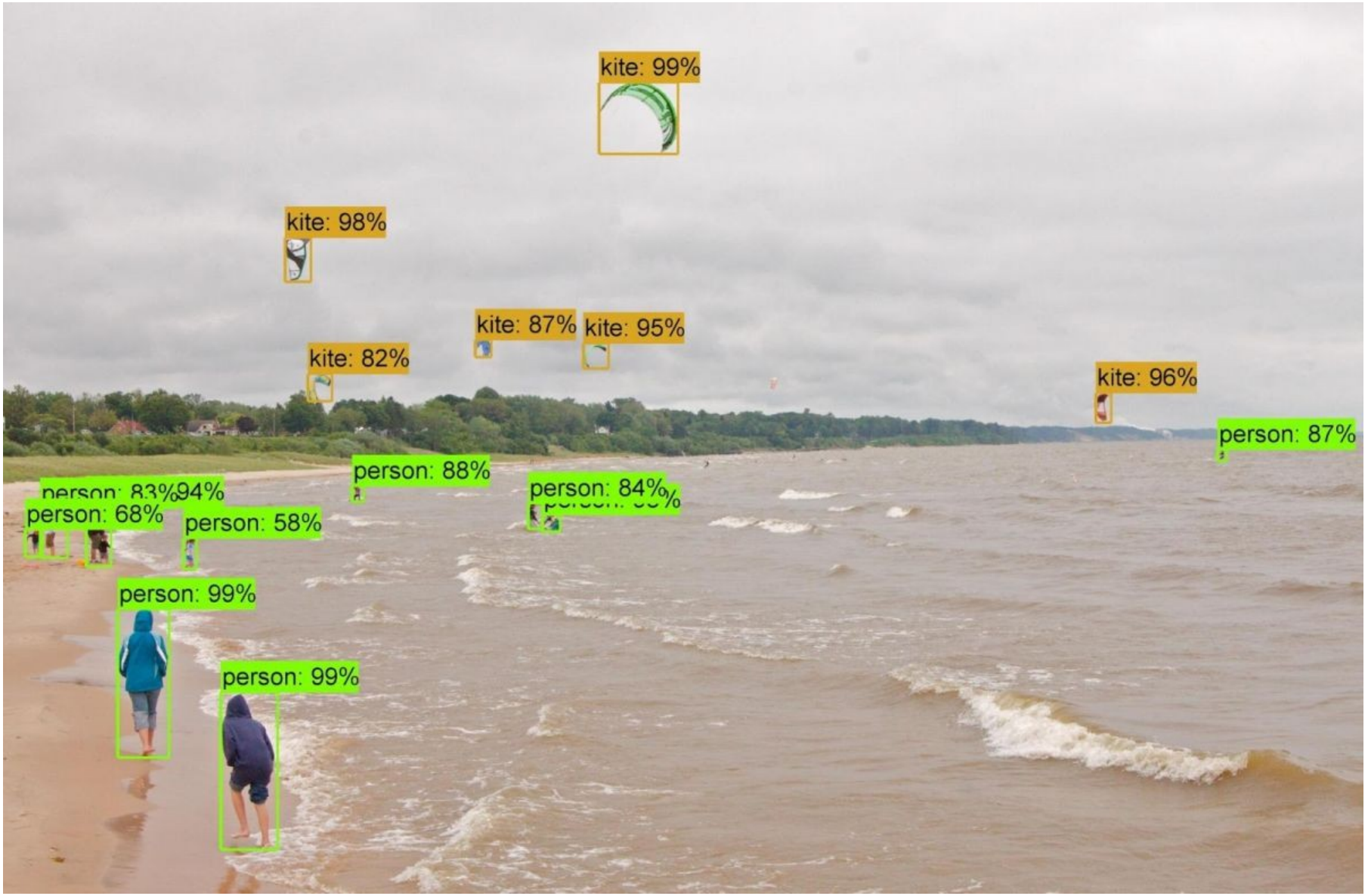
Nidal Djemam

Réseaux de neurones

E.P.S.I. Bordeaux – 2017/2018

Introduction



Introduction

- L’intelligence artificielle a longtemps eu pour but de simuler l’intelligence humaine.

Introduction

- L'intelligence artificielle a longtemps eu pour but de simuler l'intelligence humaine.

- les premiers neurones artificiels ont été définis par MacCulloch et Pitts en **1943**.

