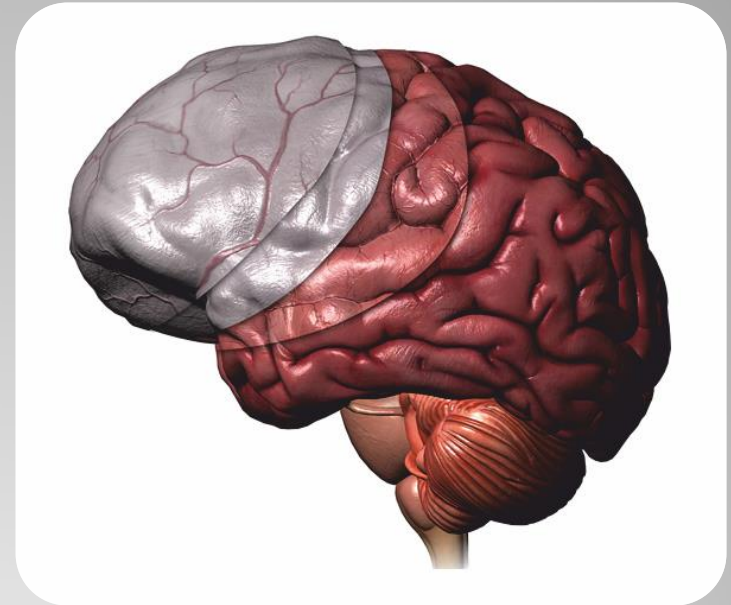


# RO et IA : Réseaux de Neurones

ESGI-PPA-4 Vidal

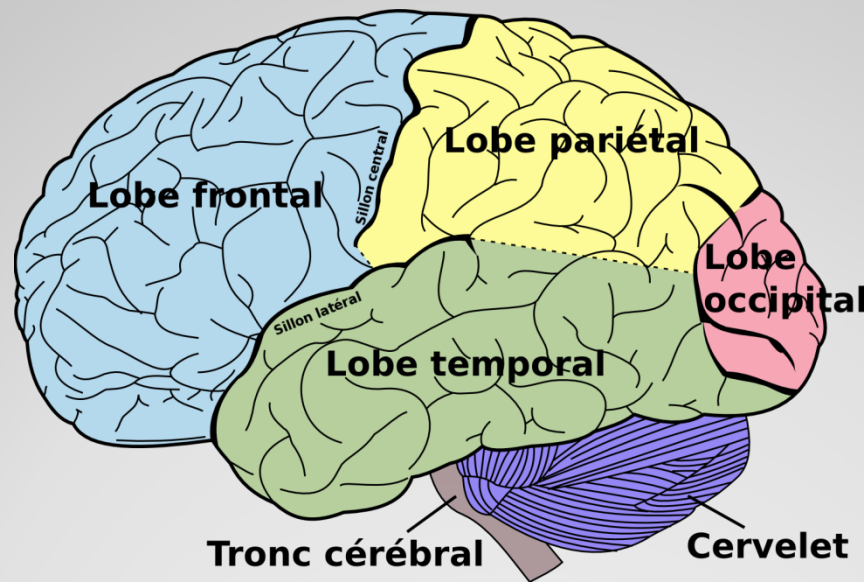
**Pourquoi les réseaux de neurones ?**

- Bonnes propriétés :
  - Parallélisme
  - Apprentissage
    - Généralisation
    - Adaptation
  - Tolérance aux pannes
    - (renvoie toujours un résultat)



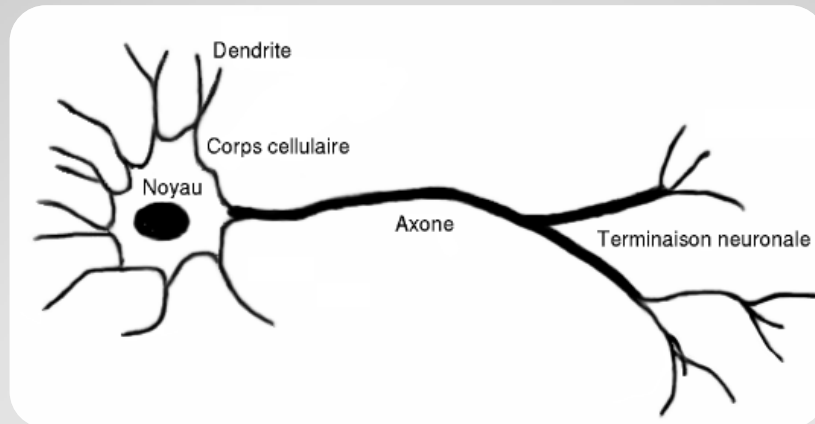
**Le cerveau humain ...**

- Pour avoir de l'intelligence artificielle ...  
« il suffit » de reproduire de manière informatique l'architecture du cerveau ?
  - Comment est organisé le cerveau ?



