

# PRÉSENTATION DU PROJET DE DEEP LEARNING

**Classification d'images**

Professeur : **M. TONDJI**





**AIDARA**  
Habib



**BAZIE**  
René Lothaire



**SANOGO**  
Steve

# Équipe

# SOMMAIRE

I

- DEEP LEARNING, C'EST QUOI?

II

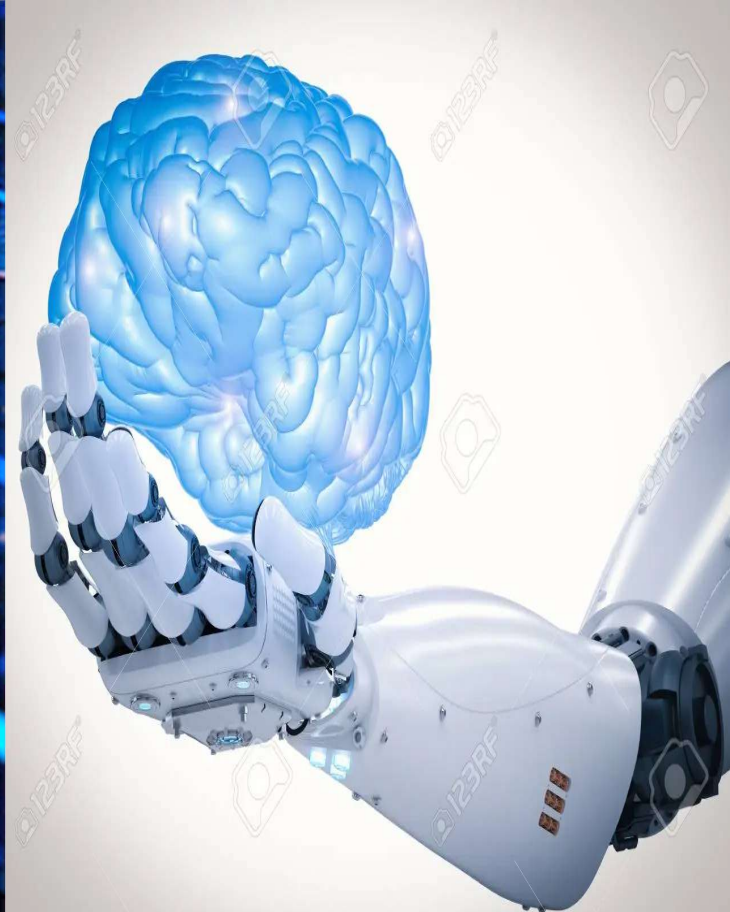
- LES COMPOSANTS DE BASE

III

- LES HYPERPARAMETRES

IV

- LES ALGORITHMES D'OPTIMISATION



# SOMMAIRE

V

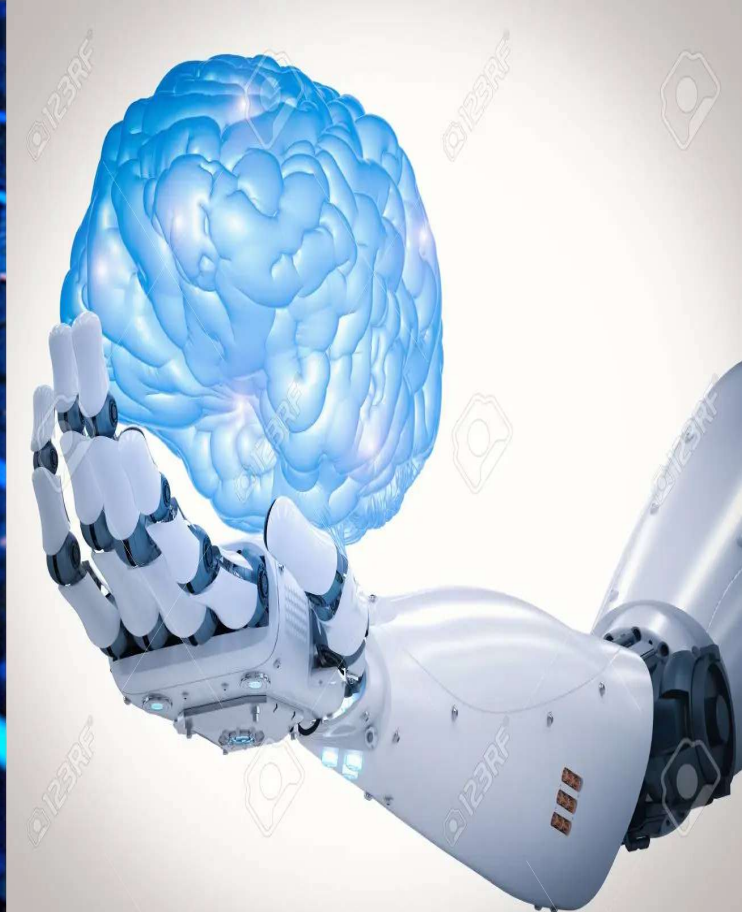
- PRINCIPE DU CNN

VI

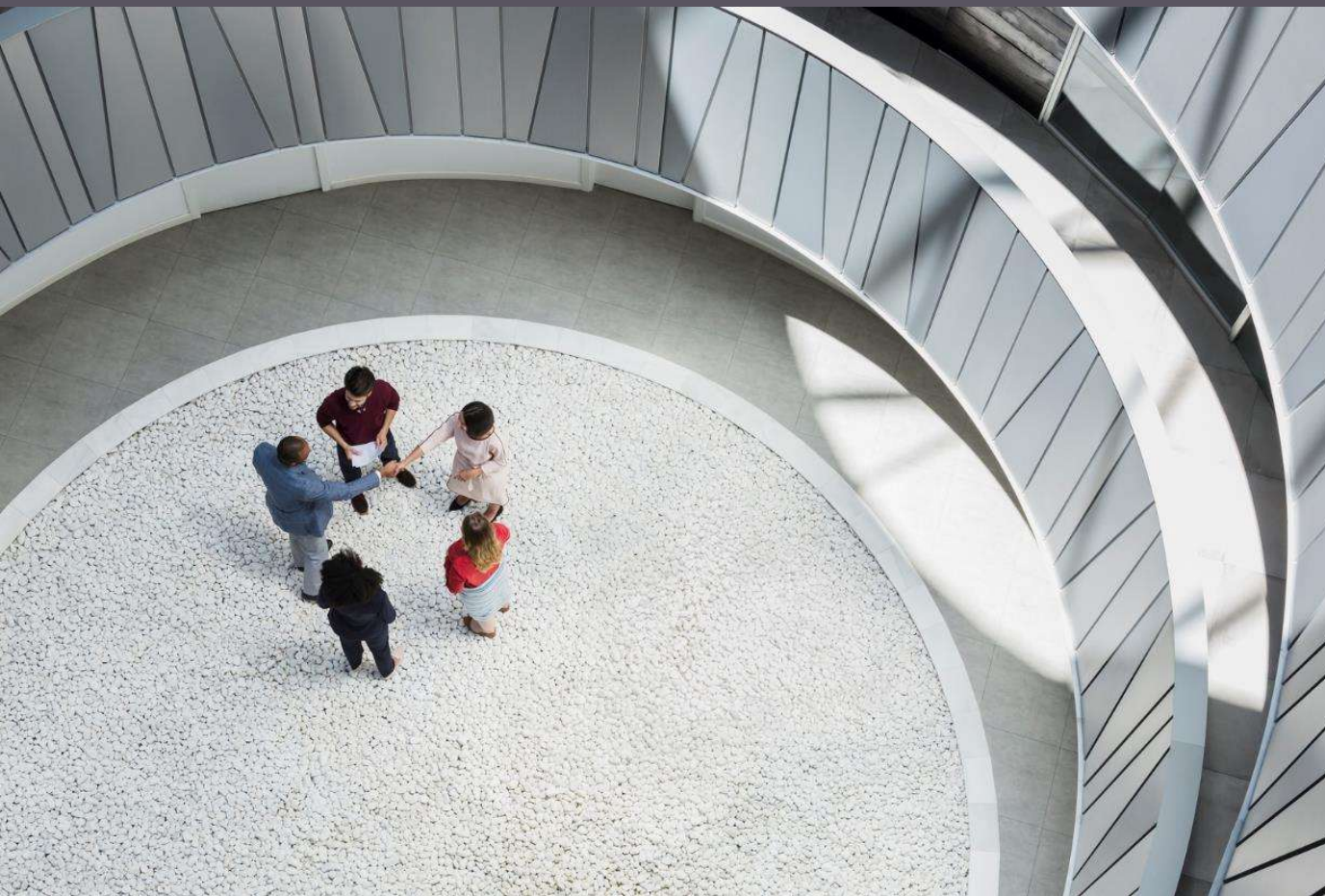
