

Réseaux de neurones artificiels

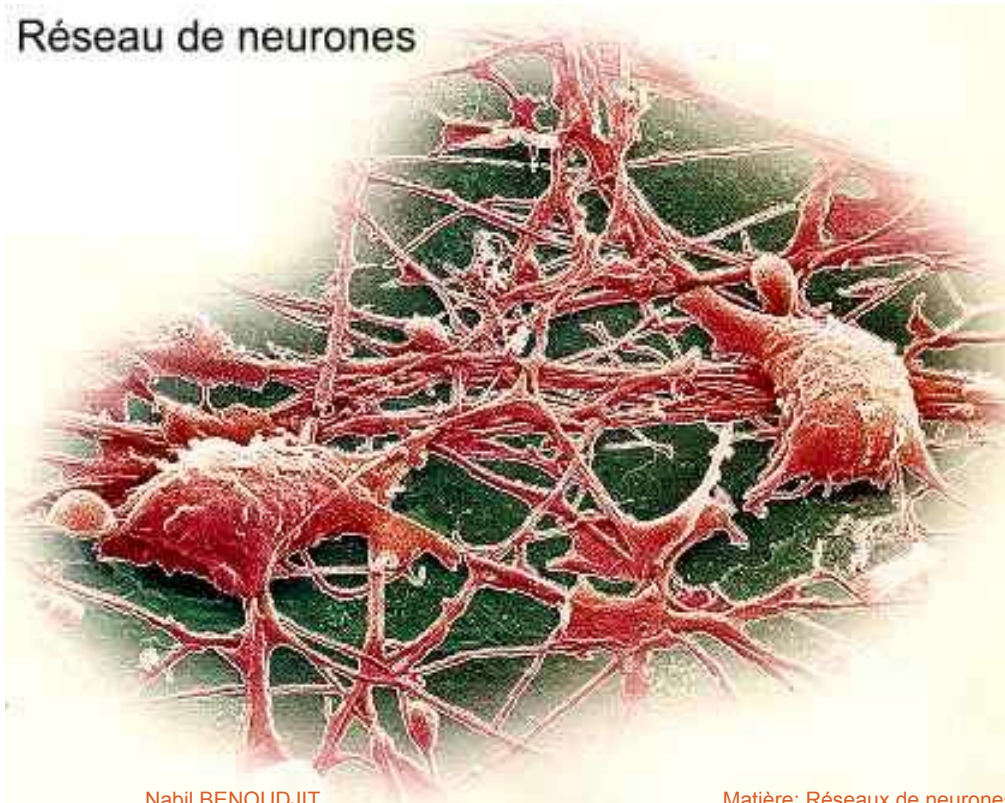
Pr. Nabil Benoudjit

Université de Batna -2-

Email: n.[benoudjit@univ-batna2.dz](mailto:n.benoudjit@univ-batna2.dz)

Introduction

Réseau de neurones



Nabil Benoudjit

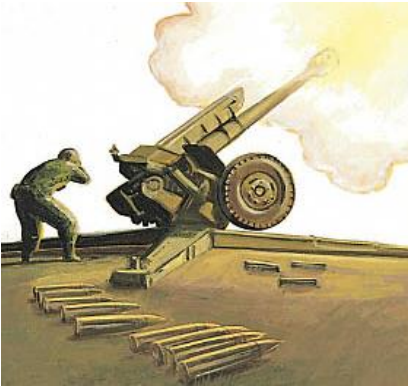
LES RNA, est-ce nécessaire?

Apport ... par rapport au conventionnel

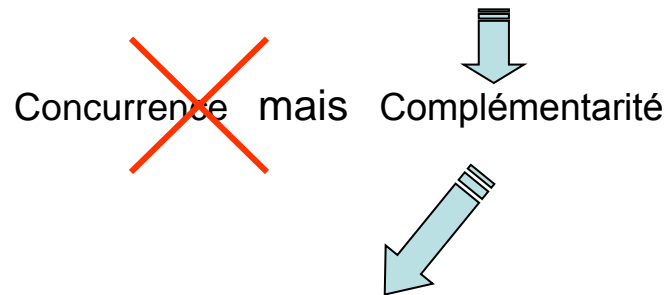
IMPORTANT

Justifier de l'utilisation des RNA :

- ✓ prouver les limites des techniques conventionnelles pour un problème donné (difficile ou impossible à appréhender le problème);
- ✓ ajouter un plus par rapport au conventionnel.



Gaspillage temps de calcul & mémoire calculateur pour résoudre un problème sans mérite



Profiter de la puissance de calcul des RNA à travers :

- ✚ **Parallélisme** (Temps de propagation de l'information)
- ✚ **Généralisation** (Pour des entrées non rencontrées auparavant)

Introduction (1)

