Mapi3 Machine Learning : Réseaux de Neurones

Contexte $(x, y) \sim \mathbb{P}$. Trouver $f \in F$ tq $y \approx f(x)$.

Réseaux de neurones : Une classe particulière de F's !

$$f_{(\theta_1,\dots,\theta_N)}(x) = f_{\theta_N}^{(N)}\left(\dots\left(f_{\theta_1}^{(1)}(x)\right)\right) \quad (\theta_1,\dots,\theta_N) \in \mathbb{R}^{d_1} \times \dots \times \mathbb{R}^{d_N}.$$

$$F = \left\{ f_{(\theta_1,\dots,\theta_N)} \; ; \; (\theta_1,\dots,\theta_N) \in \mathbb{R}^{d_1} \times \dots \times \mathbb{R}^{d_N} \right\}$$

Vocabulaire :
- $f_\bullet$ = architechture
- $(\theta_1,\dots,\theta_N)$ = poids
- N = profondeur
- $\left(f^{(n)} \circ \dots \circ f^{(1)}\right)$ = layer (ou niveau) n
- $\left(f^{(n)} \circ \dots \circ f^{(1)}\right)(x)$ = activations au layer n.

I. Quelques architechtures classiques

Principe général des réseaux de neurones :

Chaque $f^{(i)}$ est en général très simple, la complexité et l'expressivité provient de la profondeur.

1) Les couches linéaires

Une classe très simple de fonctions est celle des fonctions linéaires.

$f^{(i)}$ ( ) ( )

$\theta_i(x) = W_i x + b_i$; $\theta_i = (W_i, b_i)$.

Vocabulaire :
- W = poids
- b = biais

⚠ Le terme de "poids" a un double usage et le terme de fonction linéaire désigne en fait une fonction affine.

2) Les activations et couches non-linéaires

Simplement composer des applications affines n'est en général pas très intéressant car le résultat est aussi affine.

L'expressivité des réseaux de neurones provient de la composition d'applications linéaires et de rectifications non-linéaires.

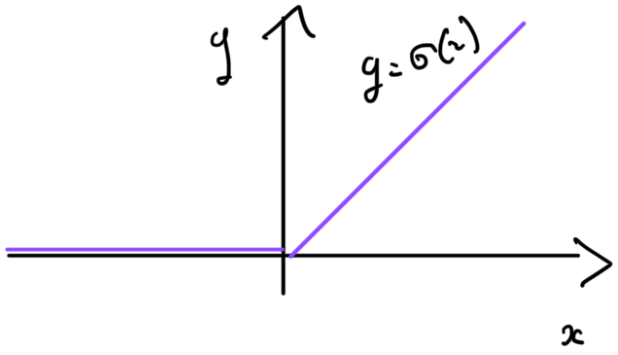
• Les activations

Soient, $P^{(i)}_{\theta_i}(x) = \sigma(W_i x + b_i)$; $\theta_i = (W_i, b_i)$

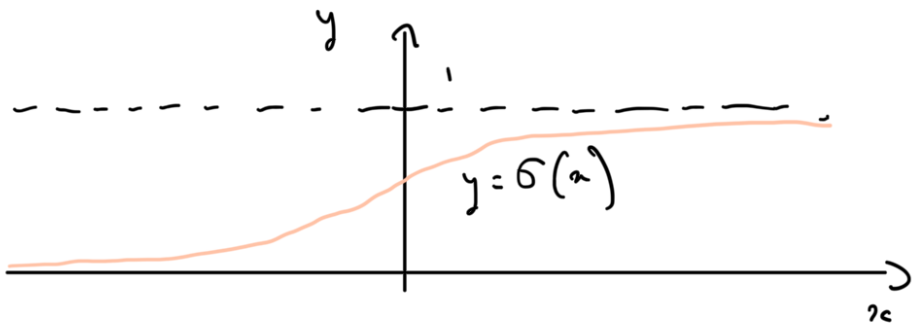
σ : fonction d'activation.

exemples classiques :