- LES MODELES PRE-ENTRAINES

VII

- IMPLEMENTATION





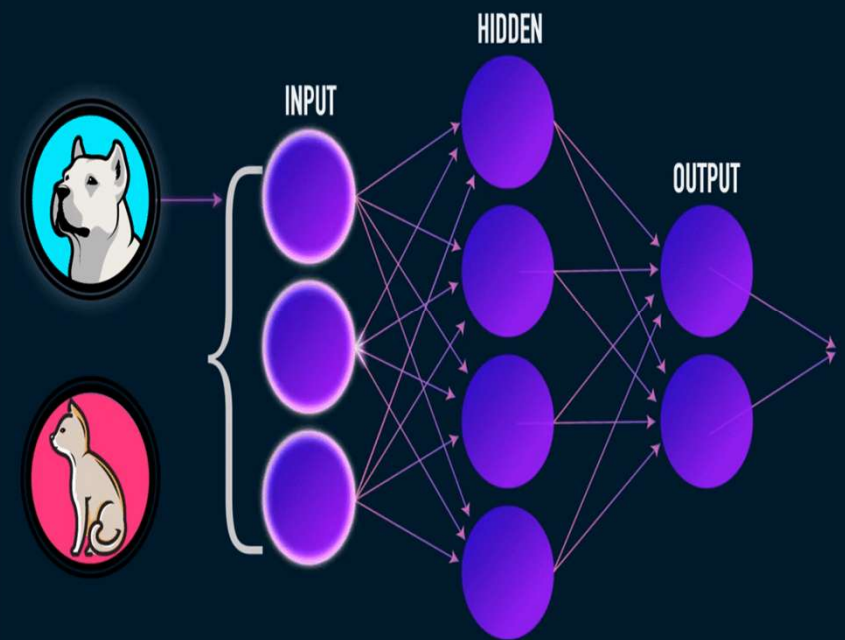


# Introduction

# Introduction

L'apprentissage profond est actuellement très important car il est plus performant que d'autres algorithmes d'apprentissage automatique pour des tâches complexes :

- ❖ Le traitement d'images et reconnaissance de formes;
- ❖ La reconnaissance vocale et le traitement du signal;
- ❖ Les technologies de télécommunications.