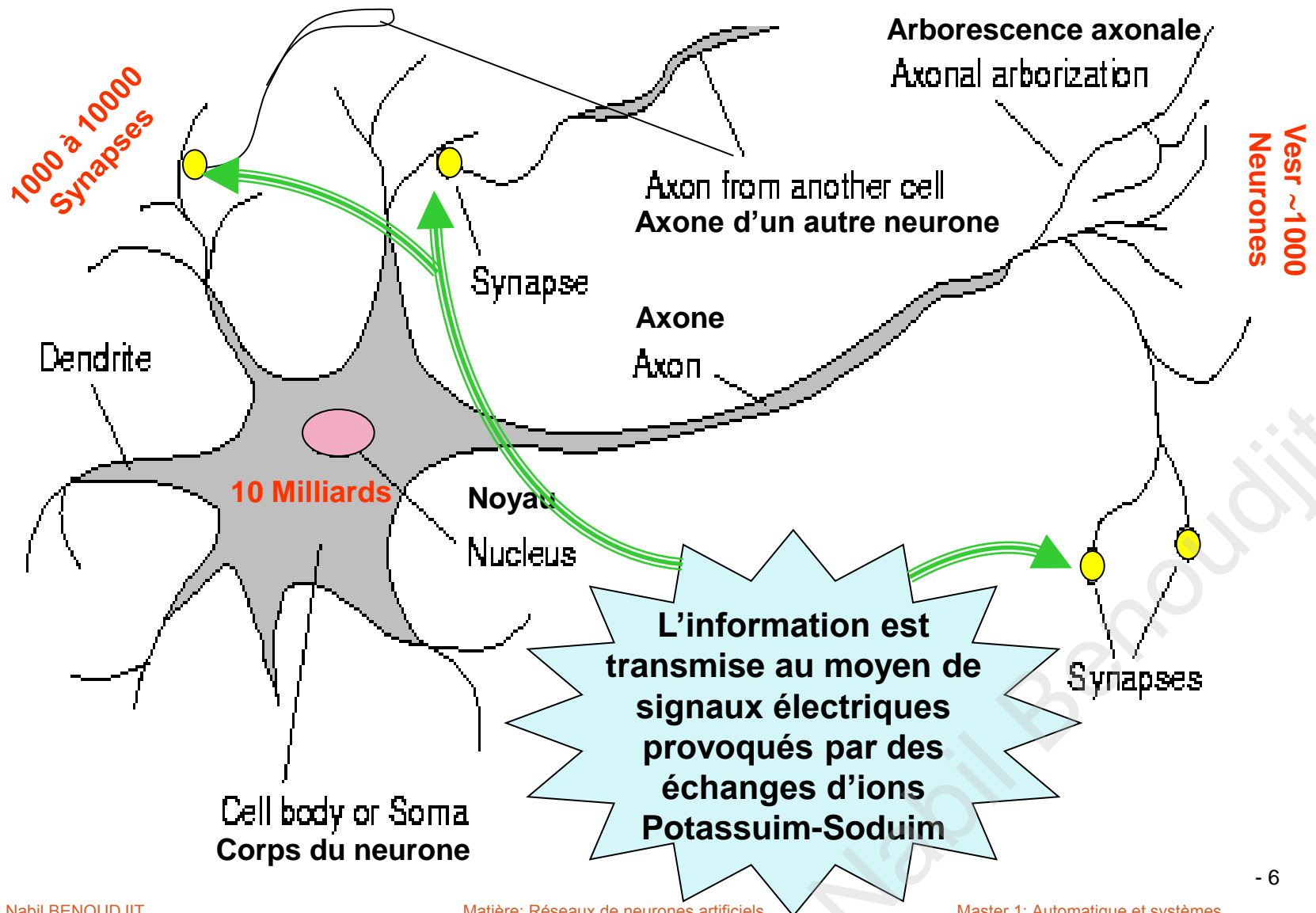
- ❑ Le cerveau humain est un modèle dans le développement de l'intelligence artificielle. De simples enchevêtrements de neurones lui donnent toutes ses capacités, et c'est ce que tentent de reproduire les chercheurs grâce à l'électronique.
- ❑ L'intelligence artificielle est en continue progression depuis l'invention de l'ordinateur et l'utilisation de programmes informatiques. Il existe en effet de nombreux programmes capables de réaliser des choses de plus en plus complexes: diriger un robot, résoudre des problèmes, jouer aux échecs... Mais ils sont très rarement capables de rivaliser avec le cerveau humain, et c'est pour cela que de nombreuses tâches sont encore irréalisables par les ordinateurs. D'où vient cette différence ? La première chose est évidemment la capacité de calcul, le cerveau est une machine assurément incroyable. Cependant, les ordinateurs n'ont pas cette faculté d'apprentissage, ils ne connaissent pas le progrès si personne ne les modifie. **Voilà le challenge pour l'intelligence artificielle: savoir apprendre.**

Introduction (2)

- D'un autre côté, la biologie a apporté un grand nombre d'informations sur le fonctionnement du cerveau, des neurones... Des mathématiciens ont alors tenté de reproduire le fonctionnement du cerveau en intégrant ces connaissances en biologie dans des programmes informatiques, et en leur donnant la possibilité d'apprendre. Cela a commencé en 1943 avec Mc Culloch et Pitts, mais en 1969, Minsky et Papert publièrent un livre pour montrer que le type de réseau élaboré à l'époque était limité. Heureusement, des progrès ont pu être réalisés par la suite. Ces recherches sur les "réseaux de neurones artificiels" ont maintenant beaucoup progressé.

