

UNIVERSITÉ LIBRE DE BRUXELLES - FACULTÉ DES SCIENCES
DÉPARTEMENT D'INFORMATIQUE

Paolo Augello, Zaky Belakbir, Maxime Brantegem, Thibault Maucq et Kolya Van Sint Jan

Intro / explication de pourquoi l'IA

La biodiversité marine

On estime qu'existe entre 500 000 et 10 millions d'espèces marines dont actuellement 240 000 ont été décrites.

Pourquoi protéger les espèces ?

Un grand nombre d'espèces jouent un rôle majeur dans les écosystèmes dans lesquels elles vivent.
Ex: le dauphin rose

Alors pourquoi utiliser l'IA ?

L'IA est un outil puissant qui peut être perfectionné afin de traiter un très grand nombre de données.

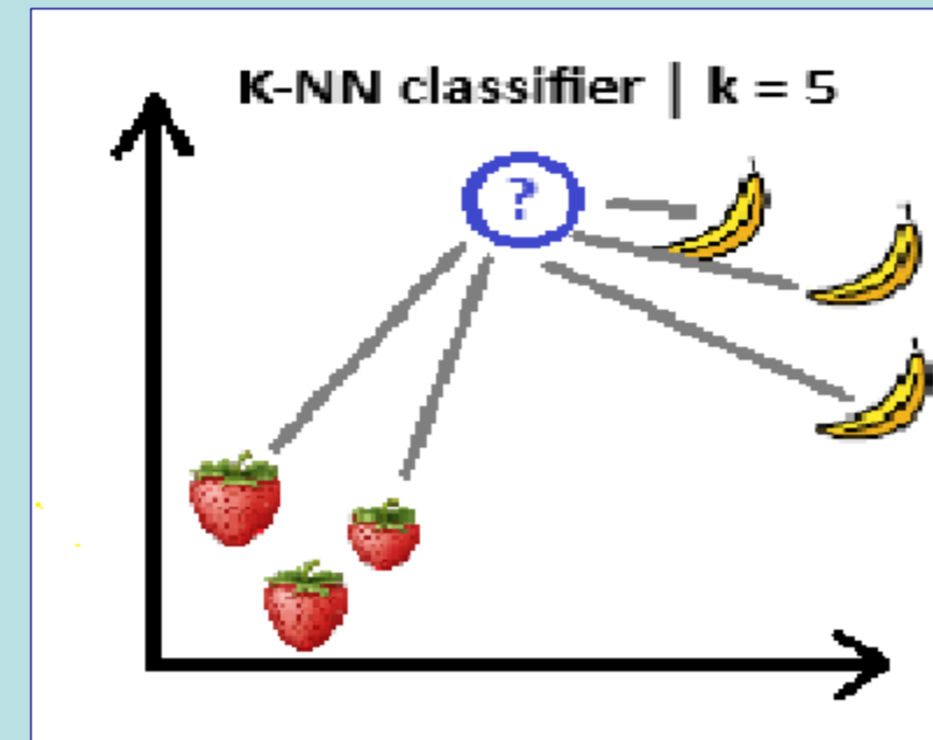
Analogies:

1. L'IA est comme le cerveau humain : elle apprend par l'exemple, elle identifie les poissons en analysant des milliers d'images



