Module Master M2

Raisonnement et Décisions
Chapitre IX : Machine et Deep Learning

Présenté par : Prof. Kholladi Mohamed-Khireddine
Département d'Informatique
Facultés des Sciences Exactes
Université Echahid Hamma Lakhdar d'El Oued
Tél. 0770314924
Email. kholladi@univ-eloued.dz et kholladi@yahoo.fr
Site Web. www.univ-eloued.dz
http://kholladi.doomby.com/ et http://kholladi.e-monsite.com/

IX –Machine et Deep learning

IX.1 - Qu'est-ce qu'une donnée ?

Le mot donnée (ou data) vient du mot grec datum qui signifie fait. Le problème consiste, à partir d'un ensemble de faits (ou des événements), de pouvoir prédire un résultat et surtout d'une manière automatique (Machine Learning).

IX.2 - Quelles sont les différents types de données ?

Il existe trois types de données :

1. Les données de type structuré : ce sont toutes les données tabulaires (Fichiers, Bases de données, etc.).
2. Les données de type semi-structuré : ce sont des données de différents types (XML, Json, Log, etc.).
3. Et les données de type non structuré : ce sont des données de différents types (image, photo, son, texte, vidéo, etc.).

IX.3 - Comment situer Maching Learning et Deep Learning par rapport aux autres disciplines?

La figure IX.1 nous donne une idée sur l'intersection et l'imbrication des sciences entre elles et surtout de situer le Machine et le Deep Learning qui sont des sciences des systèmes complexes et surtout des techniques de l'intelligence artificielle.

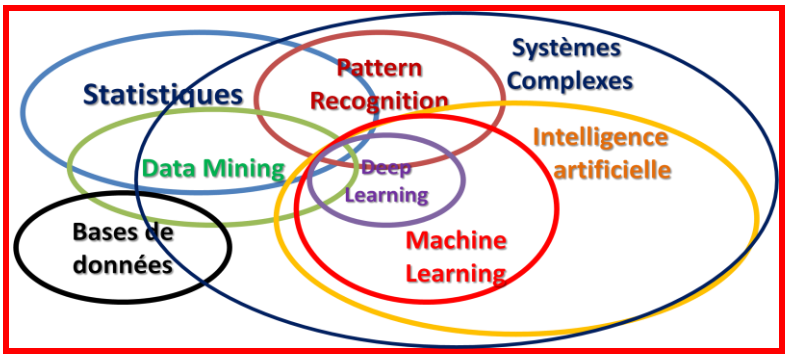

Figure IX.1 – situation du Machine et Deep Learning par rapport d'autres sciences

On voit bien que les deux disciplines (Machine Learning et Deep Learning) sont des disciplines de l'intelligence artificielle (I.A.) qui utilisent des techniques de pattern recognition et du data mining et que les Systèmes Complexes englobe toutes les sciences, où les problèmes de très grande taille n'ont pas de solutions classiques.

IX.4 - Qu'est-ce que Machine Learning?

Dans le cadre des cours sur les Systèmes Complexes, on va aborder deux aspects d'approches de certains problèmes par les techniques de Machine Learning et de Deep Learning. D'abord, Machine Learning veut dire en français apprentissage automatique. En fait, c'est comment les algorithmes apprennent en étudiant à partir des exemples. Dans différents problèmes, du genre : détection de fraude, Reconnaissance d'image ou de caractère, réseaux sociaux, etc. Ainsi, le Machine Learning, c'est de pouvoir donner à une machine la capacité d'apprendre sans vraiment qu'il y ait une programmation d'une manière explicite et claire.

IX.5 - Machine Learning : Domaines d'applications

Le tableau suivant donne une liste non exhaustive des domaines d'application du Machine Learning et du Deep Learning.

N°	Domaines d'applications	N°	Domaines d'applications
1	Recherche dans Google	9	La justice
2	Reconnaissance des photos sur Facebook	10	L'armée
3	Reconnaissance des demandeurs	11	L'agriculture
4	Industrie du transport	12	La vision ordinateur
5	Industrie aérospatiale	13	La reconnaissance de la parole et de l'écriture

6	Industrie biomécanique et la robotique	14	La sécurité
7	La santé	15	Reconnaissance d'image
8	Les finances	16	Etc.

IX.6 - Exemple d'utilisation de Machine Learning

a - Algorithme de flux Facebook ou autres réseaux sociaux

La figure IX.2 illustre le fonctionnement d'un système de prévisions des informations à partir des données en entrée en jouant sur des paramètres d'entrée et de sortie. Il est claire que si on a entrée des informations du style : qui a posté le message ?, quand est-ce qu'il a posté le message ? nature du message, le contenu du message, les différentes réactions des autres internautes au message posté, etc., le système, en jouant sur les paramètres d'entrée et réglant les paramètres de sortie, peut prédire de nouvelles informations du style : avez-vous accepté le message ?, cliquer sur le message, commenter le message, interagir avec l'envoyeur du message, etc.

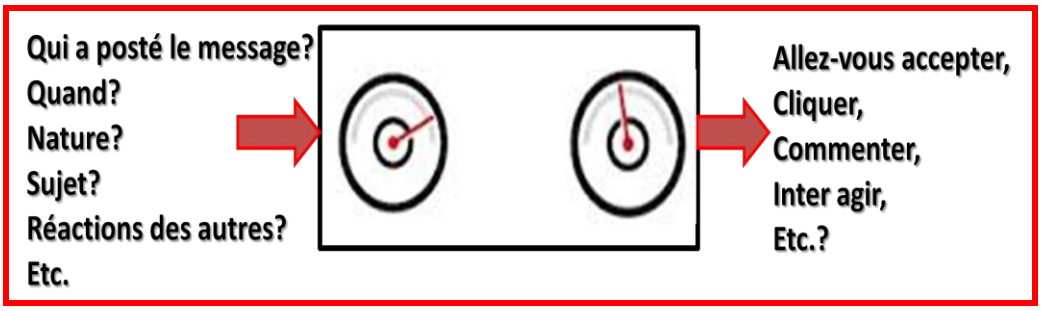

Figure IX.2 – Flux Facebook ou autres réseaux sociaux

b – Algorithme de détection des fraudes sur Internet

La figure IX.3 illustre le fonctionnement d'un système de prévisions des fraudes à partir des données en entrée en jouant sur des paramètres d'entrée et de sortie. Il est claire que si on a entrée des informations du style : nature de la transaction, localisation, site, et tout un tas d'autres données du même genre, le système, en jouant sur les paramètres d'entrée et réglant les paramètres de sortie, peut prédire s'il y a fraude ou non ?

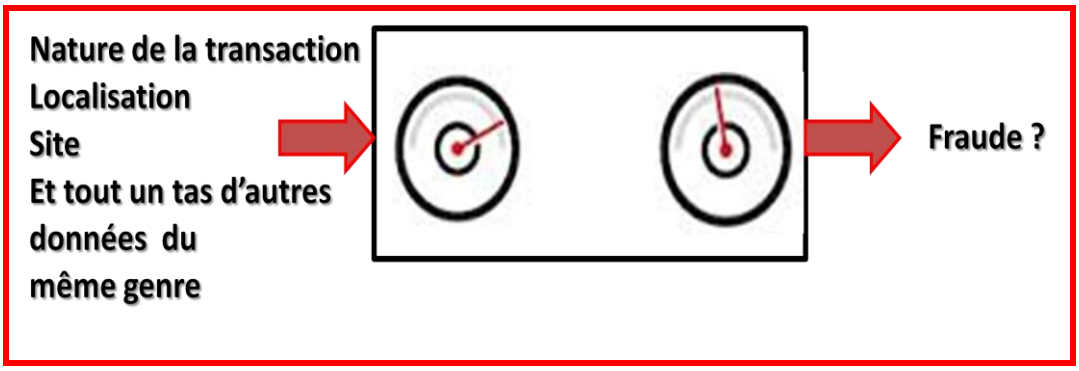

Figure IX.3 – Détection des fraudes

c - Reconnaissance d'image

La figure IX.4 illustre le fonctionnement d'un système de reconnaissance d'image à partir d'une image (à partir des caractéristiques de l'image), le système, en jouant sur les paramètres d'entrée et réglant les paramètres de sortie, peut prédire le contenu de l'image : dans notre cas : lampe, chien, robot, livre, etc.

