# Introduction

# Les prérequis

Avant tout, il va falloir préciser un certain nombre de choses. L’IA n’est pas facile, ce n’est pas un domaine auquel n’importe qui sans le maximum de volonté peut y entrer. Bien évidemment il y a quelques prérequis, deux pour être précis que sont les mathématiques et l’algorithme. Et pour ces deux domaines il va falloir être excellent dans un et avoir un bon niveau dans l’autre, si vous devenez excellent dans les deux, vous êtes ce qu’on appelle une légende. A part les mathématiques et l’informatique il y a beaucoup d’autres prérequis qui ne sont pas nécessaire mais peuvent aider dans notre objectif, et tous ses autres prérequis vont classes dans le domaine intelligence sociale.

## Les mathématiques

Quand on parle de mathématique la plupart des gens vont prendre peur, abonder voire même fuir. Mais ici, nous allons voir les concepts mathématiques qui nous serons utile à l’IA mais manière simple et concise.

Les mathématiques ne sont pas compliquées surtout quand on lui trouve une application et l’une des plus application des mathématiques c’est l’IA. Nous allons vous monter comment c’est fascinant de résoudre des problèmes mathématiques pour créer des modèles intelligents. Les mathématiques sont plus que nécessaire pour l’IA, elles sont vitales. D’ailleurs mon professeur d’intelligence artificielle nous disait à la fin d’un cours : « l’intelligence artificielle ce n’est ni plus ni moins que des calculs mathématiques ». Calculs mathématiques qui vont être faciliter avec un ordinateur, et c’est là que va utile l’ordinateur, l’informatique et l’algorithme.

En dépit du fait qu’il y a plus domaines mathématiques qui nous serons utile dans l’IA, pour ce travail de mémoire, nous allons nous concentrer sur seulement trois (3) domaines des mathématiques que sont les statistiques et probabilité, l’algèbre linéaire et l’analyse. Pour ces trois, nous allons seulement les définir mais aussi donnez leur application concrète sur l’IA. Cette approche de voir les mathématiques va être s’avérer être plus intéressant.

### Les statistiques et probabilités

Is everything in on this planet determined by randomness? This question is open to philosophy debate. What is certain is that every day thousands and thousands of engineers, scientists, business persons, manufactures, and others are using tools from probability and statistics. (Dekking, Frederik Michel, 2005).

Cette citation de Michel nous renvoie à comment sont important ses domaines dans nos vies de tous les jours et l’IA ne fait pas exception. Le statistique est domaines des mathématiques qui travaillent sur des données en les faisant parler ce qui nous permet de mieux comprendre les valeurs d’une base de données. C’est ce qu’on les statistiques descriptives. Il y a aussi les statistiques inférentielles qui comme sont l’indique va nous permettre de faire des inférences c’est-à-dire estimer l’évolution d’une population. Et c’est là que réside le lien entre les probabilités et statistique car le statistique inférentiel va avoir besoin les probabilités. La probabilité est l’étude de la chance pour qu’un évènement se produise pour faire simple.

Ceci étant dit, comment ses deux sont utiles en Machine Learning et Deep Learning ? Ils interviennent tous les deux avants et après le développement modelé d’IA.

* **Avant le développement de modèle** : les statistiques nous aident à comprendre les données, car très souvent les données brutes ne sont pas exploitables. Ici, nous vérifier les maximums des valeurs, minimum, la moyenne, le mode et l’une parte les plus importante du « Feature Engineering » la mise à l’échelle etc.
* **Apres le développement de modèle** : il va bien falloir calculer la fiabilité du modèle, ce qu’on appelle « accuracy », il faut calculer aussi, la précision, le f1-score, le recall… Ces derniers nous permettent d’apprécier la robustesse du modèle d’une fois déployer. Nous pouvons aussi faire des graphs comme la matrice de confusion par exemple.

