

Les réseaux de neurones

Le Bras – Février 2018

Problématique

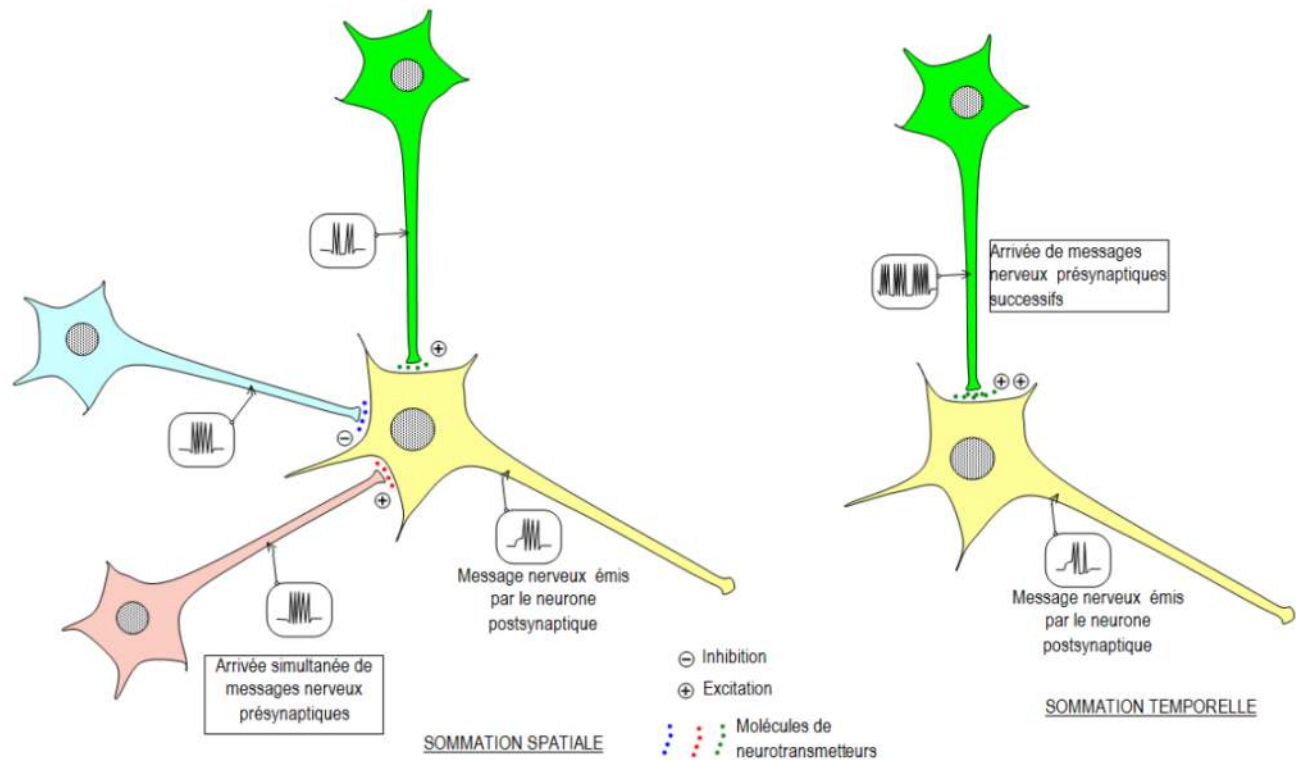
- Comment faire pour qu'une machine sache prendre des décisions, par exemple déterminer si une image est prise à l'intérieur ou à l'extérieur ?
- Il suffit de compter les pixels bleu clairs ou blanc/gris pour savoir que c'est une image de l'extérieur ... Trop facile ? au final la chambre du petit avec ses murs bleus ciel serait classée dans les images d'extérieur, ce n'est donc pas si simple.
- Imaginons que l'on ait un grand nombre de photos classées et que la machine ait mémorisé ce classement, alors quand une nouvelle photo sera présentée, elle utilisera cette mémoire pour l'affecter à la classe la plus proche. L'apprentissage (machine learning) précède l'usage et permet la prise de décision automatisée, c'est de l'intelligence artificielle.
- C'est bien proche du fonctionnement de notre cerveau, d'où l'idée d'appeler cette technologie "réseaux de neurones"... Petite et modeste visite guidée introductive à ce vaste domaine.