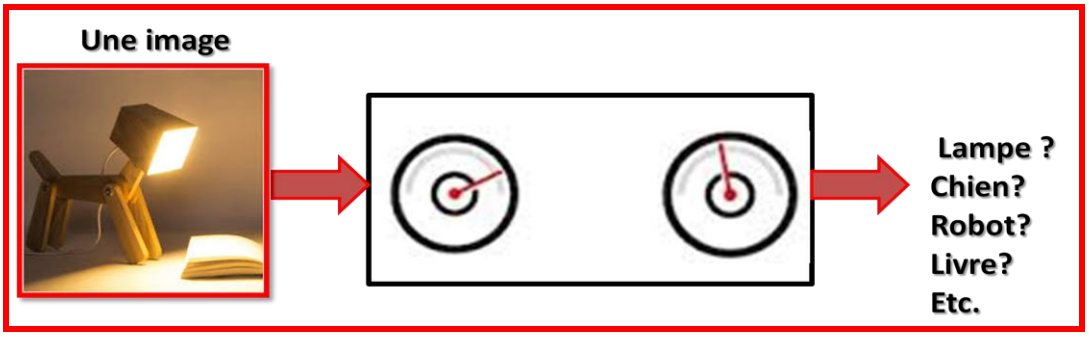

Figure IX.4 – Reconnaître d'image

d - Reconnaissance de caractères manuscrits

La figure IX.5 illustre le fonctionnement d'un système de caractères manuscrits à partir d'une image (à partir des caractéristiques de l'image), le système, en jouant sur les paramètres d'entrée et réglant les paramètres de sortie, peut prédire le contenu de l'image : dans notre cas : caractères, mots, phrases, reconstituer le document, etc. C'est un système intéressant pour la reconstitution des documents manuscrits anciens (mise en valeur et sauvegarde du patrimoine national ancien et surtout dans le domaine de la sécurité des documents financiers tel que la reconnaissance pour l'identification du porteur d'un chèque bancaire, etc.

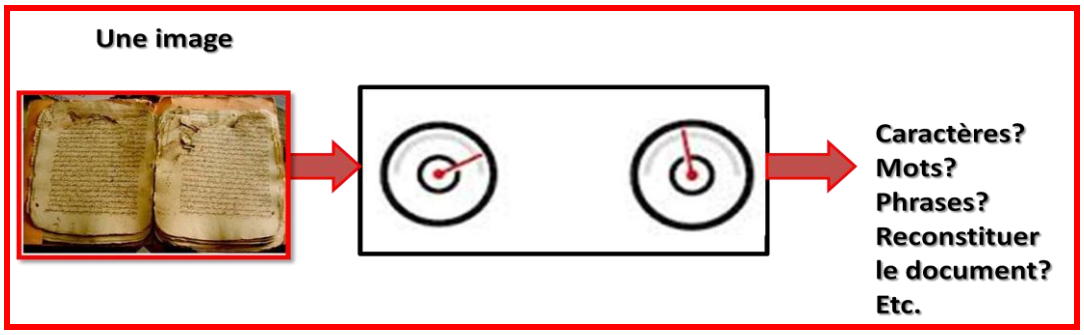

Figure IX.5 – Reconnaissance de caractères manuscrits

e - Reconnaissance de véhicule

Sachant qu'à partir de certaines caractéristiques (taille, dimension, couleur, le nombre de roues, etc.), la machine serait capable de prédire automatiquement le type de véhicule sur l'image comme sur la figure IX.6. L'expert humain du domaine est celui qui s'occupe de l'extraction des caractéristiques et les fait entrer dans la machine.

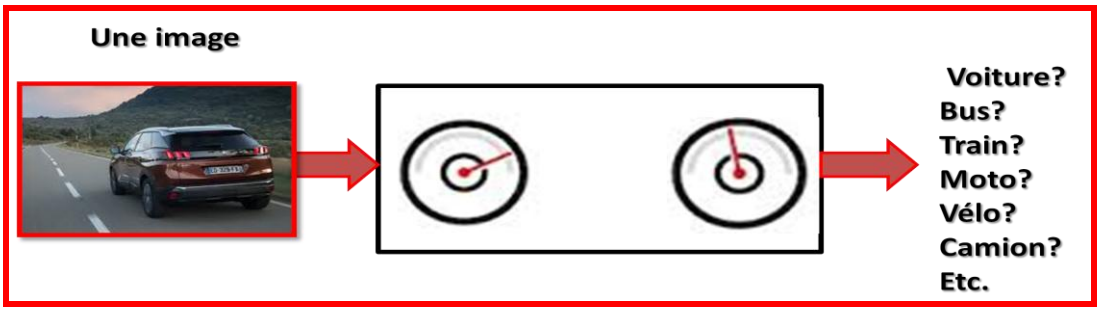

Figure IX.6 – Reconnaissance de v

IX.7 - Réseau de neurones (artificiels)

a - Un neurone artificiel

La figure IX.7 illustre la description d'un neurone artificiel. On a une sorte de cellule qui a en entrée des données X_i, elle doit permettre de prédire la sortie en jouant les pondérations P_i des X_i et les paramètres internes de la cellule (règles de fonctionnement de la cellule).

Soit : P_i = Poids de la variable X_i

On a un exemple de prise de décision du style :

$$\begin{cases} \text{Si } (P1X1 + P2X2 + \ldots + PnXn \geq S) \text{ alors } Y = 1 \\ \text{Si } (P1X1 + P2X2 + \ldots + PnXn < S) \; alors \; Y = 0 \end{cases}$$

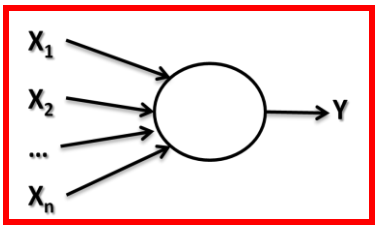

Figure IX.7 – Neurone artificiel

La figure IX.8 donne plus de détail descriptif sur la prise en compte des entrées de la cellule et décrit un peu la mécanique interne de la cellule.

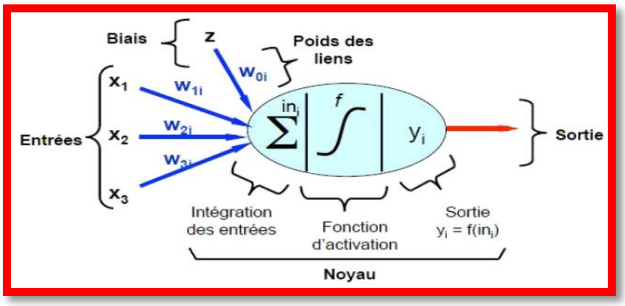

Figure IX.8 – Neurone artificiel détaillé

b - Réseau de neurones (artificiels)

La figure IX.9 illustre les deux phases d'un réseau de neurones artificiels. Lors de la phase d'apprentissage, le réseau de neurones reçoit les données d'entrée (les Xi) et les données de sortie (Y). Il doit trouver la relation qui lie Y aux différents Xi. Dans la phase de prédiction, le réseau de neurones sera capable, à partir de nouvelles entrées de déduire directement les sorties.

