# Introduction

**La machine et l’être humain :**

La vie quotidienne d’une personne exige une immense quantité de connaissances sur le monde. Une grande partie de ces connaissances sont **subjectives** et **intuitives**, et son donc difficiles à articuler de manière formelle. Le défi de l’intelligence artificielle est de :

* Savoir comment intégrer ces connaissances informelles dans un ordinateur.
* Résoudre les tâches faciles à accomplir pour les gens, mais difficiles à décrire formellement.
* **Approche fondée sur la connaissance (knowledge based approch) :**

Un ordinateur peut raisonner automatiquement au sujet des instructions rédigées dans ces langues formelle en utilisant des règles d’inférence logiques « approche fondée sur la connaissance ». L’un des projets les plus plus célèbres est le projet « Cyc », un moteur d’inférence, est une base de données d’instructions dans un langage appelé Cycl (les déclarations sont saisies par un personnel de superviseurs humains).

Faiblesses :

* Les humains ont du mal à concevoir des règles formelles pour décrire le monde avec précision.
* Les systèmes reposent sur des connaissances codées en dur.
* **Approche apprentissage automatique.**

L’IA doit acquérir ses propres connaissances en extrayant des modèles à partir de données brutes. Algorithmes :

* + Regression logistique (recommander ou non …).
  + Bayésien naïf (séparer des courriels légitimes des spams).

**Dépendance** à l’égard de la **représentation des données** qui leur sont **transmises.**

L’apprentissage des représentations : Mettre en évidence un ensemble caractéristiques pour une tache donnée. Algorithme : Auto-encodeur

La difficulté réside dans le fait que bon nombre de **facteurs de variation influent** sur les **données** (Eg. L’éclairage (jour/nuit), l’angle de vue dans une image).

* Ce qui oblige à désenchevêtrer les facteurs de variation et à rejeter les moins intéressent.
* L’apprentissage profond :

Couche visuelle

Une serie de couches cachées extrait des caractéristiques de plus en plus abstraites de l’entrée. Le modèle détermine lui-même quels concepts sont utiles pour expliquer les relations dans les données observées

Défis remportés :

**Deep Blue d’IBM** a battu le champion du **monde Garry Kasparov** en **1997**. Le jeu d’échec représente un monde très simple (64 emplacements et 32 pièces qui se déplacent de manière strictement délimitée). La difficulté n’était pas de décrire l’ensemble des pièces du jeu et le des coups autorisés)

Les échecs peuvent être décrits par une très courte liste de règles purement formelles.

Nomination de l’apprentissage profond :

1. 1940 – 1960 : cybernétique, fondée sur les **modèles linéaires simples** – f(w, x) = \* + \* = y)

* 1943 : Neurone formel de McCulloch-Pitts, modèle primitif de la fonction cérébrale. Il reconnait 2 labels différents en testant si l’entrée est positive ou négative. Les poids sont définis par des humains.
* 1958 : Perceptron de Rosenblatt, premier modèle linéaire (1 seul neurone) capable d’apprendre les poids à partir d’exemples, à l’aide de l’algorithme Descente de gradient stochastique.
* 1960 : Windrow et Hoff, élément linéaire adaptatif (ADALINE – Modèle linéaire), renvoie la valeur réelle (prévision) de f(w, x). L’algorithme d’optimisation est un cas particulier de l’algorithme Descente de gradient stochastique.
* Les limites des modèles linéaires
  + Apprendre la fonction Xor.
    - Première chute.
    - Désaffection vers l’inspiration biologique (**neuroscience**).

1. 1980 – 1990 : connexionnisme (traitement parallèle/représentation distribué), fondée sur la rétropropagation.

Le concept :

Une entrée est représentée par plusieurs variables, chaque variable doit être impliquée dans la représentation de nombreuses entrées possibles.

1. 2006 : Apprentissage profond, fondée sur les Réseaux de Neurones Artificiels.

2006 : Hinton et al.

2007 : Bengio et al.

Inspiration biologique :

L’apprentissage profond puise ses inspirations dans la neuroscience, l’algèbre linéaire, la probabilité, la théorie de l’information et l’optimisation numérique.

**Neuroscience Computationnelle** : Vise à comprendre comment le cerveau fonctionne, au niveau algorithmique, en construisant des modèles plus précis du fonctionnement réel du cerveau. Ce domaine est distinct de l’apprentissage profond, qui vise à construire des systèmes informatiques profonds capables de résoudre avec succès les tâches exigeant de l’intelligence.