Guide de la visite

- Comment marchent nos neurones ?
- La première idée de réseau de neurones : le Perceptron (avec ce nom, on est sûrs que cela date des années 40...)
- Les limites du Perceptron simple (dommage, il est vraiment simple à coder ce réseau de neurones)
- Le Perceptron multicouches (ça se corse, mais que les allergiques aux maths se rassurent, les principes s'exposent et peuvent se comprendre sans formules ...)
- Les réseaux de neurones convolutifs (au point où on en est, ce n'est pas forcément beaucoup plus compliqué)
- N'y avait-il pas plus simple ?
- Conclusion

Comment marchent nos neurones ?

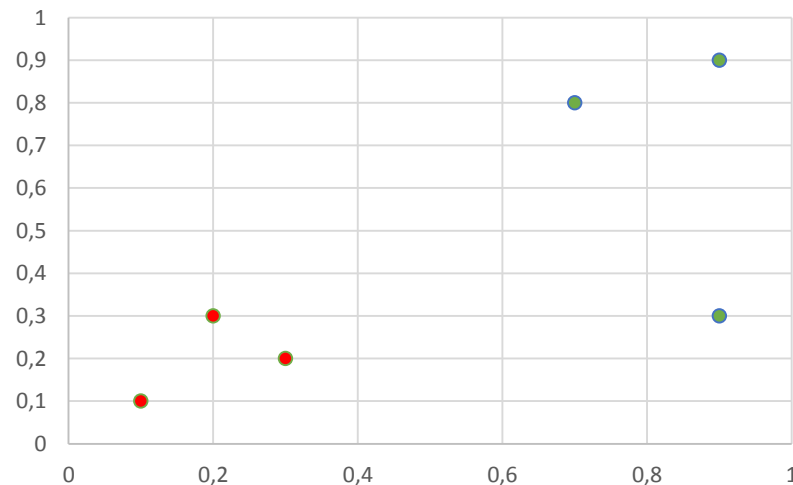
- Dans notre cerveau de 86 à 100 milliards de neurones ... interconnectés
- A chaque connexion, le message nerveux est transmis a un autre neurone amplifié ou atténué.
- Le neurone récepteur transmet aux neurones suivants si l'ensemble des signaux reçus est supérieur à un seuil.
- C'est cette mécanique de seuil innée ou acquise qui constitue notre mémoire, notre capacité de décision.



Le Perceptron 1/3

- M. Rosenblatt, mathématicien, en s'inspirant du fonctionnement des neurones, a cherché un algorithme permettant de déterminer si un élément appartenait à une catégorie ou à une autre en faisant une simple somme pondérée des différentes caractéristiques de cet élément (et il a trouvé !)
- Donc, par exemple, prenons un ensemble de points répartis en 2 catégories (les verts et les rouges) et le but est de calculer la droite permettant de séparer ces deux classes au mieux afin de savoir si un nouveau point est à mettre en vert ou en rouge en fonction de sa position.