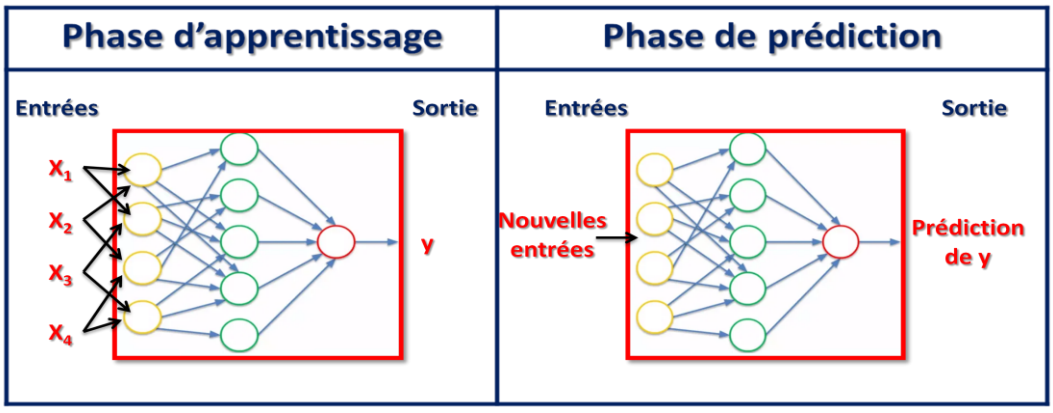

Figure IX.9 – Réseau de neurones artificiels

IX.8 - Comparaison entre Machine et Deep Learning

A travers les figures IX.10 et IX.11, on va essayer de montrer la différence le Machine Learning (apprentissage automatique) et le Deep Learning (apprentissage approfondi). Dans le Machine Learning, Il faut un expert du domaine pour extraire la liste des caractéristiques (nombre de roues, longueur, largeur, hauteur, nombre de porte, couleur, etc.). Par contre, dans le Deep Learning, la liste des caractéristiques est extraite automatiquement par la machine.

a - Machine Learning

La figure IX.10 illustre le fonctionnement de l'apprentissage automatique. En effet, à partir de l'extraction des caractéristiques de l'image d'entrée par un expert, la machine va procéder à la classification pour prédire en sortie l'objet trouvé (dans notre cas voiture).

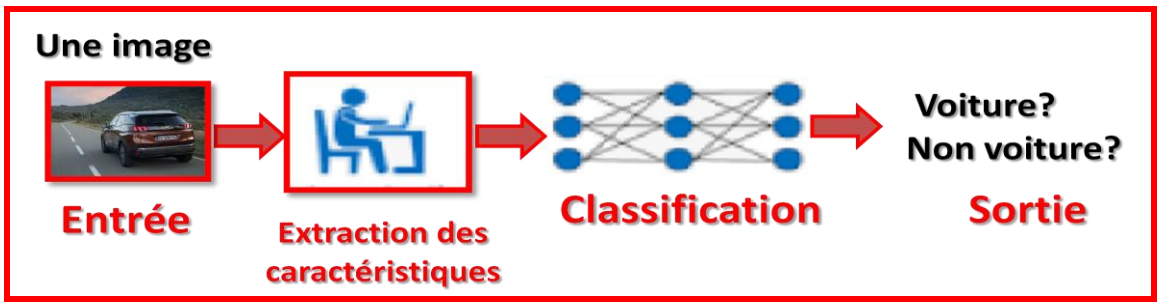

Figure IX.10 – Apprentissage automatique (Machine Learning)

b - Deep Learning

La figure IX.11 illustre le fonctionnement de l'apprentissage approfondi. En effet, à partir de l'image d'entrée, la machine va procéder à l'extraction des caractéristiques et à la classification pour prédire en sortie l'objet trouvé (dans notre cas voiture) d'une manière directe sans intervention de l'expert grâce à la puissance de son réseau de neurones, qui peut contenir plusieurs couches. Dans le Deep Learning, il n'y a pas d'expert humain. Et c'est la machine qui procède directement à l'identification des objets dans une image et pourquoi pas dans une vidéo. Dans ce cas, le réseau de neurones a plusieurs couches internes.

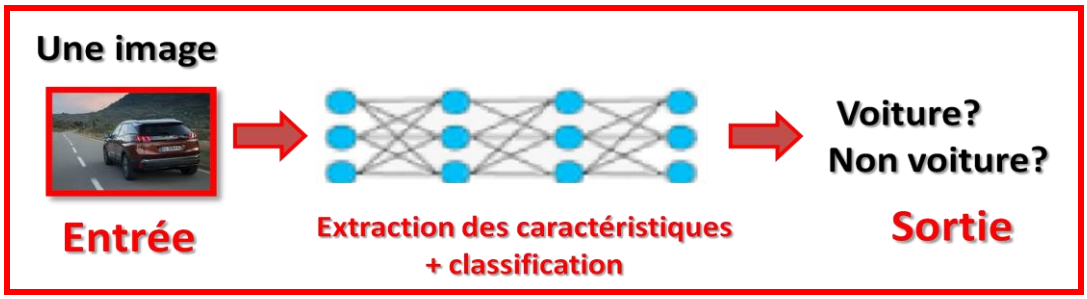

Figure IX.11 – Apprentissage approfondi (Deep Learning)

IX.9 – Exemple de reconnaissance de véhicule

Pour effectuer l'apprentissage automatique de reconnaissance, la machine a besoin de moyens de comparaisons (liste des caractéristiques) fournit par l'expert humain à partir d'une image brute. C'est l'abstraction du problème. A partir de là, la machine effectue la reconnaissance de véhicule grâce à son réseau de neurones comme sur la figure IX.12.

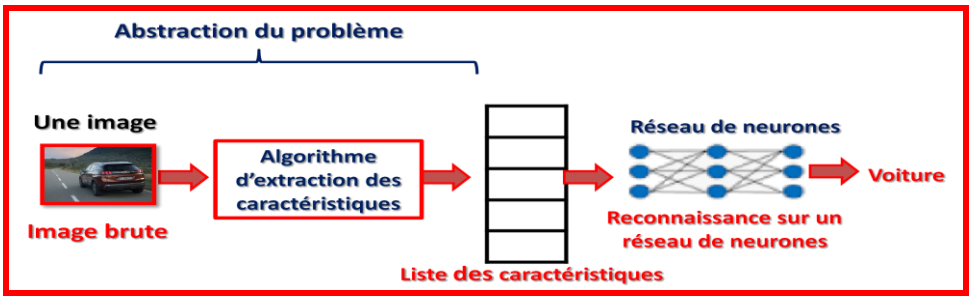

Figure IX.12 – Exemple de reconnaissance de véhicule

Dans le Machine Learning, Il faut un expert du domaine pour extraire la liste des caractéristiques (nombre de roues, longueur, largeur, hauteur, nombre de porte, couleur, etc.). Par contre, dans le Deep Learning, la liste des caractéristiques est extraite automatiquement par la machine.

IX.10 - Comment la machine va-t-elle apprendre et surtout comprendre?

La technique d'apprentissage la plus courante s'appelle l'apprentissage supervisé. On donne des exemples pour que la machine puisse apprendre, c'est la notion de Dataset (ou de tableau de données). On suppose qu'on a un tableau de X et de Y. Il va falloir trouver la relation f qui relie X à Y, telle que f(X) soit proche de Y, c'est-à-dire f(X) $\approx$ Y. La figure IX.13 donne une idée sur la nature de la fonction f de régressions linéaires ou non linéaires.

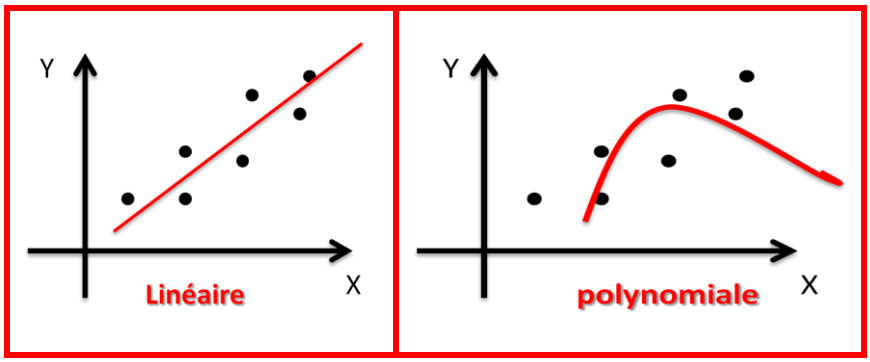

Figure IX.13 – Exemple de régressions linéaire et non linéaire

D'un côté, on a des problèmes de régression : variables continues et nombre fini de valeurs comme sur la figure IX.14a. De l'autre côté, on a des problèmes de classifications : variables discrètes et nombre fini de valeurs comme sur la figure IX.14b. Par exemple, on peut avoir :

$$Y = \begin{cases} o \rightarrow pas\ spam\ x \\ 1 \rightarrow spam\ o \end{cases}$$ comme sur la figure IX.14c.