### L’algèbre linéaire

Au fait, il trois grandes parties dans le développement d’un réseau de neurone et a titre illustré, nous pour pouvons dire le travail a posteriori, le modèle et travail a priori. Pour le modèle il y a deux parties le Feed-forward et le Backpropagation et l’agréable linéaire va intervenir dans ces deux parties.

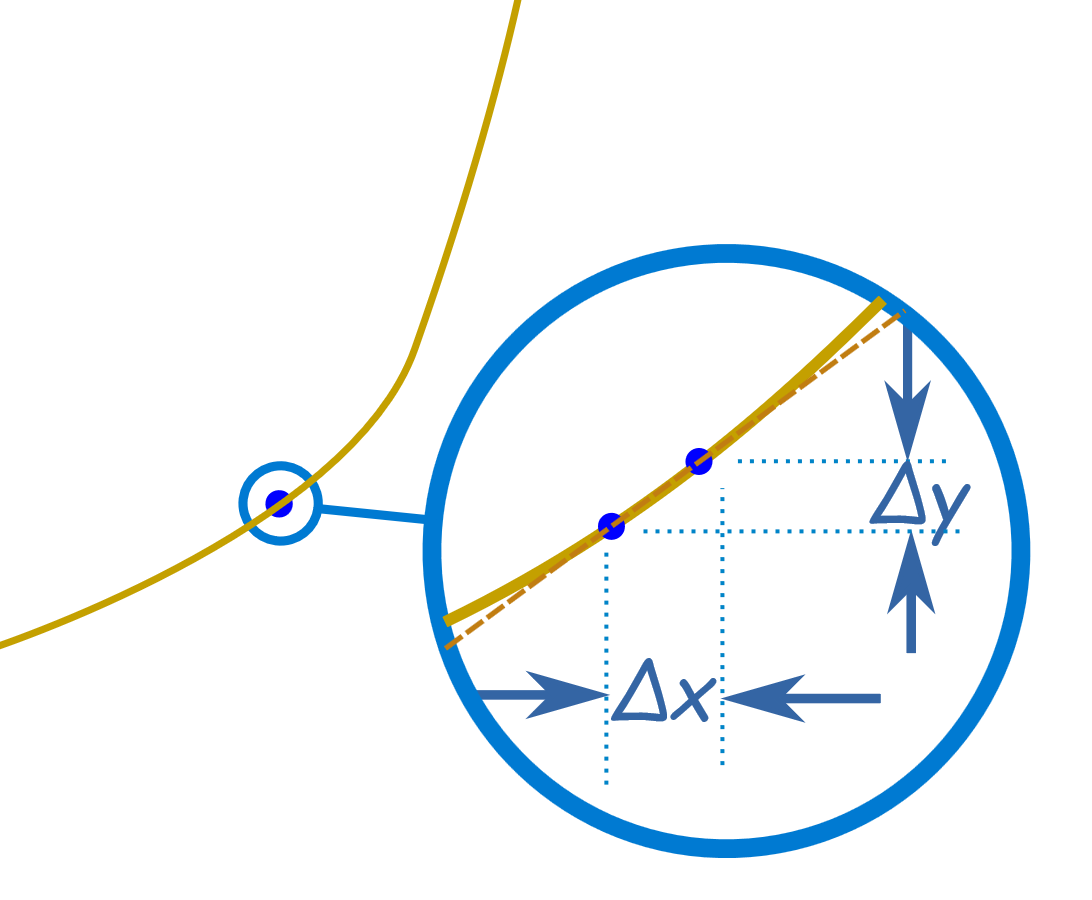
L’algèbre linéaire est la branche des mathématiques qui s'intéresse à l'étude des espaces vectoriels (ou espaces linéaires), de leurs éléments les vecteurs, des transformations linéaires et des systèmes d'équations linéaires (théorie des matrices). (Algèbre linéaire - Définition, 2024)

Ainsi, la plus grande utilité de l’algèbre linéaire est le calcul de poids, elle va nous permettre de d’automatiser les calculer lourds et couteux, de ce fait nous permettant de gagner du temps. Sans elle nous aurions passé beaucoup de temps sur ces calculs. Je rappelle qu’un réseau de neurone a des milliers de neurones d’inputs, plusieurs couches cachées qui peuvent avoir elle-même des milliers de neurones. C’est juste impossible de calculer de ceci d’une séquentielle.