# DEFINITION DU DEEP LEARNING



# QU'EST-CE QUE LE DEEP LEARNING ?

- ❑ L'apprentissage profond est une technologie qui consiste à enseigner aux ordinateurs ce que les humains sont naturellement capables de savoir.
- ❑ Avec le Deep Learning, nous parlons d'algorithmes capables de mimer les actions du cerveau humain grâce à des réseaux de neurones artificielles.



- ❑ Ces techniques ont permis de réaliser des progrès importants et rapides dans les domaines de l'analyse des signaux sonores, de la reconnaissance vocale, de la vision par ordinateur et du traitement automatique des langues.

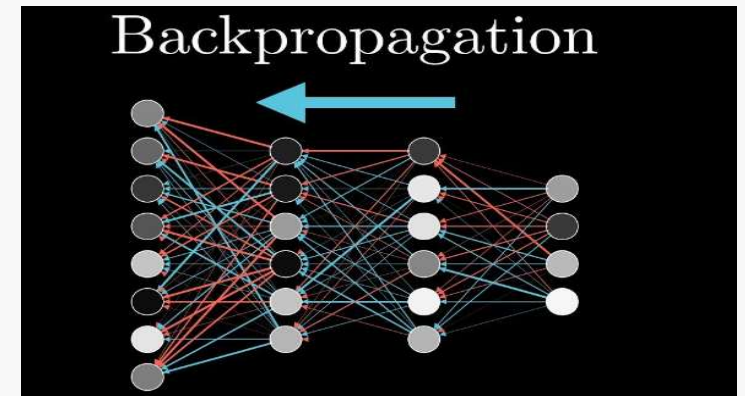
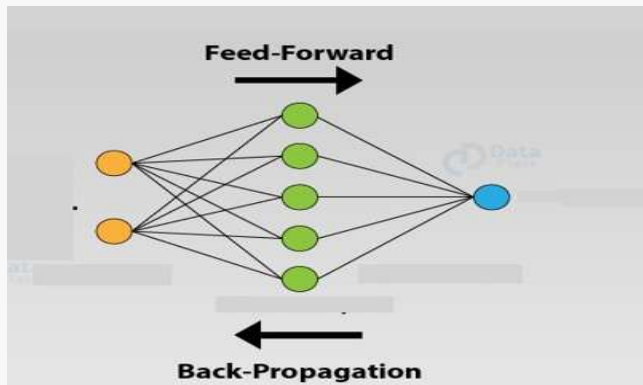


# COMMENT FONCTIONNE LE DEEP LEARNING ?

- ❑ Les réseaux neuronaux d'apprentissage profond, ou réseaux neuronaux artificiels, tentent d'imiter le cerveau humain par une combinaison d'entrées de données, de pondérations et de biais. Ces éléments fonctionnent ensemble pour reconnaître, classer et décrire avec précision les objets présents dans les données.
- ❑ Les réseaux neuronaux profonds se composent de plusieurs couches de nœuds interconnectés, chacune s'appuyant sur la couche précédente pour affiner et optimiser la prédiction ou la catégorisation. Cette progression des calculs dans le réseau est appelée propagation vers l'avant.

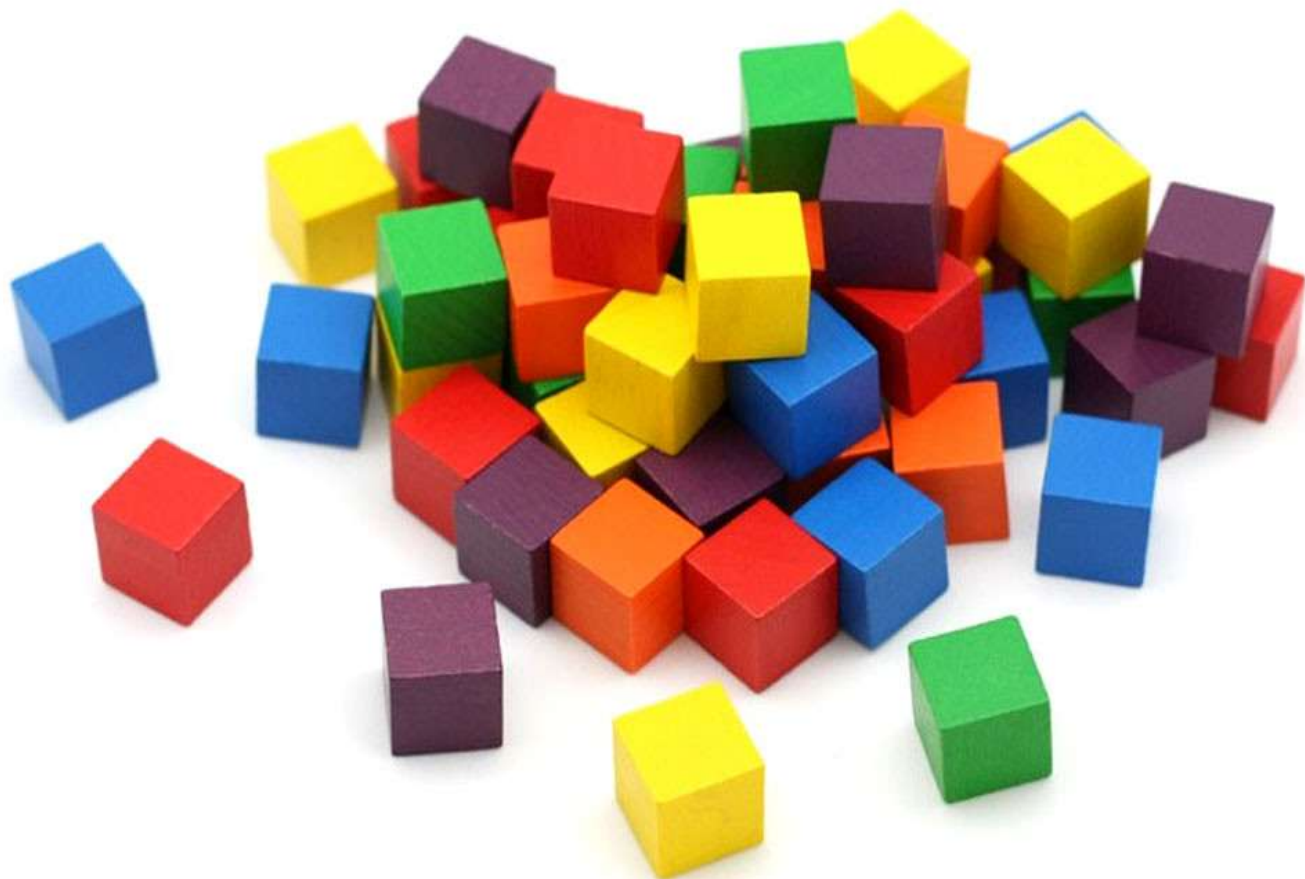
# COMMENT FONCTIONNE LE DEEP LEARNING ?