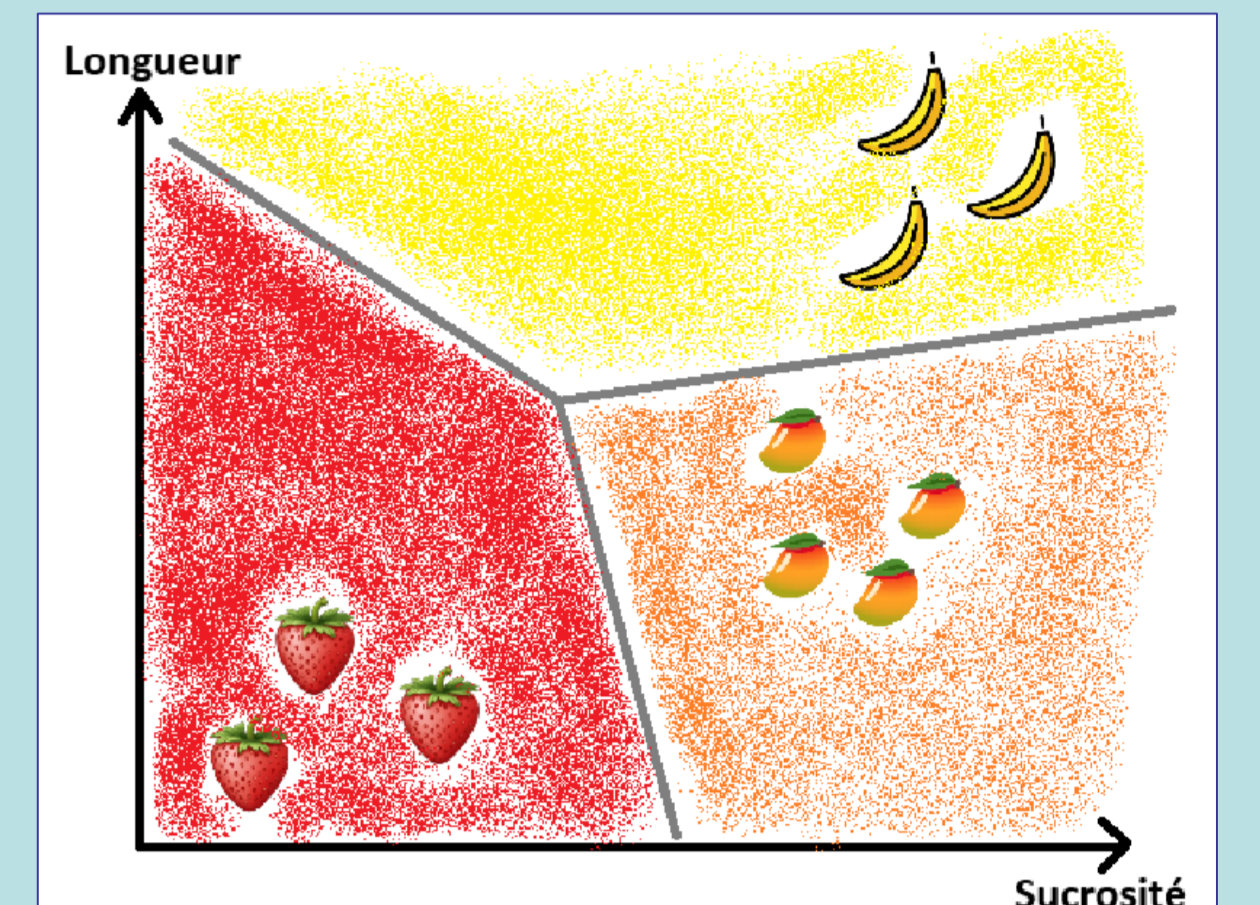
K-Nearest Neighbors

K-NN est un algorithme simple utilisé pour classer des données ou faire des prédictions en se basant sur la proximité des points dans un espace.



Comment ça marche ?

- On mesure les caractéristiques d'un nouvel élément (ex : sucré et longueur pour un fruit).
- On cherche ses **K plus proches voisins** parmi les données existantes.
- L'élément est classé selon la majorité de ses voisins (ex : si 3 voisins sur 5 sont des bananes, alors c'est une banane!).
- On peut aussi déterminer les surfaces de décision et obtenir un diagramme de Voronoi.



Neural Network

Un réseau de neurones artificiels est un modèle informatique inspiré du cerveau humain. Il traite des données en ajustant des connexions appelées poids pour apprendre et faire des prédictions.