### L’analyse

Quand on parle d’apprentissage en IA ou le terme très populaire anglais « Learning », c’est à cause du domaine des mathématiques l’analyse avec le calcul des dérivés. Si l’IA est devenue ce qu’elle est devenue aujourd’hui avec les performances qu’on la connait, c’est en très grande partie cause des calculs d’analyse mathématique.

La dérivée d’une fonction nous informe sur la variation d’une de la fonction en un point. Pour être plus claire, elle nous permet de calculer la pente d’une fonction sur n’importe quel point de la fonction. Le calcule de dérivée est très important dans beaucoup de domaine notamment dans de Deep Learning.



Maintenant comment se passe l’apprentissage dans un réseau de neurones ? Nous allons demander à l’IA de faire une assomption, n’importe laquelle. Au premier coup, elle fera certainement une erreur, ensuite il va falloir rectifier cette erreur. Puis nous allons lui demander de faire une seconde assomption et après on rectifie en cas d’erreur. Si nous répétons ces actions autant de fois que nécessaire, l’erreur d’assomption va se réduire au minimum et l’ « accuracy » va se maximiser. Ce qu’il faut comprendre par là c’est que le calcul de la dérivée de la fonction l’erreur va nous permettre de rectifier cette erreur.

De manière pratique, on calcul l’erreur en premier lieu, puis on calcul la dérivée de la fonction d’erreur. La manière dont la rectification va se faire, c’est on va donnez à chacune des poids une valeur correspondante à leur responsabilité dans l’erreur et c’est ça le Backpropagation ou la rétropropagation en français.

## L’informatique

L’information c’est la science de l’automatisation de l’information, d’ailleurs son nom vient de la : une contraction entre information et automatisation. Chez les anglosaxons, on parle plutôt de de Computer science qui se traduit littéralement par science de l’ordinateur.

Plus haut, nous avions attester que l’IA est une science purement mathématique avec que des calculs que l’on pourrait même faire sur feuille. Des lors, que représente l’informatique pour l’IA : elle a rôle d’une calculatrice géante capable de faire des super calculs en temps record. Ajouté a cela l’informatique nous permet en outre de présenter les résultats dans un interface graphique pour que n’importe puisse y avoir accès.

C’est ainsi que nous allons voir les domaines, de ce vaste étendu qu’est l’informatique, qui vont nous intéresser pour développer des modèles intelligents.

### L’algorithme

Bien évidemment, la première des choses que l’on va voire c’est l’algorithme. Nous pouvons attester sans prendre beaucoup de risques que l’algorithme est l’informatique, et que l’informatique est l’algorithme.

L’algorithme est l’ensemble des étapes auxquelles il va falloir passer pour résoudre un problème informatique. Il est souvent fait l’analogie de la recette de cuisine pour illustrer l’algorithme et juste titre.

La raison pour laquelle il faut maitriser l’algorithme c’est que : pour implémenter un problème mathématique dans un ordinateur, il faut savoir comment s’y pendre quelles étapes à suivre, sinon beaucoup de frustration nous attends.

Exemple : écrivons un programme qui résous un polynôme du second dégrée, un algorithme simple serai la suivante :

* Afficher : Donner les valeurs a, b et c.
* Stocker a, b et c dans des variables.
* Calculer delta (delta = b2 – 4 \* a \* c)
* Si delta positif alors x1 = (-b – racine(delta) / 2 \* a) et x1 = (-b + racine(delta) / 2 \* a)
* Si delta nul alors x = racine(delta) / 2 \* a
* Si delta négatif alors il n’y a pas de solution dans R.