x	y
0,1	0,1
0,3	0,2
0,2	0,3
0,9	0,9
0,7	0,8
0,9	0,3



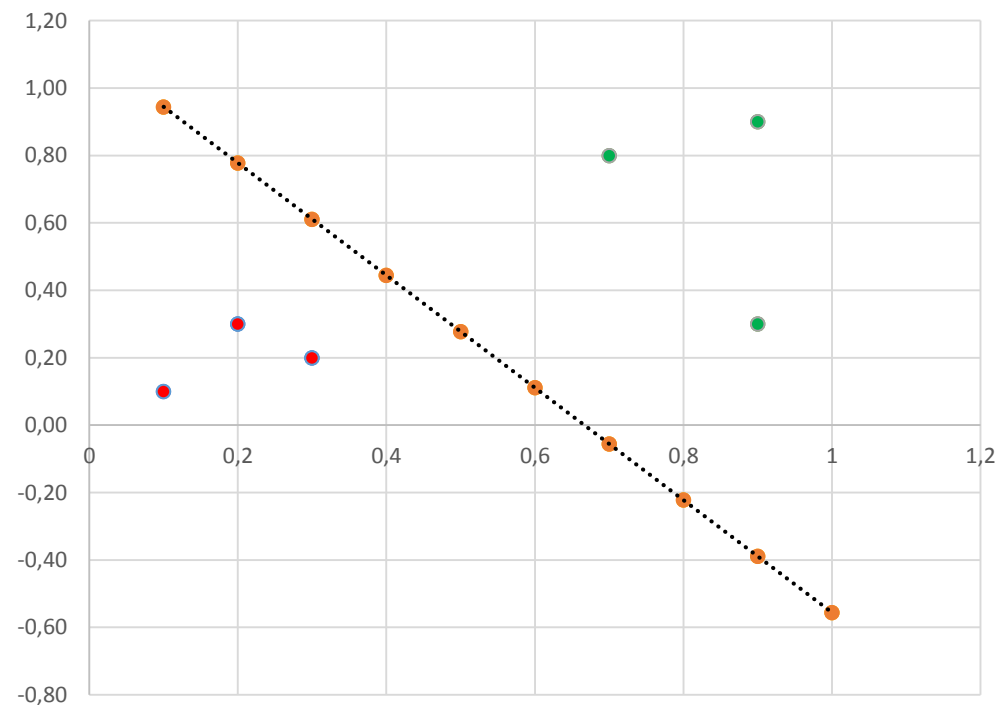
Le Perceptron 2/3, l'apprentissage

- Donc quels sont les coefficients a , b , d , tels que $s = a.x_i + b.y_i + d$ est > 0 pour les verts et < 0 pour les rouges.
- L'idée est de commencer en prenant des valeurs aléatoires pour a , b et d , de mesurer l'erreur de classement que ces coefficients génèrent, de les modifier de façon à diminuer cette erreur, et d'itérer jusqu'à trouver des valeurs qui conviennent.
- Considérons que l'on doit avoir un résultat (s) proche de 1 pour les verts et de -1 pour les rouges, l'erreur c'est $(s - c)$ (ou c est la catégorie : -1 ou 1), alors :
 - $a = a + n.(c - s). x_i$
 - $b = b + (c - s). y_i$
 - $d = d + n.(c - s)$et on continue jusqu'à ce que a , b , d ne bougent plus (c'est lorsque $s = c$),
- n est un coefficient de convergence compris entre 0 et 1

Le Perceptron 3/3, le résultat

- En prenant $a = b = d = 0$ en point de départ, le Perceptron converge sur les données de l'exemple précédent en 4 itérations avec un coefficient de convergence de 0,6 : $a = -1,8$, $b = -1,08$, $c = 1,2$
- Les résultats ne sont pas indépendants des valeurs de départ
- Par exemple avec $a = b = d = 0,1$, en 4 itérations, on trouve la solution :
 - $a = -1,46$
 - $b = -1,58$
 - $d = 1,3$

Perceptron

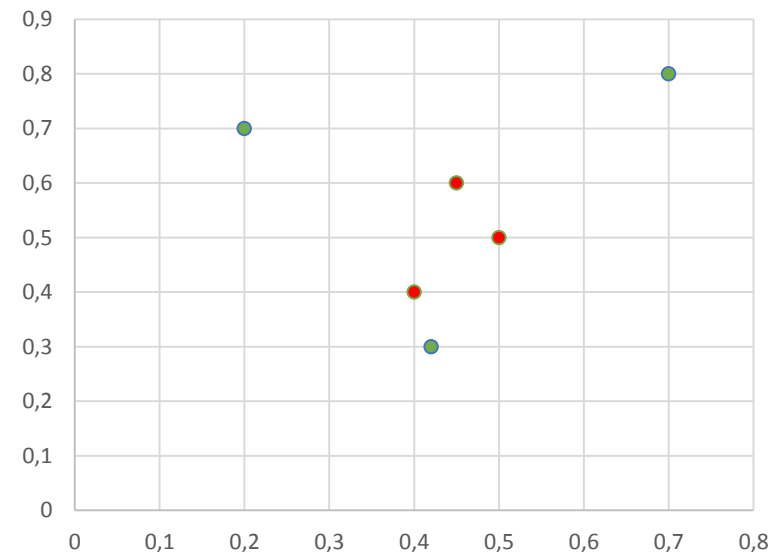


Grandeurs et limites du Perceptron monocouche

- L'exemple considère des données d'entrées à 2 caractéristiques (l'abscisse et l'ordonnée), mais c'est généralisable à n caractéristiques (c'est juste moins facile à dessiner). L'algorithme calcule les n+1 coefficients.
- C'est tout simple, cela prend 10 lignes de code et ça marche (c'est-à-dire cela converge) ... uniquement et uniquement si les catégories sont séparables par un hyperplan (qui n'est qu'une droite dans notre exemple) !