Introduction

- L'intelligence artificielle a longtemps eu pour but de simuler l'intelligence humaine.

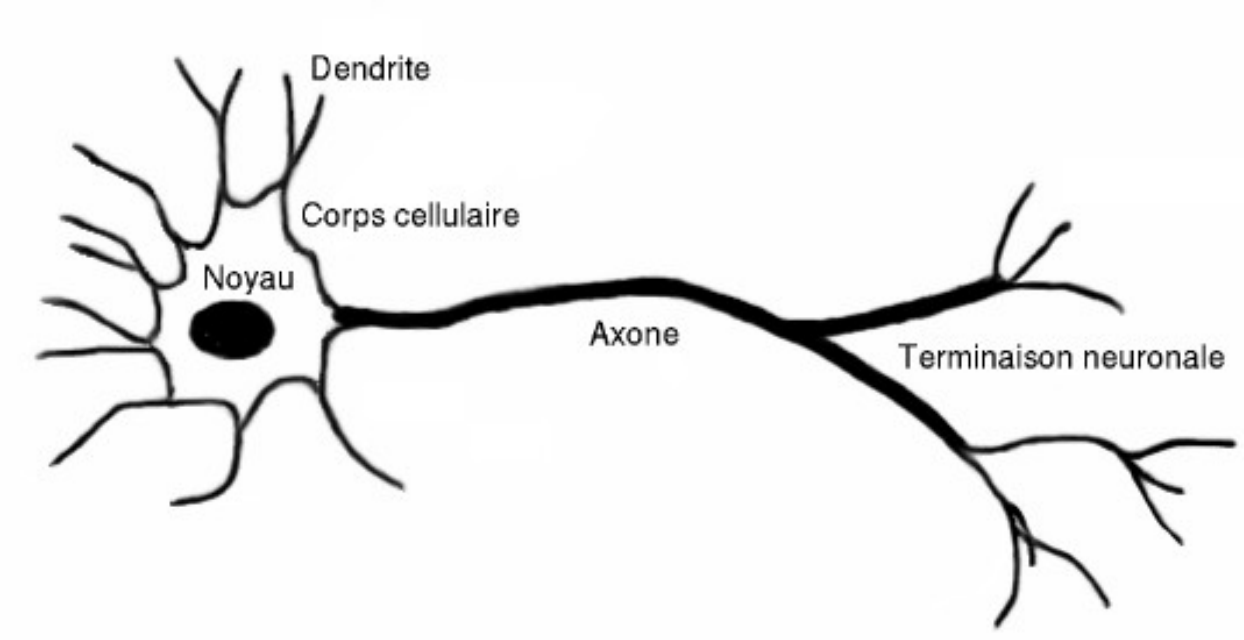
- les premiers neurones artificiels ont été définis par MacCulloch et Pitts en **1943**.

- **Aujourd'hui**, on ne cherche plus à créer des cerveaux avec toutes leurs capacités, mais à avoir des systèmes pouvant résoudre certains problèmes complexes sur lesquels les systèmes classiques sont limités.

Biologique

Chaque neurone possède donc autour de son coeur (nommé Noyau) :

- Des dendrites, qui sont ses entrées.
- Un long axone lui servant de sortie.

