Cours-TD d'introduction à l'Intelligence Artificielle

Partie III

Le Perceptron

Simon Gay

Introduction à l’Intelligence Artificielle

- **Menu :**

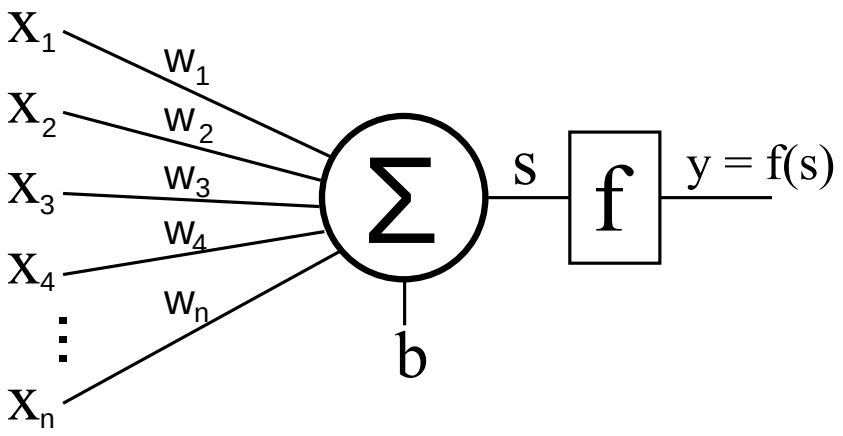
  - Théorie :
    - le neurone formel, pourquoi ça marche
    - Le principe du perceptron

  - Pratique :
    - implémentation d’un percpetron
    - Optimisation du réseau

Introduction à l'Intelligence Artificielle

- **Le neurone formel**

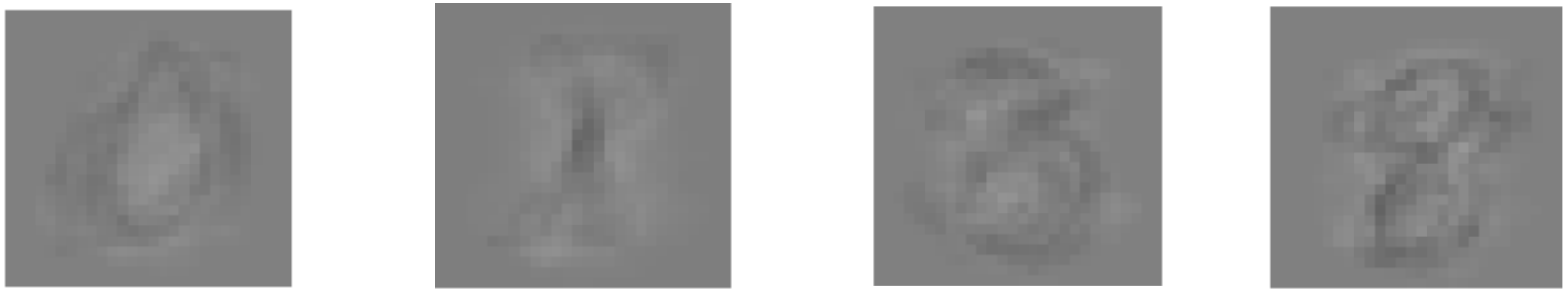
  - Un ensemble de poids
  - Apprentissage sur des exemples
  - Règle de Widrow-Hoff
  - Résultats concluants sur des exemples non connus



Introduction à l'Intelligence Artificielle

- **Le neurone formel**

  - Le neurone formel fait une moyenne des résultats positifs et des résultats négatif pour former une 'image moyenne' faisant ressortir les caractéristiques de l'élément à détecter

