

Plan

Introduction générale

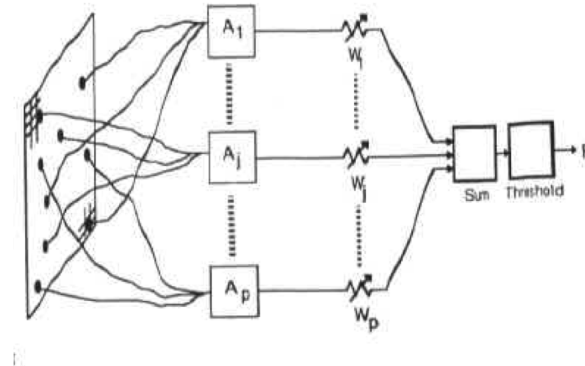
Le perceptron simple

Le perceptron multicouches

11 / 43

Le perceptron de Rosenblatt (1957)

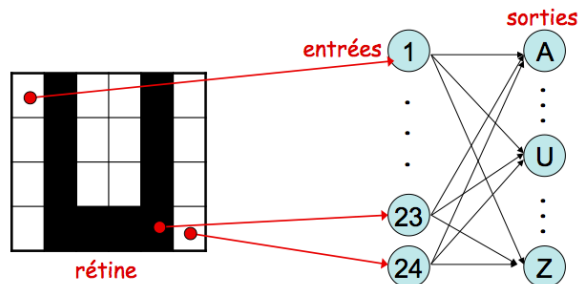
Architecture simple : une entrée (la rétine) et une sortie.



12 / 43

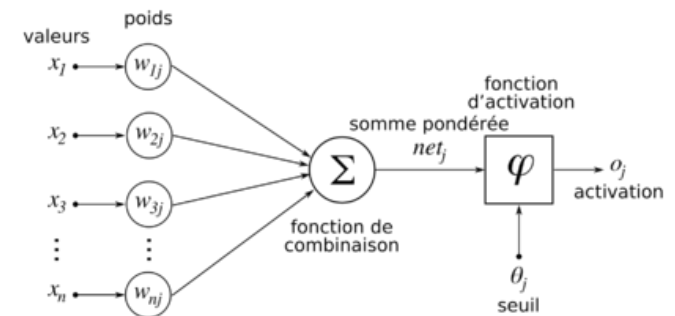
Exemple : associer des formes à des classes

Rétine de 4 x 6 éléments (image d'une lettre) avec 26 éléments en sortie



13 / 43

Le Perceptron



14 / 43

Le Perceptron

Cas à un seul neurone de sortie

Modèle de McCulloch et Pitts :

- Soit $\mathbf{x} = (x_1, \dots, x_p)$ dans \mathbb{R}^p le signal d'entrée
- Chaque x_i est associé à un poids w_i
- Le vecteur \mathbf{w} est appelé coefficients synaptiques
- On ajoute une constante w_0 appelée le biais

15 / 43

Le Perceptron

Cas à un seul neurone de sortie

Modèle de McCulloch et Pitts :

- Soit $\mathbf{x} = (x_1, \dots, x_p)$ dans \mathbb{R}^p le signal d'entrée
- Chaque x_i est associé à un poids w_i
- Le vecteur \mathbf{w} est appelé coefficients synaptiques
- On ajoute une constante w_0 appelée le biais

Signal post-synaptique

$$s(\mathbf{x}) = w_0 + \sum_{i=1}^p (w_i \times x_i)$$

15 / 43

Le Perceptron

Cas à un seul neurone de sortie

Modèle de McCulloch et Pitts :

- Soit $\mathbf{x} = (x_1, \dots, x_p)$ dans \mathbb{R}^p le signal d'entrée
- Chaque x_i est associé à un poids w_i
- Le vecteur \mathbf{w} est appelé coefficients synaptiques
- On ajoute une constante w_0 appelée le biais

Signal post-synaptique

$$s(\mathbf{x}) = w_0 + \sum_{i=1}^p (w_i \times x_i)$$

ou, si $\mathbf{x} = (1, x_1, \dots, x_p) : s(\mathbf{x}) = \sum_{i=0}^p w_i x_i = \mathbf{w} \cdot \mathbf{x}$

Fonction d'activation

$\hat{y}_{\mathbf{w}}(\mathbf{x}) = h(s(\mathbf{x}))$ où h peut être définie de différentes manières

