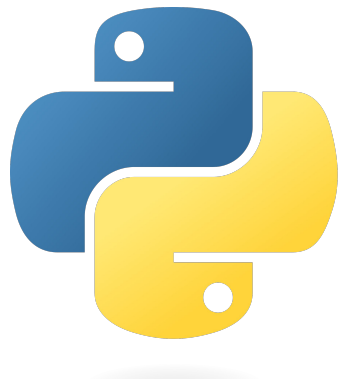


Python avancé

Revoir les bases du langage

Découvrir des notions avancées du langage

Introduction à la data-science et au machine learning avec Python



Intervenant

Alexandre Chevalier - formateur depuis 2018

- Ingénieur développement web
 - Depuis plus de 9 ans
 - En ESN et en start-up
 - 100% freelance depuis 2021
- Compétences principales :
 - Back **Java** (Spring)
 - Front **JavaScript** (Vue, React)



dev.alexandre.chevalier@gmail.com



<https://www.linkedin.com/in/alexandre-chevalier-610555ba>



alexandrechevalier.fr



Contacts



dev.alexandre.chevalier@gmail.com



4lexandre#8029



GitHub

@AlexandreChevalier



GitLab

@alexandre.chevalier

Références et ressources

- Documentation officielle
 - <https://docs.python.org/fr/3/>
- Documentations simples pour Python, ML, numpy, Django, etc.
 - <http://www.python-simple.com/>
- Cours Python pour la data-science
 - <https://www.pythonds.linogaliana.fr/>
- Actualité Python (payant)
 - <https://realpython.com/>

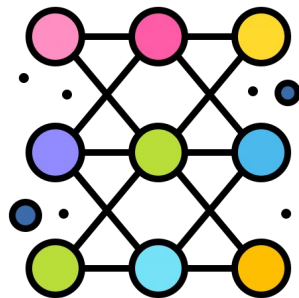
Références ML

- Vidéo d'introduction au *machine learning*
 - <https://www.youtube.com/watch?v=UGPOauhD7Ks>
- Cours sur OpenClassrooms
 - <https://openclassrooms.com/fr/courses/4011851-initiez-vous-au-machine-learning>
- Les bases du *machine learning*
 - <https://machinelearnia.com/machine-learning/>
 - Avec Python : <https://machinelearnia.com/python-machine-learning/>
- Introduction au ML avec Python
 - <https://www.data-transitionnumerique.com/machine-learning-python/>
- Modèles ML
 - <https://moncoachdata.com/blog/modeles-de-machine-learning-expliques/>
- Apprentissage supervisé ou non
 - <https://experiences.microsoft.fr/articles/intelligence-artificielle/apprentissage-supervise-et-non-supervise-quelles-differences/>
- Classification
 - <https://datascientest.com/algorithmes-de-classification-definition-et-principaux-modeles>
- Deep Learning
 - <https://datascientest.com/deep-learning-definition>
 - [https://www.wikiwand.com/fr/Apprentissage profond](https://www.wikiwand.com/fr/Apprentissage_profond)



Objectifs de la semaine

- Approfondir Python pour la **data-science**
- Suivre le cours suivant :
 - <https://www.pythonds.linogaliana.fr/>
- Découvrir des notions de **machine learning**
- Tester des librairies comme scikit-learn (ou autres) :
 - <https://scikit-learn.org/stable/>



Machine Learning

Références et ressources

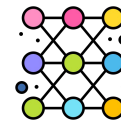
Machine Learning

- Introduction au machine learning
 - <https://www.youtube.com/watch?v=PeMlggyqz0Y>