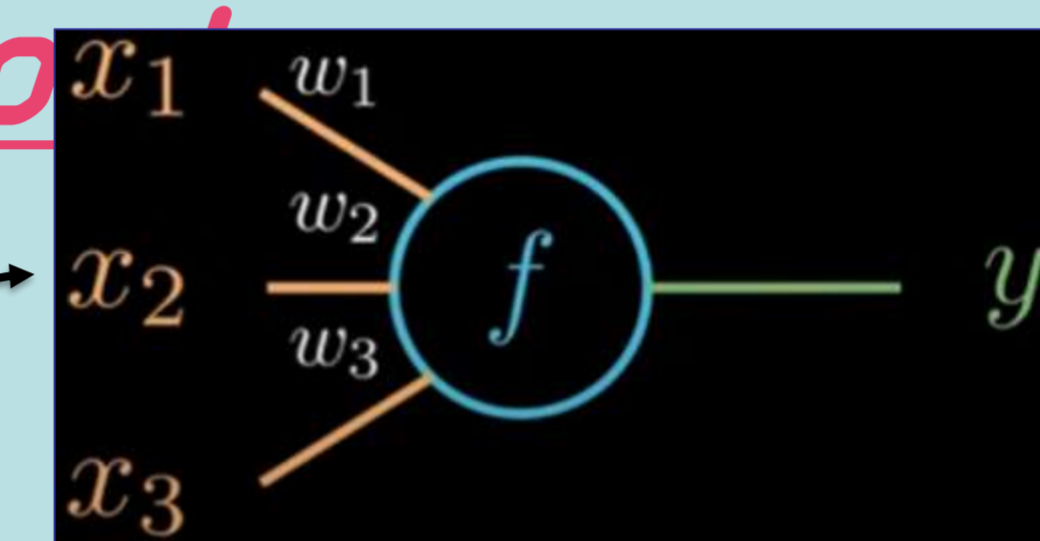
Neurone Artificiel (1943) : Dans un simple neurone artificiel, on a les différentes entrées **X** avec chacune un paramètre **poids W**. On passe les entrées dans une fonction qui nous donne une valeur de **sortie Y**.
Agrégation : $f = w_1 * x_1 + w_2 * x_2 + w_3 * x_3$
Activation : $y = 1$ si $f \geq 0$, $y = 0$ sinon.

Perceptron (1957) : C'est un neurone qui apprend ! Il ajuste ses **poids W** pour améliorer ses réponses à l'aide d'un algorithme d'apprentissage. Si le résultat en sortie est différent que le résultat attendu, on utilise cette formule :

Perceptron Multicouche (1986) : Un simple perceptron résout des problèmes linéaires. Pour résoudre des problèmes plus complexes, on empile plusieurs neurones en **couches**.