15 / 43

Exemples de fonctions d'activation

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multi-layer NN	

16 / 43

Neurone à deux entrées

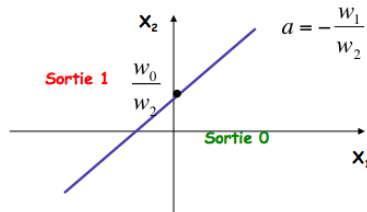
Règle d'activation linéaire

Si $s = w_1 \cdot x_1 + w_2 \cdot x_2 > w_0$ alors $\hat{y}=1$

Si $s = w_1 \cdot x_1 + w_2 \cdot x_2 \leq w_0$ alors $\hat{y}=0$

Géométriquement

Cela signifie qu'on a divisé le plan en deux régions par une droite d'équation $w_1 \cdot x_1 + w_2 \cdot x_2 - w_0 = 0$



17 / 43

Lien avec la régression logistique

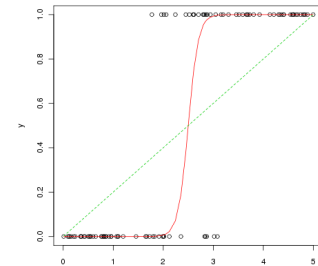
Principe

- on cherche à modéliser de façon linéaire $\ln \frac{P(Y=1|\mathbf{X}=\mathbf{x})}{1-P(Y=1|\mathbf{X}=\mathbf{x})}$:

$$\ln \frac{P(Y=1|\mathbf{X}=\mathbf{x})}{1-P(Y=1|\mathbf{X}=\mathbf{x})} = \underbrace{\ln \frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}}_{\text{logit}(\pi(\mathbf{x}))} = \beta_0 + \sum_{j=1}^p \beta_j x_j$$

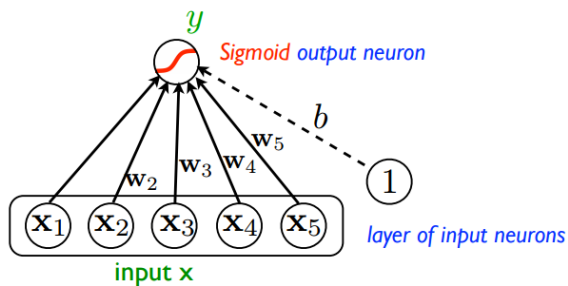
ce modèle s'écrit aussi

$$\pi(\mathbf{x}) = \frac{\exp(\beta_0 + \sum_{j=1}^p \beta_j x_j)}{1 + \exp(\beta_0 + \sum_{j=1}^p \beta_j x_j)}$$



18 / 43

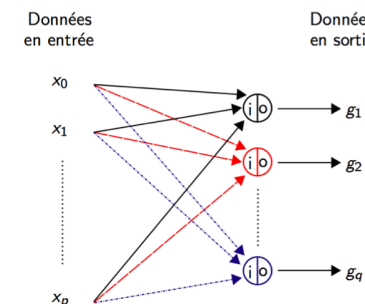
Une autre représentation bien utile



19 / 43

Remarques sur le perceptron

- Sans couche cachée, on retrouve les modèles linéaires
- Avec la fonction sigmoïd, il s'agit d'une régression logistique et on interprète les sorties comme des probabilités
- Pour le problème multiclass, on traite q problèmes binaires indépendants :



20 / 43

A nouveau : relations entre LR et ANN

Pour la catégorisation en $q = 2$ classes, on utilise un seul perceptron de sortie et la fonction sigmoïde :

$$g(x) = h(s(x)) = \frac{1}{1 + \exp(-w^T \cdot x)}$$

La régression logistique polytomique est équivalente à un ANN avec q perceptrons en parallèle et des fonctions d'activation softmax :

$$g_k(x) = h(\mathbf{w}_k^T \cdot \mathbf{x}) = \frac{\exp(\mathbf{w}_k^T \cdot \mathbf{x})}{\sum_j \exp(\mathbf{w}_j^T \cdot \mathbf{x})}$$

Le softmax permet de garantir une probabilité en sortie.

21 / 43

Cas de la fonction booléenne “AND”

X^1	X^2	AND
1	1	1
1	0	0
0	1	0
0	0	0

22 / 43

Cas de la fonction booléenne “AND”

X^1	X^2	AND
1	1	1
1	0	0
0	1	0
0	0	0

Signal post-synaptique et fonction d'Heaviside :
 $w_0 + w_1 X^1 + w_2 X^2 \geq 0$

Cas de la fonction booléenne “AND”

X^1	X^2	AND
1	1	1
1	0	0
0	1	0
0	0	0

Signal post-synaptique et fonction d'Heaviside :
 $w_0 + w_1 X^1 + w_2 X^2 \geq 0$
 $\begin{cases} w_0 + w_1 + w_2 > 0 \\ w_0 + w_1 \leq 0 \\ w_0 + w_2 \leq 0 \\ w_0 \leq 0 \end{cases}$