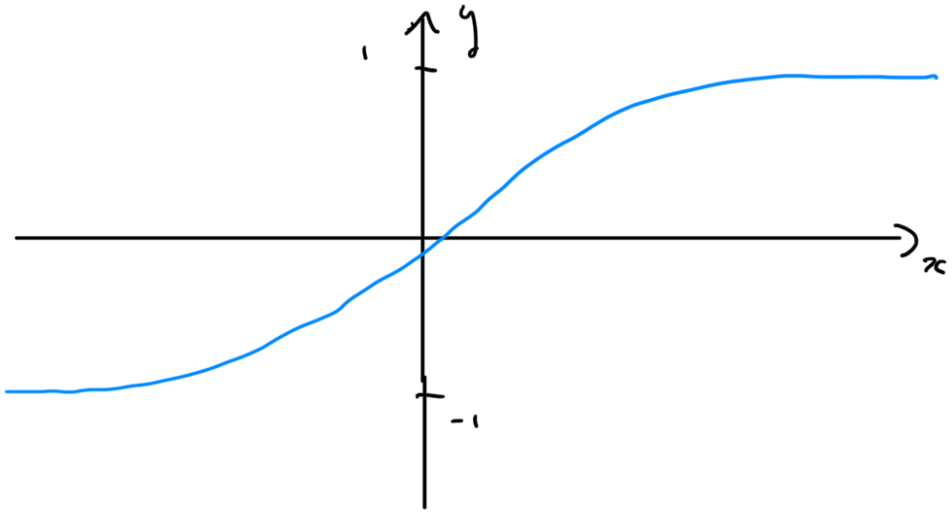
• ReLu : $\sigma(x) = (x_+) = \max(0, x)$.



• Sigmoid : $\sigma(v) = \frac{1}{1 + e^{-v}}$



• Tanh $\qquad \sigma(v) = \dfrac{e^{v} - e^{-v}}{e^{v} + e^{-v}}$



• Le Softmax

En classification, on veut $f(x) \in \Delta_{d-1}$ avec

$$\Delta_{d-1} = \left\{ x \in \mathbb{R}^d \text{ st } x \geq 0 \text{ et } \mathbb{1}^T x = 1 \right\}.$$

L'interprétation est que $f(x)$ encode les probabilités d'être dans telle ou telle classe.

Il faut alors convertir un vecteur de $\mathbb{R}^d$ en un point de Δ_{d-1}. C'est réalisé grâce au Softmax.

$$\text{Softmax}\left(\begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}\right) = \begin{pmatrix} \dfrac{e^{x_1}}{\sum_i e^{x_i}} \\ \vdots \end{pmatrix}$$

$$\begin{pmatrix} \vdots \\ e^{x_d} \\ \hline \sum_i e^{x_d} \end{pmatrix}$$

3) Les fonctions de perte

Un réseau est ensuite entraîné avec une fonction de perte $\ell(\cdot,\cdot)$

• En régression

La fonction de perte est souvent la fonction de perte quadratique

$$\ell(y, f(x)) = \|y - f(x)\|_2^2 .$$

• En classification

En classification, il faut pouvoir comparer deux vecteurs de probabilités de Δ_{d-1}.
C'est effectué avec la cross-entropy loss

$$\ell\left(\begin{pmatrix} y_1 \\ \vdots \\ y_d \end{pmatrix}, \begin{pmatrix} p_1 \\ \vdots \\ p_d \end{pmatrix} = f(x)\right) = -\sum_{i=1}^{d} y_i \log(p_i) .$$

Proposition $\forall p, q \in \Delta_{d-1}$, $\ell(p,q) \geq 0$ et $\ell(p,q) = 0$ si et seulement si $p = q$.

Preuve : $\ell(p,q) - \ell(p,p) = -\sum_i p_i \log(q_i) - \left(-\sum_i p_i \log(p_i)\right)$

$$= -\sum_i p_i \log\left(\frac{q_i}{p_i}\right)$$

$$\geq -\log\left(\sum_i p_i \frac{q_i}{p_i}\right)$$

Jensen

$$= -\log\left(\sum_i q_i\right) = -\log(1) = 0.$$

Il y a égalité dans l'inégalité de Jensen ssi $\forall i,\ p_i = q_i$.

La suite au prochain cours...