Voici ci-dessus un algorithme qui marche pour un polynôme du second dégrée et cette même manière de réflexion pour peut implémenter n’importe quel problème déjà résolu en mathématique en algorithme informatique.

### Les structure de données

D’abord, les structures de données sont le terme utilisé pour représenter toutes les différentes façons en informatique pour modéliser les données avec lesquelles nous travaillons. Très souvent, pour ne pas dire tout le temps, nous avions directement la façon optimale de gestions de données.

Ces structures peuvent partir d’un simple tableau dans un langage de programmation jusqu’à atteindre les graphs (structure de données de complexe et très puissante). Comme nous l’avons dit et redit l’IA travail sur des données. Citions quelques exemples de structures de données :

* Les listes chainées
* Les tables de hachages
* Les arbres
* Les piles et files
* Les graphs
* …

### Les langages de programmation

Les langages de programmation sont les langages aussi dits informatique qui traduise les algorithmes d’une manière compréhensible a l’ordinateur. Il faut préciser que l’ordinateur ne comprend pas le texte, il comprend seulement les chiffres (nombre binaire en l’occurrence). Ce que le langage de programmation fait c’est de convertir sont syntaxe en langage binaire compréhensible par l’ordinateur et chaque langage a sa propre syntaxe.

L’importance des langages de programmation va être évidente pour le monde, de ce fait nous allons présenter quelques-uns ici.

* C/C++ (important pour l’IA)
* Python (important pour l’IA)
* Java
* PHP
* JavaScript
* …

## L’intelligence sociale

Tous les domaines qui restent qui ne sont forcément liées à la science mais qui apporter une plus-value importante, nous allons les classer dans l’intelligence sociale.

* Repérage de problème

Celui ou celle qui prétend à travailler dans l’IA doit à tout prix être mesure de repérer les problèmes auxquels sont confrontés sa communauté. C’est ici que réside le vrai intérêt de l’IA rendre la vie des gens mieux. Et nous n’avons pas besoin chercher besoin loin, des problèmes sont trouvables partout, il suffit juste de bien observer son environnement.

* La résolution de problème

Apres avoir desceller de potentielles problèmes, il va falloir naturellement proposer des solutions. Il faut plusieurs modèle et algorithme d’IA qui excellent dans diffèrent domaines, modèle et algorithme que nous allons voire dans partie suivante. Il noter aussi que c’est le problème qui définit le modèle utilisé mais pas l’inverse.

* La pédagogie

Pourquoi la pédagogie ? même on n’est pas tous appeler à être professeur mais en tant qu’ingénieur dans l’IA, nous allons nous retrouver très souvent en train d’expliquer des concepts. Maintenant, il va falloir être capable d’expliquer des concepts complexes d’une manière simple. Apres le développement d’un modèle il va bien falloir l’expliquer au client et aux utilisateurs.

# Les algorithmes d’intelligence artificielle

Nous y voilà, l’une des parties les plus importante de ce travail de mémoire. Nous avons parlé de l’IA dans ce document mais cette fois nous voir comment elle fonctionne en parcourant diffèrent des plus importants algorithmes d’IA, celle qui sont vraiment utiliser dans par les grandes entreprises. Donc pour cette partie, je vais vous demander une attention particulière car ce sera très intéressant.

Alerte âme sensible !!! il y aura beaucoup de calculs mathématiques dans cette partie.

## Machine Learning

Littéralement, Machine Learning veut dire apprentissage des machines. Comme nous êtres humains, nous naissons sans connaissance dans notre tète, mais en regardant notre environnement et en imitant nos parents, nous apprenons. Ce processus peut être répliquer sur un ordinateur, c’est la Machine Learning, il y en a deux : Supervised Machine Learning (SML), Unsupervised Machine Learning (UML).