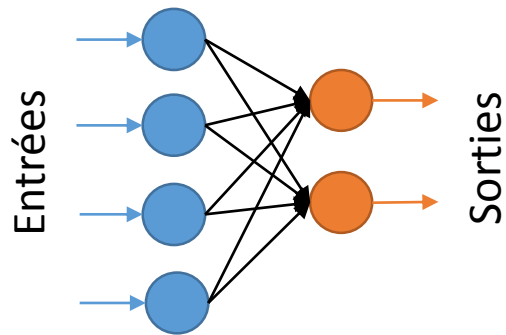
- Avec cet autre exemple, l'algorithme ne converge pas

x	y
0,4	0,4
0,45	0,6
0,5	0,5
0,2	0,7
0,7	0,8
0,42	0,3

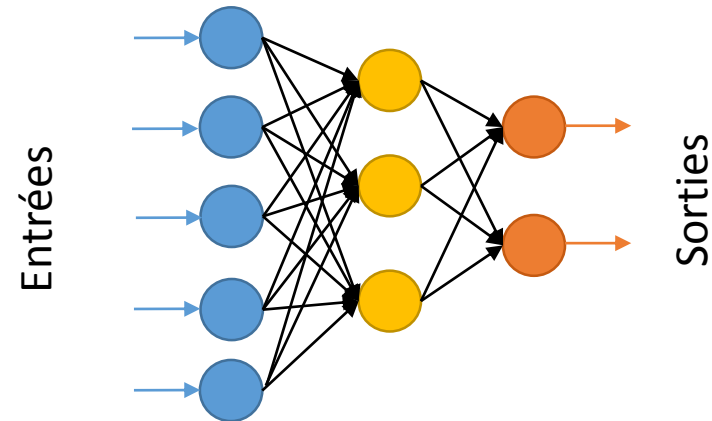


Le Perceptron multicouches 1/3

- L'idée est de se rapprocher un peu plus du fonctionnement du cerveau en insérant des couches intermédiaires entre l'entrée et la sortie du réseau.



Réseau monocouche



Réseau multicouches (une couche "cachée")

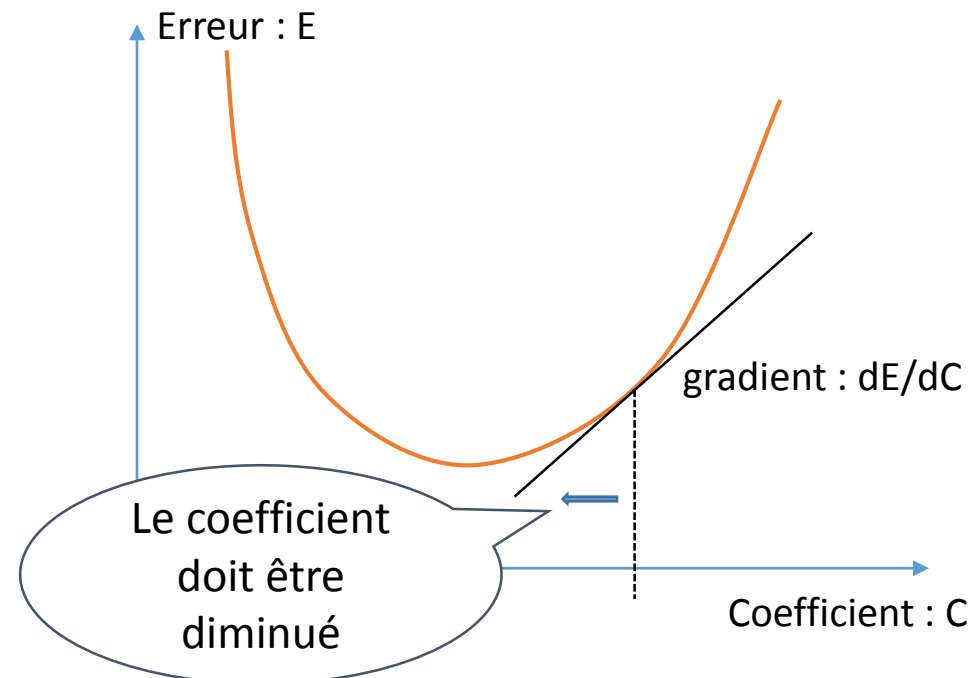
- Autant de neurones d'entrée que de caractéristiques observées, autant de neurones de sortie que de catégories et un nombre de neurones de la couche intermédiaire qui peut varier en fonction de la précision vs la souplesse dont on a besoin .