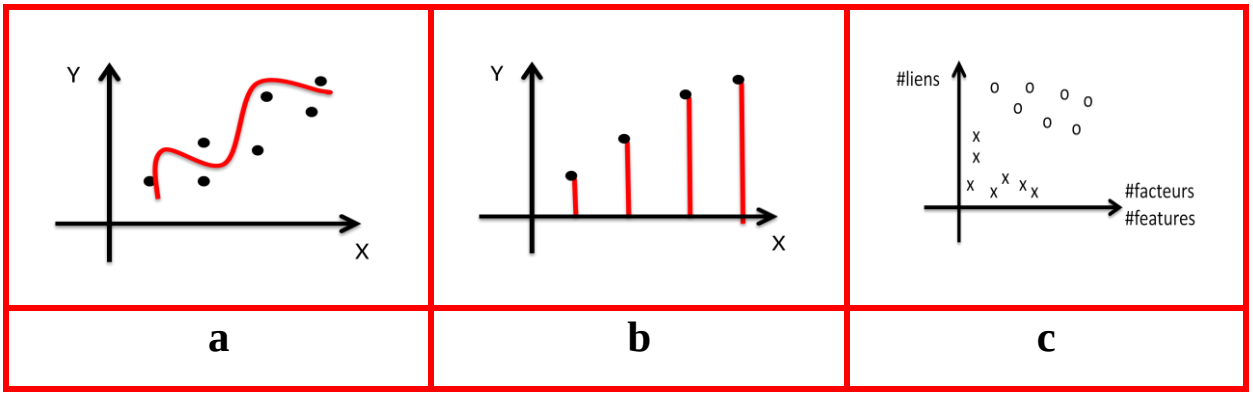

Figure IX.14 – Exemples de régression et de classification

Dans l'exemple d'avant de la figure IX.15c, la machine va créer une frontière de décisions pour scinder l'ensemble en deux parties claires comme sur la figure IX.15. On voit bien que la frontière peut prendre plusieurs formes : droite ou une courbe quelconque selon la nature de la f(X), qui lie les liens avec les facteurs (ou features).

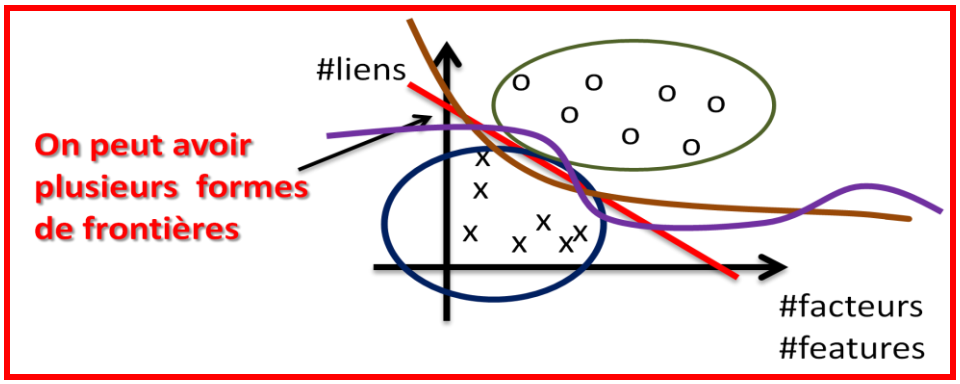

Figure IX.15 – Frontière de décisions

Supposant qu'un forestier s'intéresse à l'étude de certaines caractéristiques particulières d'arbres de la forêt. Il prend le diamètre des troncs et les hauteurs des arbres selon le tableau suivant :

N°	Diamètre cm)	Hauteur (m)
1	10	2.8
2	17	4.6
3	14	4.0
4	9	2.6
5	21	5.8
6	25	6.9
7	31	8.8
8	12	3.4
9	23	6.6
10	32	9.0
11	31	8.7
12	11	3.0
13	13	3.8

La figure IX.16.a nous donne le nuage de points de notre tableau de données. Avec, cette droite rouge de la figure IX.16b, on peut prédire, si je prends un nouvel arbre et si on a le diamètre de son tronc, on peut avoir directement la valeur de sa hauteur.

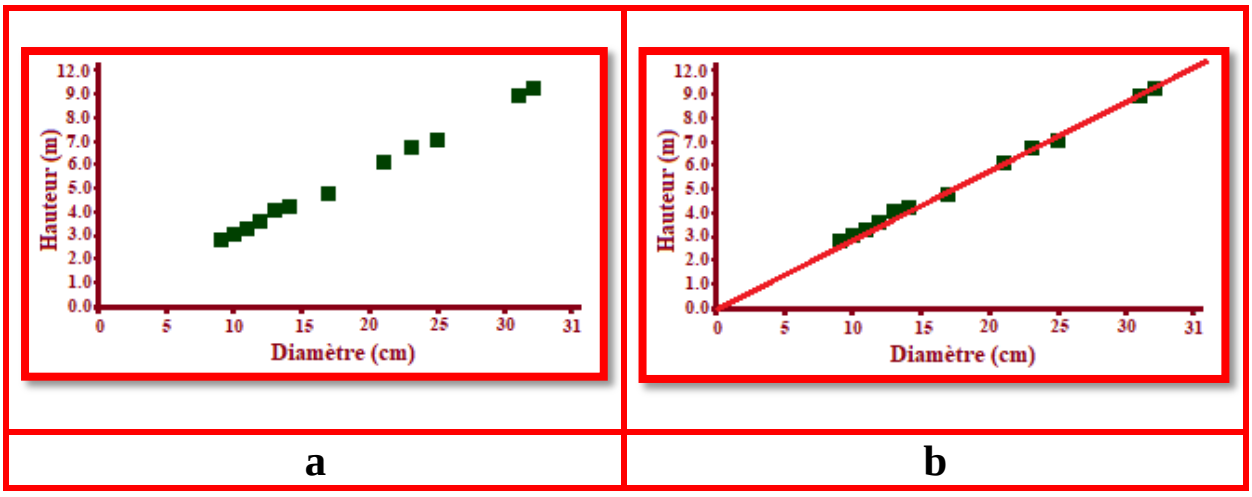

Figure IX.16 – Régression linéaire

IX.11 - Phase d'apprentissage

Dans le Machine Learning, on donne à la machine des exemples (Datasets), et elle doit créer un modèle de prévision, grâce à l'apprentissage supervisé comme sur le tableau de la figure IX.17a. À partir des données de X et Y, on va d'abord procéder au calcul la valeur des paramètres a et b. Ensuite, on va jouer sur les paramètres a et b pour trouver la meilleure frontière qui sépare les deux ensembles de données, qui optimise au mieux les erreurs comme sur la figure IX.17b.

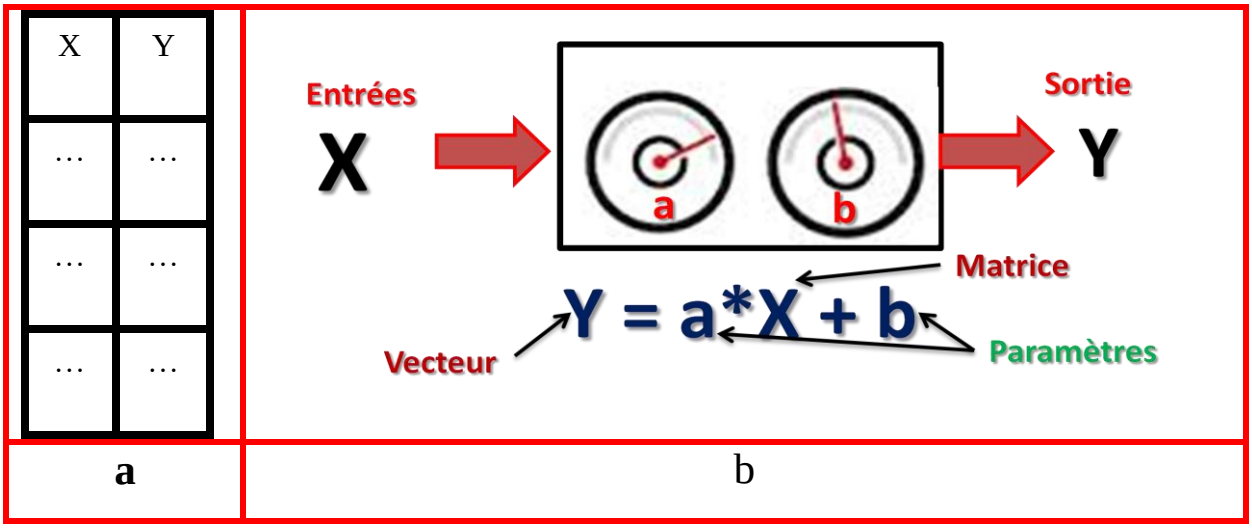

Figure IX.17 – Fonction de prévisions

IX.12 - Apprentissage supervisé (Supervised Learning)