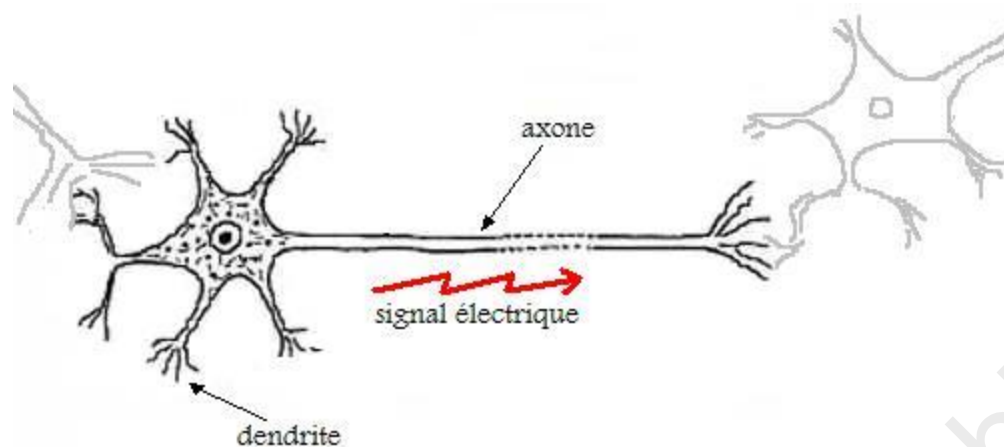
Le Neurone Biologique (1)

Unité fondamentale, fonctionnelle et anatomique du tissu nerveux,



Le Neurone Biologique (2)

- ❑ Les modèles de réseaux de neurones artificiels sont, à l'origine, une imitation du fonctionnement du cerveau. Il contient, chez l'homme, environ 10 milliards de neurones, et chacun est connecté à environ 10.000 autres neurones. On voit ainsi sa complexité étonnante. Les connexions permettent le transfert d'informations sous forme d'impulsions électriques entre les neurones.



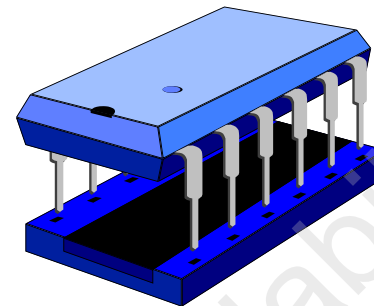
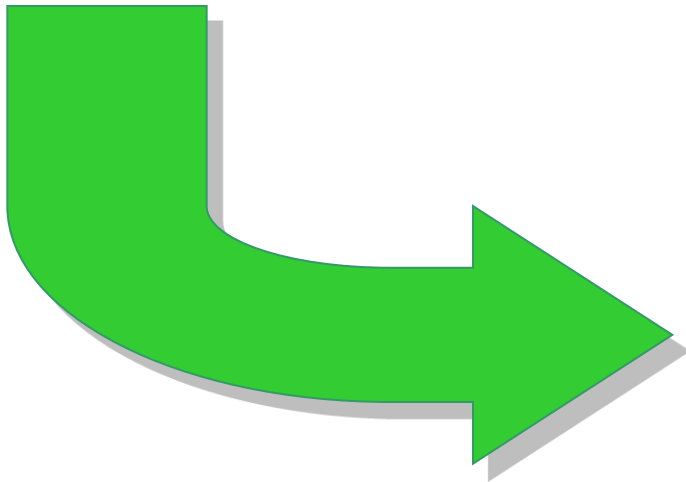
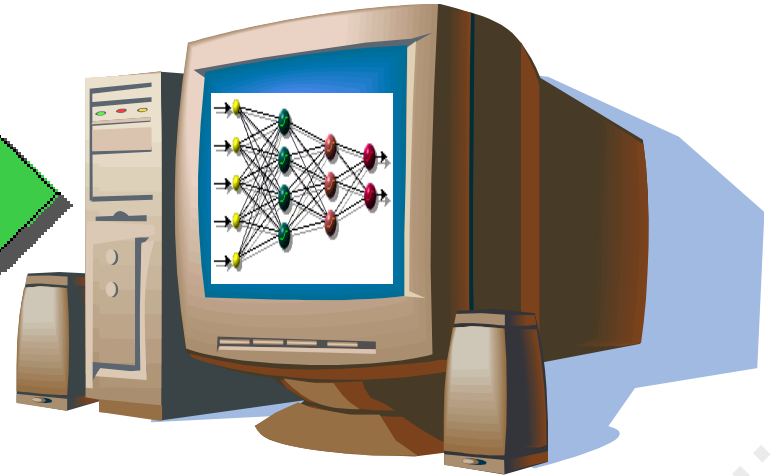
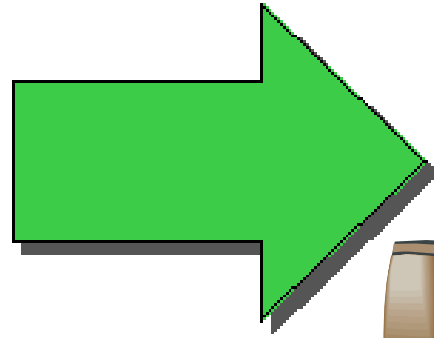
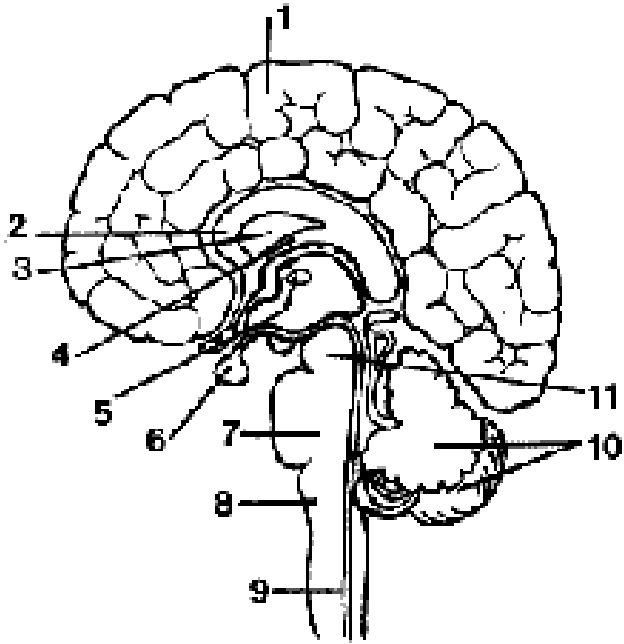
Le Neurone Biologique (3)