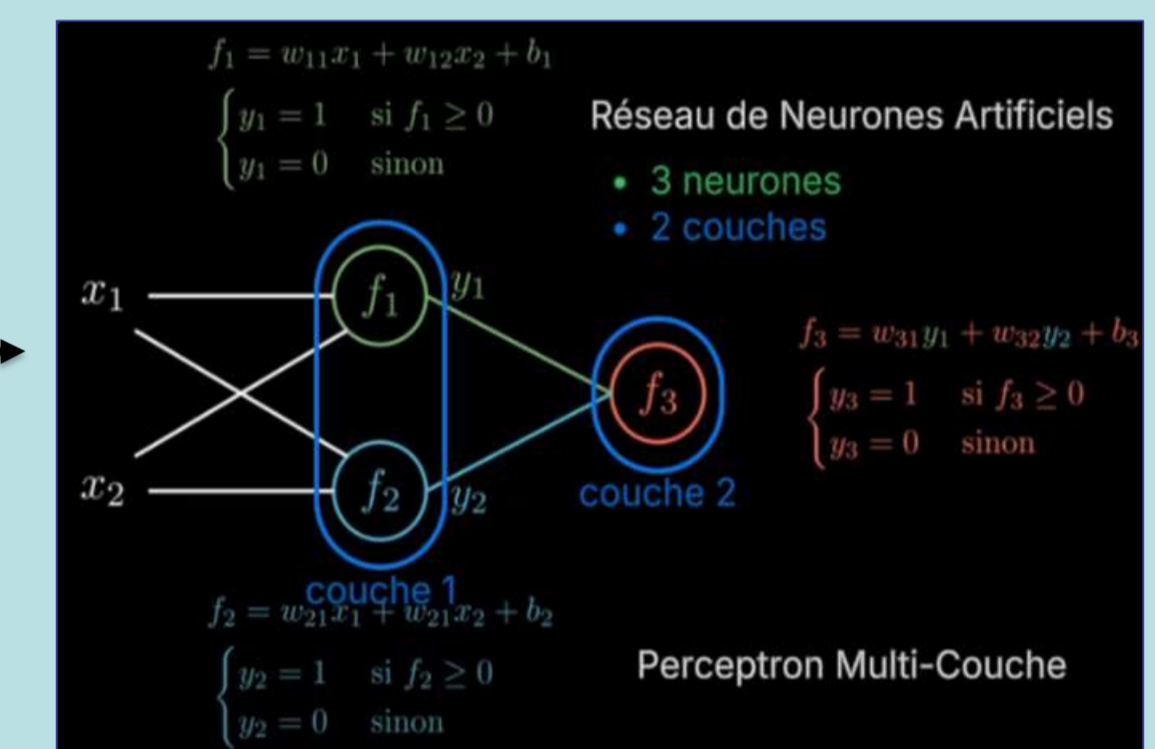
Apprentissage avec Backpropagation : On compare la sortie du réseau avec la valeur attendue. On **ajuste les poids** pour minimiser l'erreur en utilisant un algorithme appelé **descente de gradient**. Ensuite il suffit de réaliser la boucle d'apprentissage :