Le principe consiste à monter à une machine des exemples de X et de Y, et de lui demander de trouver l'association qui relie X à Y. En fait, trouver F telle que Y = F(X). Les

quatre notions fondamentales de l'apprentissage supervisé sont : le Dataset (ou le tableau de données), le modèle, la fonction coût et l'algorithme d'optimisation.

IX.12.1 - Apprentissage supervisé

a – Dataset (ou tableau de données)

Ici dans notre exemple, on a un Dataset formé de deux types de variables pour illustrer un peu le problème. Dans les problèmes concrets, on peut avoir des Datasets formés de n variables. Exemple de Dataset sur des appartements d'une agence immobilière (voir la figure IX.18).

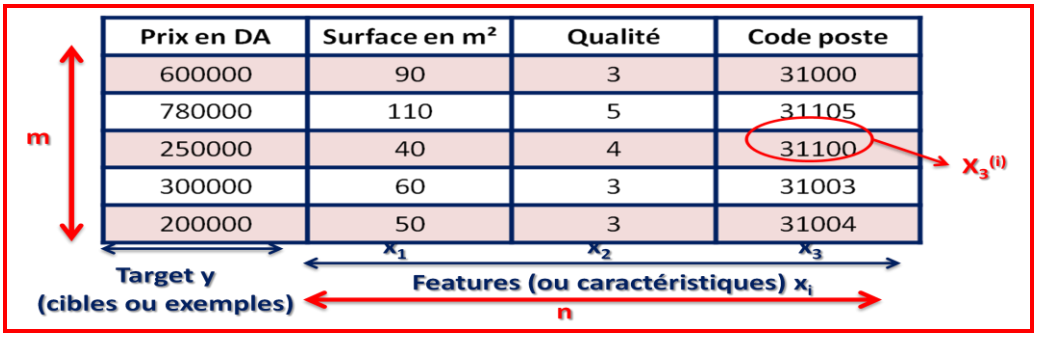

Figure IX.18 – Dataset sur des appartements d'une agence immobilière

Par convention, on note m = nombre d'exemples et n = nombre de features comme sur la figure IX.19.

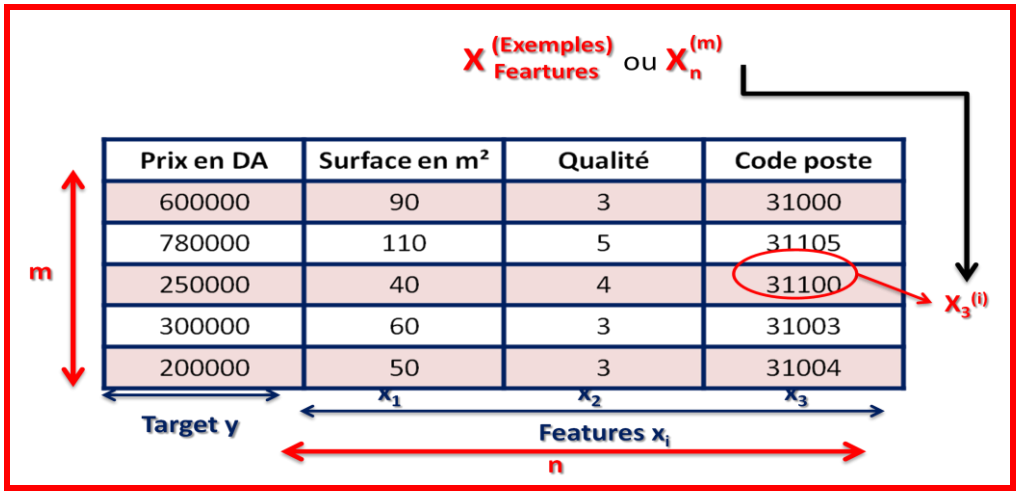

Figure IX.19 – Dataset détaillé avec les conventions d'écriture

La figure IX.20 illustre le Dataset (X, Y), avec le vecteur Y et la matrice X de données.

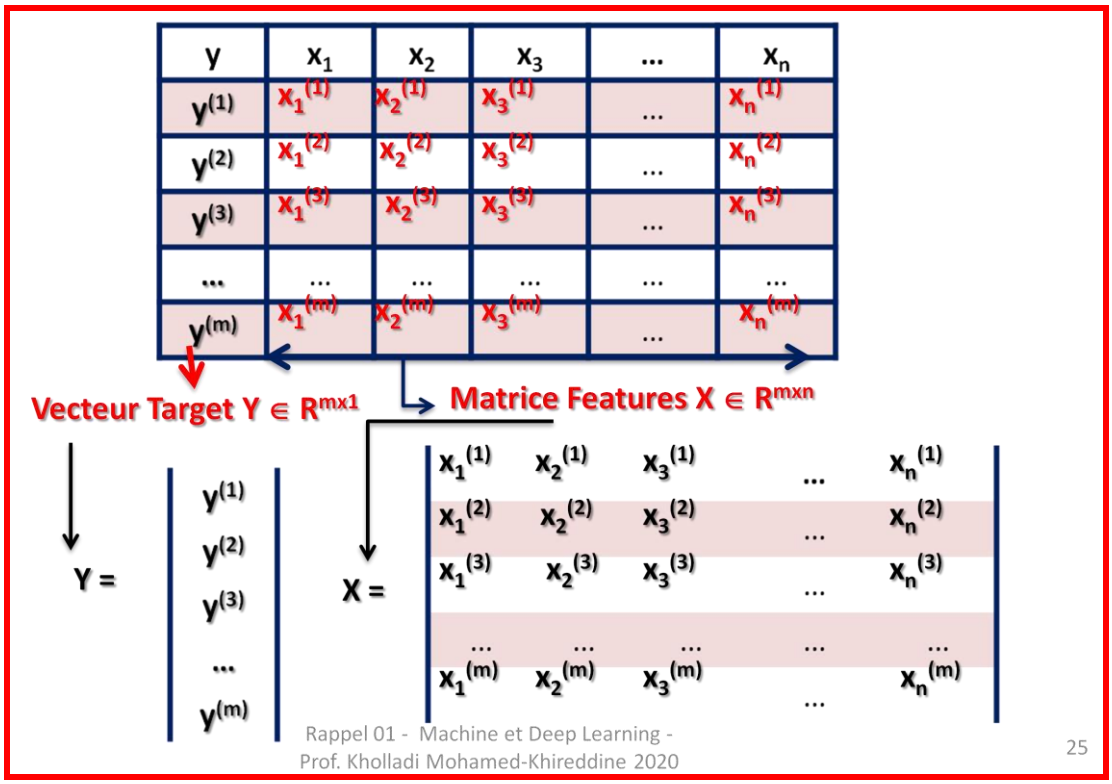

Figure IX.20 – Dataset (X, Y)

b – Modèle

A partir du Dataset (X, Y) où les X représentent les Features et le Y représente le Target. On peut prendre différents modèles selon la nature du problème traité.

- Modèle linéaire → $f(x) = a*x + b$
- Modèle polynomial → $f(x) = a*x^2 + b*x + c$

$$f(x) = a*x^3 + b*x^2 + c*x + d$$

- Modèle quadratique, etc.