- ❑ Un neurone reçoit des impulsions de ses voisins par l'intermédiaire des "dendrites".
- ❑ Si la somme des signaux dépasse un certain *seuil*, il renvoie un signal vers d'autres neurones, par l'intermédiaire de son "axone". Ce mécanisme représente la façon dont les informations sont transmises:
- ❑ Un neurone ne se borne pas à faire passer l'information, il la filtre.
- ❑ Pour résumer, un neurone peut être schématisé ainsi: il fait la somme de toutes les informations qu'il reçoit et il émet un signal à condition que la somme soit suffisamment élevée.

IMITATION DU CERVEAU : SOFT & HARD

Réseaux Neuro-mimétiques ;

Réseaux de Neurones Artificiels (RNA)



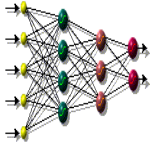
HISTORIQUE

- ❖ **Av. 1940: von Hemholtz, Mach, Pavlov, etc.**
 - Théories générales d'apprentissage, vision, traitement;
 - Pas de modèle mathématique spécifique pour le fonctionnement du neurone;
- ❖ **1940s: Hebb, McCulloch & Pitts**
 - Modèle simple de neurone biologique (sortie binaire, poids fixes);
 - Mécanisme pour l'apprentissage des neurones biologique;
 - Neurone formel (le **Perceptron** : fonctions logiques);
- ❖ **1959: Rosenblatt, Widrow and Hoff**
 - Premiers réseaux pratiques et règles d'apprentissage;
- ❖ **1969: Minsky and Papert**
 - Démonstration des limites des Réseaux existants & pas de nouveaux algorithmes d'apprentissage, Recherches suspendues;
- ❖ **1970s: Amari, Anderson, Fukushima, Grossberg, Kohonen**
 - Progrès continus, bien que à pas lents;
- ❖ **1980s: Grossberg, Hopfield, Kohonen, Rumelhart, etc.**
 - Nouveaux développements importants causent une résurgence (Algorithme de la rétro propagation en 1986).

DEFINITIONS (1)

RNA

Les réseaux de neurones (ou neuronaux) artificiels (**RNA**) consistent en des modèles plus ou moins inspirés du fonctionnement cérébral de l'être humain (cerveau), en se basant sur le concept de neurone⁽¹⁾.

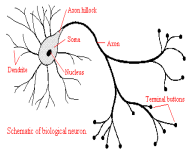


Un réseau de neurones (ou neuronal) est un processeur massivement **parallèle** et **distribué** constitué d'unités de traitement simples (neurones) qui ont la propriété naturelle de **stocker l'information** (connaissances) et de la mettre disponible pour l'utilisation ⁽²⁾.

Neurone

Le neurone est l'unité élémentaire de traitement. Il est doté d'entrées (**dendrites**) pouvant recevoir des signaux et une sortie (**axone**) distribuant le résultat du traitement.

Les échanges de signaux se font au niveau des **synapses** reliant les axones aux dendrites des autres cellules (neurones).



1. Def. Algorithmes génétiques et réseaux de neurones, J.M. Rendres, Ed. Hermes
2. Def. Neural Networks, S. Haykin, Ed. Prentice Hall Inc.,

Réseaux de neurones: définitions et propriétés

- ❑ Un neurone est une fonction algébrique non linéaire, paramétrée, à valeurs bornées.

- ❑ *Un réseau de neurones est un processus distribué de manière massivement parallèle, qui a une propension naturelle à mémoriser des connaissances de façon expérimentale et de les rendre disponibles pour utilisation. Il ressemble au cerveau en deux points:*
 - 1 - *la connaissance est acquise au travers d'un processus d'apprentissage;*
 - 2 - *les poids des connections entre les neurones sont utilisés pour mémoriser la connaissance.*

Définition proposé par l'auteur Simon Haykin en 1994.