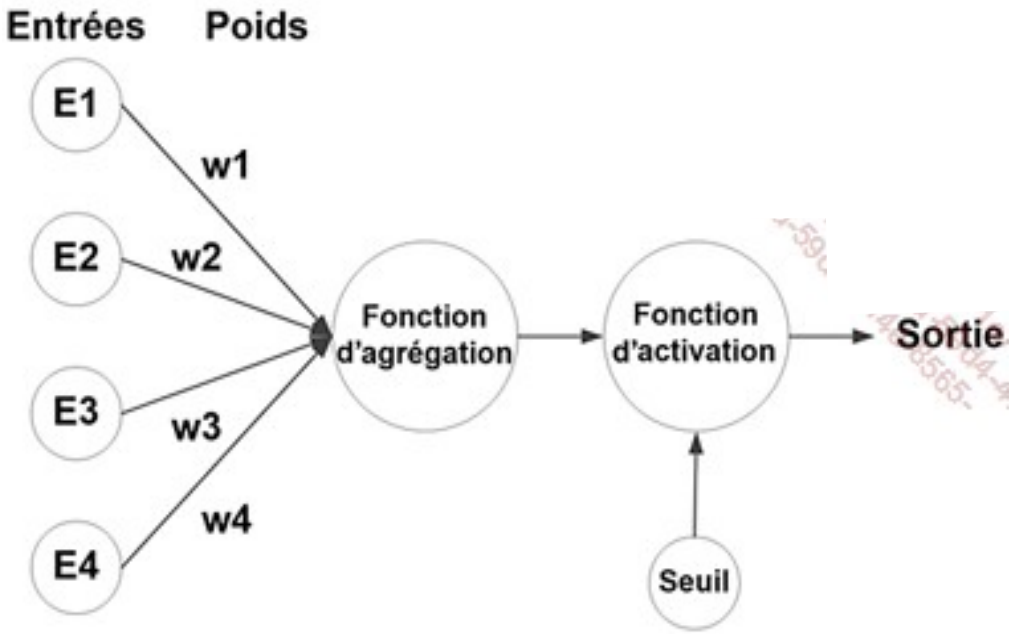


Le neurone formel

Fonctionnement général :

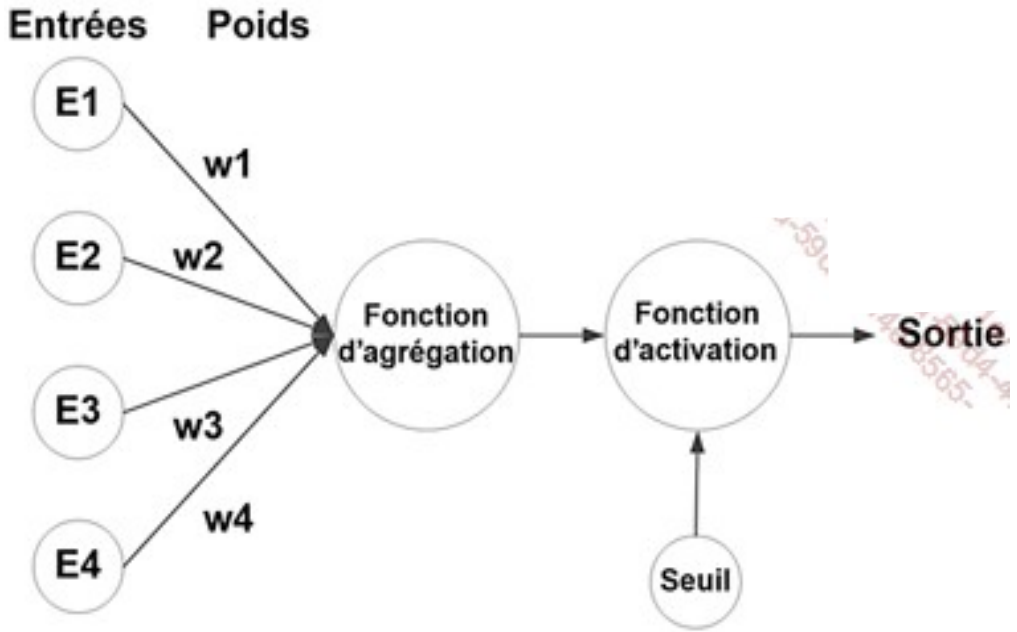
Un neurone reçoit des entrées et fournit une sortie, grâce à différentes caractéristiques :

1) Des poids.
2) Une fonction d'agrégation.
3) Un seuil (ou biais).
4) Une fonction d'activation.