- ❑ Un autre processus appelé rétropropagation utilise des algorithmes, comme la descente de gradient, pour calculer les erreurs dans les prédictions, puis ajuste les poids et les biais de la fonction en remontant les couches dans le but d'entraîner le modèle.



- ❑ Ensemble, la propagation avant et la rétropropagation permettent à un réseau neuronal de faire des prédictions et de corriger les erreurs en conséquence. Au fil du temps, l'algorithme devient progressivement plus précis.

# LES COMPOSANTS DE BASE





# LES COMPOSANTS DE BASE



INPUTS

WEIGHTS

BIAIS

OUTPUT

ACTIVATION  
FUNCTION

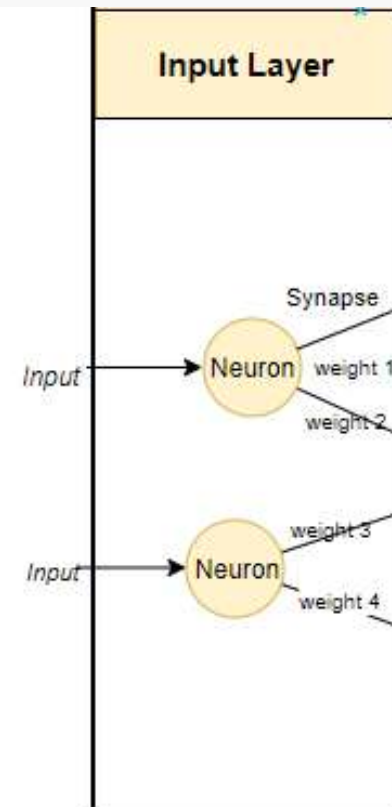
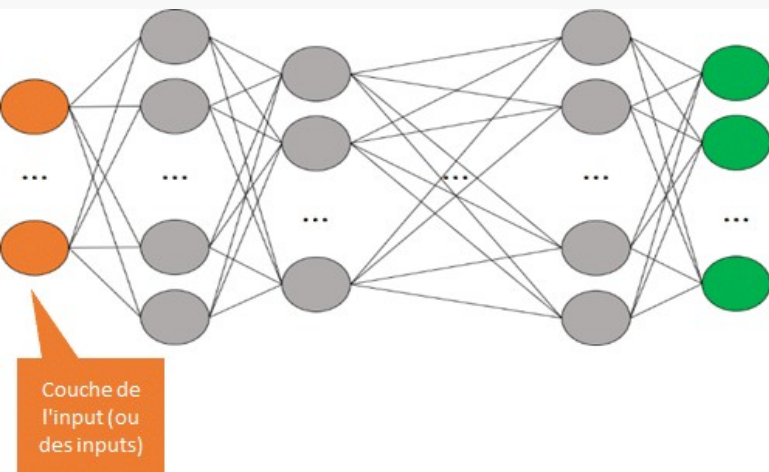




# INPUTS



Il s'agit d'un composant principal qui accepte les données initiales dans le système pour un traitement. Le contenu de cette couche varie en fonction du contexte, du cas d'usage concerné par le réseau de neurones.