Il s'agit de calculer les valeurs des paramètres **a**, **b**, **c** et **d** pour pouvoir prédire de nouvelles valeurs comme sur la figure IX.21.

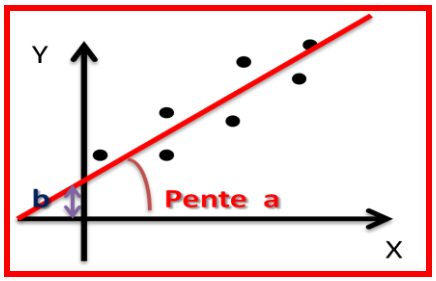

Figure IX.21 – Modèle linéaire

c - Fonction coût

Un bon modèle doit permettre la généralisation. La fonction coût représente la somme des erreurs entres les valeurs réelles et les valeurs estimées.

d - Algorithme d'apprentissage

L'algorithme d'apprentissage permet de minimiser les erreurs. Il développe une stratégie d'optimisation des paramètres a, b, c, etc. qui vont minimiser au mieux la fonction coût. Pour cela, on utilise la méthode de descente du Gradient (Gradient Descent) ou la méthode des moindres carrés.

e - Problème de classification

Par exemple, on veut savoir si une cellule est cancéreuse ou non. Ainsi, on aura un Dataset des cellules observées sur le tableau suivant : (x_1 = taille de la cellule, x_2 = taille du noyau, etc. et y = 1 si cellule cancéreuse ou y = 0 si cellule saine). On cherche la meilleure courbe ou droite et même dans certain cas, le meilleur plan qui sépare au mieux nos deux ensembles en jouant sur les valeurs des paramètres (a, b, c, etc.) comme sur la figure IX.22.

y	x_1	x_2	...	x_n
1				
0				
1				
...				
1				

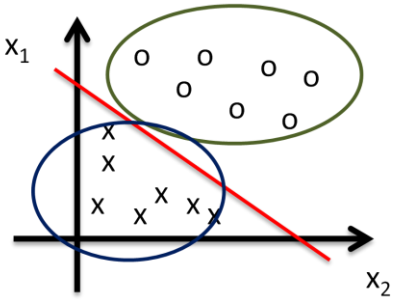

Figure IX.22 – Exemple de classification

d – le modèle de régression linéaire

La figure IX.23 illustre la démarche pour construire le modèle de régression linéaire.

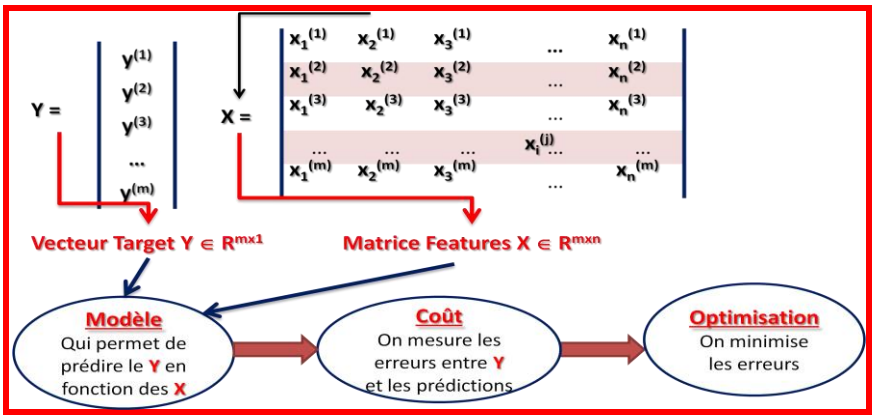

Figure IX.23 – Modèle de régression linéaire

d.1 - Exemple d'application de la régression linéaire

Soit le Dataset (x, y) avec m = 7, n = 1, $x^{(i)}$, on a le modèle linéaire avec des paramètres a et b avec f(x) = a*x + b. C'est le rôle de machine de trouver a et b. La fonction coût donne la mesure des erreurs e_i entre y et les prédictions. Et enfin, l'optimisation va jouer sur les valeurs des paramètres a et b pour trouver la meilleure droite qui représente au mieux le nuage de points et minimise les erreurs $\mathbf{e_i}$. comme sur la figure IX.24.

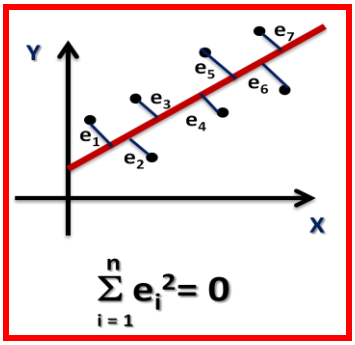

Figure IX.24 – Application de la régression linéaire

e - Fonction coût

Soit on la valeur absolue de → $\mathbf{|f(x^{(i)}) - y^{(i)}|}$ **et** Et soit le carré de la différence (norme euclidienne → $\mathbf{(f(x^{(i)}) - y^{(i)})^2}$, **o**n calcule l'erreur quadratique moyenne (mean squared Error). La fonction coût sera donc comme sur la figure IX.25 :

$$J(a, b) = 1/2m \Sigma (f(x^{(i)}) - y^{(i)})^2$$

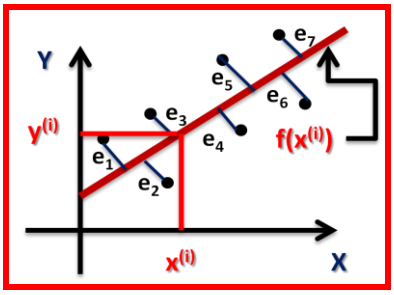

Figure IX.25 – La fonction coût

f - Algorithme de minimisation

Il y a deux méthodes d'optimisation :

1. La méthode des moindres carrés, qui utilise une inversion de matrice sur notre Dataset, ce qui est mauvais pour les très grandes matrices et peut avoir des problèmes de taille mémoire de la machine.
2. La méthode de la descente du Gradient, qui est la plus simple à utiliser, puisque c'est une méthode itérative de descente ou de la montée du Gradient selon que l'on se trouve d'un côté ou de l'autre de la courbe à partir d'une valeur (a_0 ou b_0) des paramètres a ou b) comme sur la figure IX.26.

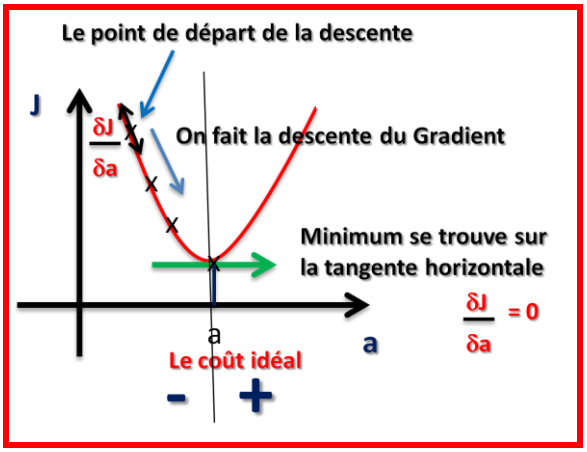

Figure IX.26 – Algorithme du Gradient

C'est un algorithme, qui est au cœur du Machine Learning et du Deep Learning.

f1. -Algorithme de descente du Gradient

La descente du Gradient (Gradient Descent) est un algorithme qui calcule la convergence vers le minimum d'une fonction convexe. Il permet de trouver le minimum de n'importe quelle fonction convexe comme sur la figure IX.27. En apprentissage supervisé (Supervised Learning), la descente du Gradient permet de minimiser la fonction coût qui est justement une fonction convexe (erreur quadratique moyenne). En effet, c'est de cette façon que la machine apprend. En fait, cela revient à trouver le meilleur modèle pour notre programme de Machine Learning pour n'importe quel type d'application.