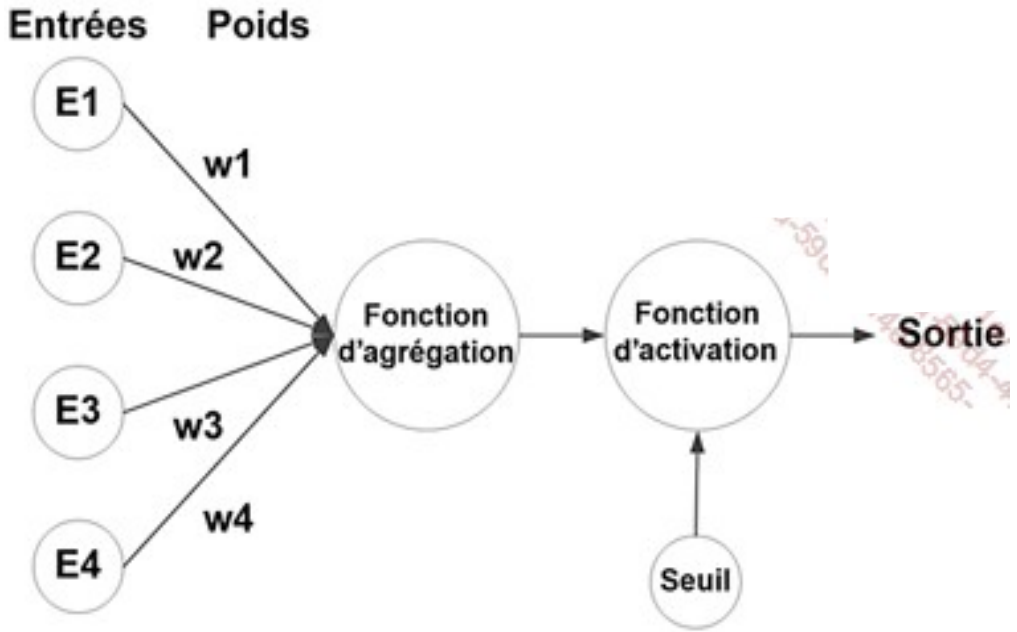
Le neurone formel

- **<u>Des poids</u>** accordés à chacune des entrées, permettant de modifier l'importance de certaines par rapport aux autres.



Le neurone formel

- **<u>Une fonction d'agrégation :</u>** qui permet de calculer une unique valeur à partir des entrées et des poids correspondants.



Le neurone formel

Fonctions d'agrégation :

- **<u>La somme pondérée :</u>** on va simplement faire la somme de toutes les entrées multipliées par leur poids.

  –
$$\sum_{i=1}^{n} E_i * w_i$$

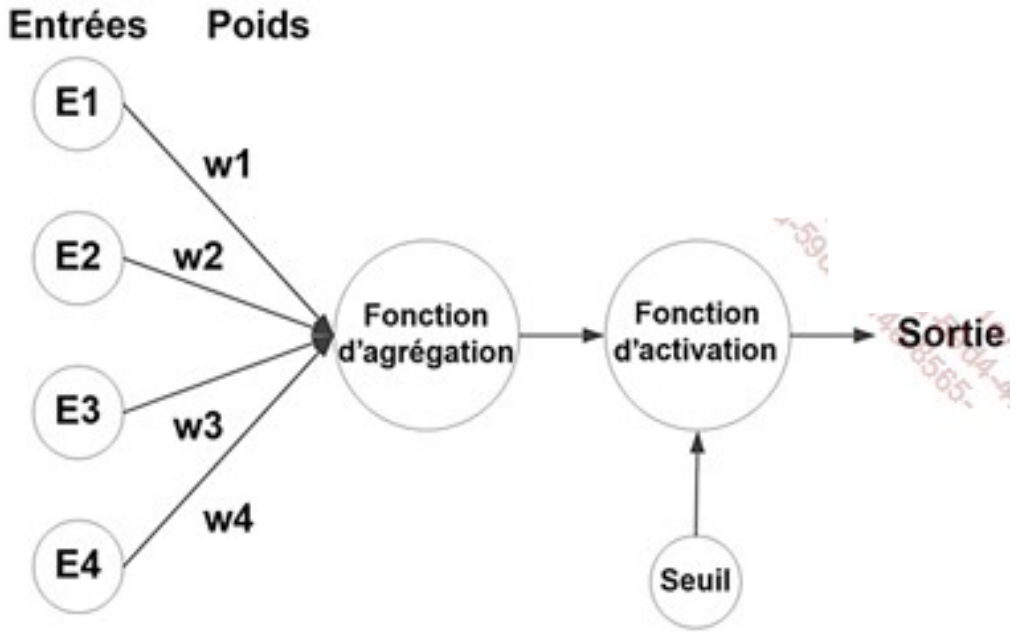
Le neurone formel

- **<u>Le calcul de distance :</u>** on va comparer les entrées aux poids (qui sont les entrées attendues par le neurone), et calculer la distance entre les deux.