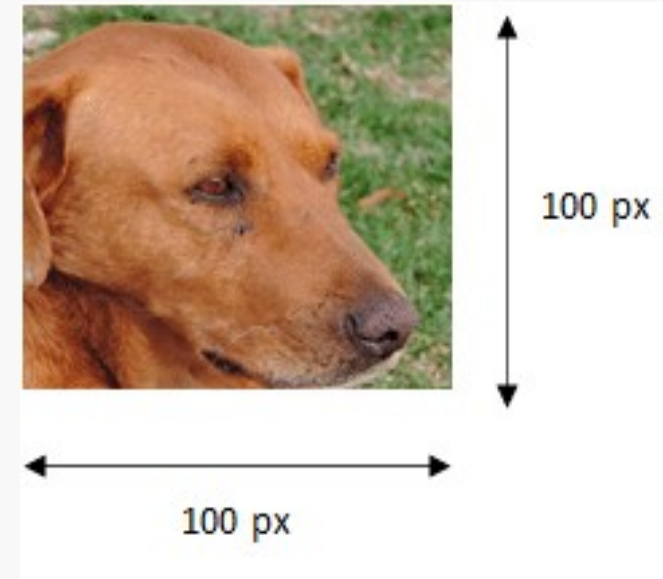


# 1er contexte : la reconnaissance d'images



## 1.1. L'image en couleur

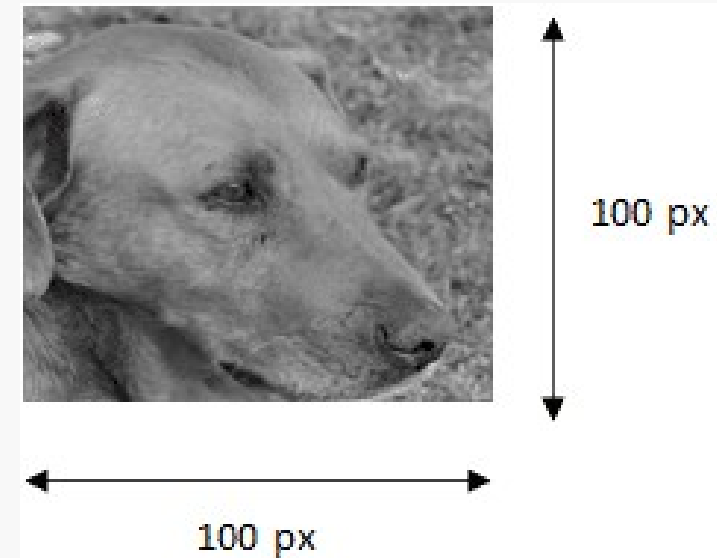
Cette image a une taille de 100 px sur 100 px. Un pixel contient 3 valeurs (R, G, B). Cette image est donc composée de  $100 \times 100 \times 3 = 30\,000$  valeurs. Ces 30 000 valeurs vont représenter notre input layer (couche d'entrée). Il y a aura donc 30 000 neurones dans la première couche.



# 1er contexte : la reconnaissance d'images

## 1.2. L'image en noir et blanc

Cette image a une taille de 100 px sur 100 px. Un pixel contient 1 valeur comprise entre 0 et 1 (0 = blanc, 1 = noir, gris = 0,5...). Cette image est donc composée de  $100 \times 100 = 10\,000$  valeurs. Ces 10 000 valeurs vont représenter notre input layer (couche d'entrée). Il y aura donc 10 000 neurones dans la première couche.





## 2<sup>ème</sup> contexte : la couleur d'une fleur



Dans cette situation, notre réseau pourrait avoir en entrée  
2 inputs (1<sup>er</sup> input = longueur, 2<sup>ème</sup> input = largeur).







# WEIGHTS AND BIAIS



Les poids et biais sont des variables du modèle qui sont mises à jour pour améliorer la précision du réseau.

- ❑ Un poids est appliqué à l'entrée de chacun des neurones pour calculer une donnée de sortie. Les réseaux de neurones mettent à jour ces poids de manière continue. Il existe donc une boucle de rétroaction mise en œuvre dans la plupart des réseaux de neurones.



# WEIGHTS AND BIAIS



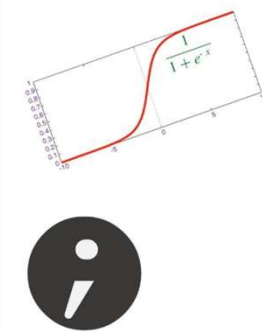
- ❑ Les biais sont également des valeurs numériques qui sont ajoutées une fois que les poids sont appliqués aux valeurs d'entrée. Les poids et les biais sont donc en quelque sorte des valeurs d'auto-apprentissage de nos réseaux de neurones.



# ACTIVATION FUNCTIONS



Une fonction d'activation est une formule mathématique (algorithme) activée dans certaines circonstances. Lorsque les neurones calculent la somme pondérée des valeurs d'entrée + le biais, elles sont transmises à la fonction d'activation, qui vérifie si la valeur calculée est supérieure au seuil requis.



**Activation  
Functions**

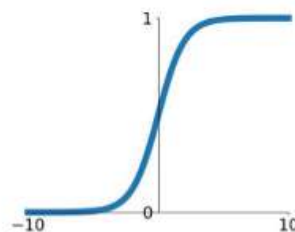
Si la valeur calculée est supérieure au seuil requis, la fonction d'activation est activée et une valeur de sortie est calculée.

Les fonctions d'activation introduisent la non-linéarité dans les réseaux de neurones, nécessaire pour résoudre des problèmes complexes.

