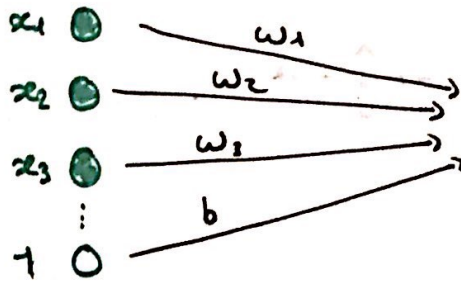


Régression Logistique (0 couches cachées)



Calcul de la loss.

- si $x, y \in \mathbb{R}$, MSE
- si $y \in \{0, 1\}$, BCE
- si $y \in \{1, \dots, K\}$, CE

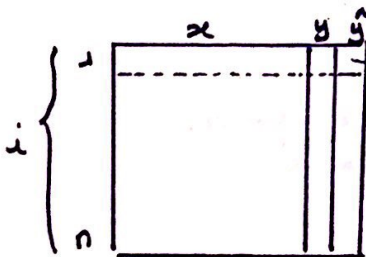
"A quel point \hat{y} s'écarte de y "

couche d'entrée

couche de sortie

$$x \longrightarrow x^T w + b \longrightarrow \sigma(z) = \hat{y} = a$$

=> objectif: trouver les paramètres $\Theta = (w_1, \dots, w_d, b)$ du Réseau tel qu'on approche au maximum la vérité terrain en veut que $\hat{y}^{(i)} \approx y^{(i)}$



une fois qu'on a calculé la loss sur tous les exemples, on peut calculer la coût global.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)})$$

=> en veut que ce coût global soit minimal.

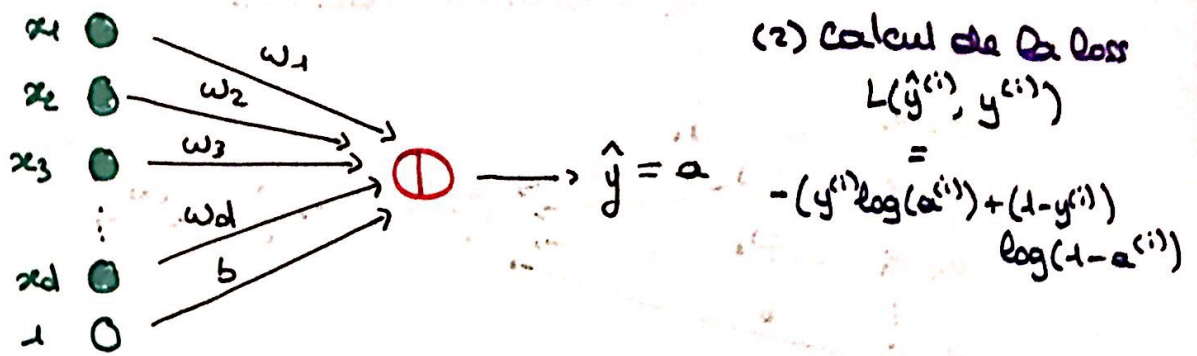
Pour ça, on utilise la descente de gradient: on veut aller dans sa direction opposée car c'est le point dans la direction de plus forte augmentation de la fonction.

Mise à jour des poids.

$$w \leftarrow w - \alpha \frac{\partial J(w, b)}{\partial w} \quad b \leftarrow b - \alpha \frac{\partial J(w, b)}{\partial b}$$

α = Learning rate

Algorithme.



(1) Forward propagation

on calcule la prédiction \hat{y}

$$z^{(i)} = x^{(i)T}w + b \quad a^{(i)} = \sigma(z^{(i)}) \quad \hat{y}^{(i)} = a^{(i)}$$

(3) Backward propagation

on calcule les gradients $\frac{\partial L}{\partial w}$ et $\frac{\partial L}{\partial b}$

Pour cela, on calcule d'abord $da = \text{gradient on sortie}$

Puis $dz = \text{gradient au niveau du neurone}$

Puis dw et $db = \text{gradients des entrées}$.

Pour tous les exemples ; on veut minimiser le coût global.

$$\frac{\partial J(\theta)}{\partial \theta_1} = \frac{1}{m} \sum_{i=1}^m \frac{\partial L(\hat{y}^{(i)}, y^{(i)})}{\partial \theta_1}$$

Le gradient du coût global est la moyenne des gradients des pertes sur tous les exemples.

=> A la fin on met à jour chaque poids et le biais.

Régression Logistique / Perceptron = classifieur linéaire, on ne peut pas résoudre le XOR.

Avec les réseaux de neurones, on compose des fonctions pour rendre le pb linéairement séparable.