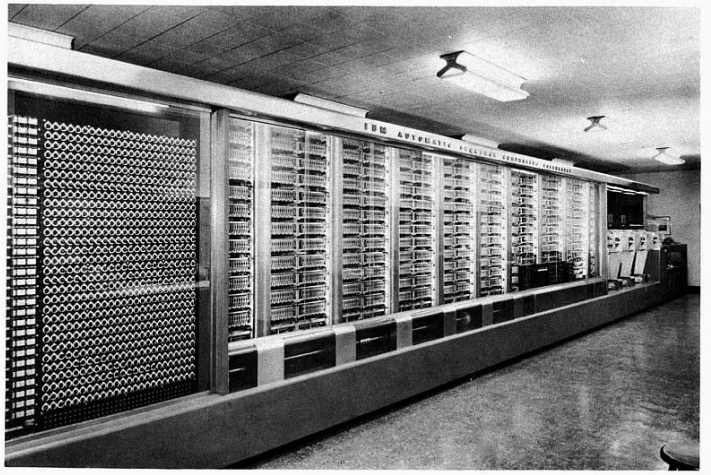


  - Cette 'image moyenne' permet de détecter l'élément sur un vecteur d'entrées non connu

  - **Mais pourquoi ça marche ?**

Introduction à l'Intelligence Artificielle

- **Le perceptron**



II Front View of the Calculator

  - Un des plus ancien réseau supervisé
    - inventé en 1957 par Frank Rosenblatt
    - Un ou plusieurs neurones formels
    - Règle de Hebbs (puis Widrow-Hoff plus tardivement)

  - Classifieur binaire : chaque neurone retourne 0 ou 1
    - Fonction d'activation à seuil

Introduction à l'Intelligence Artificielle

- **Fonction linéaire dans un plan**

  exemple :

  3.y + 2.x + 1 = 0 (y = -(2/3).x – 1/3)

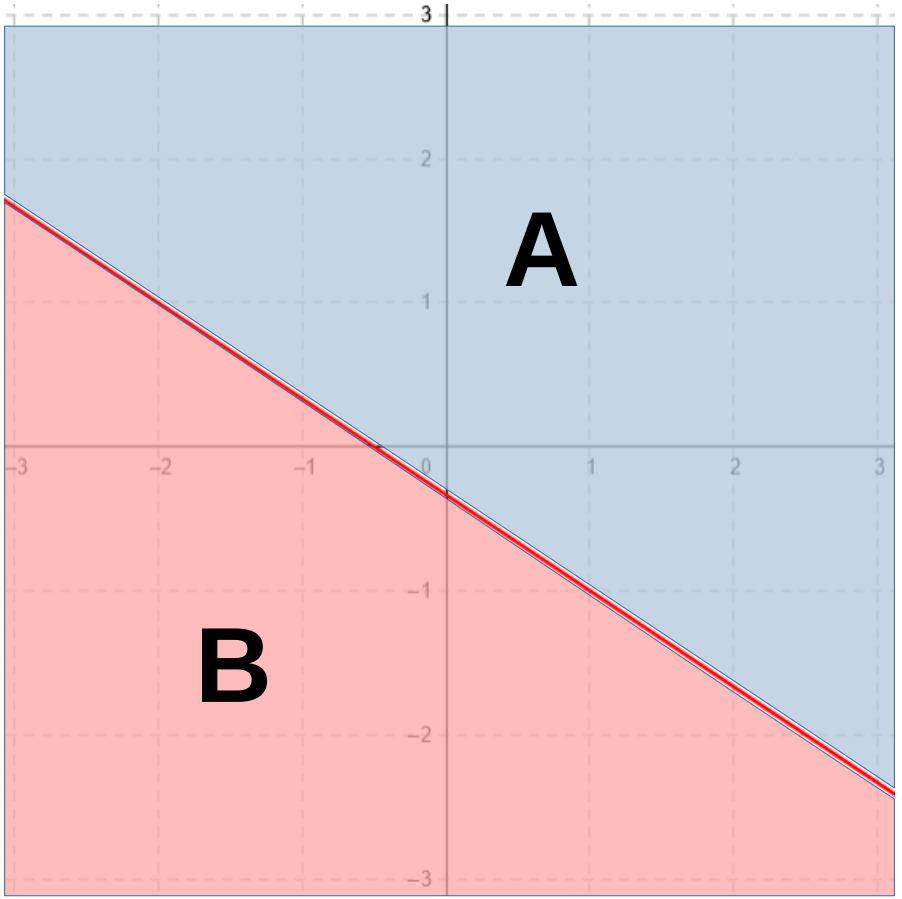
  Cette fonction sépare le plan en 2 :