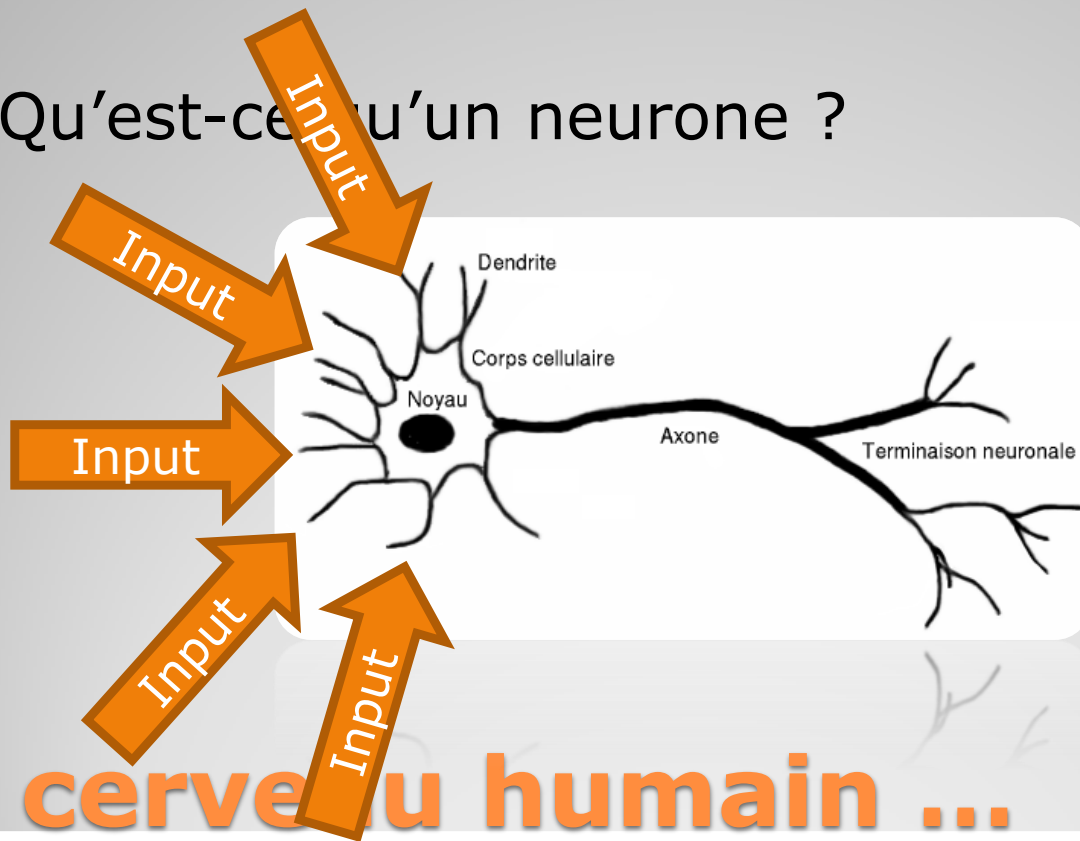
**Le cerveau humain ...**

- Comment chacune de ces zones fonctionne-t-elle ?
  - Un ensemble de neurones interconnectés
  - Qu'est-ce qu'un neurone ?