1. Forward Propagation
2. Cost Function
3. Backward Propagation
4. Gradient Descent

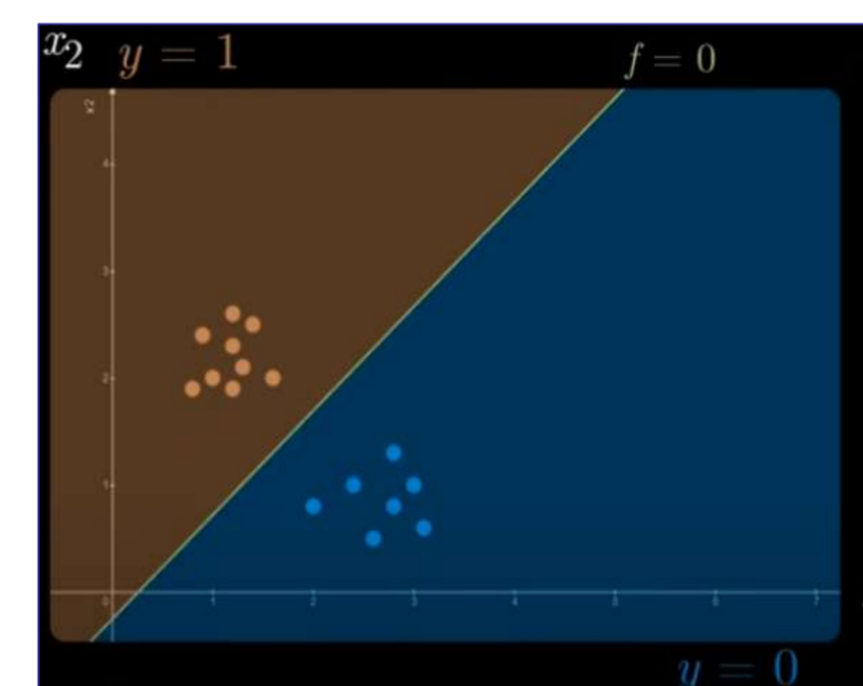


$$W = W + \alpha(y_{true} - y)X$$

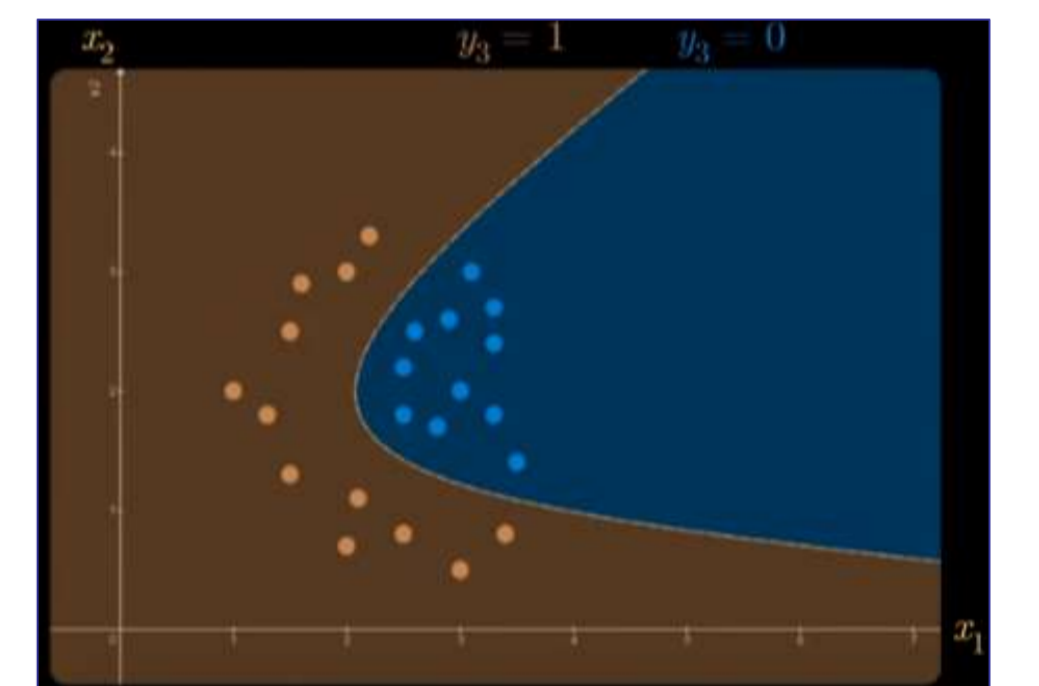
y_{true} : sortie de référence
 y : sortie produite par le neurone
 X : entrée du neurone
 α : vitesse d'apprentissage



Problème linéaire
On sépare deux ensembles avec une droite.