$$\sqrt{\sum_{i=1}^{n} (E_i - w_i)^2}$$

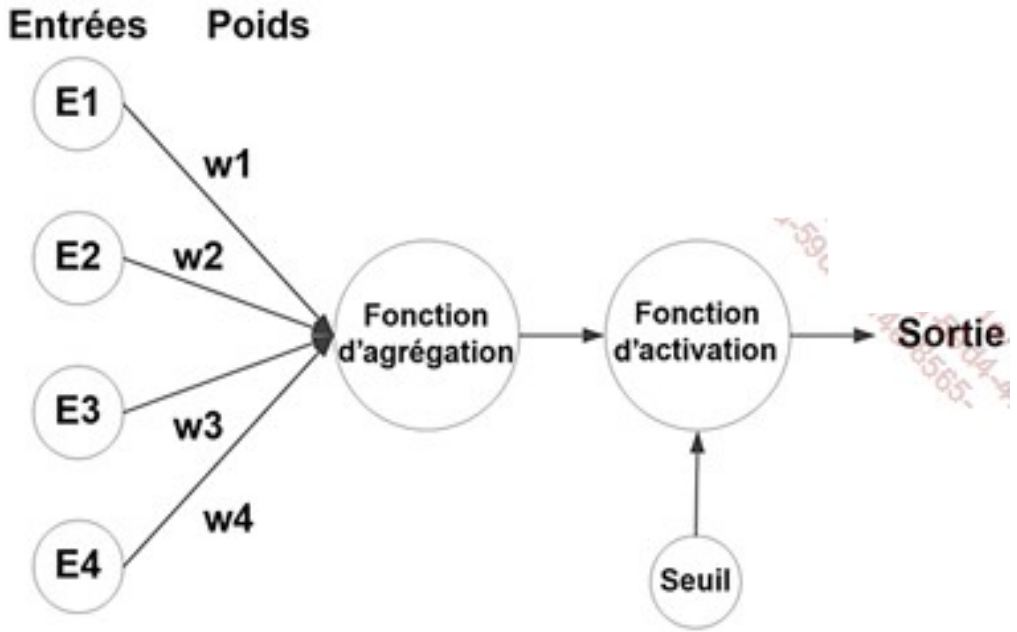
Le neurone formel

- **<u>Un seuil</u>** (ou biais) : permettant d'indiquer quand le neurone doit agir.



Le neurone formel

- **<u>Une fonction d'activation</u>** : qui associe à chaque valeur agrégée une unique valeur de sortie dépendant du seuil.



Le neurone formel

Fonctions d'activation :

Une fois une valeur unique calculée, le neurone compare cette valeur à un seuil et en décide la sortie. Pour cela, plusieurs fonctions peuvent être utilisées.

- Fonction heavyside (signe)