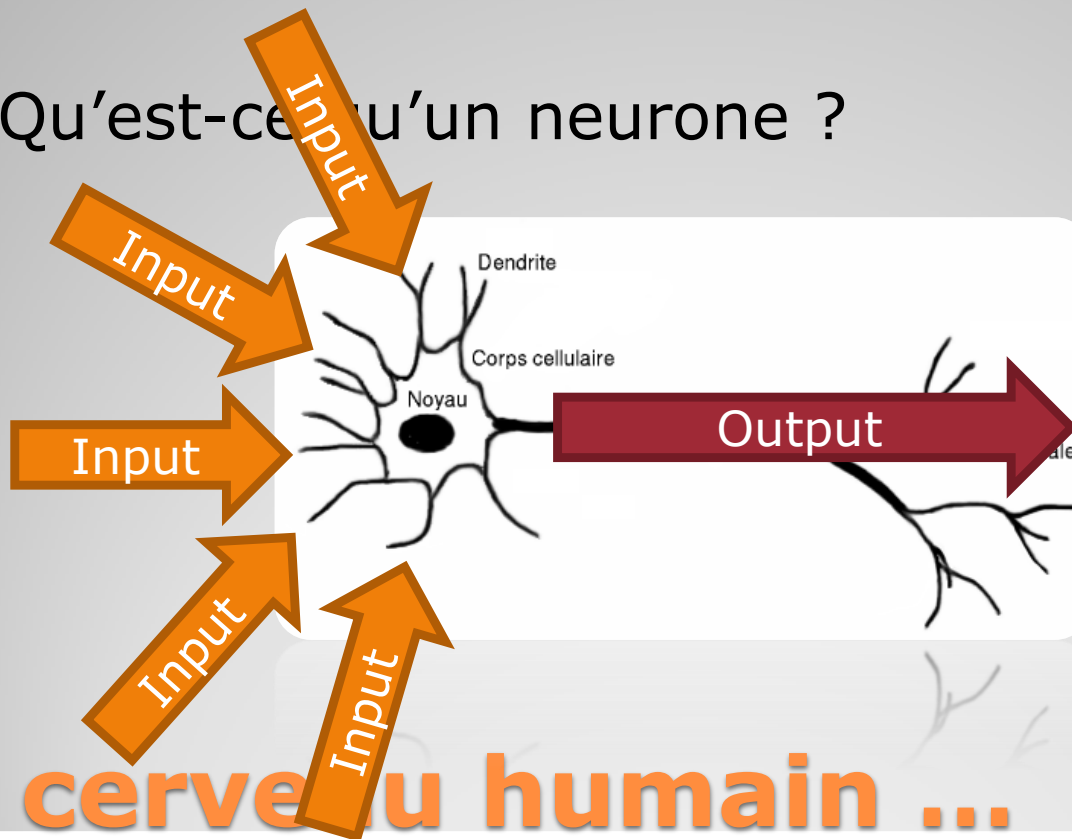
**Le cerveau humain ...**

- Comment chacune de ces zones fonctionne-t-elle ?
  - Un ensemble de neurones interconnectés
  - Qu'est-ce qu'un neurone ?



**Le cerveau humain ...**

- Comment chacune de ces zones fonctionne-t-elle ?
  - Un ensemble de neurones interconnectés
  - Qu'est-ce qu'un neurone ?



**Le cerveau humain ...**

- Peut-on simuler ce fonctionnement par une fonction mathématique ?
  - Le neurone envoie un signal électrique à partir du moment où il est suffisamment stimulé (la somme des signaux électriques des dendrites est d'une intensité suffisante).
  - On cherche donc une fonction de seuil sur la somme des entrées.
  - On souhaite aussi qu'elle soit continue/différentiable (apprentissage).

## Neurone artificiel

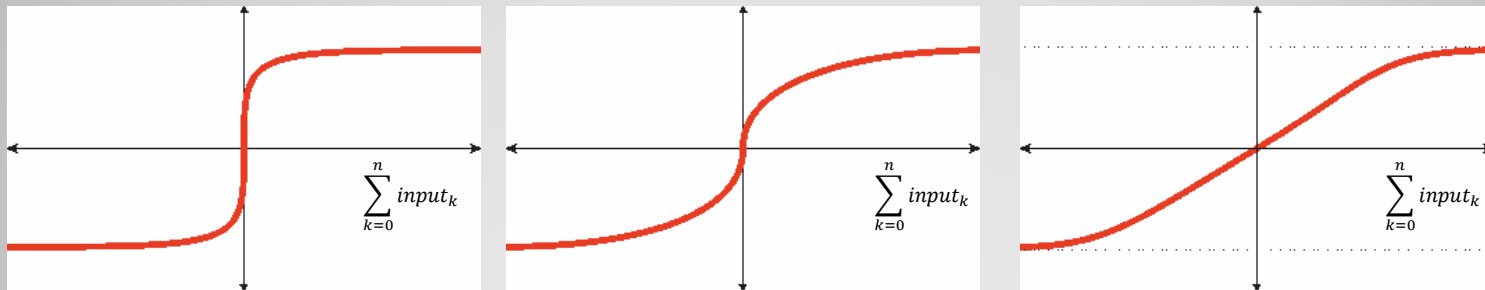


- Peut-on simuler ce fonctionnement par une fonction mathématique ?

