

Perceptrons

ARN

2 - Perceptrons

Abstract

Table des matières

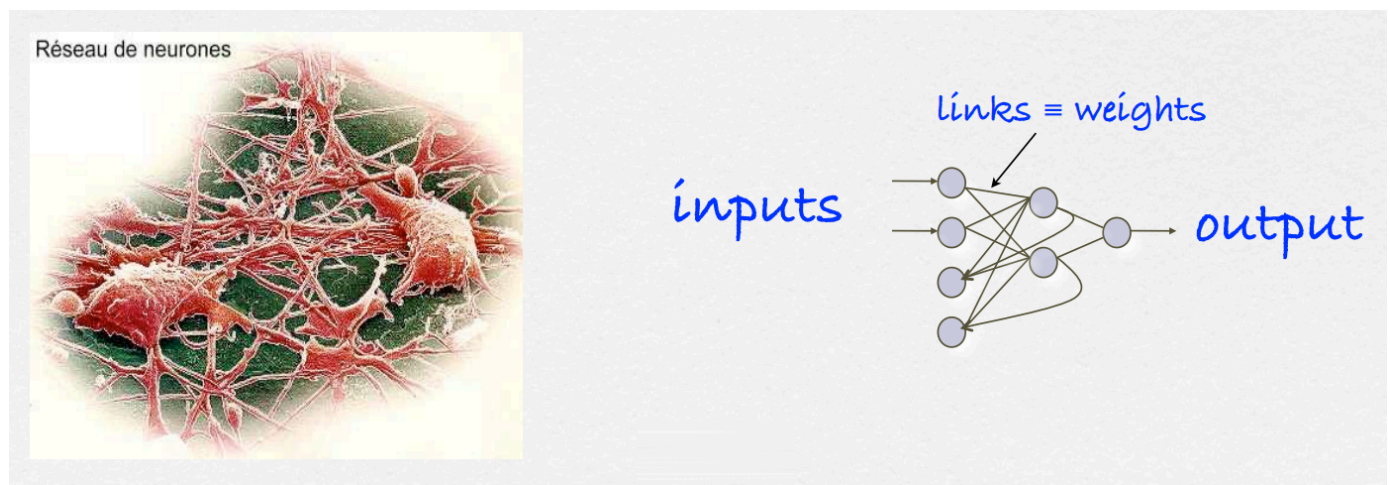
1. Perceptrons	2
1.1. Connexionnisme	2
1.2. Neurone Artificiel	2
1.2.1. Formule	3
1.3. Perceptron de Rosenblatt (1958)	3
1.4. Séparation Linéaire	3
1.5. Biais	3
1.6. Algorithme d'Apprentissage du Perceptron	3
1.7. Descente de Gradient (Widrow-Hoff / Delta Rule)	3
1.8. Fonction d'Activation Sigmoidale	3
1.9. Classification à Deux Classes	3
1.10. Problèmes Non-Linéairement Séparables	3

1. Perceptrons

Caractéristique	Ordinateur von Neumann	Cerveau
Processeur	Complexe (GHz)	Simple (Hz)
Fréquence	Élevée	Basse
Nombre d'unités	Quelques cœurs	Très nombreuses
Mémoire	Séparée, adressable	Intégrée, distribuée
Calcul	Centralisé, séquentiel	Distribué, parallèle
Programmation	Programmable	Apprentissage
Fiabilité	Vulnérable	Très robuste