22 / 43

22 / 43

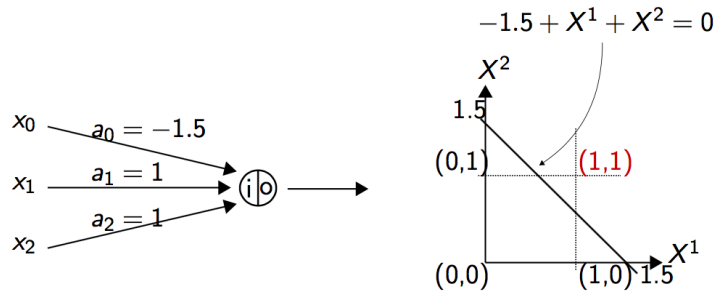
Cas de la fonction booléenne "AND"

X^1	X^2	AND
1	1	1
1	0	0
0	1	0
0	0	0

Signal post-synaptique et fonction d'Heaviside :

$$w_0 + w_1 X^1 + w_2 X^2 \begin{cases} \leq 0 \\ > 0 \end{cases}$$

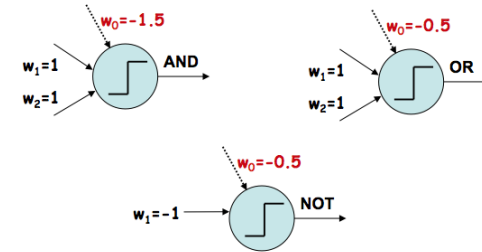
$$\begin{cases} w_0 + w_1 + w_2 > 0 \\ w_0 + w_1 \leq 0 \\ w_0 + w_2 \leq 0 \\ w_0 \leq 0 \end{cases}$$



22 / 43

Note au sujet du biais w_0

L'une des motivations initiales était de pouvoir représenter des fonctions logiques :



23 / 43

Cas de la fonction booléenne "XOR"

X^1	X^2	XOR
1	1	0
1	0	1
0	1	1
0	0	0

Signal post-synaptique et fonction d'Heaviside :

$$w_0 + w_1 X^1 + w_2 X^2 \leq 0$$

X^1	X^2	XOR
1	1	0
1	0	1
0	1	1
0	0	0

24 / 43

24 / 43

Cas de la fonction booléenne "XOR"

Signal post-synaptique et fonction d'Heaviside :

X^1	X^2	XOR
1	1	0
1	0	1
0	1	1
0	0	0

$$w_0 + w_1 X^1 + w_2 X^2 \stackrel{\leq}{\geq} 0$$

$$\begin{cases} w_0 + w_1 + w_2 \leq 0 \\ w_0 + w_1 > 0 \\ w_0 + w_2 > 0 \\ w_0 \leq 0 \end{cases}$$

Cas de la fonction booléenne "XOR"

Signal post-synaptique et fonction d'Heaviside :

X^1	X^2	XOR
1	1	0
1	0	1
0	1	1
0	0	0

$$w_0 + w_1 X^1 + w_2 X^2 \stackrel{\leq}{\geq} 0$$

$$\begin{cases} w_0 + w_1 + w_2 \leq 0 \\ w_0 + w_1 > 0 \\ w_0 + w_2 > 0 \\ w_0 \leq 0 \end{cases}$$

Ensemble de contraintes **incompatibles**.

24 / 43

24 / 43

Cas de la fonction booléenne "XOR"

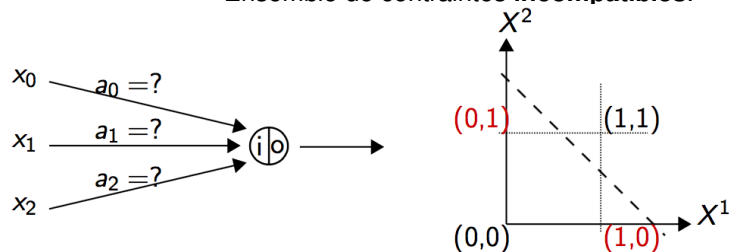
Signal post-synaptique et fonction d'Heaviside :

X^1	X^2	XOR
1	1	0
1	0	1
0	1	1
0	0	0

$$w_0 + w_1 X^1 + w_2 X^2 \stackrel{\leq}{\geq} 0$$

$$\begin{cases} w_0 + w_1 + w_2 \leq 0 \\ w_0 + w_1 > 0 \\ w_0 + w_2 > 0 \\ w_0 \leq 0 \end{cases}$$

Ensemble de contraintes **incompatibles**.



Et pourtant, une solution existe...

On peut apprendre le XOR avec un réseau à une couche cachée :

24 / 43

25 / 43

Et pourtant, une solution existe...

On peut apprendre le XOR avec un réseau à une couche cachée :

