




Introduction aux réseaux de neurones

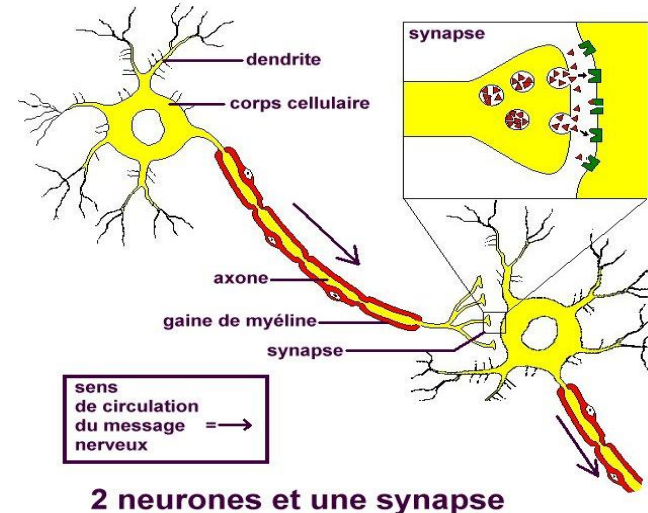
2017-2018



Introduction aux réseaux de neurones

<http://www.erappa.univ-lille3.fr/~gilleron/PolyApp/node15.html>

- Van Leeuwenhoek (1718) : première description fidèle des axones.
- Dutrochet (1824) : observation du corps cellulaire des neurones.
- Valentin (1836) : découverte des dendrites.
- Deiters (1865) : image actuelle de la cellule nerveuse.
- Sherrington (1897) : synapses.
- 1900--1950 : neurotransmetteurs.



Propriétés des neurones humains

Neurones humains

- Un cerveau contient 100 milliards de neurones.
- Seulement quelques dizaines de catégories distinctes.
- Aucune catégorie de neurones n'est propre à l'homme.
- Propagation de l'influx nerveux de l'ordre de 100m/s (très inférieur à la vitesse dans un circuit électronique = $\frac{2}{3}$ vitesse lumière).
- Plusieurs dizaines de milliers de contacts synaptiques au maximum par neurone.
- Nombre total de connexions d'environ 10^{15} .
- La connectique du cerveau est impossible à coder dans une structure biologique (ADN) pour des raisons de combinatoire.
- Il nécessite donc une phase d'apprentissage (contacts avec l'environnement).

Historique des neurones formels

Neurones formels

- 1943 : première définition par McCulloch et Pitts.
- 1949 : association de 2 neurones en entrée d'un troisième par Hebb (associationnisme).
- 1958 : perceptron de Rosenblatt, ajout d'un processus d'apprentissage.
- 1969 : critique dans [*Perceptrons: an introduction to computational geometry*](#) (Minsky(1927-2016) et Papert(1928-2016)), sonne le glas des RN jusqu'en 1980.
- 1970--1980 : RAS.

Historique des neurones formels

- 1980 : réseaux multi-couches et rétropropagation du gradient par Rumelhart et McClelland, Parker, Hinton, Le Cun.
- 1982 : réseaux complètement connectés par Hopfield (attracteurs, mémoires associatives).
- 1984: cartes de Kohonen avec algorithme non supervisé basé sur l'auto organisation.
- 1985 : machines de Boltzmann, utilisant Monte--Carlo.
- 1985--2015: RAS.
- 2015 : deep learning (Facebook, Google) avec des réseaux de convolutions à très grand nombre de couches, distribués sur des clusters de calcul.
- 2016 : annonce de traducteurs automatiques comparables à la traduction humaine [GNMT](#).

AlphaGo Deepmind

- 2016 : AlphaGo bat 4-1 le champion de Go Lee Sedol classé au niveau maximal (9e dan professionnel).
- 2017 : AlphaGo bat 3-0 le champion de Go Ke Jie.
- 2017 : AlphaGo Zero bat la version précédente 100-0 sans accès à des parties jouées par des humains.

[AlphaGo Zero: learning from scratch](#)

Perceptron

Perceptron (Rosenblatt 1958) :

Un perceptron linéaire à seuil prend en entrée p valeurs x_1, \dots, x_p et calcule une sortie o .