  +1 ou 0

- Fonction sigmoïde

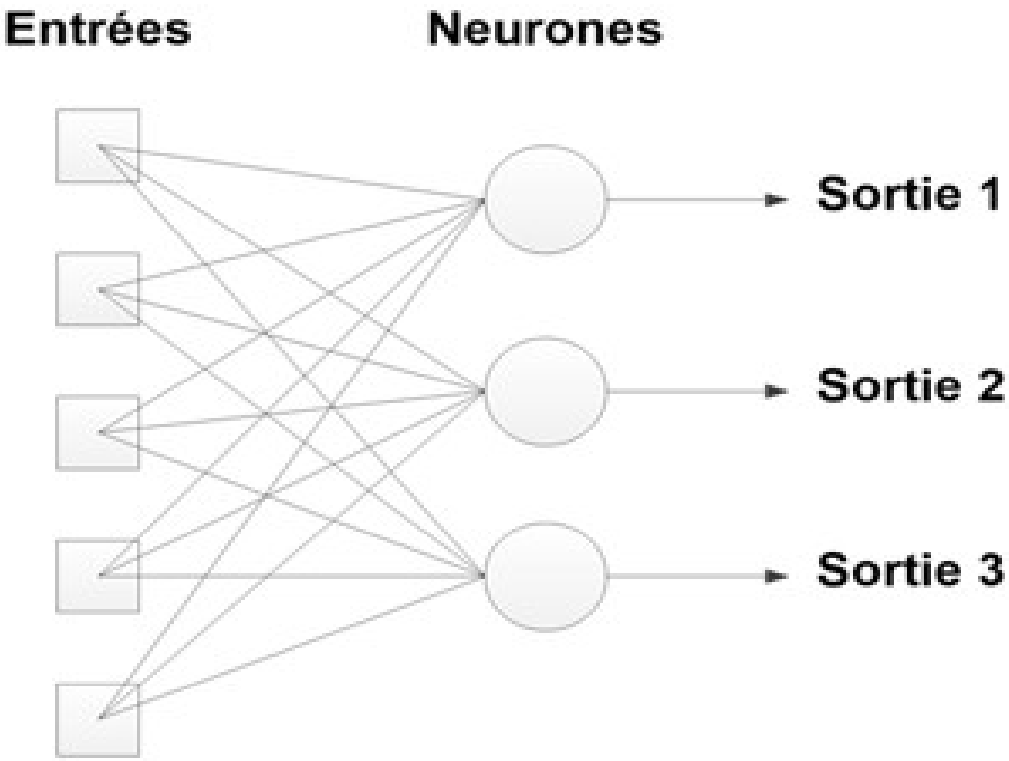
$$f(x) = \frac{1}{1 + e^{-x}}$$

- Fonction gaussienne

$$f(x) = k.e^{-x^2/k\prime}$$

Perceptron

Le perceptron est le plus simple des réseaux de neurones.



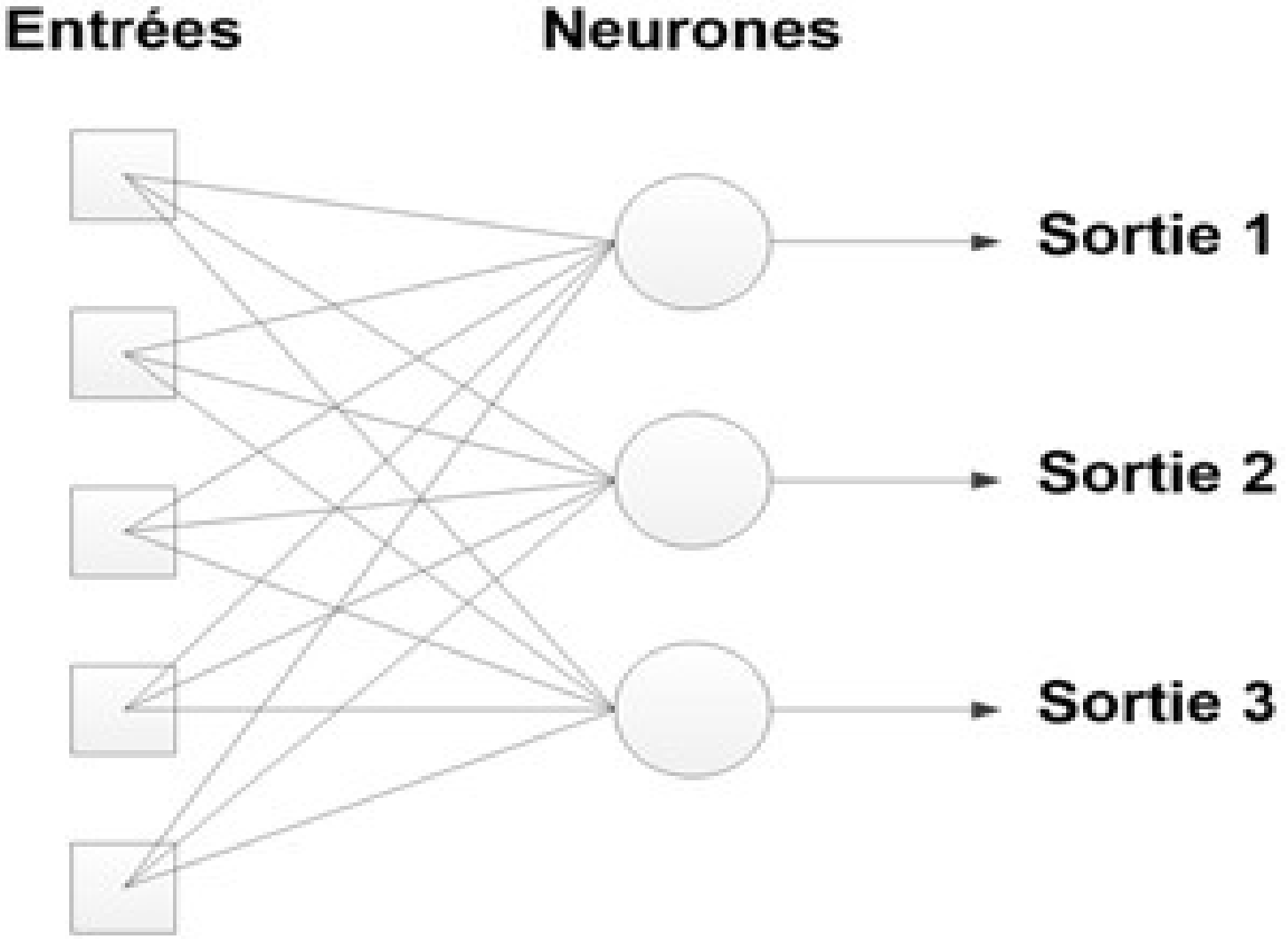
Un perceptron est un réseau contenant **P** neurones.