  - Si 3.y+2.x+1 > 0

    → espace A au dessus de la courbe

  - Si 3.y+2.x+1 < 0

    → espace B en dessous de la courbe



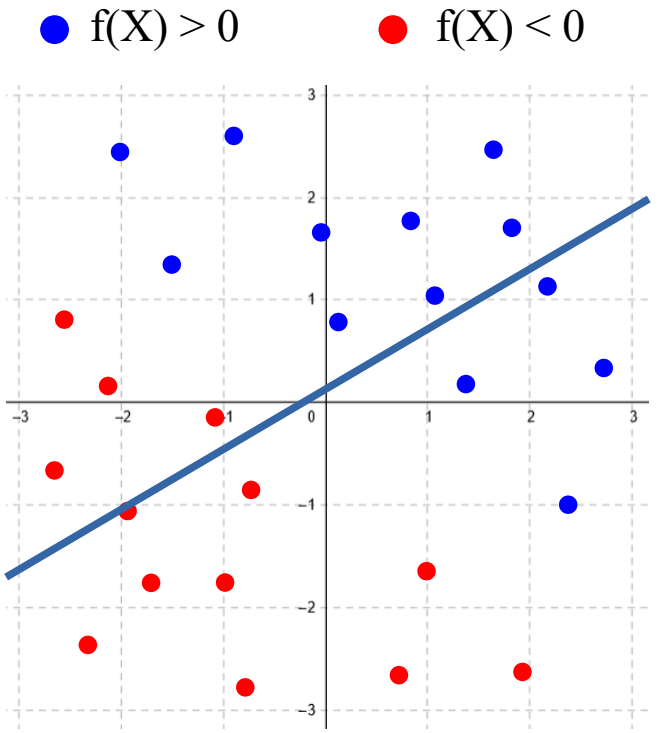
Introduction à l'Intelligence Artificielle

- **Et si on ne connaît pas la fonction ?**

  Soit une fonction a.x + b.y + c = 0

  Avec un ensemble de points X_i = [x_i , y_i] tels que f(X_i) > 0 ou f(X_i) < 0

  → On doit trouver un triplet (a , b , c) respectant toutes les contraintes



- On prend un triplet au hasard
- On teste chaque exemple
  - Si f(X_i)>0 mais $ax_i+by_i+c<0$
    - Il faut augmenter ax_i+by_i+c
  - Si f(X_i)<0 mais $ax_i+by_i+c>0$
    - Il faut réduire ax_i+by_i+c
  - Sinon, on ne fait rien

Introduction à l'Intelligence Artificielle

- **Et si on ne connaît pas la fonction ?**

  Pour chaque paramètre a, b et c, il faut légèrement augmenter ou diminuer la valeur proportionnellement, respectivement, à x, y et 1