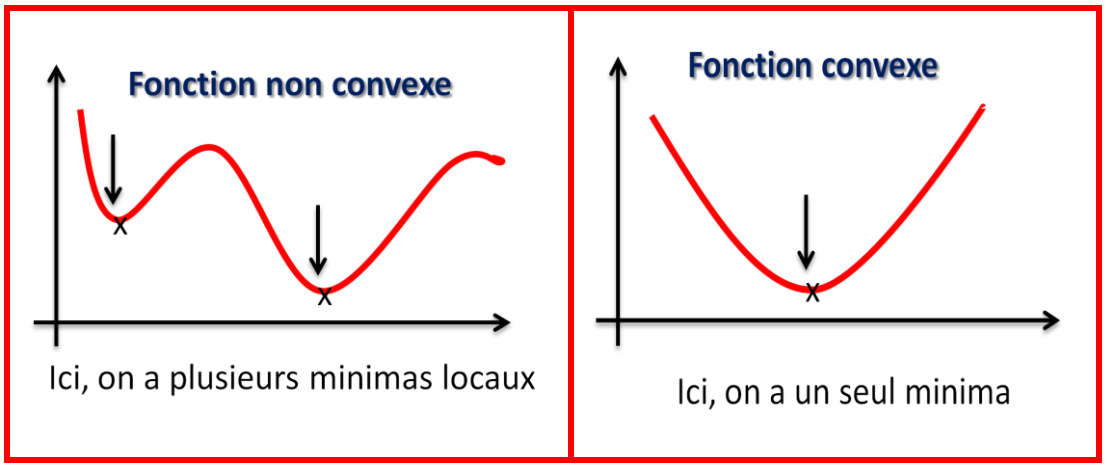

Figure IX.27 – Fonction non convexe et fonction convexe

On commence l'algorithme avec des paramètres de a et de b choisis au hasard par exemple, on prend a_0 et b_0. Le paramètre Alpha (α) pour la paramètre a (et Béta (β) pour le paramètre b) appelé Learning Rate définit la vitesse de convergence de l'algorithme vers le minimum. On dit que ce sont des hyper-paramètres comme sur la figure IX.28. Et on recommence en boucle :

1. Calcul de la pente au point suivante a_i
2. Petit pas α dans la descente
3. Etc. pour tous i
4. Jusqu'à converger vers a.

On fait la même chose avec le paramètre b. On calcule la dérivé partielle de J(a, b) par rapport à chaque paramètre pour trouver la pente.

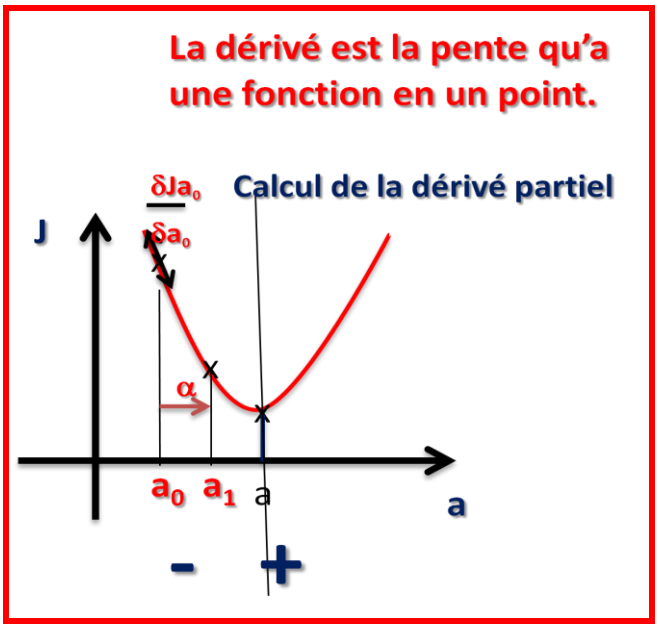

Figure IX.28 – Algorithme de descente du Gradient

On fait, en utilisant la courbe en fonction des deux paramètres a et b, on a une hyper courbe en 3D. Vu d'en haut, on a une sorte de courbes de niveaux comme sur les cartes géographiques comme sur la figure IX.29.

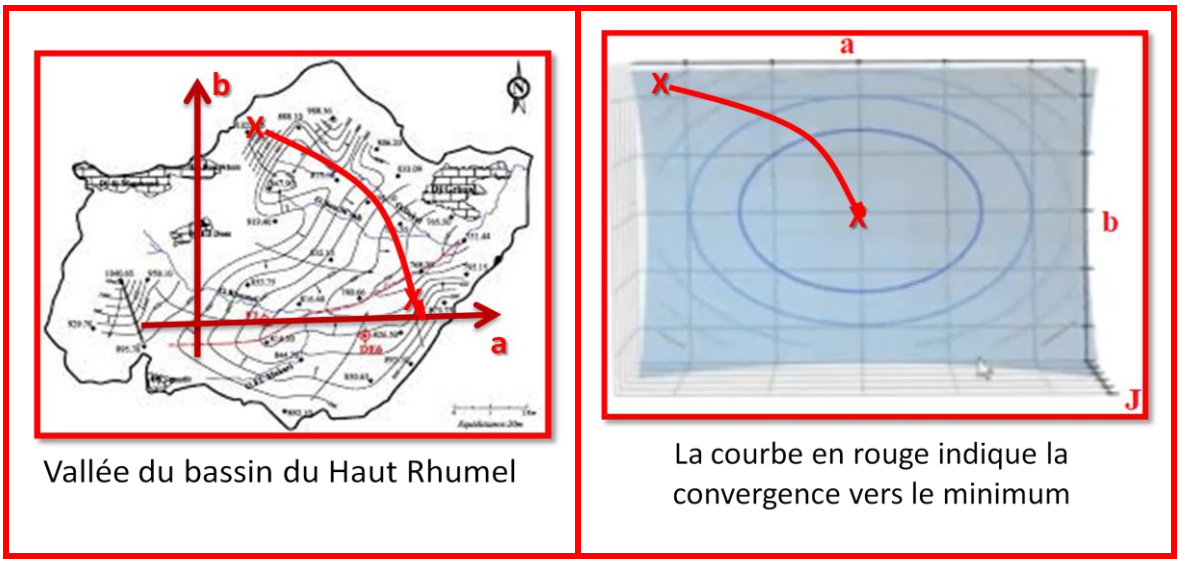

Figure IX.29 – Les courbes de niveaux

Vue de profil de par rapport à a et celui de b ou des deux à la fois a et b, on a les coupes suivantes par profil de la fonction coût comme la figure IX.30.

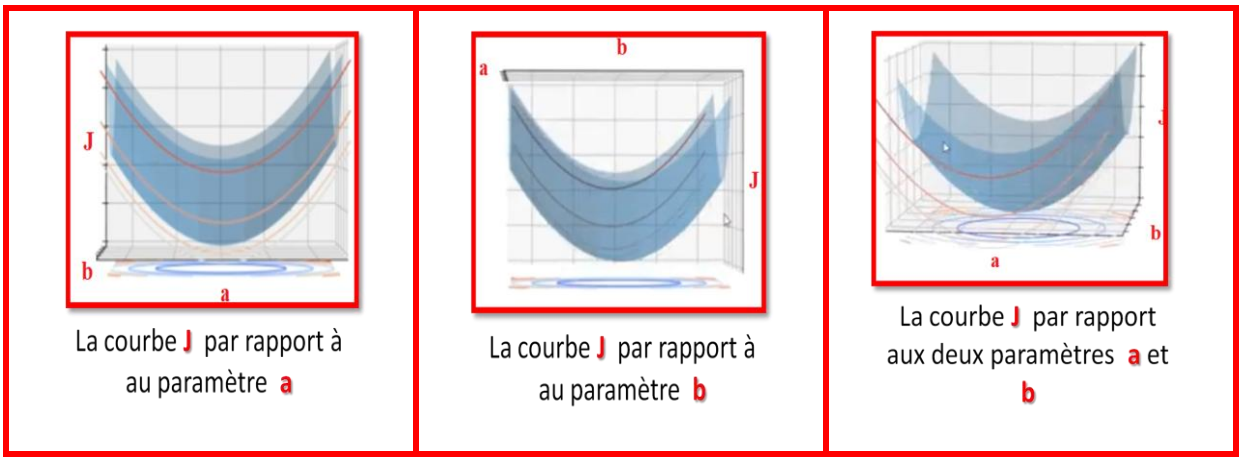

Figure IX.30 – Les différentes coupes de profil des courbes selon a, b et ab

$$a_{i+1} = a_i - \alpha \left(\frac{\delta J(a_i)}{\delta a} \right)$$

On met cette formule dans une loupe et l'algorithme devient itératif.

$$a_1 > a_0$$
$$a_1 = a_0 - \alpha \left(\frac{\delta J(a_0)}{\delta a} \right) \quad <0$$

Si α est trop grand, on fait de trop grands pas et on risque d'avoir le phénomène d'oscillation autour de notre objectif sans jamais l'atteindre. Si α est trop petit, alors on risque de mettre un temps infini pour atteindre l'objectif. D'où l'importance du choix de la valeur de α (Learning Rate) comme sur la figure IX.31.