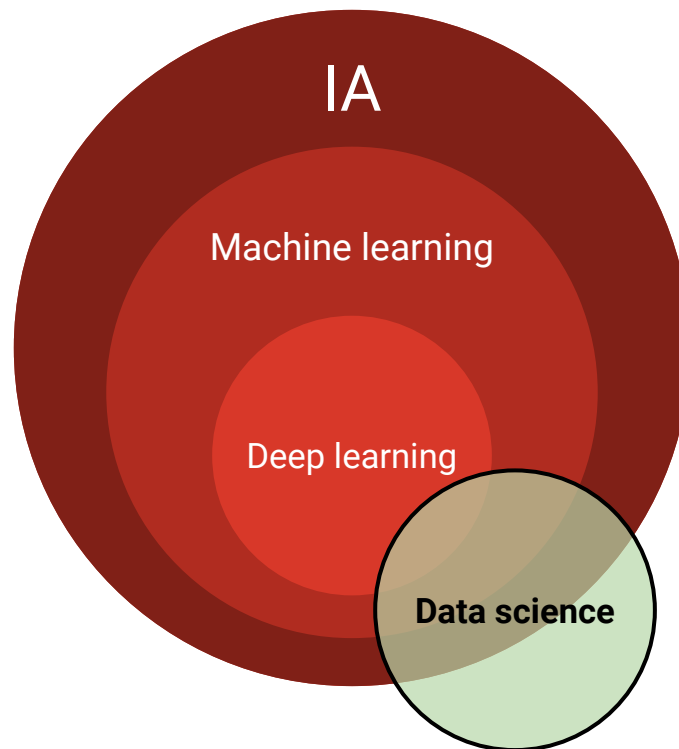
Deep Learning

- Vidéo de vulgarisation sur le deep learning
 - <https://www.youtube.com/watch?v=trWrEWfhTVg>
- Formation en vidéo sur le deep learning
 - https://www.youtube.com/playlist?list=PLO_fdPEVIfKoanjvTJblbd9V5d9Pzp8Rw

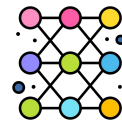
Machine learning ou deep learning ?



- Intelligence artificielle
 - Ensemble des techniques permettant de simuler l'intelligence du vivant
- Machine learning
 - Apprentissage automatique
- Deep learning
 - Spécialisation du ML en utilisant les réseaux de neurones

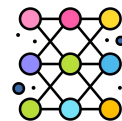


Pourquoi faire du ML ?



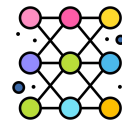
- Quand les données sont abondantes (relativement)
- Mais les connaissances peu accessibles ou peu développées
- **Classification**
 - Construire un modèle de classement
 - Classer des données dans des catégories
- **Prédiction**
 - Modéliser sur des valeurs connues
 - “Deviner” quelles pourraient être des valeurs inconnues

A quoi ça sert ?



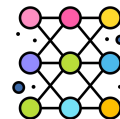
- Objectif global : généraliser à partir du jeu de test
- **Classification** : classer des données
 - Cette image représente tel animal
 - Détection de visages sur une photo
 - Ce texte parle de tel sujet
- **Prédiction** : imaginer un résultat
 - Autocomplétion de ChatGPT
 - Ce dessin ressemble à telle photo
 - Recommendations Youtube ou Tiktok

Notions de base



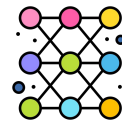
- Les algorithmes d'apprentissage
 - Les procédures qu'on **entraîne sur les données**
 - Pour produire **un modèle**
- Les **données** :
 - Des exemples pour entraîner les algorithmes
- L'**entraînement** :
 - Exécuter un algorithme d'apprentissage
 - Sur un jeu de données exemple

Un modèle



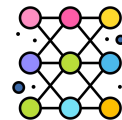
- L'objectif du ML est d'obtenir un **modèle**
- Un modèle c'est quoi ?
 - Un algorithme customisé (entraîné)
 - Il est entraîné (customisé) sur des données d'exemples
 - Pour apprendre les relations entre :
 - Les données d'entrée
 - Les données en sortie
 - Quand on lui donnera une nouvelle entrée :
 - Il **prédira** un résultat probable
 - D'après ce sur quoi il a été entraîné

Quels types de problèmes ?



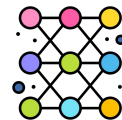
- Qu'on ne sait pas résoudre classiquement
- Qu'on sait résoudre
 - Mais on ne sait pas formaliser les algorithmes
- Qu'on sait résoudre
 - Mais les techniques classiques sont trop lourdes
 - Trop gourmandes en ressources
 - Exemple : simulations physiques

Programme classique



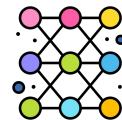
- Un programme :
 - En entrée : des données
 - En sortie : des résultats
 - C'est comme une fonction
- Programmé explicitement pour produire ce résultat
 - Il n'est pas capable de s'adapter à un nouveau contexte