Caractéristiques (1)

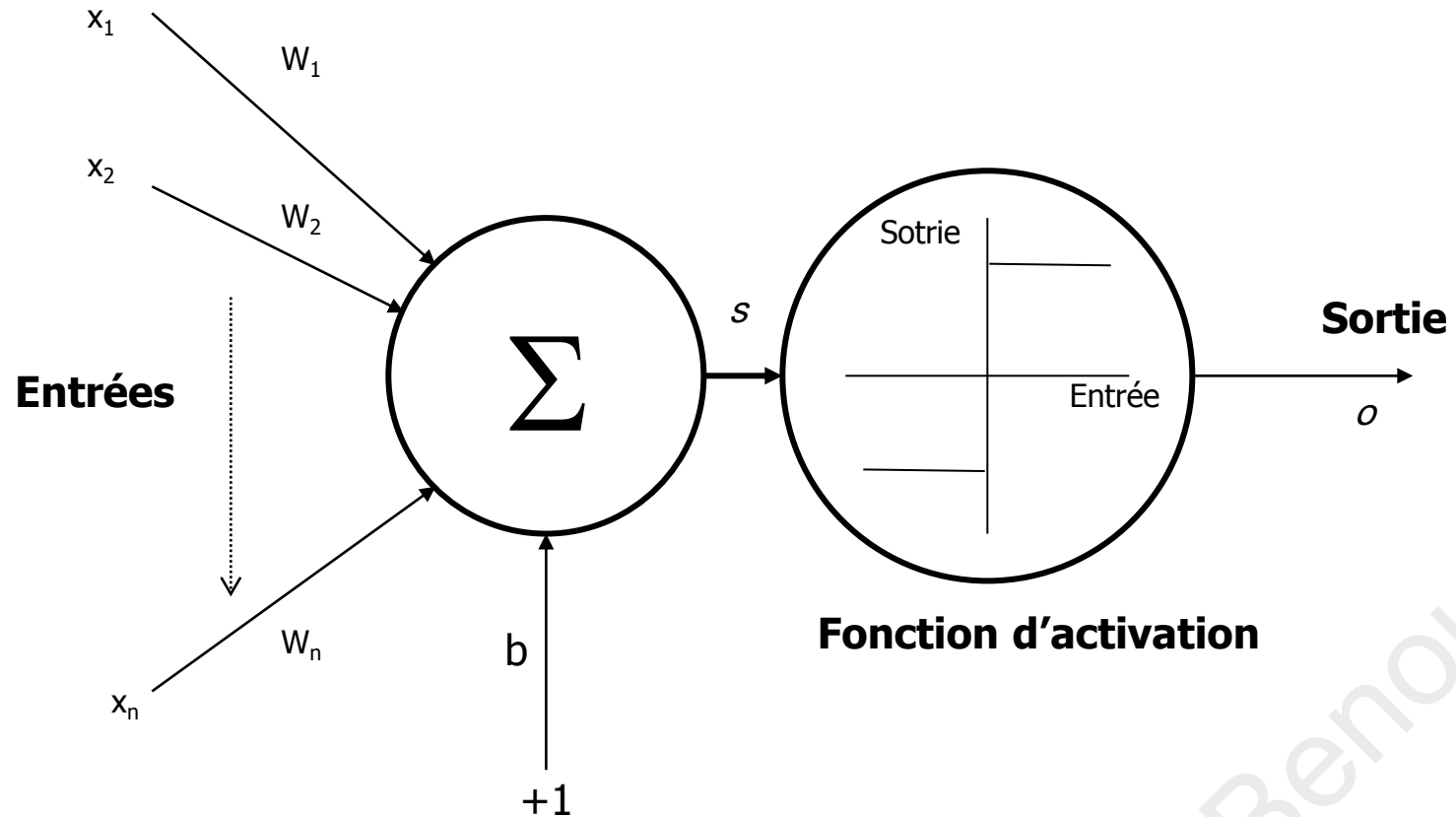
- **L'unité de base ou “neurone artificiel”** (modélisation simpliste d'un neurone biologique): unité qui effectue un calcul simple sur ses “entrées” (x_i), généralement une somme pondérée, dont le résultat est soumis à une transformation souvent non linéaire $F[\cdot]$, encore appelée **fonction d'activation**. Le résultat ou valeur de sortie (notée y) est encore appelée *niveau d'activation*.

$$y = F \left[\sum_{i=1}^n w_i x_i + w_0 \right] = F \left[\sum_{i=0}^n w_i x_i \right] = F \left[\mathbf{w}^T \mathbf{x} \right]$$

Où $\mathbf{x} = [\mathbf{1}, x_1, x_2, \dots, x_n]^T$ est le vecteur d'entrée “augmenté” avec $x_0 = 1$ pour prendre en compte un terme indépendant (w_0) dans le vecteur poids: $\mathbf{w} = [w_0, w_1, w_2, \dots, w_n]^T$