### Supervised learning

Si nous reprenons l’analogie de l’enfant, dans sa phase d’apprentissage ses parents vont être derrière et le guider. Si, l’enfant commet des erreurs ses parents vont de rectifier s’il fait une bonne chose ses parents vont le récompenser ou l’encenser.

Dans le domaine des ordinateurs, pour faire en sorte qu’une machine apprenne, on aura besoin de données, beaucoup de données. Et chaque ligne de donnée va être étiqueté, on parle input et d’output. Maintenant, le modèle va essayer de s’adapter à tous les inputs et leurs outputs.

Nous allons voir dans la suite les différents types d’apprentissage supervisé et leurs algorithmes.

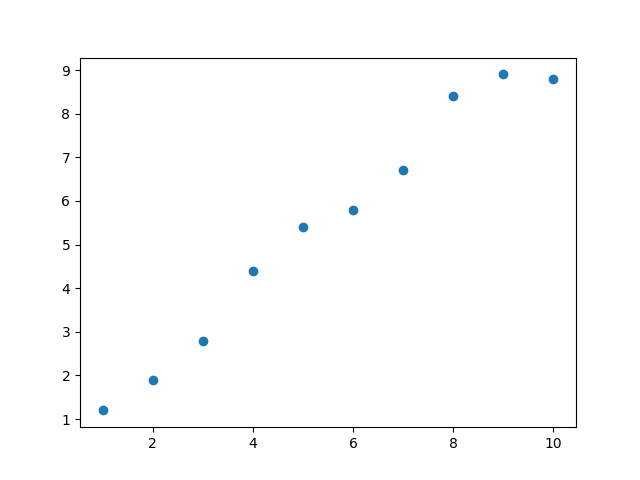
#### La régression

La régression est méthode statistique qui nous permet de d’approximer la valeur d’une variable a partie des valeurs déjà présentes et connues. Elle va se faire en traçant une courbe qui représentée le mieux la relation en les points dans un repère orthonormal. Il y a plusieurs types de régression mais nous en voir trois (3).

##### La régression linière

Ci-après un tableau de valeur et sa représentation graphique.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Y | 1.2 | 1.9 | 2.8 | 4.4 | 5.4 | 5.8 | 6.7 | 8.4 | 8.9 | 8.8 |



Prenons cette courbe, ci-dessus nous voyons la représentation d’un certain nombre de points. Maintenant si nous voulons tracer une droite qui va au mieux représenter l’évolution de ces points point, qu’allons-nous faire. Il y a la méthode des moindre carrés, élaboré par le légendaire Carl Friedrich Gauss, qui est une méthode purement statistique mais allons utiliser une méthode d’IA avec la descente des gradients. Cette dernière méthode peut être divisée en trois parties :

* Forward propogation (essaie au hasard)

D’abord la courbe que l’on veut tracer va être de la forme , mais dans le jargon on va parler w0 et w1 qui vont représenter les poids respectifs, l’équation revients . Le but de jeu est de trouver les w0 et w1 qui vont au mieux correspondre à nos points. Dans un premier temps on va les donnez des valeurs aléatoires d’où l’essai au hasard.

* Calculer l’erreur

Nous voyons que les poids pris aléatoirement, le premier essaie fut une erreur, puisque dans le tableau y = 1,2 pour une valeur x = 1. De de fait il faut calculer l’erreur, nous allons la fonction suivante :

MSE : Mean Square Error (la moyenne des erreurs au carré)

Y : la sortie attendue

Ŷ : la sortie observée

N : le nombre d’élément dans le tableau

Nous pouvons maintenant faire une application de cette fonction avec le premier essaie fait.

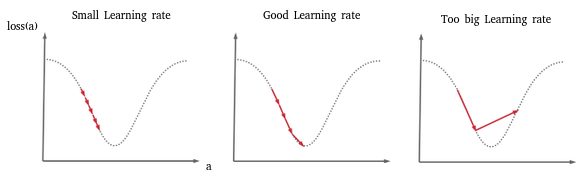
Attention ! il ne faut oublier que de la même manière que l’on a calculer pour x = 1, il faut aussi le calculer pour tous les autres x et ainsi avoir toutes les erreurs pour pouvoir l’appliquer formule générale.