- Somme des entrées =>

$$\sum_{k=0}^n input_k$$

- Fonction seuil sur la somme des entrées :



# Neurone artificiel

- Peut-on simuler ce fonctionnement par une fonction mathématique ?

- Somme des entrées =>

$$\sum_{k=0}^n w_k \times input_k$$

- Fonction seuil sur la somme des entrées :

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

$$f(x) \rightarrow \arctan(Kx) \times \frac{2}{\pi}$$

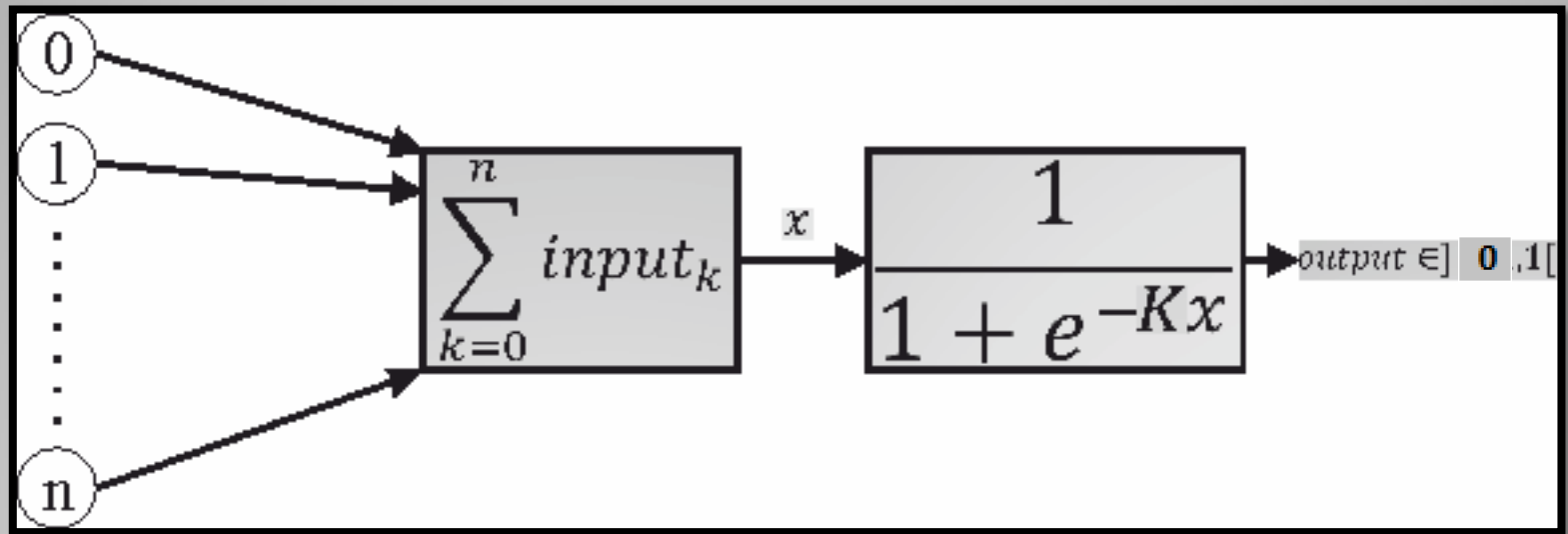
## Neurone artificiel

- Peut-on simuler ce fonctionnement par une fonction mathématique ?
  - Au final la valeur de sortie est dictée par :

