

TP semaine 10 : Perceptron simple

On travaille dans ce TP avec un perceptron simple à deux entrées, avec la fonction de Heaviside comme fonction de transfert, qui réalise une classification binaire ($\hat{y} \in \{0, 1\}$), et on va étudier les capacités de ce perceptron d'apprendre les fonctions booléennes **and**, **or**, **nor**, ou **xor**.

and	x_1	x_2	y	or	x_1	x_2	y	nor	x_1	x_2	y	xor	x_1	x_2	y
	0	0	0		0	0	0		0	0	1		0	0	0
	0	1	0		0	1	1		0	1	0		0	1	1
	1	0	0		1	0	1		1	0	0		1	0	1
	1	1	1		1	1	1		1	1	0		1	1	0

1. En supposant un neurone simple à 2 entrées et un biais, et des poids déjà initialisés (on prendra les valeurs $(-1, 1.4, 1)$ Pour commencer), écrire une fonction **prediction()** qui retourne la classe prédite par le perceptron.
2. Écrire une fonction **epoque()** qui, pour chacun des triplets d'un ensemble d'apprentissage, calcule la prédiction faite par le neurone, calcule l'erreur et modifie les poids en appliquant la règle de mise à jour vue en classe $\Delta a_j = \mu (y - \hat{y}) x_j$.
On va supposer que $\mu = 0.1$.
3. Écrire une fonction d'apprentissage qui initialise les poids, fait un apprentissage en n époques (on suppose $n = 5$) et affiche la valeur de l'erreur globale.
4. Faire en sorte que la fonction **epoque()** calcule une erreur globale, et définir une nouvelle fonction d'apprentissage qui arrête le calcul une fois que l'erreur globale est nulle.
5. Ajouter une initialisation aléatoire des poids (par exemple avec des valeurs comprises entre -1 et 1 , puis avec des valeurs comprises entre -100 et 100), et étudier la convergence de l'apprentissage.
6. Étudier la façon dont le taux d'apprentissage peut influencer sur la vitesse de convergence.

Une combinaison données de 3 poids correspond à une droite séparatrice (p. ex. les poids $(-1, 1.4, 1)$ correspondent à l'équation $-1 + 1.4x + y = 0$, donc à l'équation $y = -1.4x + 1$). On peut donc assez facilement afficher la succession des droites produites à chaque époque pour visualiser la façon dont le calcul aboutit à une droite séparatrice.