# LES DIFFERENTS TYPES DE FONCTIONS D'ACTIVATION

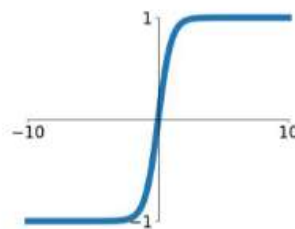
## Sigmoid

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



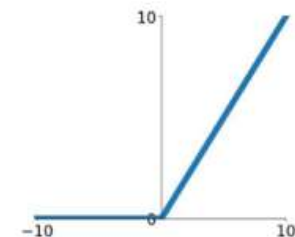
## tanh

$$\tanh(x)$$



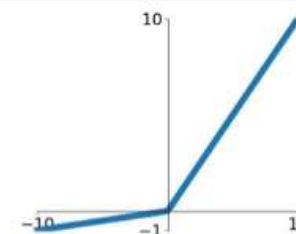
## ReLU

$$\max(0, x)$$



## Leaky ReLU

$$\max(0.1x, x)$$

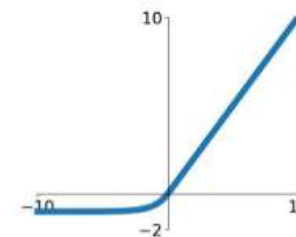


## Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

## ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

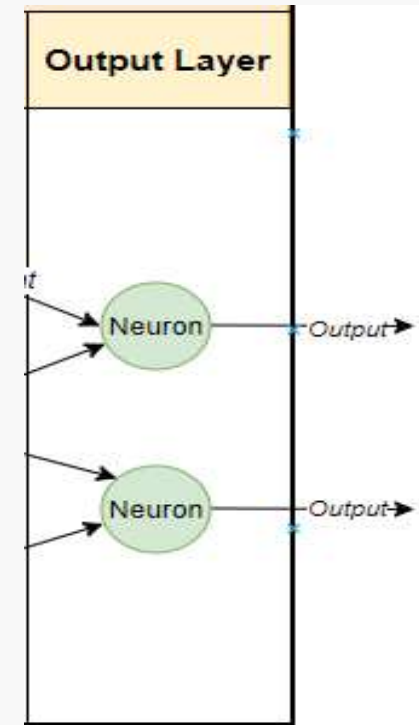
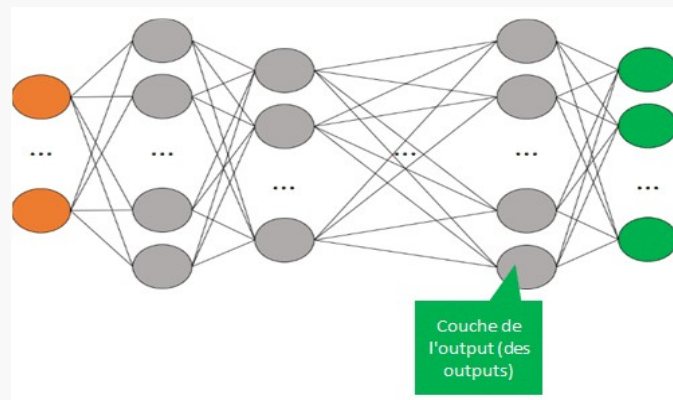




# OUTPUT

La couche de sortie est responsable de la production du résultat final. Il doit toujours y avoir une couche de sortie dans un réseau de neurones.

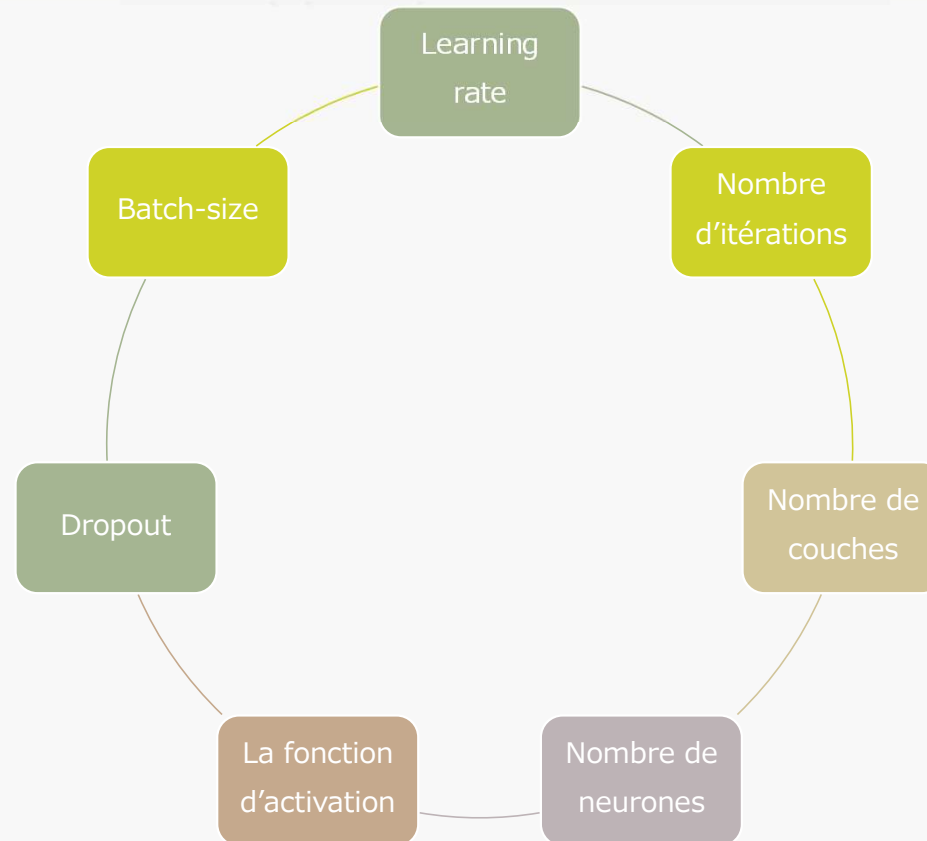
La couche de sortie reçoit les entrées transmises par les couches qui la précèdent, effectue les calculs via ses neurones, puis la sortie est calculée.