LE NEURONE FORMEL

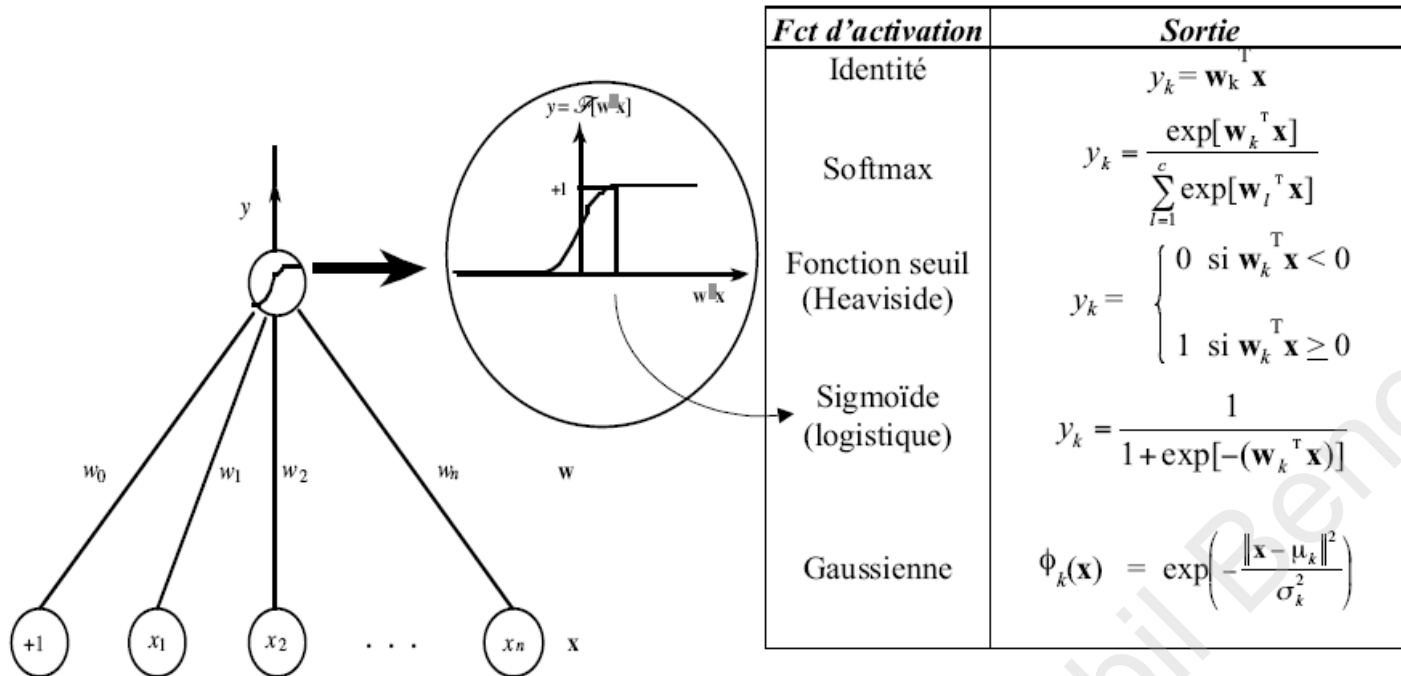
LE PERCEPTRON



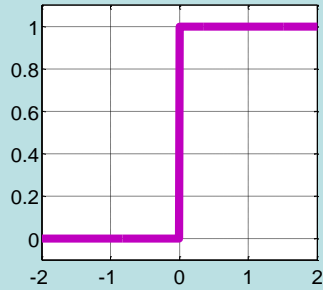
$$o = f\left(\sum_{i=1}^n w_i x_i + b\right) = f\left([w_1 \dots w_n \ b] \cdot [x_1 \ x_2 \dots x_n \ 1]^T\right)$$

Caractéristiques (2)

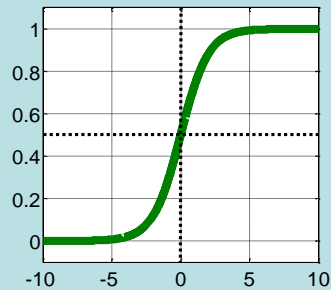
- Les **poids** sont des **paramètres adaptatifs**, dont la valeur est à déterminer en fonction du problème via un algorithme d'apprentissage (ou entraînement).



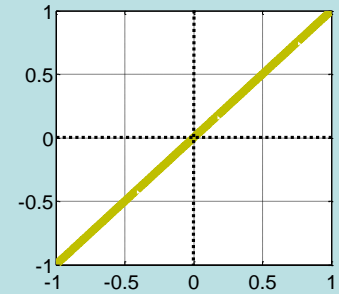
FONCTIONS D'ACTIVATION



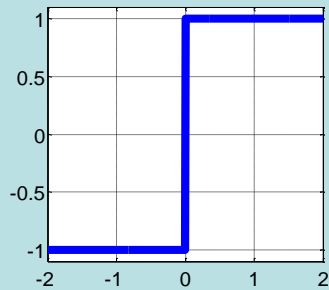
$y=\text{hardlim}(x)$



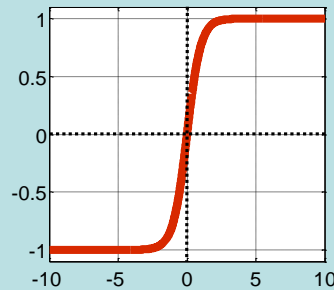
$y=\text{logsig}(x)$



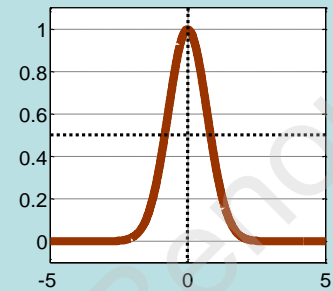
$y=\text{purelin}(x)$



$y=\text{hardlims}(x)$



$y=\text{tansig}(x)$



$y=\text{radbas}(x)$

Caractéristiques (3)