$$\frac{1}{1 + e^{-k \sum_{k=0}^n input_k}}$$

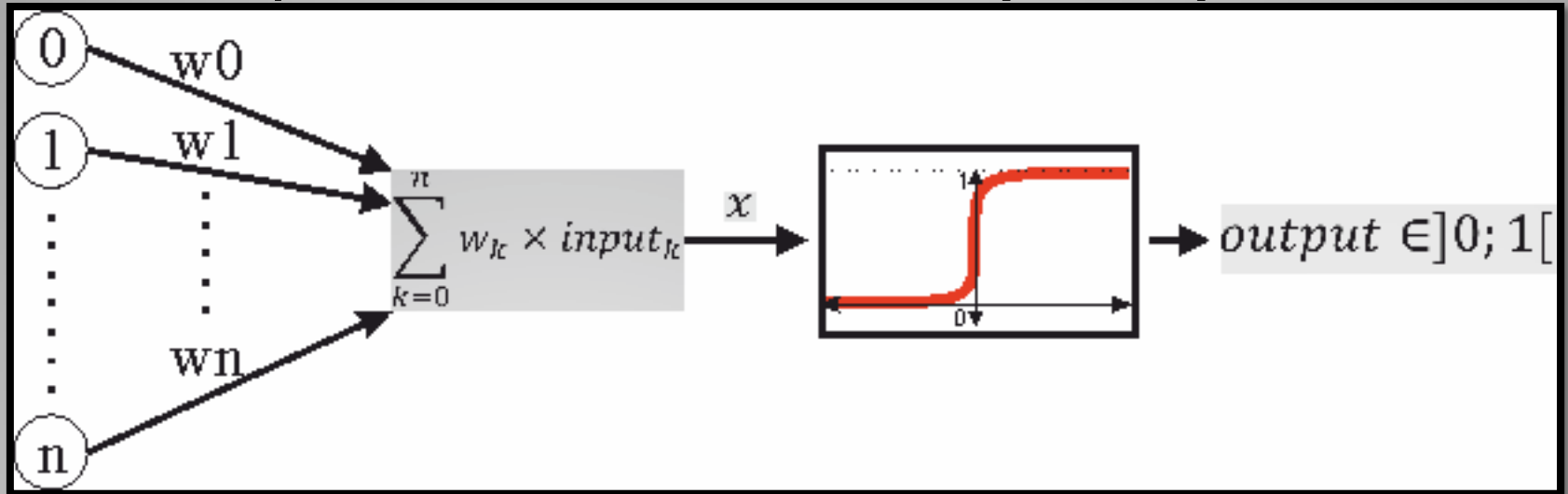
**Neurone artificiel**

- Neurone artificiel fonctionnel :



**Neurone artificiel**

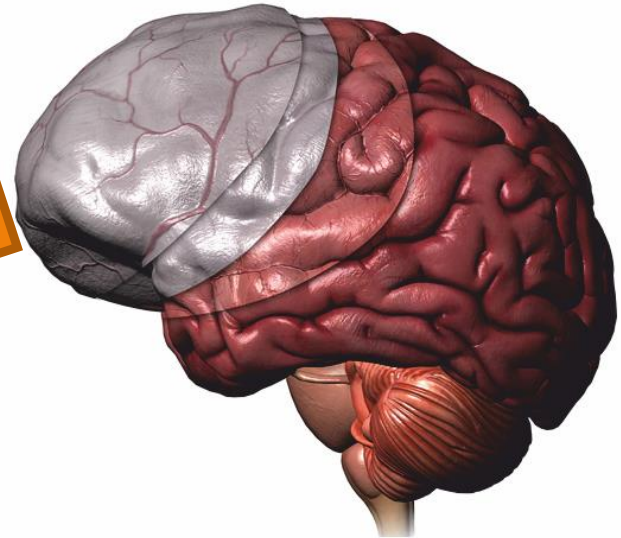
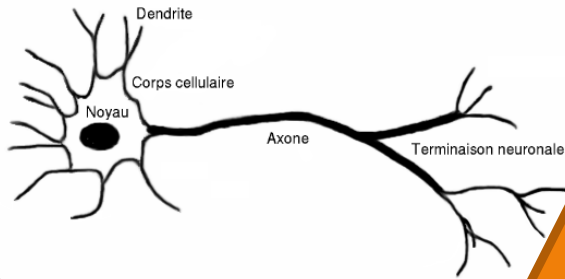
- Perceptron de Rosenblatt (1957) :