Cependant, les plus curieux vont se demander pourquoi élever l’erreur au carré. C’est une bonne question. La raison est simple car une erreur de -1 est égale à une erreur qui vaut 1. Et le fait de l’élever au carrée va nous aider dans la mise à jour des poids ou nous allons utiliser l’algorithme de la descente des gradients.

* Backpropagation (rétropropagation qui met à jour les poids)

Maintenant que nous l’erreur nous pouvons enfin mettre à jour nos poids w0 et w1. Cela veut chaque va prendre une part de l’erreur qui est égale à sa responsabilité de cette même erreur et se rectifier lui-même. Pour se faire nous allons calculer la dérivée de toutes les fonctions qui nous mener à cette erreur de manière suivante :

Dans la descente des gradients, il y a ce qu’on appelle le pas, il va déterminer à quelle vitesse la descente va se faire. Si le pas est trop petit l’apprentissage va lent et si le pas est trop grand, allons déplacer le point qui minimise le l’erreur, ce pas le « lr » dans les deux fonctions cela signifie « Learning Rate ».



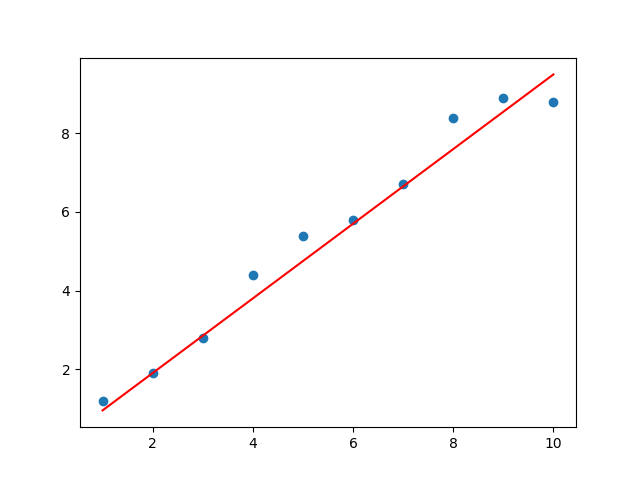
Représente quant a elle, la dérivée de la fonction MSE par rapport au poids concerné, c’est ce qu’on appelle une dérivée partielle.

Exemple :

Donc

Nous avons trouvé 3.6 il reste qu’a le multiplier avec le Learning rate et w0 est près à être mise à jour. Il va falloir faire de même pour w1 et c’est pour le premier epochs (terme anglais qui signifie le parcours en le tous les ligne de jeu de données).

En revanche, le travail ne s’arrête pas là, tous ces trois actions précédant, il falloir les répéter autant fois que nécessaire pour avoir le modèle le plus fiable possible, généralement on parle de milliers d’épochès. Si le travail est bien fait nous pouvons retrouver avec une courbe comme la suivante :



##### La régression logistique

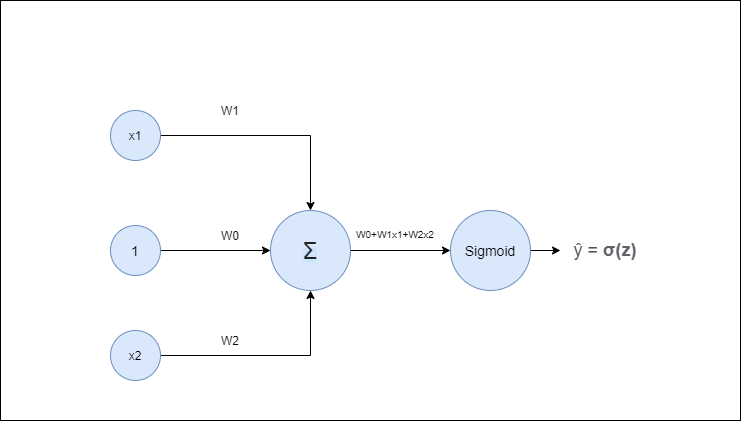
La régression logistique, contrairement à celle dite linéaire, n’a pas pour vocation prédire une future. Sa prédiction est de type binaire : oui ou non, bon ou mauvais, 0 ou 1 etc. Ce genre va s’avérer être très important dans beaucoup de domaines, nous l’utilisons dans nos vies de tous les jours sans s’en rendre compte. Par exemple, détecter si email un spam ou non, si information est fakes news ou non, si un investissement va être rentable ou pas…