Chacun est relié aux **N** entrées. Ce réseau permet d'avoir **P** sorties.

Généralement, chacune représente une décision ou une classe, et c'est la sortie ayant la plus forte valeur qui est prise en compte.

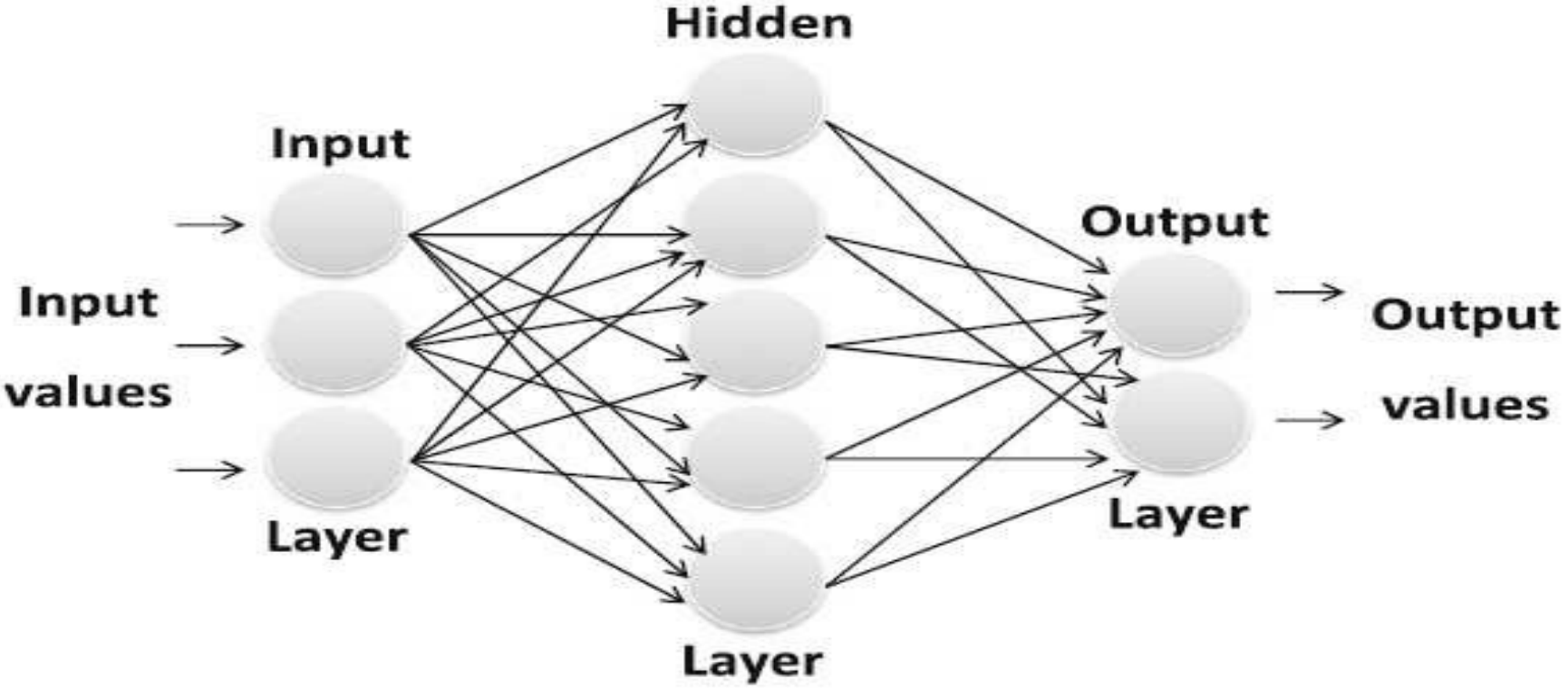
Perceptron

Avec 3 neurones et 5 entrées, on a donc 3 sorties. Voici la structure obtenue dans ce cas :