Il est défini via p constantes : les coefficients synaptiques w_1, \dots, w_p et un seuil (ou biais) q .

Sa valeur est calculée par la formule :

$$o = 1 \text{ si } \sum_1^p w_i x_i > q$$

$$o = 0 \text{ sinon}$$

Perceptron

Perceptron (Rosenblatt 1958) :

On préférera :

$$o = 1 \text{ si } \sum_0^p w_i x_i > 0 \text{ avec } w_0 = -q \text{ et } x_0 = 1$$

Il s'agit donc d'une fonction linéaire affine, qui nous permet de ne pas avoir à traiter le biais (q) comme un cas particulier.

Perceptron pour la fonction OR

Soit le réseau suivant :

$$w_0 = -0.5, w_1 = 1, w_2 = 1$$

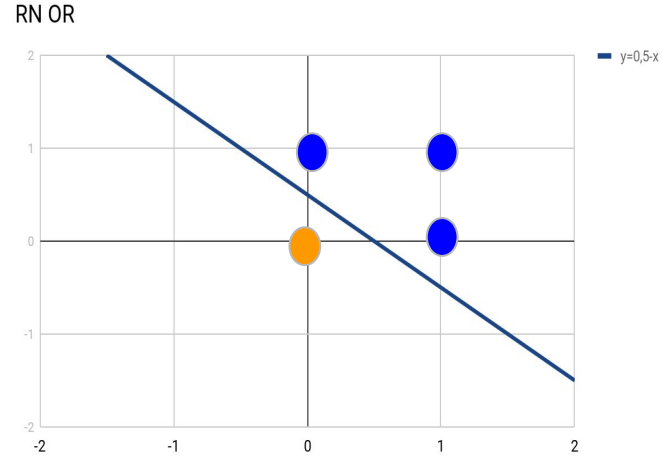
Exercice :

Dessiner le réseau ci--dessus.

Vérifier qu'il correspond bien au OR (avec $F=0$ et $V=1$)

Interprétation géométrique

$$y = -x + 0.5$$



Théorème : L'ensemble des données est linéairement séparable dans \mathbb{R}^n ssi cet ensemble est discriminable via un perceptron.

Corollaire : Le XOR ne peut pas être calculé par un perceptron.

Apprentissage des réseaux de neurones

Algorithme par correction d'erreur

```
initialisation des poids (par exemple aléatoire)
repete jusqu'à critère d'arrêt (convergence ou nb iter)
  sélectionner un exemple (x,c)
  calculer o pour x
  pour i de 0 à p
     $w_i := w_i + (c - o) x_i$ 
  fpour
frepete
```

Propriétés des réseaux de neurones à une couche

- **Correction** : convergence garantie en temps fini
- **Inapplicable** : si l'ensemble d'apprentissage est non linéairement séparable (pas de détection de la non convergence).
- **Robustesse** : non garantie même en cas de convergence.
 - l'hyperplan de séparation peut être très proche d'un des exemples.
 - un nouvel exemple ou une variation minime d'un exemple existant peut remettre en cause l'apprentissage.

Descente du gradient

Algorithme descente du gradient :

```
initialisation des poids
repete
  pour i de 0 a p
     $D_{wi} := 0$ 
  fpour
  pour tout exemple (x,c)
    calculer o pour x
    pour i de 0 a p
       $D_{wi} := D_{wi} + e(c-o)x_i$ 
    fpour
  fpour
  pour i de 0 a p
     $w_i := w_i + D_{wi}$ 
  fpour
frepeter
```

Intérêt : permet de résoudre le problème de la robustesse.

Algorithme Adaline (gradient stochastique)

Algorithme Adaline (ajustement adaptatif des poids) :

```
initialisation des poids
repete
  selectionner un exemple (x,c)
  calculer o pour x
  pour i de 1 a p
     $w_i := w_i + e(c-o)x_i$ 
  fpour
fpour
frepete
```

Problème : on ne règle pas le problème des ensembles non linéairement séparables.

Solution : rajouter des couches de neurones.