Apprentissage



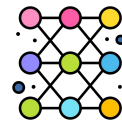
- Modification d'un comportement
 - Sur la base d'une expérience
- Apprentissage automatique
- On souhaite faire un programme qui apprend automatiquement
 - Sur une base de données exemple
- Il modifiera son comportement
 - En se basant sur des exemples
 - Mais aussi sur les résultats produits

Machine Learning



- Un programme “apprend” automatiquement
 - Sans être explicitement programmé pour
- Sur des tâches **précises** et **simples**
- A partir de **données** sur lesquelles **s’entraîner**
- C’est une boîte noire
 - On ne sait pas expliciter mais on sait construire
 - Voir plus tard réseaux de neurones en *deep learning*
- Problème mathématique d’**optimisation** (descente de gradient)

Machine Learning

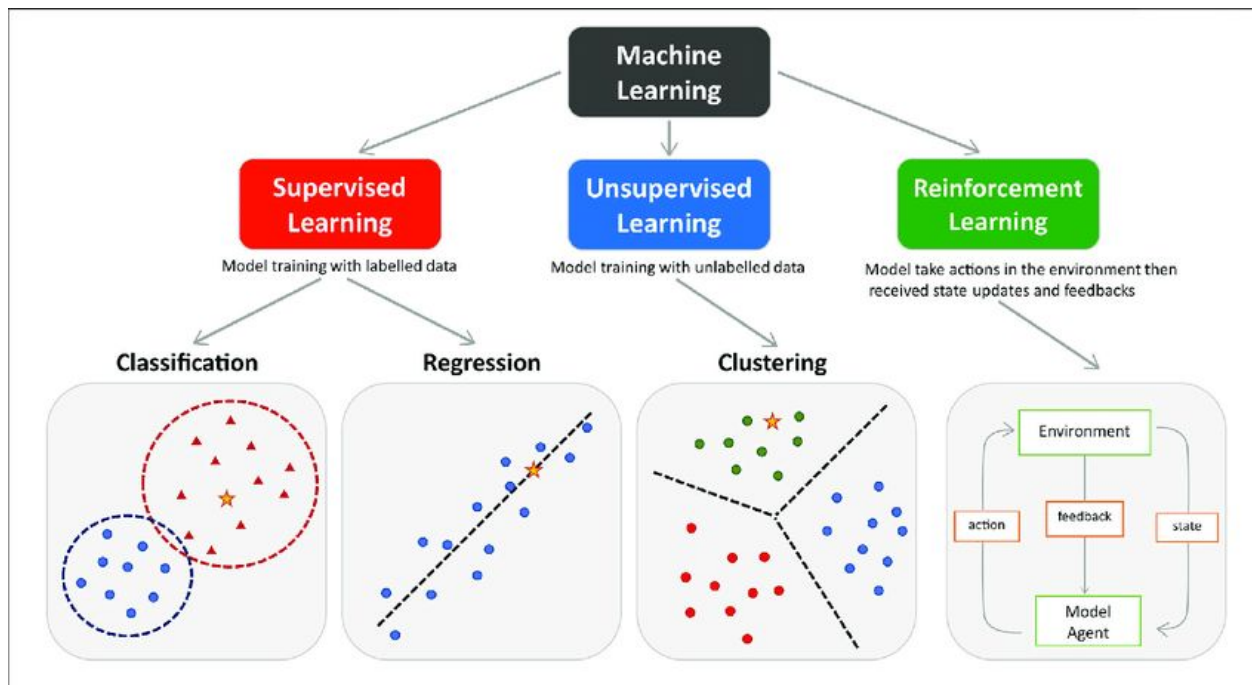


- Branche de l'IA
 - Il n'y a pas d'intelligence
 - Simulation d'intelligence
 - Une IA ne sait pas inventer
 - Tout ce qu'elle sait, elle l'a déjà vu
- Exemples connus :
 - Reconnaissance d'images
 - Traitement du langage naturel
 - Conduite autonome
 - ...