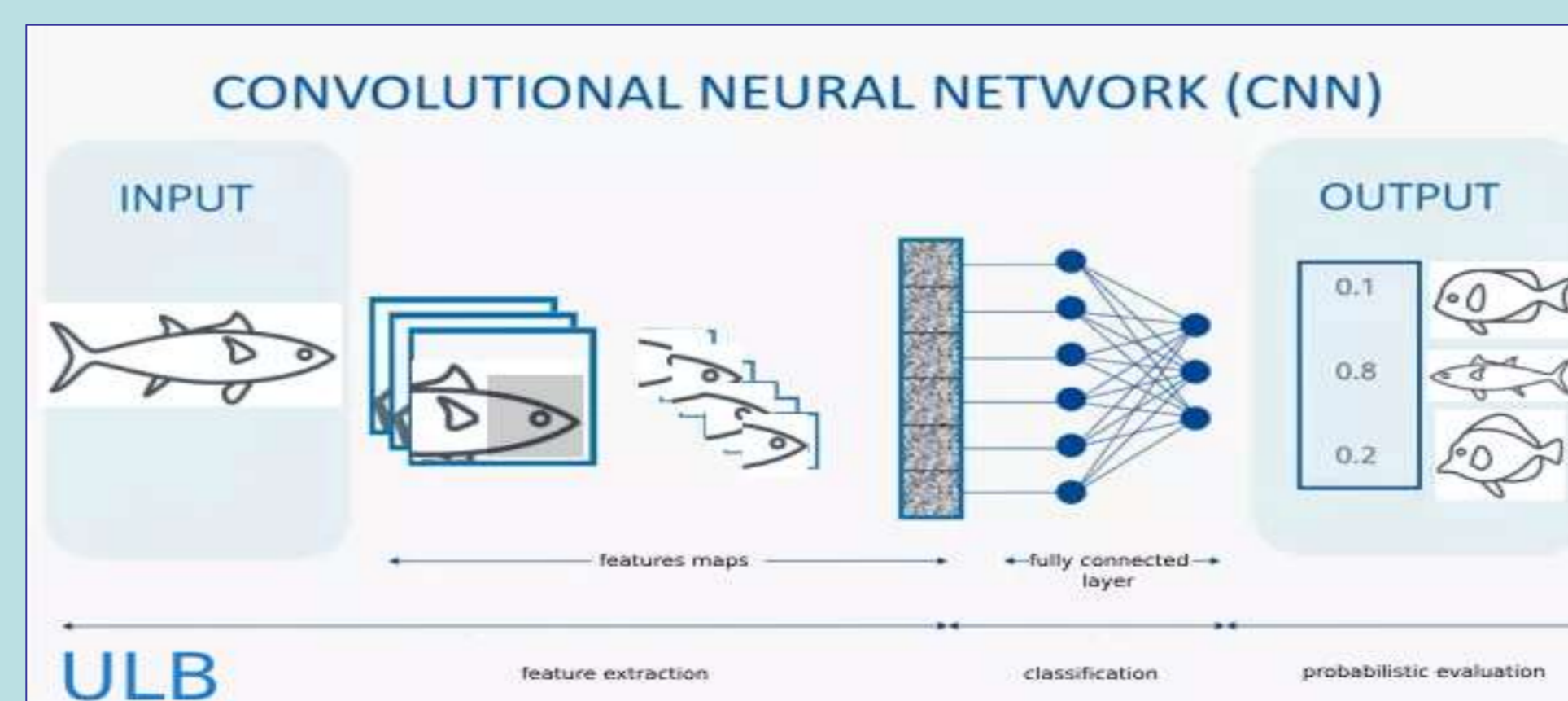


Problème non-linéaire
On sépare deux ensembles avec une courbe.



Comment un réseau de neurones reconnaît un poisson ?
- On montre à l'algorithme des images de poissons avec leurs noms.
- Chaque image est décomposée en pixels et analysée par le réseau de neurones.
- Le réseau **ajuste ses poids** pour apprendre à associer une image à un poisson.
- Une fois entraîné, il peut **prédire** quel poisson est sur une nouvelle image !

Convolutional Neural Network



Un **CNN** est un type de réseau de neurones artificiels conçu pour :
- analyser des images
- extraire des caractéristiques automatiquement
- utile en classification d'images

Pourquoi utiliser un CNN ?

- Capable de reconnaître des formes complexes dans les images sous différents angles et éclairages
- Apprend à distinguer les espèces marines en s'entraînant sur une large base d'images
- Automatisation efficace de la classification des photographies marines

Fonctionnement

Bloc filtreur

Couches de convolution pour détecter des motifs dans l'image
Couche de pooling pour réduire la taille des données sans perdre les informations importantes

Bloc classificateur

Couche de correction ReLU pour transformer les caractéristiques extraites en valeurs numériques
Couche fully-connected pour utiliser ces valeurs et classer l'image dans une catégorie