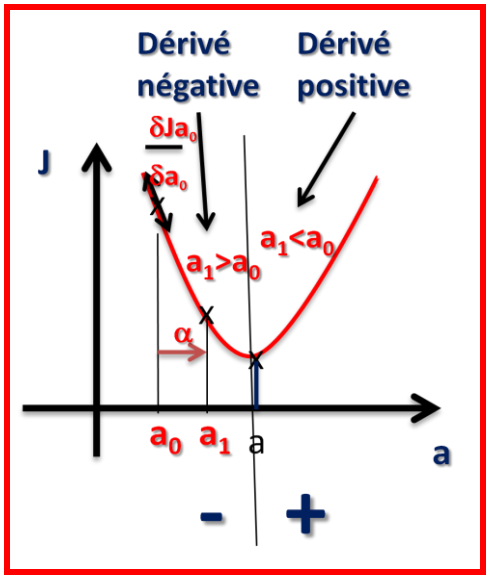

Figure IX.31 – Calcul de la dérivé

f2. -Calcul du gradient

$$a = a - \alpha\left(\frac{\delta J}{\delta a}\right) \quad \text{et} \quad b = b - \alpha\left(\frac{\delta J}{\delta b}\right)$$

Notre fonction coût, si c'est une erreur quadratique moyenne, ce qui nous donne :

$$J(a, b) = 1/2m \, \Sigma \, (a*x + b - y)^2$$

J(a, b) est en faite, une fonction composée de deux fonctions telles que : (g°f)(a, b), avec :

$$f(a, b) = a*x + b$$
$$g(f) = f\text{^}2$$

Rappel : La dérivé de (g°f)' = f' + g'(f)

$$\frac{\delta J}{\delta a} = 1/m \, \Sigma \, x^* \, (a*x + b - y)$$

et

$$\frac{\delta J}{\delta b} = 1/m \, \Sigma \, (a*x + b - y)$$

La fonction coût est égale à :

$$f(a, b) = 1/2m \sum_{i=1}^{m} (a*x^{(i)} + b - y^{(i)})^2$$

Les Gradients sont égaux à :

$$\frac{\delta J(a, b)}{\delta a} \quad \text{et} \quad \frac{\delta J(a, b)}{\delta b} = 1/m \sum_{i=1}^{m} (a*x^{(i)} + b - y^{(i)})$$

Algorithme de descente des Gradients

$$a = a - \alpha \left(\frac{\delta J(a, b))}{\delta a} \right)$$

et

$$b = b - \alpha \left(\frac{\delta J(a, b))}{\delta b} \right)$$

g. - Régression linéaire

La figure IX.32 illustre le schéma de principe de la démarche de calcul de la régression linéaire.

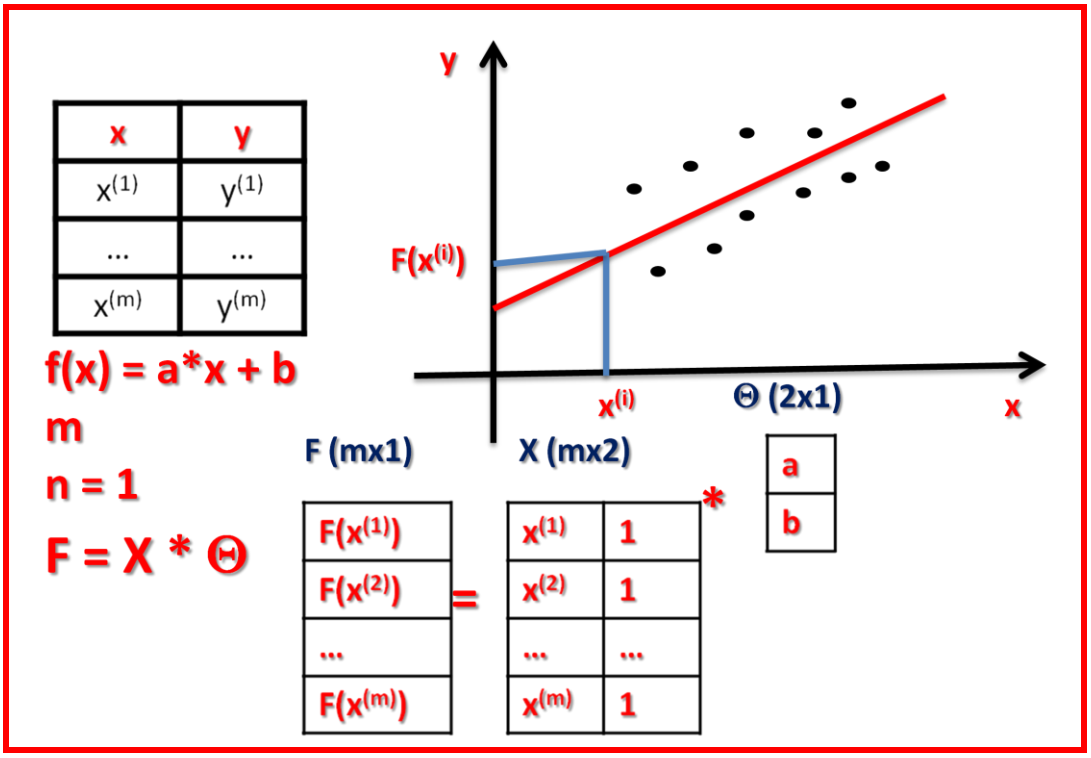

Figure IX.32 – La régression linéaire

La fonction coût sous forme de calcul matriciel :

$$J(a, b) = 1/2m \sum_{i=1}^{m} (a*x^{(i)} + b - y^{(i)})^2$$

(1x1)

$a*x^{(i)} + b$: $X * \Theta$ (mx1)

$y^{(i)}$: Y (mx1)

$$J(\Theta) = 1/2m \sum (X * \Theta - Y)^2$$

Le gradient sous forme matriciel :

$$\left.\begin{array}{l} \dfrac{\delta J}{\delta a} = 1/m \, \Sigma \, x^* \, (a^*x + b - y) \\ \text{et} \\ \dfrac{\delta J}{\delta b} = 1/m \, \Sigma \, (a^*x + b - y) \end{array}\right\} \quad \frac{\delta J(\Theta)}{\delta \Theta} = 1/m \, * \, X^t \, (X^* \, \Theta \, - Y)$$

La descente du Gradient :

$$\left.\begin{array}{l} a = a - \alpha \left(\dfrac{\delta J}{\delta a}\right) \\ \text{et} \\ b = b - \alpha \left(\dfrac{\delta J}{\delta b}\right) \end{array}\right\} \quad \Theta = \Theta - \alpha \left(\frac{\delta J}{\delta \Theta}\right)$$

Régression polynômiale :

$$f(x) = a^*x^2 + b^*x + c \quad \Rightarrow$$

$$\Theta = \text{Vecteur} \begin{array}{|c|} \hline a \\ \hline b \\ \hline c \\ \hline \end{array}$$

Donc, on aura le calcul de trois Gradients.

h. - Classification linéaire & Classification non linéaire

Dans la classification linéaire, pour minimiser la fonction coût, on peut utiliser diverses méthodes : la moindre carré, l'analyse discriminante linéaire, l'approche probabiliste générative, l'approche probabiliste discriminante : régression logistique qui utilise la descente du Gradient, etc. Dans la classification non linéaire, on va utiliser diverses méthodes basées sur les méthodes à noyau.