Types de machine learning

- Plusieurs sous-catégories

Voir <https://moncoachdata.com/blog/modeles-de-machine-learning-expliques/>



Apprentissage supervisé ou non

- Apprentissage supervisé

- Des exemples annotés (étiquetés ou labellisés)
- Sont la base d'apprentissage
- But : **généraliser** à partir d'exemples
- Régression et classification



- Apprentissage non supervisé

- Données non annotées
- But : **découvrir des structures** dans ces données
- Plus complexes et moins prévisibles (voire inattendu ou indésirable)
- Association et clustering



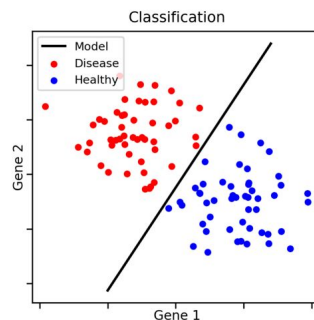
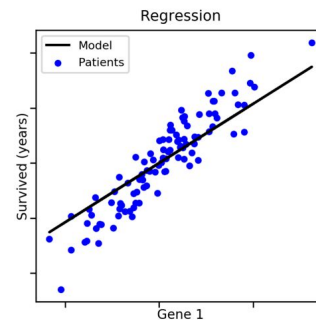
- Apprentissage semi-supervisé

- Avec des données étiquetées et non étiquetées

Apprentissage supervisé

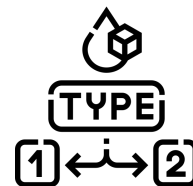


- Apprendre une fonction de prédiction
 - A partir d'exemples annotés (supervision)
- Deux grandes familles de problèmes
 - Problèmes de **régression**
 - Trouver une valeur numérique
 - Problèmes de **classification** (ou de classement)
 - Classer dans des catégories
- **Régression** → variables **quantitatives**
- **Classification** → variables **qualitatives**



Qualitative ou quantitative ?

- **Classification** : trouver la variable **qualitative**
 - Définit une qualité, une caractéristique, un attribut ontologique
 - Répond à une question du style : c'est **quoi** ?
 - Exemples : espèce animale, forme, type d'aliment, ...
- **Régression** : trouver la variable **quantitative**
 - Définit une quantité, un attribut numérique
 - Répond à une question du style : c'est **combien** ?
 - Exemples : prix, quantité, surface, date, altitude, ...

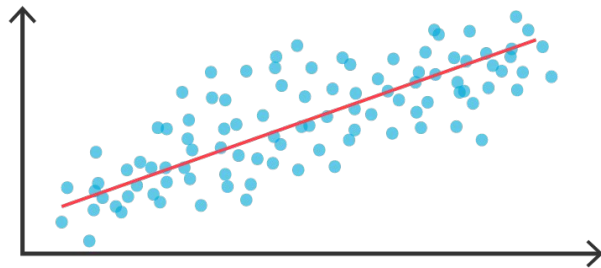


Classification

- Famille d'algorithmes permettant de **classer** des données
 - Dans plusieurs catégories
- Beaucoup de modèles différents
- Eager learner (impatient) :
 - Le modèle est entraîné sur la donnée dès le début
 - Exemples : arbres de décision, SVM, réseaux de neurones
 - Use cases : reconnaissance d'images, classifieurs classiques
- Lazy learner (paresseux) :
 - Le modèle s'entraîne sur de la donnée entrante
 - Exemples : K-means, KNN, perceptron
 - Use cases : recommandations Netflix ou Amazon, détection de spam...

Régression

- Méthode d'**approximation** d'une donnée à partir d'exemples
 - Mathématiquement avec la Méthode des moindres carrés
- La plus connue : la régression linéaire
 - Dans un ensemble de points : **trouver une droite**
 - Qui passe le mieux et en moyenne par le plus de points



Régression linéaire

- Exemple : trouver une relation dans les données, entre :
 - L'**altitude** (d'une station de ski)
 - La **température** dehors
 - En imaginant qu'une telle relation existe : $\text{température} = f(\text{altitude}) = ax+b$
- Objectif
 - **Prédire** la température grâce à l'altitude de la station
- Limites
 - Modélisation approximative sur un seul phénomène
 - Soumise à l'imprécision de la mesure des données
 - Trop simplificatrice (la relation n'est peut-être pas linéaire)