Le Perceptron multicouches 2/3

- Comment calculer la contribution de chaque neurone dans l'erreur globale du réseau de façon à pouvoir la minimiser.
- L'algorithme est le suivant :
 1. Déterminer aléatoirement les coefficients pour le premier passage (entre -0,5 et +0,5)
 2. Pour chaque élément de l'ensemble d'apprentissage :
 1. Calculer la sortie
 2. Calculer l'erreur par rapport à la sortie attendue
 3. Rétro-propagation de l'erreur jusqu'aux entrées
 4. On boucle en 2 jusqu'à ce que l'erreur globale soit inférieure à un seuil
- Dans le cas d'un réseau de neurones à une couche les poids sont modifiés à partir d'une fonction d'activation en forme d'échelon (valant -1 ou 1 selon la catégorie), ce qui permettait de modifier les poids dans le "bon sens". Cette méthode ne permet pas de calculer les poids de la couche intermédiaire d'un réseau multicouches...

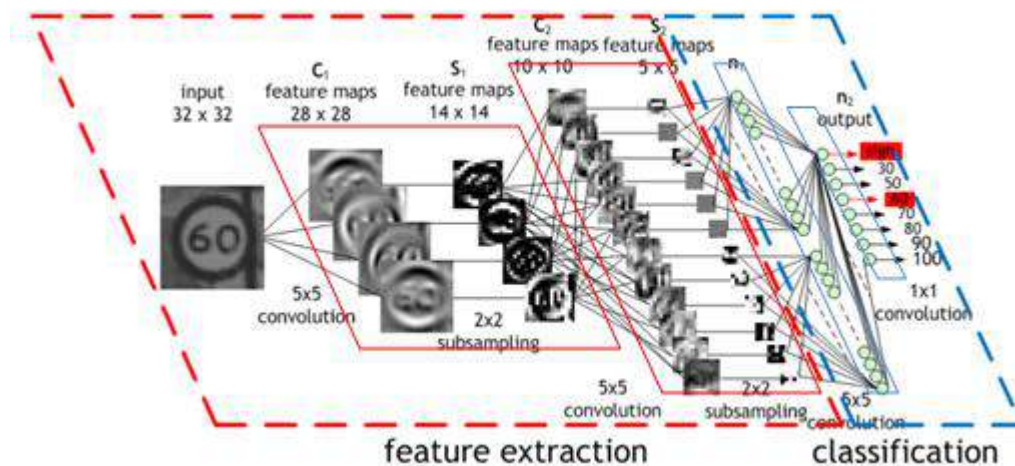
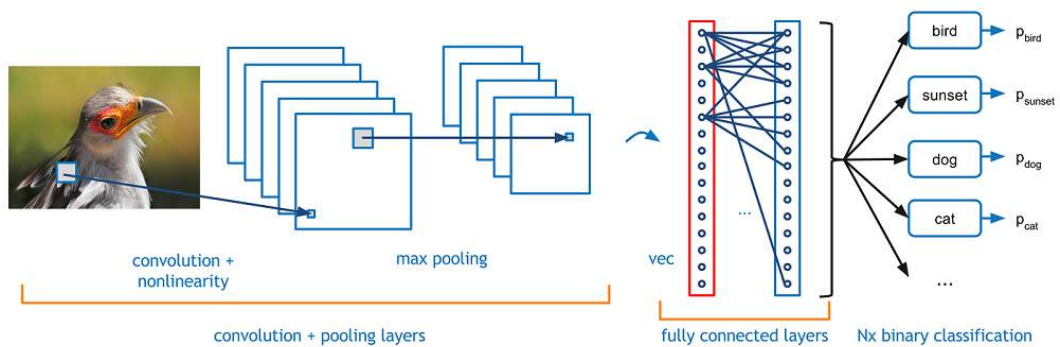
Le Perceptron multicouches 3/3

- Pour calculer la variation à appliquer sur chaque coefficient il faut calculer l'évolution de l'erreur en fonction de ce coefficient (le gradient). En "descendant" le long du gradient on s'approche de l'optimum qui est le minimum de la fonction d'erreur.
- Le gradient c'est la dérivée partielle de l'erreur totale par rapport au coefficient considéré. Il faut donc que la fonction d'activation qui détermine la sortie du neurone, soit dérivable (donc ce n'est pas une fonction échelon, mais par exemple une fonction tangente hyperbolique).
- L'algorithme s'appelle rétro-propagation du gradient, parce que les coefficients des couches intermédiaires et de la première sont calculés en intégrant les valeurs de la couche suivante. La couche de sortie est calculée en premier avec une formule qui diffère.