La principale raison de la diminution du rôle des neurosciences dans la recherche sur l’apprentissage profond aujourdhui est le manque d’information sur le cerveau.

# Algèbre linéaire

### Scalaires vecteurs matrices et tenseurs

|  |  |
| --- | --- |
| Scalaire | Un nombre unique |
| Vecteur | V est écrit sous la forme de colonne.  Si chaque élément est dans R, et que le vecteur n éléments, alors le vecteur se trouve dans un ensemble formé en prenant le produit cartésien de R n fois avec lui-même.  On peut considérer les vecteurs comme des points d’identification dans l’espace, chaque élément donnant la coordonnée le long d’un axe différent.  Produit scalaire est le produit entre deux vecteurs : s y = si = 0, alors perpendiculaire (produit vectoriel => colinéaire) |
| Matrice | Est un tableau bidimensionnel.  Un vecteur est considéré comme une matrice à une seule colonne, sa transposition est une matrice à une seule ligne.  Un scalaire est une matrice avec une seule entrée  Propriétés :  Additionner :  Matrice et scalaire : = + b  Matrice et vecteur :  C = + , le vecteur v est ajouté à chaque **ligne** de la matrice.  C = A + v, est une copie implicite de v à plusieurs endroits « Diffusion »  Produit terme à terme :A B  Distributivité : A(B+C) = AB + AC  Associativité : (BC) = (AB)C  Transposition d’un produit matricielle :  **Non commutative** : AB != BA, **n’est pas vérifié contrairement à la multiplication scalaire** |
| Tensor | Est un tableau multidimensionnel |

Système d’équation :

Sachant : multiplication à gauche puis à droite,

A.x = b

Savoir si l’équation à une solution, revient à tester si b est dans le sous-espace vectoriel engendré par les colonnes de A.

Pour que Ax = b ait une solution pour toutes les valeurs de b , il faut que l’espace colonne de A soit , si un point dans est exclu de cette espace, ce point est une valeur potentielle de b qui n’a pas de solution.

La contrainte sur l’espace colonne de A vis-à-vis de implique immédiatement que A doit avoir au moins m colonnes, **n >= m (condition nécessaire),** sinon la dimension de l’espace colonne serait inférieure à m.

Exemple :

A une matrice 3X2 et b en 3D, x est donc en 2D.

Modifier les valeurs de x nous permet au mieux de tracer un plan 2D dans . L’équation a une solution si et seulement si b se trouve dans ce plan.

Condition suffisante et nécessaire :

Les colonnes ne doivent ni etre redondantes ni avoir de dépendance linéaire, sinon elle n’ajoute aucun point à l’espace vectoriel engendré par l’ensemble. Elles doivent être linéairement indépendante pour englober la totalité de .

Matrice singulière : matrice carrée et que les colonnes soient linéairement indépendantes.

Inverse :

Si la matrice n’est pas carrée ou est carrée mais singulière, la résolution de l’équation est possible, mais on ne peut pas utiliser la méthode d’inversion matricielle pour trouver la solution.

La norme :

Pour mesurer la taille d’un vecteur, on utiliser la fonction F norme avec p , qui associe un vecteur à un scalaire non négatif. Intuitivement, la norme d’un vecteur x mesure la distance entre l’origine et le point x.

La fonction norme F, vérifie :

F(x) = 0 🡺 x = 0

F(x + y) <= f(x) + f(y), inégalité triangulaire

F(a.x= |a|.F(x), a

: la norme euclidienne, distance euclidienne de l’origine au point identifié par x, désignée par ||x|| =

En ML, on travaille plus avec la norme L2 au carré qu’avec une norme **L2** en elle-même. Car la **dérivée** des éléments de x de la norme L2 au carrée ne dépend que de l’élément correspondant de x, tandis que toutes les dérivées de la norme L2 dépendent du vecteur entier. La norme L2 au carré peut être **indésirable** parce qu’elle **augmente très faiblement près de zéro**. Dans ce cas, se tourner vers une norme L1 pour faire la **différence entre les éléments nuls et non nuls.** Chaque fois qu’un élément de c s’éloigne de 0 avec un écart , L1 augmente de

La norme L0, mesure la taille d’un vecteur selon le nombre d’élément non nul (terminologie fausse, utiliser la norme L1)

La norme uniforme L.

La norme de Frobenius : norme d’une matrice