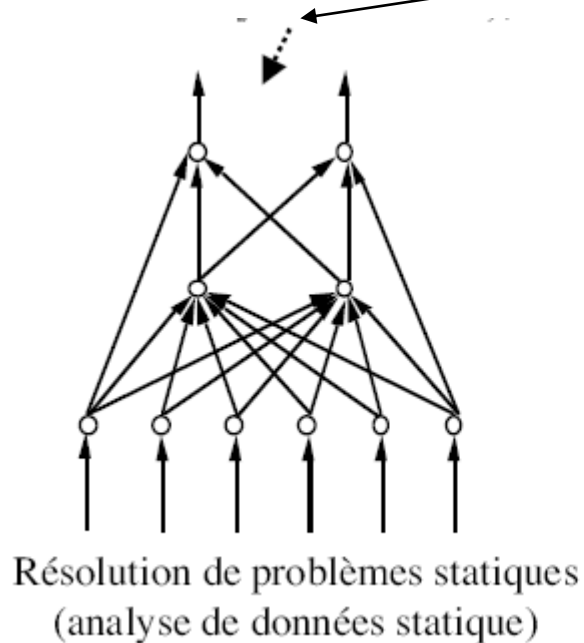
- ❑ Remarque: Lors de l'usage de sigmoïdes comme fonction d'activation, il est fortement conseillé de normaliser les valeurs des entrées dans un intervalle restreint ($[-1, 1]$, ou $[0, 1]$, suivant le type de sigmoïde) afin de centrer la majorité des données dans la zone de linéarité de la sigmoïde (aspects pratiques).

- ❑ **L'architecture du réseau:** fixe le schéma d'inter-connexion des unités. Chaque connexion est généralement caractérisée par un sens et un **poids de connexion** (noté $w = \textit{weight}$) indiquant la force de la connexion (à déterminer).

- 1. On distingue 3 types d'unités: les **unités d'entrée**, activées par un signal d'entrée (*input*), les **unités cachées** (*hidden*) n'ayant pas de connexion avec "l'extérieur".

Caractéristiques (4)

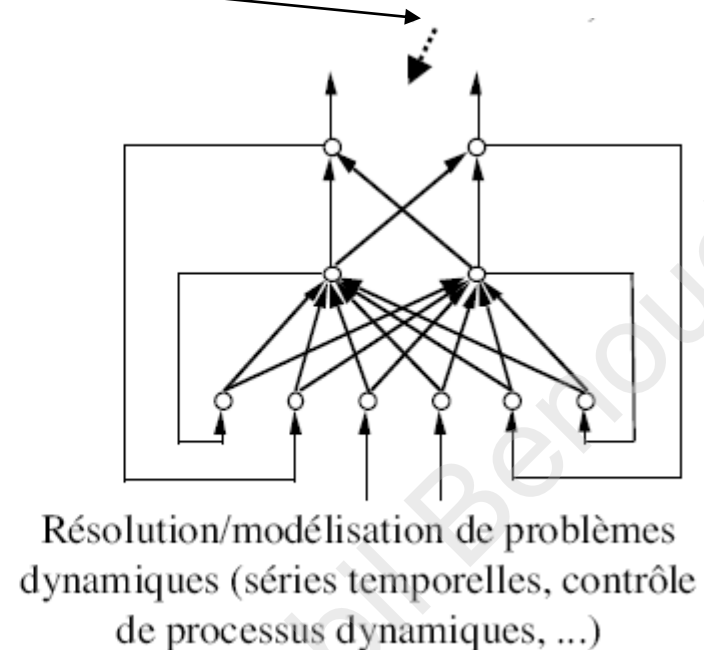
2. Le signal se propage des entrées vers les sorties via les unités cachées.
3. On distingue 2 type d'architecture: sans boucle de rétroaction et souvent organisé en couches (*feedforward*), ou incluant des boucles de rétroaction (*récurrent*).



sorties

unités
cachées

entrées



Caractéristiques (5)

- **Dénomination** pour les réseaux en couches:

Ex: **réseaux à L couches ou L -layer** (en réalité compte $L + 1$ couches):

couche 0 = unités d'entrée, couche 1 = 1^{ère} couche cachée, couche 2 = 2^{ème} , ...,
couche L = couche de sortie.

Propagation du signal de couche en couche: $y_k^{l+1}(\mathbf{x}) = F \left[\sum_{i=1}^{n_l} w_{ik}^{l+1} y_i^l(\mathbf{x}) \right]$

sortie de la: $k^{\text{ème}}$ unité de la couche $l+1$ $i^{\text{ème}}$ unité de la couche l
poids de connexion entre les 2 unités

- **L'algorithme d'apprentissage:** ajuste les valeurs des poids (et éventuellement l'architecture) afin de réaliser une tâche donnée le mieux possible (ex: problème de classification, supervisée ou non, de régression,...).

Apprentissage des RNA

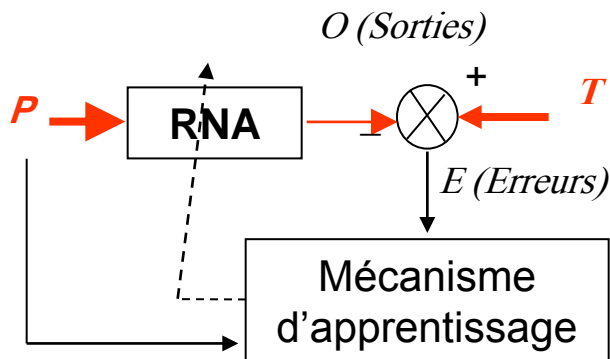
L'apprentissage pour les RNA (et cerveau) est une étape inéluctable.

Ce processus requiert des exemples (généralement de l'expérience) à présenter au réseau (**étape cruciale**). Ces exemples peuvent être :

Entrées-Sorties désirées : (P , T) ;

où P matrice pattern et T Matrice target

App. Supervisé



Entrées : (P) ;

App. Non Supervisé

En fonction des différentes excitations, le réseau organise les P en catégories.