# LES HYPERPARAMETRES



# Hyperparamètres



# LES ALGORITHMES D'OPTIMISATION







# LES ALGORITHMES D'OPTIMISATION



## Descente du gradient:

La descente de gradient est un algorithme d'optimisation itératif du premier ordre pour trouver le minimum d'une fonction.

- |              |                 |                  |
|--------------|-----------------|------------------|
| 1. par batch | 2. stochastique | 3. avec momentum |
| 4. Adagrad   | 5. RMSprop      | 6. Adam          |

# PRINCIPE DU CNN





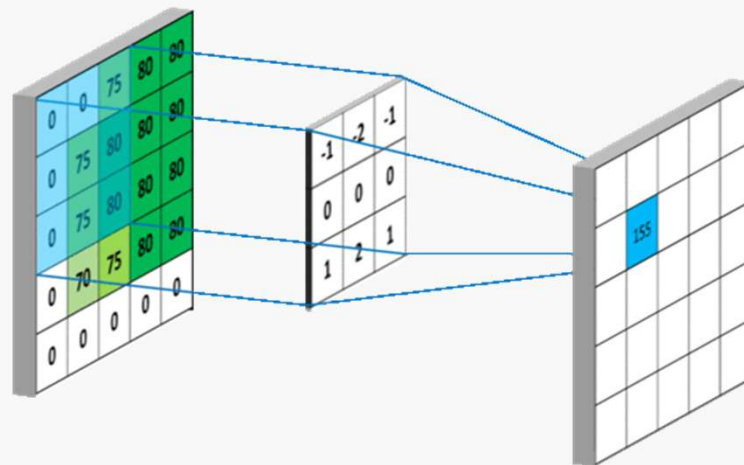
# Convolutional Neural Network



Les réseaux de neurones convolutionnels sont un type de réseau de neurones profond utilisé pour la reconnaissance d'images et d'autres tâches liées à l'analyse d'images.

En 1998, Yann LeCun et ses collègues ont présenté le premier modèle de réseau de neurones convolutionnel pour la reconnaissance de chiffres manuscrits.

En 2012, AlexNet, un modèle de CNN, a gagné la compétition ImageNet avec une meilleure précision de 15% par rapport à la deuxième place, démontrant la puissance des CNN pour la reconnaissance d'images et stimulant la recherche dans ce domaine.





## Composant d'un CNN

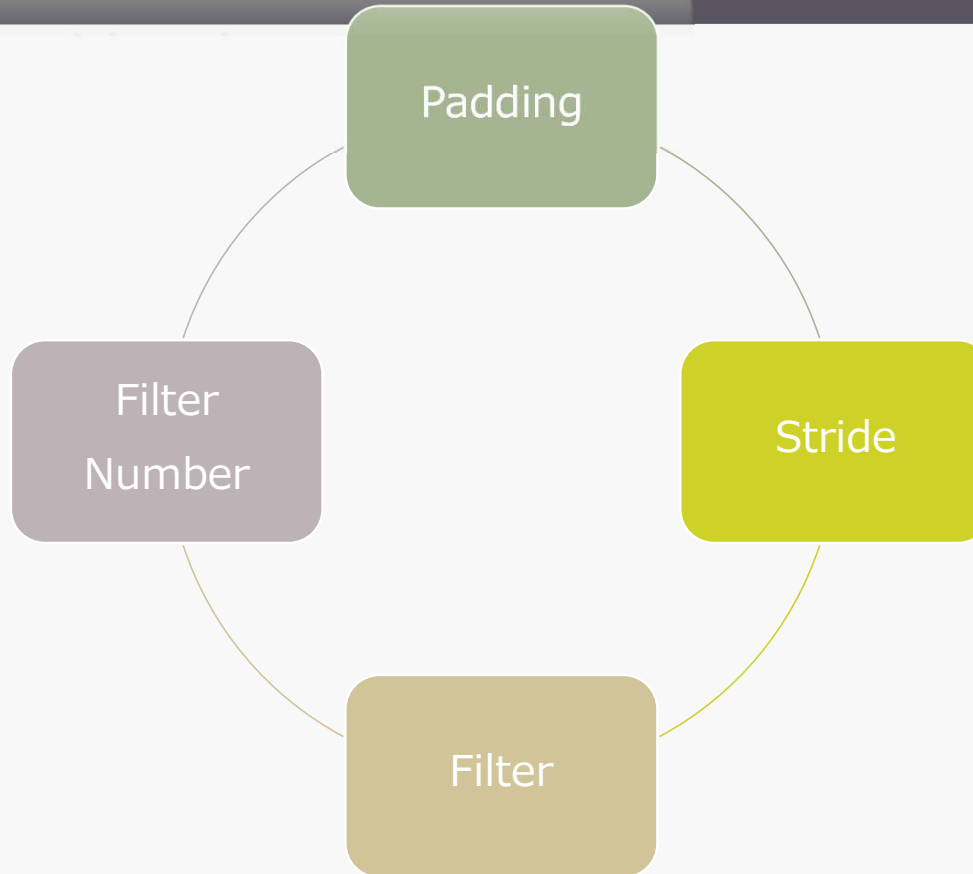


- ❑ **Convolution layer:** C'est la couche principale qui effectue la convolution des entrées avec des filtres pour extraire des caractéristiques.
- ❑ **ReLU layer:** C'est la couche d'activation qui ajoute une non-linéarité au réseau pour rendre le modèle plus flexible.
- ❑ **Pooling layer:** C'est la couche qui effectue un pooling spatial qui réduit la dimensionnalité des caractéristiques extraites.
- ❑ **Fully connected layer:** C'est la couche qui connecte tous les neurones d'une couche précédente pour produire un vecteur d'activation pour la couche suivante.
- ❑ **Output layer:** C'est la couche de sortie qui produit la prédiction finale du réseau.