- Apprentissage :
  - Trouver les  $w_i$  minimisant l'erreur.
- Exemple

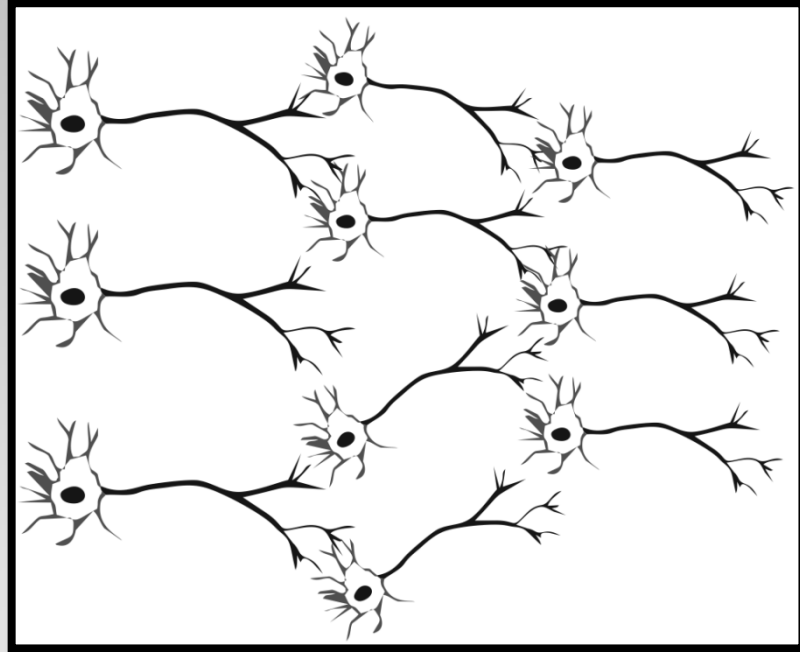
**Neurone artificiel**

- Du neurone au réseau de neurones



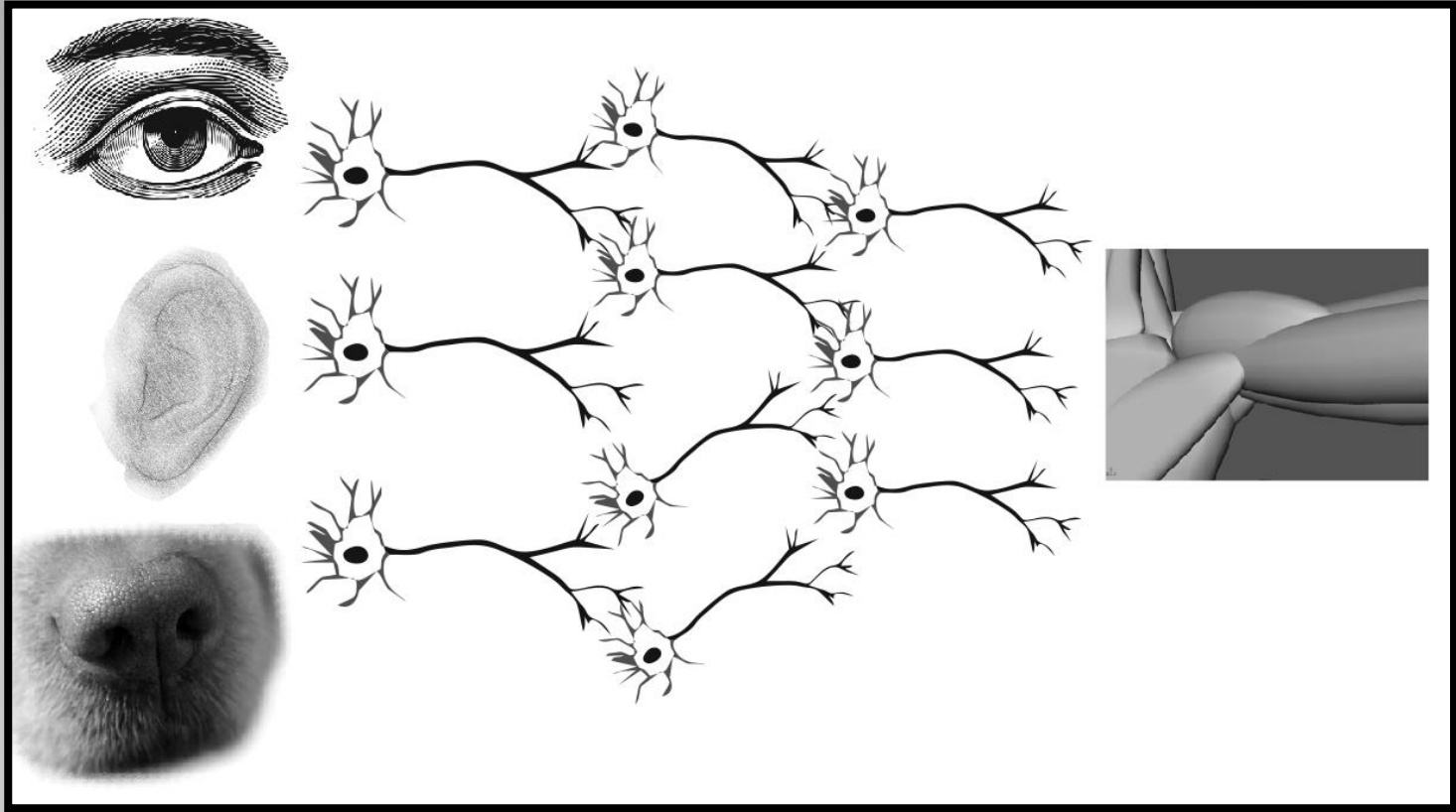
**Réseaux de neurones artificiels**

- Du neurone au réseau de neurones



**Réseaux de neurones artificiels**

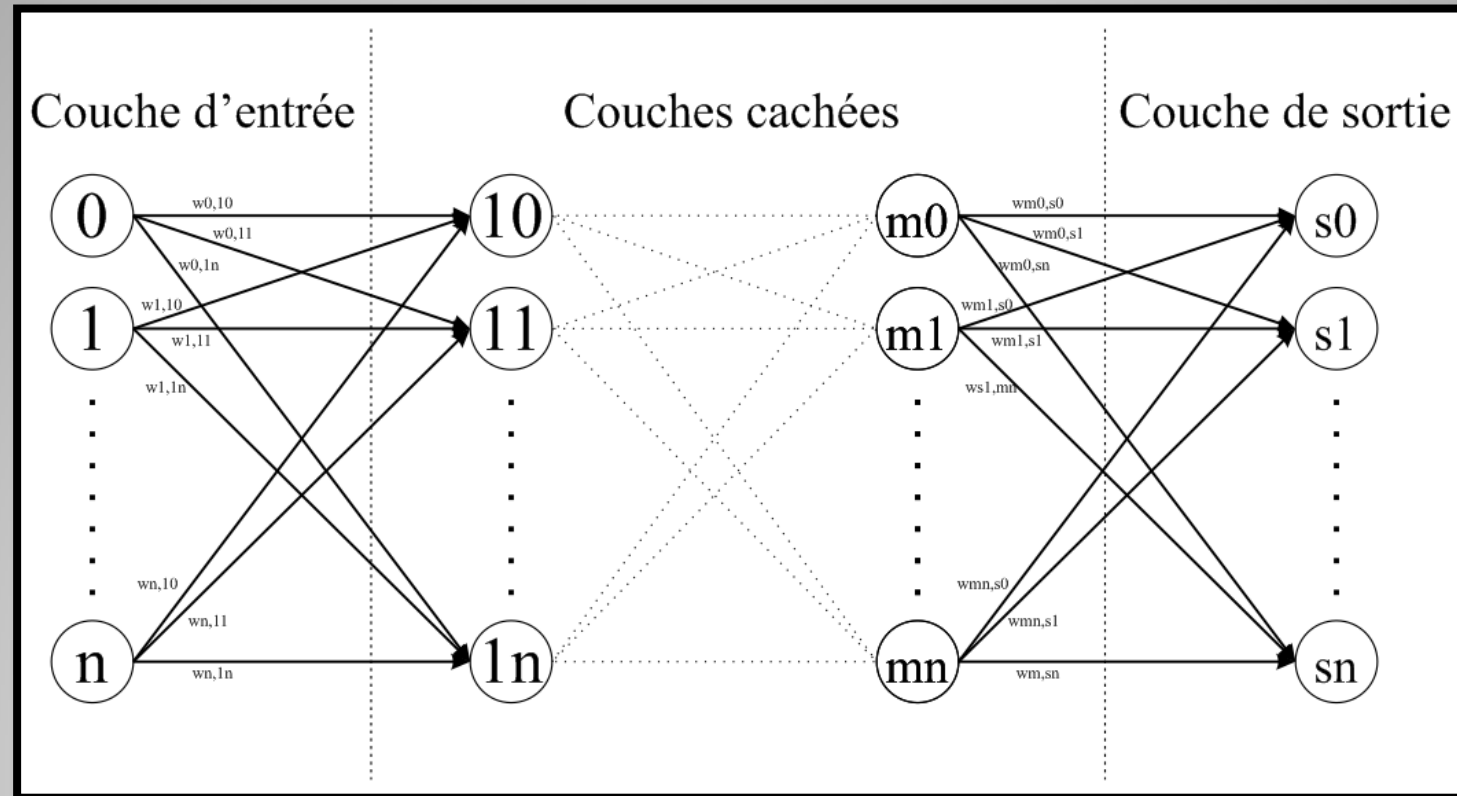
- Du neurone au réseau de neurones



**Réseaux de neurones artificiels**



- Perceptron Multi-Couches (PMC ou MLP)



Réseaux de neurones artificiels

- Théorème d'approximation universelle
  - Cybenko 89
    - Pour toute fonction  $F$  continue définie et bornée sur un ensemble borné, et pour tout  $\varepsilon$ , il existe un réseau de neurones à 1 couche cachée de neurones sigmoïdes qui approxime  $F$  à  $\varepsilon$  près.
  - Sussman 92
    - Les réseaux à une couche cachée forment une famille d'approximateurs parcimonieux : à nombre égal de paramètres on approxime correctement plus de fonction qu'avec des polynômes.

**Réseaux de neurones artificiels**

- Théorème d'approximation universelle
  - Cybenko 89