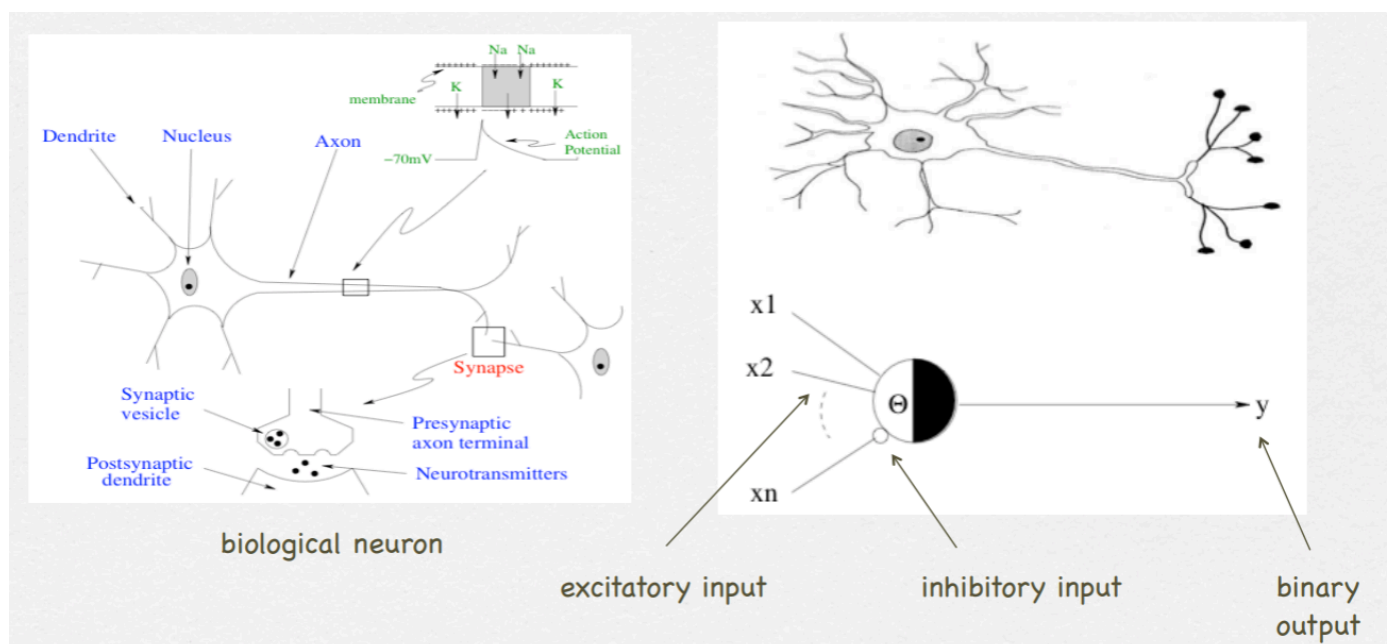
1.1. Connexionnisme

- Modélisation des phénomènes mentaux/comportementaux comme des processus émergents de réseaux interconnectés.
- Architecture massivement parallèle : chaque neurone corrèle les entrées avec des poids synaptiques.
- Changement de paradigme → remplace la programmation par l'apprentissage.



1.2. Neurone Artificiel

Le neurone artificiel, inventé par Warren McCulloch et Walter Pitts en 1948, est une unité de calcul qui prend des entrées, les pondère et les somme pour produire une sortie.



1.2.1. Formule

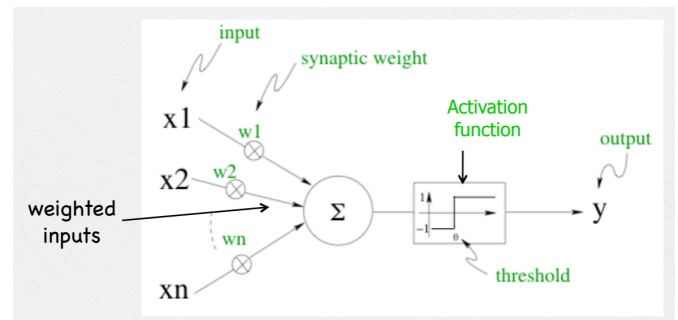
Si $(x_1 + x_2 + \dots + x_n) \geq \Theta$ alors $y = 1$, sinon $y = 0$

- Entrées : excitatrices (poids = 1) et inhibitrices (poids = -1).
- Sortie binaire.

1.3. Perceptron de Rosenblatt (1958)

Si $\sum w_i x_i \geq \Theta$ alors $y = 1$, sinon $y = -1$

- Somme pondérée des entrées.
- Fonction d'activation : discontinue.



1.4. Séparation Linéaire

- Le perceptron sépare les données par un hyperplan :
 - 2D : une ligne.
 - 3D : un plan.
 - nD : un hyperplan.

1.5. Biais

- Se débarrasse du seuil en ajoutant un poids supplémentaire appelé "biais" :

Si $\sum w_i x_i + w_0 \geq 0$ alors $y = 1$, sinon $y = 0$

1.6. Algorithme d'Apprentissage du Perceptron

1. Initialisation aléatoire des poids.
2. Calcul de la sortie y pour une entrée x .
3. Mise à jour des poids :

$$W_j(t+1) = W_j(t) + \eta(d - y)x_j$$

4. Répéter jusqu'à ce que l'erreur soit inférieure à un seuil.

1.7. Descente de Gradient (Widrow-Hoff / Delta Rule)

- Minimise la fonction d'erreur :

$$E = 1/2 \sum (y_i - d_i)^2$$

- Mise à jour des poids :

$$\Delta w_j = \eta(d - y)x_j$$

1.8. Fonction d'Activation Sigmoidale

$$y = 1 / (1 + e^{-(S_j)})$$

- Activation non linéaire souvent utilisée.
- Autres fonctions courantes : tanh, ReLU.

1.9. Classification à Deux Classes

- Le perceptron classe les données selon un seuil fixé.
- Équilibre entre taux de vrais positifs et taux de faux positifs.

1.10. Problèmes Non-Linéairement Séparables

- Le perceptron ne peut pas séparer certaines classes avec une ligne simple.

Dis-moi si tu veux que j'ajoute ou clarifie des parties spécifiques !