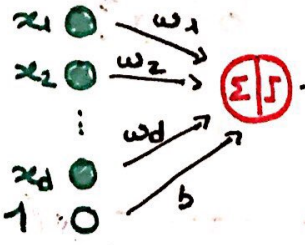
Du perceptron à la régression logistique (cas de la classification binaire)

Architecture du perceptron (0 couches cachées)



$$\hat{y} = R_w(x) = \begin{cases} 1 \text{ si } \Sigma \geq 0 \\ 0 \text{ si } \Sigma < 0 \end{cases}$$

sortie

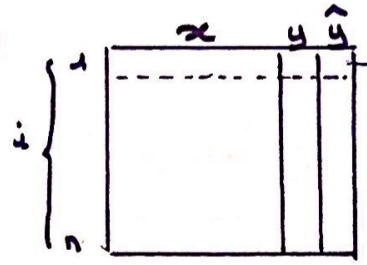
couche d'entrée

=> on modélise la décision en utilisant la combinaison linéaire des poids et des entrées suivi d'une fonction de seuil.

$$\hat{y} = R_w(x) = \text{Threshold}(x^T w)$$

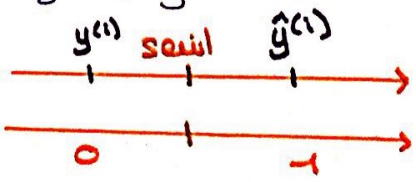
donc on va chercher les meilleurs poids w et b pour obtenir la meilleure prédiction (proche de la vraie sur le jeu d'entraînement).

Algorithme



pour chaque paire $x^{(i)}, y^{(i)}$
- calcul de la prédiction $\hat{y}^{(i)}$
- comparer $y^{(i)}$ à $\hat{y}^{(i)}$

Idée =
- si $y^{(i)} = \hat{y}^{(i)}$ => bonne prédiction pas de maj.
- si $y^{(i)} \neq \hat{y}^{(i)}$



$y^{(i)} - \hat{y}^{(i)} = -1$
$x^{(i)T} w$ trop grand

$y^{(i)} - \hat{y}^{(i)} = 1$
$x^{(i)T} w$ trop petit

Formalisation en termes de coût.

$$L(y^{(i)}, \hat{y}^{(i)}) = -(y^{(i)} - \hat{y}^{(i)}) x^{(i)T} \cdot w$$

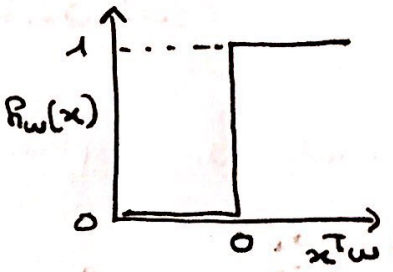
Règle de mise à jour des poids (descente de gradient)

$$w_d \leftarrow w_d - \alpha \frac{\partial L(y^{(i)}, \hat{y}^{(i)})}{\partial w_d}$$

on diminue les poids ou on ↗ les poids en fonction de "combien on s'est trompé"

Mais la dérivée de la fonction seuil en 0 n'existe pas.
donc cela peut causer des pb pendant l'entrainement.

=> au lieu de prédire la classe, on va prédire une probabilité car continue entre 0 et 1.

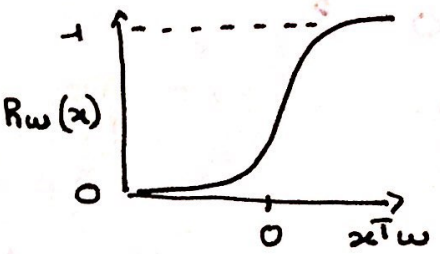


Pour cela, on n'utilise pas la fonction seuil mais la fonction sigmoïde.

$$\hat{y} = h_w(x) = \text{logistic}(x^T w)$$

$$= P(Y = 1 | x)$$

$$= \frac{1}{1 + e^{-x^T w}}$$



Fonction de perte associée — Binary cross entropy.

$$L(y^{(i)}, \hat{y}^{(i)}) = -y^{(i)} \log(h_w(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_w(x^{(i)}))$$

Mise à jour des poids.

$$w_d \leftarrow w_d - \alpha \frac{\partial L(y^{(i)}, h_w(x^{(i)}))}{\partial w_d} = w_d + \alpha (y^{(i)} - h_w(x^{(i)})) x_d$$

identique à celle du perceptron.
On a juste changer la loss et la règle de décision.