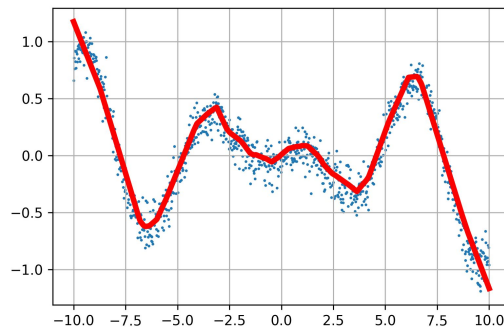
Régression linéaire

- Exemple : trouver une relation dans les données, entre :
 - Le **prix** d'une maison dans une ville donnée
 - La **superficie** de la maison
 - En imaginant qu'une telle relation existe : $\text{prix} = f(\text{superficie}) = ax+b$
- Objectif
 - **Prédire** le prix d'une maison grâce à sa superficie
- Limites
 - Modélisation approximative sur un seul phénomène
 - Soumise à l'imprécision de la mesure des données
 - Trop simplificatrice (la relation n'est peut-être pas linéaire)



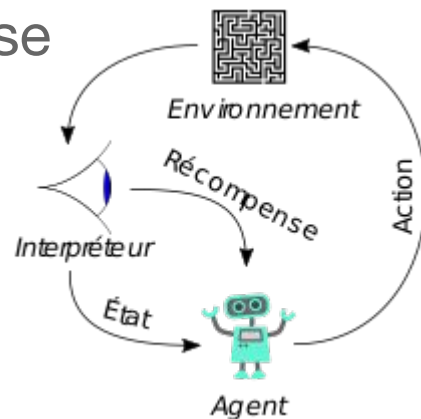
Régressions non linéaires

- Plusieurs techniques
 - K-NN, SVM, random forest, paramétrique, polynomiale, ...
- Le deep learning est parfois plus efficace
- Régression non linéaire obtenue par réseau de neurones



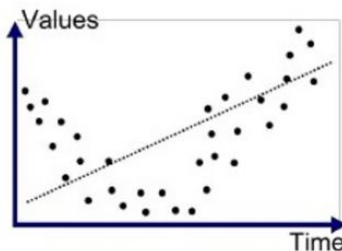
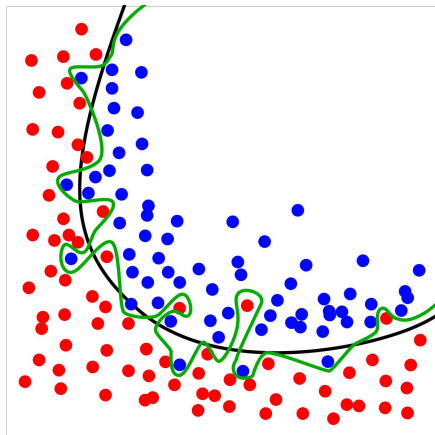
Apprentissage par renforcement

- Agent autonome : programme autonome, comme un robot
- Apprend par l'échec avec optimisation jusqu'à la réussite
- Un peu comme dans un jeu de type "*die and retry*"
- L'agent (algorithme ou robot) essaie par itération
- Si l'essai est meilleur → meilleure récompense
- Et il retente jusqu'à réussir
 - En partant de la meilleure itération précédente
- But : maximiser les récompenses
 - Jusqu'à la réussite

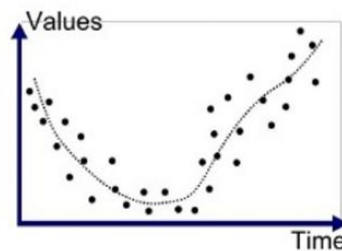


Surapprentissage (overfitting)

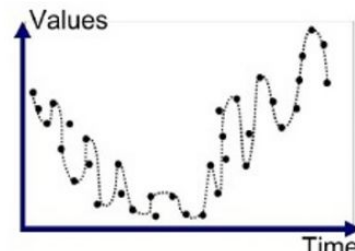
- Quand l'algorithme apprend les erreurs statistiques
 - Dans son modèle
- Trop entraîné sur un jeu de données précis