Les Réseaux de neurones convolutifs 1/2

- En analyse d'image et a fortiori de vidéo, le nombre de connexions d'un Perceptron multicouches est exponentiel jusqu'à devenir rédhibitoire.
- Les réseaux de neurones convolutifs (en anglais Convolutional Neural Networks) font précéder une couche de type Perceptron par une couche de filtres appliqués sur des sous-ensembles de l'image. Cette couche de convolutions (dont on rappelle qu'elle s'opère en effectuant une somme pondérée sur le voisinage de chaque point, c'est donc aussi une sorte de réseau de neurones) permet l'extraction de caractéristiques synthétiques (comme par exemple les frontières des objets) et donc la réduction du nombre de points d'entrée nécessaires pour le Perceptron.
- Exemples d'applications :



Les Réseaux de neurones convolutifs 2/2

- Autre avantage, une caractéristique sera détectée indépendamment de sa localisation dans l'image (les filtres sont appliqués de la même manière sur toute l'image, alors qu'un Perceptron aurait des coefficients différents).
- Les réseaux neuronaux convolutifs ont de nombreuses applications dans la reconnaissance d'images, de vidéos ou le traitement du langage naturel.
- Le réglage et l'apprentissage d'un réseau CNN est long et délicat, il existe des CNN pré-entraînés avec des milliers d'images. Leur adaptation à des classifications spécifiques est possible avec un minimum d'ajustement des coefficients lors d'une phase d'apprentissage ad hoc. Il faut cependant prendre garde à ne modifier les coefficients du Perceptron qu'à la marge pour qu'il ne "perde pas la mémoire".

Les autres méthodes 1/2

- Lorsqu'une seule donnée sert à prendre une décision, un simple seuil suffit (le radar flashe au dessus d'une certaine vitesse), mais lorsque le nombre de données augmente, il faut faire appel à des algorithmes plus complexes pour affecter une nouvelle donnée à une catégorie. A chaque catégorie correspond une décision (quel médicament prescrire en fonction des symptômes, quelle correction apporter à l'image ...). La prise de décision par les algorithmes, c'est l'intelligence artificielle.
- Il y a deux approches dans les techniques de classification :
 - Les méthodes supervisées, où les catégories sont connues et le sujet est de savoir affecter une nouvelle entrée à une catégorie (c'est typiquement l'objet des réseaux de neurones)
 - Les méthodes non supervisées où les données sont analysées de façon à identifier des catégories ou des caractéristiques

Les autres méthodes 2/2

- Une méthode supervisée qui donne des résultats, même s'ils ne sont pas forcément très pertinents, c'est la méthode Bayésienne : on calcule la fonction de répartition de chaque classe dans l'ensemble d'apprentissage puis on affecte la nouvelle entrée à la classe pour laquelle elle est la plus probable.
- Les méthodes non supervisées sont les techniques d'agglomération (de clustering) qui peuvent être ascendantes (au début, chaque point est une classe puis on réduit jusqu'à ce qu'un seuil de cohésion soit atteint) ou alors descendantes (on part d'un nombre de classe a priori puis on agrège en faisant bouger le centre de chaque classe). Il existe un très grand nombre d'algorithmes, plus ou moins coûteux en temps machine, qui pourront faire l'objet d'une présentation spécifique.
- **Attention, les méthodes de classification, ne donnent pas l'information de la pertinence qu'il y a à considérer que deux groupes sont distincts ! Des méthodes spécifiques, radicalement différentes sont à employer pour savoir si à l'issue d'une expérience les différences (de moyennes par exemple) sont significatives ou non.**

Conclusion

- Les réseaux de neurones informatiques, aussi puissants soient-ils sont loin d'avoir la dimension d'un cerveau humain. Il n'en reste pas moins qu'ils sont capables dans un contexte bien circonscrit de prendre de bonnes décisions bien plus rapidement et efficacement qu'un humain.
- Comme l'humain, ils partent d'un acquis (l'apprentissage initial) et peuvent s'améliorer au fur et à mesure.
- Comme tous les algorithmes, s'ils sont utilisés pour réagir au comportement humain, ils peuvent constituer la base d'une atteinte à la vie privée, mais ça c'est une toute autre histoire ...