Réseaux de neurones multi-couches

Exemple : XOR (1 couche cachée de 4 neurones => 3+4+1=8 neurones)

couche d'entrée (x_0, x_1, x_2), couche cachée (x_3, x_4, x_5, x_6), couche sortie (x_7)

$$w_{13} = 1 \quad w_{14} = -1.5 \quad w_{24} = 1 \quad w_{25} = 1 \quad w_{04} = 1 \quad w_{06} = 1 \quad w_{37} = -0.5 \quad w_{47} = -2 \quad w_{57} = 1 \quad w_{67} = 1.5$$

Exercice : Dessiner le réseau ci-dessus avec (biais $x_0 = 1$).

Vérifier qu'il correspond bien au XOR (toujours avec $F=0$ et $V=1$).

Propriétés des réseaux multicouches

- Cellules réparties en q couches (entrée couche $k+1$ = sortie couche k , sauf pour $k=0$ et $k=q$).
- Toute fonction booléenne peut être calculée par un réseau de neurones à une couche cachée.
- Nouveau problème : au 2^p maximum neurones dans la couche cachée.
- Deux nouvelles questions se posent :
 - Comment choisir l'architecture (nombre et tailles des couches cachées) : pas de solution garantie en 2017.
 - Quel algorithme d'apprentissage choisir : rétropropagation du gradient depuis 1980 (backpropagation)

Rétropropagation du gradient

Algorithme de rétropropagation du gradient :

```
initialisation des poids
repete
  selectionner un exemple (x,c)
  calculer o pour x
  pour toute cellule i de sortie
    di := oi(1-oi)(ci-oi)
  fpour
  pour chaque couche de q-1 a 1
    pour toute cellule i de la couche courante
      di := oi(1-oi)*somme(dk*wik) avec k=succ(i)
    fpour
  fpour
  pour tout poid wij
    wij := wij + eps*di*xij
  fpour
frepeter
```

Problème : trouver les valeurs de eps (learning rate)

Rétropropagation du gradient avec momentum

Algorithme de rétropropagation du gradient avec momentum :

- pondération de la mise à jour des poids en fonction du nombre d'itérations déjà effectuées.
- limitation des oscillations liées aux optimums locaux.
- amélioration de la vitesse de convergence.

Problème : trouver les valeurs de ϵ et α

```
initialisation des poids
repete
  selectionner un exemple (x,c)
  calculer o pour x
  pour toute cellule i de sortie
    di := oi(1-oi)(ci-oi)
  fpour
  pour chaque couche de q-1 a 1
    pour toute cellule i de la couche courante
      di := oi(1-oi)*somme(dk*wik) avec k=succ(i)
    fpour
  fpour
  pour tout poids wij
    wij := wij + Dwij(t) avec Dwij(t)=eps*di*xij+alpha*Dwij(t-1)
  fpour
frepeter
```

Bibliographie

Yann LE CUN

[profil quora](#)

[cours au Collège de France](#)

[MOOC de Geoff HINTON](#)

Jeremy KUN

[The perceptron and all the things it can't perceive](#)

[Neural networks and the backpropagation algorithm](#)

Bibliographie

[Neural networks](#) par 3blue1brown

[Neural networks demystified](#) par Welch Labs

[How to create a mind](#) par Ray KURZWEIL
[site du livre](#)

TED

Ken JENNINGS: Watson, Jeopardy and me, the obsolete know-it-all

Sebastian SEUNG: I am my connectome

Fei-Fei LI: How we're teaching a computer to understand pictures

Jeremy HOWARD: The wonderful and terrifying implications of computers that can learn

Garry KASPAROV: Don't fear intelligent machines. Work with them

Stuart RUSSELL: 3 principles for creating safer AI

Sam HARRIS: Can we build AI without losing control over it?

Ray KURZWEIL: Get ready for hybrid thinking

Hugo LA ROCHELLE

Turing

Réseaux de neurones en 1948

Article

Bibliographie

Science Étonnante

Computerphile

Logiciels

playground.tensorflow.org

[TensorFlow et l'apprentissage profond sans les équations différentielles](#)

Torch

New theory cracks open the blackbox of deep neural networks

<https://www.wired.com/story/new-theory-deep-learning/>