MultiLayer Perceptron

- Une couche d'entrée
- Une couche de sortie
- Toutes les entrées sont connectées à toutes les sorties



Apprentissage

- L'étape la plus importante dans l'utilisation d'un réseau de neurones est l'apprentissage des poids et seuils. Cependant, les choisir ou les calculer directement est impossible sur des problèmes complexes.

- Il est donc nécessaire d'utiliser des algorithmes d'apprentissage.

Apprentissage

1. Apprentissage non supervisé.

2. Apprentissage par renforcement.

3. Apprentissage supervisé

Apprentissage

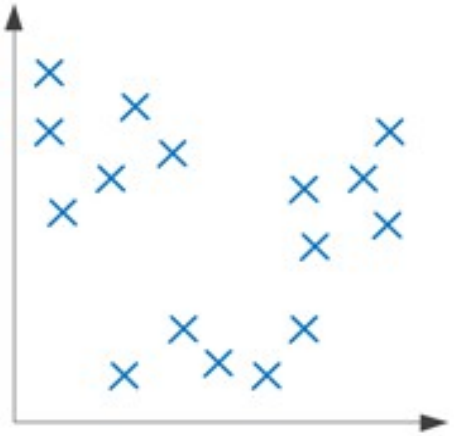
1. Apprentissage non supervisé :

L'apprentissage non supervisé est la forme la moins courante d'apprentissage :

- Pas de résultat attendu.
- On utilise cette forme d'apprentissage pour faire du clustering.

Clustering :

on a un ensemble de données,
et on cherche à déterminer des
classes de faits.



Apprentissage

2. Apprentissage par renforcement :

Dans l'apprentissage par renforcement,

- On indique à l'algorithme si la décision prise était bonne ou non.
- On a donc un retour global qui est fourni. Par contre, l'algorithme ne sait pas exactement ce qu'il aurait dû décider.

L'apprentissage non supervisé peut se faire grâce aux **métaheuristiques**. elles permettent d'optimiser des fonctions sans connaissances a priori.

la technique la plus employée est l'utilisation des **algorithmes génétiques**. Ils permettent, grâce à l'évolution, d'optimiser les poids et de trouver des stratégies gagnantes, sans informations particulières sur ce qui était attendu.

Apprentissage

3. Apprentissage supervisé :

   L'apprentissage supervisé Il est utilisé pour des tâches d'estimation, de prévision, de régression ou de classification.

- On sait quelle sortie on attend.
- Si la sortie obtenue est différente de celle attendue, on doit adapter les poids des connexions.
- On répète l'opération jusqu'à ce que le réseau fournisse des sorties acceptables.

Apprentissage

3. Apprentissage supervisé :

   L'apprentissage supervisé Il est utilisé pour des tâches d'estimation, de prévision, de régression ou de classification.

- On sait quelle sortie on attend.
- Si la sortie obtenue est différente de celle attendue, on doit adapter les poids des connexions.
- On répète l'opération jusqu'à ce que le réseau fournisse des sorties acceptables.

Apprentissage

Soit un réseau possédant X entrées, et N exemples. On note s_i la sortie obtenue sur le énième exemple, et y_i la sortie attendue. L'erreur commise sur un point s'exprime donc :

$$Erreur = y_i - s_i$$

Au début de chaque passe, on initialise à 0 les modifications à appliquer aux poids w_i .

La variation est notée δw_i . À chaque exemple testé, on va modifier celle-ci de la manière suivante :

$$\delta w_i = \delta w_i + \tau . (y_i - s_i) . x_i$$

Apprentissage

Le taux d'apprentissage dépend lui du problème à résoudre :

- trop faible, il ralentit énormément la convergence.
- Trop grand, il peut empêcher de trouver la solution optimale.
- Ce taux sera généralement choisi fort au début de l'apprentissage puis sera réduit à chaque passe.
- Après avoir passé tous les exemples une fois, on applique la modification totale aux poids :

$$w_i = w_i + \delta w_i$$