y='sepal_width', data=iris, hue='species')
      7 # Ajouter des étiquettes et un titre
      8 plt.xlabel('Longueur des sépales')
      9 plt.ylabel('Largeur des sépales')
     10 plt.title('Longueur des sépales vs largeur des sépales')
     11 # Affichage
     12 plt.show()
     13
```