  $a \leftarrow a + \alpha \cdot x_i$ ou $a \leftarrow a - \alpha \cdot x_i$

  $b \leftarrow b + \alpha \cdot y_i$ ou $b \leftarrow b - \alpha \cdot y_i$

  $c \leftarrow c + \alpha \cdot 1$ ou $c \leftarrow c - \alpha \cdot 1$

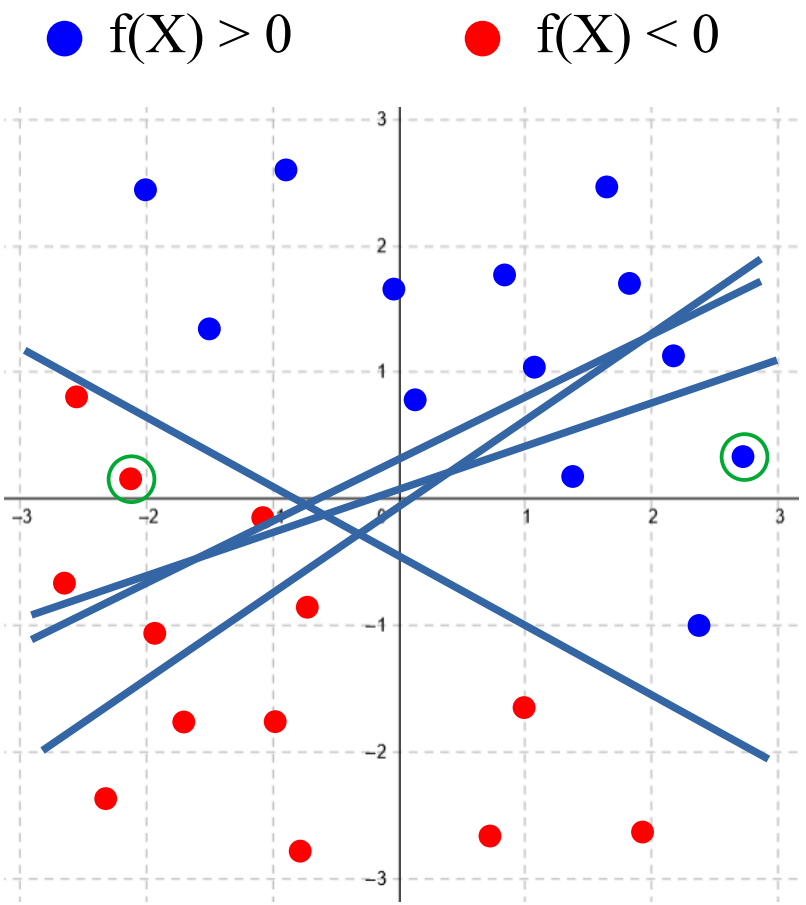
- **Si on note $r_i = 1$ si $f(X_i)>0$ et $r_i = -1$ si $f(X_i)<0$**
  - Alors
    - $a \leftarrow a + \alpha \cdot r_i \cdot x_i$
    - $b \leftarrow b + \alpha \cdot r_i \cdot y_i$
    - $c \leftarrow c + \alpha \cdot r_i$
  - On teste les points Xi, jusqu'à ce qu'ils soient tous du bon côté de la courbe

Introduction à l'Intelligence Artificielle

- **Les paramètres vont évoluer jusqu'à converger vers une solution**

- Algorithme :

erreurs = vrai
tant que erreurs faire
    erreurs = faux
    pour chaque Xi faire
        si $f(X_i) \cdot (a.x_i+b.y_i+c) < 0$ faire
            $a \mathrel{+}= \alpha \cdot r_i \cdot x_i$
            $b \mathrel{+}= \alpha \cdot r_i \cdot y_i$
            $c \mathrel{+}= \alpha \cdot r_i$
            erreurs = vrai
        fin si
    fin pour
fin tant que



- On ne met à jour que si il y a erreur

Introduction à l'Intelligence Artificielle

- **Et le perceptron dans tout ça ?**

- Écrivons :
  - $a . x + b . y + c = 0 \quad \rightarrow \quad w_1 . x_1 + w_2 . x_2 + b = 0$

- On généralise à un espace à n dimensions :

$$\sum_k w_k . x_k + b = 0$$

- Les poids d'un neurone formel forment l'équation d'un hyperplan
- L'apprentissage modifie les poids pour séparer l'espace en deux pour séparer deux groupes de points de cet espace

Introduction à l'Intelligence Artificielle

- **Quelques propriétés :**
- **Si l'ensemble d'exemples peut être séparé par un plan, alors :**

  - **Convergence assurée en un nombre fini d'étapes**
    - **Quel que soit le nombre d'exemples**
    - **Quelle que soit la distribution**
    - **Quel que soit le nombre de dimensions de l'espace**

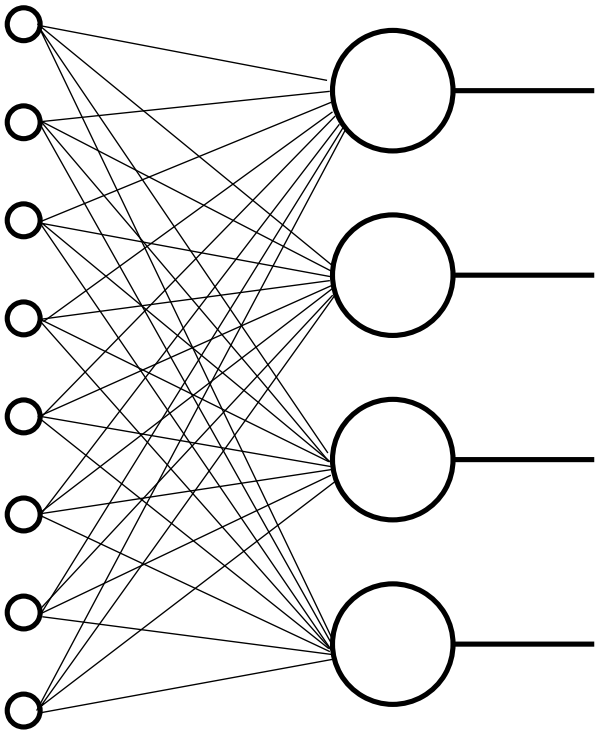
Introduction à l'Intelligence Artificielle

