

## Réseaux de neurones

### Exercice 1 : Perceptron simple

On considère un perceptron simple avec deux entrées et une sortie, et une fonction d'activation suivante :

$$a(x) = \begin{cases} 1 & \text{if } x - w_0 > 0 \\ 0 & \text{sinon} \end{cases}$$

1. trouvez les poids pour que le perceptron calcule la fonction ET logique
2. même question avec la fonction OU logique
3. essayer de trouver des poids pour la fonction XOR.

Construisez un *réseau* de neurones qui calcule la fonction XOR.

### Exercice 2 : mise à jour des poids pour réseaux multicouches

On a vu en cours comment mettre à jour les poids dans le cas d'un seul neurone en utilisant la descente de gradient. Le but de l'exercice est de traiter le cas des réseaux avec deux couches : une couche de neurones qui est reliée aux entrées et une couche de neurones pour la sortie.

**Notations :**

- $x_{ji}$  désigne l'entrée  $j$  de l'unité  $i$
- $w_{ji}$  désigne le poids associé à l'entrée  $j$  de l'unité  $i$
- $net_j = \sum_i w_{ji} x_{ji}$  est la somme pondérée des entrées
- $o_j$  est la valeur de sortie de l'unité  $j$
- $t_j$  est la valeur attendue de l'unité  $j$
- $\sigma$  est la fonction sigmoïde
- $out$  est l'ensemble des neurones de la couche de sortie
- $downstream(j)$  est l'ensemble des neurones qui utilisent l'unité  $j$  en entrée

On mesure l'erreur d'une instance  $d$  avec la fonction suivant

$$E_d(\vec{w}) = \frac{1}{2} \sum_{k \in out} (t_k - o_k)^2$$

1. Quelle est la dérivée de la fonction sigmoïde  $\sigma(x) = \frac{1}{1 + e^{-x}}$  ?
2. En utilisant la descente de gradient, écrire la formule de mise à jour
  - (a) pour les unités en sortie
  - (b) pour les autres unités