Underfitted



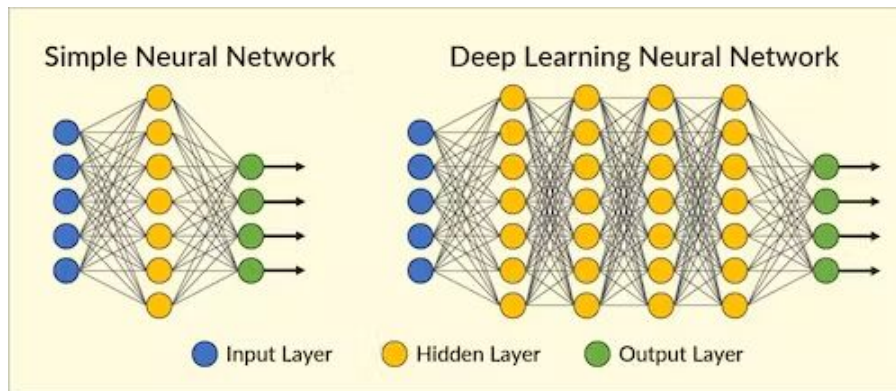
Good Fit/Robust



Overfitted

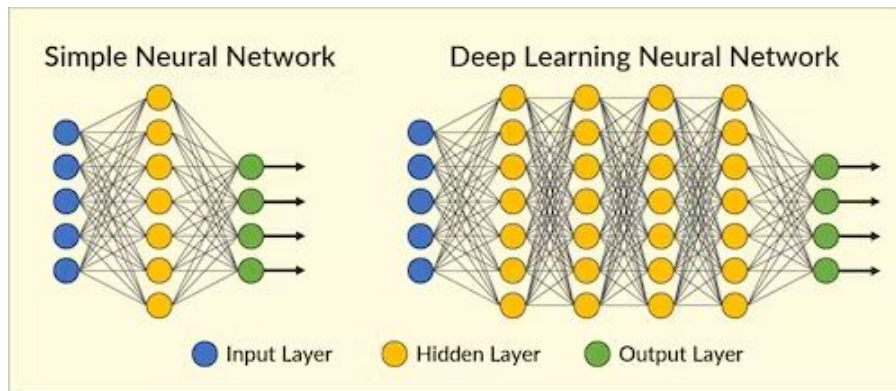
Deep learning

- Machine learning à base de réseaux de neurones
- Imiter le cerveau avec des “neurones” artificiels
 - Réseaux de plusieurs couches de “neurones”
 - Chaque couche essaye d'identifier une caractéristique à reconnaître



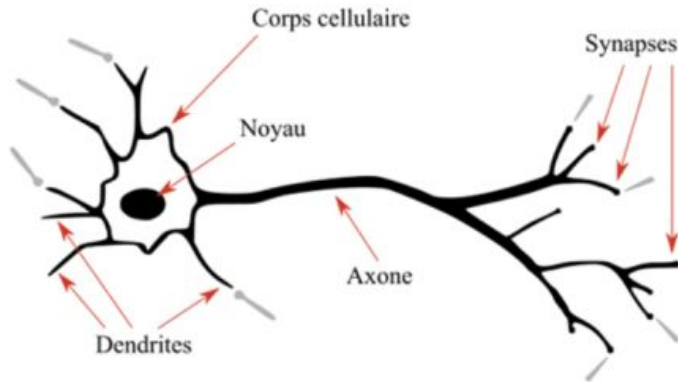
Deep learning

- Pendant l'apprentissage :
 - Ajuster les poids (valeurs) des neurones
 - Pour minimiser l'écart entre le résultat obtenu et le résultat attendu
 - C'est une boîte noire, les poids ne représentent rien pour un humain

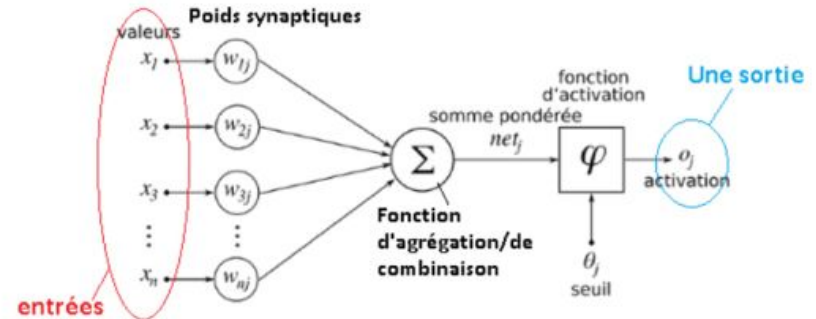


Neurone

- Peut être vu comme une fonction
 - Prend un vecteur en entrée et renvoie une valeur
 - Déterminée par son poids et un seuil d'activation
- Ces données sont modifiées et apprises
 - Pendant l'entraînement du modèle



NEURONE BIOLOGIQUE



NEURONE ARTIFICIEL