- **Perceptron avec plusieurs neurones (réseau simple couche)**

- **Utilisé pour définir plus de deux classes**

  - Possibilité d'une sortie sur plusieurs bits
    - Exemples : code ascii d'une lettre, conversion binaire vers afficheur 7 segment…
  - Un neurone par classe
    - Compétition entre les neurones, peut utilisée car possibilité d'égalité entre deux neurones

      → neurones à fonction d'activation continue



Introduction à l'Intelligence Artificielle

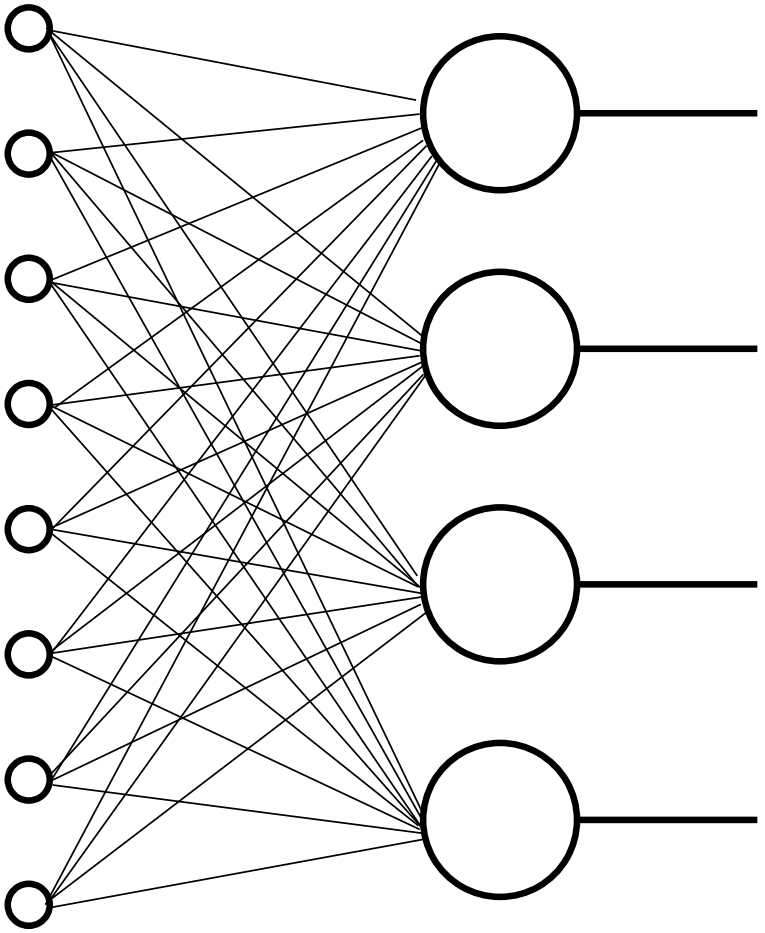
- **Perceptron avec fonction d'activation continue**

- Fonctions sigmoïde, tanh, linéaire, RELU ...
  - Ajoute une information supplémentaire : le niveau de confiance dans le résultat
    - Plus on est proche de 0 ou de 1, plus on est sûr du résultat
    - Compétition entre les neurones : on considère le neurone le plus actif (pas de seuillage des résultats, résolution d'ambiguïté)

(1 : 0,013) (2 : 0,658) (3 : 0,553) (4 : 0,350) (5 : 0,112) (6 : 0,092)...

- Apprentissage par la méthode Widrow-Hoff
  - Prend en compte l'erreur → recherche plus efficace de solutions

Introduction à l'Intelligence Artificielle



Passons à la pratique !