    - Pour toute fonction  $F$  continue définie et bornée sur un ensemble borné, et pour tout  $\varepsilon$ , il existe **un réseau de neurones à 1 couche cachée** de neurones sigmoïdes qui approxime  $F$  à  $\varepsilon$  près.
    - → Comment trouver un tel réseau ?

**Réseaux de neurones artificiels**

- Apprentissage supervisé
  - On connaît pour un nombre conséquent d'exemples les valeurs d'entrée et de sortie
  - Idée générale :
    - 1 : On divise notre ensemble d'exemples en deux
      - L'ensemble d'apprentissage : A
      - L'ensemble de test : T
    - 2 : Pour chaque exemple de A :
      - On présente au réseau de neurone un exemple de A
      - On compare la sortie obtenue avec la sortie attendue
      - On corrige (**rétropropagation du gradient**) les poids du réseau pour qu'il tende vers la sortie attendue.
    - 3 : On évalue le score du réseau sur les exemples de T
    - 4 : On recommence si le réseau n'obtient pas le score désiré, on arrête dans le cas où le score obtenu est satisfaisant.

# Apprentissage

- Cas d'utilisations
  - Reconnaissance de formes
  - Traitement du signal
    - Classification d'image radars
    - Reconnaissance vocale
    - ...
  - Peu à peu remplacés par d'autres structures de réseaux neuronaux

# Apprentissage

- Limites...

- Paramètres du réseau de neurones
  - Dimensionnement
  - Structure
- Apprentissage de la multiplication de deux nombres
- Gestion du temps
- Sur-apprentissage
- Nécessité d'une base d'exemples représentative
- Phénomène 'boite noire'
- ...

# Apprentissage

- Apprentissage supervisé
  - Radial Basis Function Network
  - Linear Vector Quantization
  - Support Vector Machines
  - ...
- Apprentissage non supervisé
  - Cartes de Kohonen
  - Adaptive Resonance Theory
  - ...
- Prise en compte du temps
  - Time-delay neural network
  - Réseau de neurones bouclés/récurents
  - ...

# Apprentissage

**Métaheuristiques et RdN**



- Le cas NEAT (Neuroevolution of Augmented Topologies)
- Apprentissage indirect
- Un individu = un réseau (ensemble des poids + structure)
- Exemple 1 : SmartSweepers
- Exemple 2 : Galactic Arms Race
  - <http://www.youtube.com/watch?v=QiB0k6ar1mg>

**Métaheuristiques et RdN**