Après apprentissage, lorsque une entrée est présentée au réseau, une sortie est activée indiquant son appartenance à une classe, sinon une nouvelle classe est Générée.

L'apprentissage peut être en ligne (On line) ou hors ligne (Off line)

Caractéristiques (6)

❑ L'apprentissage nécessite:

- ❑ un ensemble de données sur lesquelles se baser, encore appelé ensemble **d'apprentissage/entraînement** (*learning / training set*); dans les problèmes supervisés, ces données doivent préciser les valeurs de la variable à prédire → **sorties désirées** du réseaux (notée y^*) pour les exemples d'apprentissage;
- ❑ un **critère de performance** à optimiser;
pour les problèmes supervisés (classification ou régression), le critère le plus utilisé est **le critère des moindres carrés (à minimiser)**.

Caractéristiques (7)

Cas d'un réseau avec une seule sortie y (problème de régression ou de classification en 2 classes):

$$C = \sum_i (y(\mathbf{x}_{(i)}) - y^*(\mathbf{x}_{(i)}))^2$$

valeur de sortie \nearrow \uparrow vecteur d'entrée \nwarrow valeur de **sortie désirée**
(ième cas d'apprentissage) (valeur de la var. à prédire pour les cas d'apprentissage)

Cas d'un réseau avec plusieurs sorties y_1, y_2, \dots, y_m (classification en m classes) \rightarrow un vecteur de sorties $\mathbf{y}(\mathbf{x}_{(i)})$ et de sorties désirées $\mathbf{y}^*(\mathbf{x}_{(i)})$:

$$C = \sum_i \|\mathbf{y}(\mathbf{x}_{(i)}) - \mathbf{y}^*(\mathbf{x}_{(i)})\|^2 = \sum_i \sum_j (y_j(\mathbf{x}_{(i)}) - y_j^*(\mathbf{x}_{(i)}))^2$$

Caractéristiques (8)

- Différentes **méthodes d'optimisation** sont disponibles, en général: algorithmes itératifs basés sur des méthodes de **descente de gradient**. → méthode d'apprentissage dite "globale": adaptation de tous les poids du réseau sur base de l'optimisation d'un critère comparant la sortie estimée du réseau et la sortie désirée.

Principe de la descente de gradient:

Ex: minimisation d'une fonction d'erreur E dans l'espace des poids (similaire pour un critère à maximiser)

→ Recherche d'un minimum, il faut: $\frac{\partial E}{\partial w_i} = 0 \quad \forall i$

→ Algorithme généralement itératif: pour chaque vecteur d'entrée $\mathbf{x}_{(k)}$, calcul de sa contribution à l'erreur E_k

Caractéristiques (9)

(= $[y(\mathbf{x}_{(k)}) - y^*(\mathbf{x}_{(k)})]^2$, dans le cas des moindres carrés),

→ modification des poids:

$$w'_i = w_i + \Delta w_i \quad \text{avec} \quad \Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

avec η = “pas” de gradient (suffisamment petit, sinon oscillations)

→ aller à “petits pas” vers la région où E est minimum.

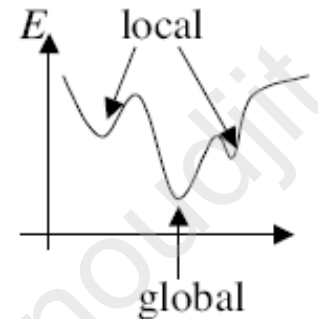
Caractéristiques (10)

Remarque: La méthode de descente de gradient est une méthode “approchée” (heuristique) qui assure de trouver le minimum (global) d’une fonction en *l’absence de minima locaux*, mais peut-être “piégée” par ces derniers

→ différents développements pour éviter ou sortir de ces pièges

→ choix de la fonction d’erreur est important !!

- D’autres méthodes d’apprentissage sont dite “**locales**”: seuls certains poids de connexion sont adaptés en fonction, par ex., de l’entrée du réseau et des unités (neurones) les plus activées par cette entrée (encore appelé “*apprentissage compétitif*”).



Caractéristiques (11)

- ❑ Les méthodes globales sont plutôt utilisées pour des problèmes supervisés et les méthodes locales plutôt pour des problèmes non-supervisés (ex: clustering).
- ❑ Dans tous les cas, l'algorithme vise à converger vers une solution, c'est-à-dire la détermination des valeurs de poids qui assurent de bonnes propriétés aux sorties, en accord avec le problème posé (ex: proches de valeurs désirées en mode supervisé).

Remarques générales

- ❑ Il est (presque toujours) nécessaire d'initialiser les poids de connexion. Différentes stratégies possibles suivant le type de réseau: petites valeurs tirées au hasard (perceptron et MLP) ou méthodes statistiques (réseaux RBF ou à prototypes).
- ❑ Les algorithmes d'apprentissage sont (presque toujours) itératifs: adaptation des poids de connexion après la présentation de chaque vecteur d'entrée. Il est nécessaire de repasser plusieurs fois l'ensemble des données d'entrée avant que les poids ne convergent vers des valeurs stables.
- ❑ Une façon de mesurer la convergence/stabilisation du réseau est d'observer une stabilisation du (d'un) critère d'erreur C entre 2 passages successifs de l'ensemble des données.

→ règle d'arrêt: $\Delta C < \varepsilon$

Merci pour votre attention