**On peut aussi rencontré des couches de normalisation et des couches de dropout.**

# Hyperparamètres

**Common NN  
hyperparameters**



# LES MODELES PRE-ENTRAINES







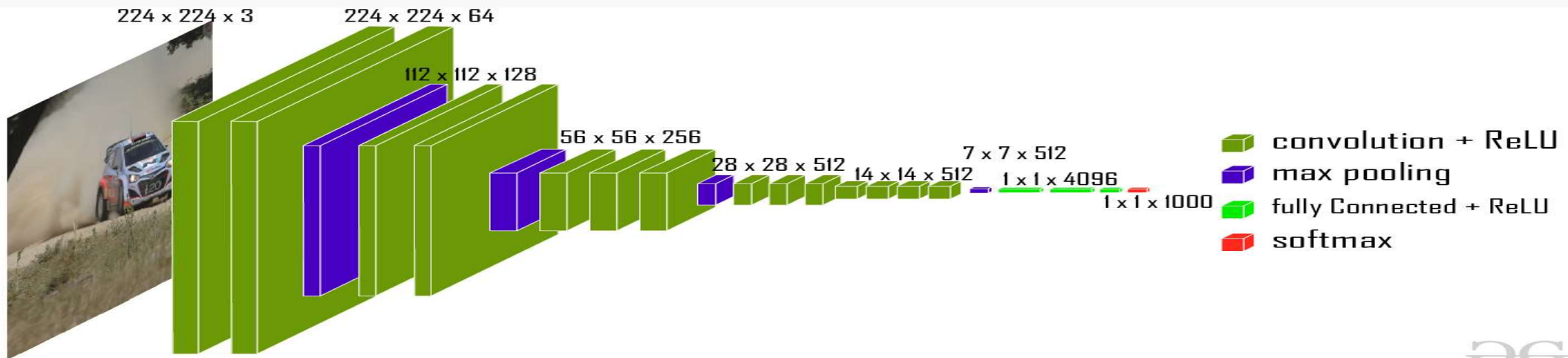
# Modèles pré-entraînés



- ❑ Un modèle préformé fait référence à un modèle ou à un réseau enregistré créé par quelqu'un d'autre et formé sur un grand ensemble de données pour résoudre un problème similaire.
- ❑ Il existe plusieurs modèles préformés utilisés. Chacun d'eux atteint des niveaux de performance différents et est utilisé pour différentes tâches. Voici quelques exemples bien connus :
  - ❑ VGG-16
  - ❑ ResNet50
  - ❑ EfficientNet

# VGG-16

S'appuyant sur les travaux d' AlexNet , VGG se concentre sur un autre aspect crucial des réseaux de neurones convolutifs (CNN), la profondeur. Il a été développé par Simonyan et Zisserman. Il se compose normalement de 16 couches convolutives.






## VGG-16



VGG-16 comporte principalement trois parties : convolution, pooling et fully connected layers.

- ❑ **Couche de convolution** → Dans cette couche, des filtres sont appliqués pour extraire les caractéristiques des images.
- ❑ **Couche de regroupement** (pooling) → Sa fonction est de réduire la taille spatiale pour réduire le nombre de paramètres et de calculs dans un réseau.
- ❑ **fully connected layers** → Ce sont des connexions entièrement connectées aux couches précédentes comme dans un simple réseau de neurones.



**LA MEILLEURE  
MANIÈRE DE  
COMMENCER,  
C'EST  
D'ARRÊTER DE  
PARLER ET DE  
S'Y METTRE.**

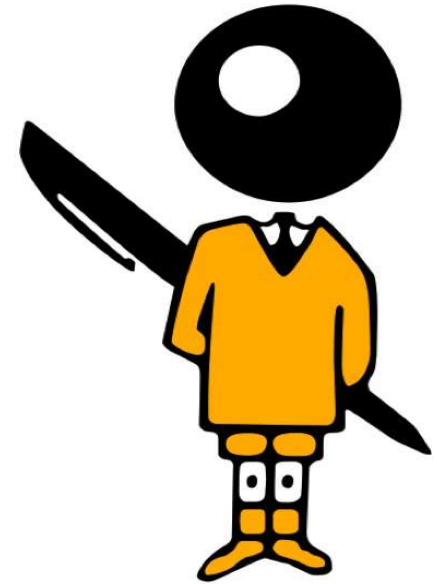
**Walt Disney**

Projet de DEEP LEARNING

3/02/2023

34

**MERCI !**



Projet de DEEP LEARNING

3/02/2023

**35**

# PRÉSENTATION DU PROJET DE DEEP LEARNING

**Classification d'images**

Professeur : **M. TONDJI**