Nous allons pour la suite faire l’exemple de la fonction logique OÙ :

|  |  |  |
| --- | --- | --- |
| X1 | X2 | OU |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Les mêmes étapes, que la régression linéaire, vont revenir avec intronisation d’un nouveau concept : la fonction d’activation (hautement important).

* Forward propogation



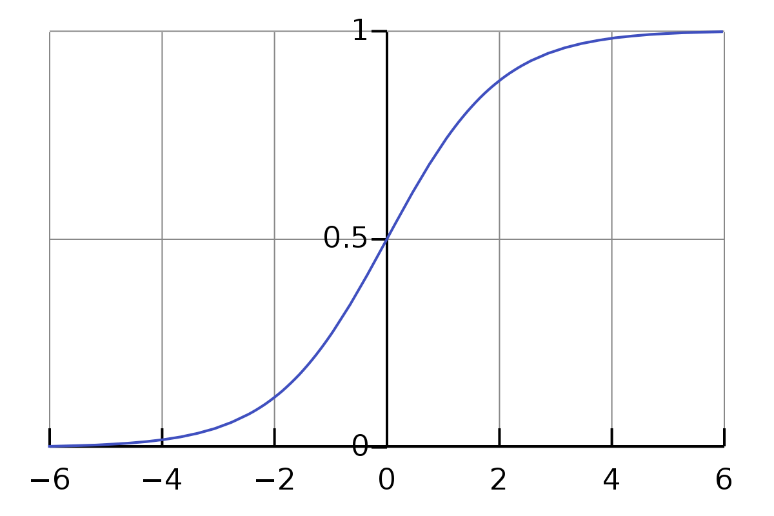
Voici a quoi va ressemble notre réseau de neurone, on va un autre input en plus x1 et x2, c’est le biais qui va toujours être égale a 1, son utilité est d’éviter que certains ne meurt durant l’entrainement si x1 = 0 et x2 = 0.

Commandons par initialiser

Puisqu’on dit que les valeurs de sortie doivent 0 ou 1, nous trouver un moyen toujours mettre à l’échelle la sortie observée, c’est là qu’intervient les fonctions d’activation. Pour les problèmes de régression logistique il y’en deux très populaire : la fonction a seuil et sigmoid.

|  |  |  |
| --- | --- | --- |
| Fonction | Formule | Sortie possible |
| Seuil(x) |  | 0, 1 |
| Sigmoid (x) |  | Tout réel compris en 0 et 1 |

Nous continuer avec la fonction sigmoid :



Donc pour x1 = 1 et x2 = 0, y = 0.12, donc il y a une erreur puisque la sortie doit être 1.

* Calculer l’erreur

Pour l’erreur rien ne va changer nous allons utiliser la Mean Square Error :

* Backpropagation

Nous voici près pour la rétropropagation, seulement ici nous allons mettre a jour trois poids a savoir w0, w1, w2.

Même si les formules restent les mêmes, ne peuvent encore rien pour acquis, ici la valeur de la dérivée partielle va changer étant donné qu’on introduit une nouvelle fonction, celle d’action, nous allons de facto nous retrouver avec trois membres dans le calcul de dérivée partielle.

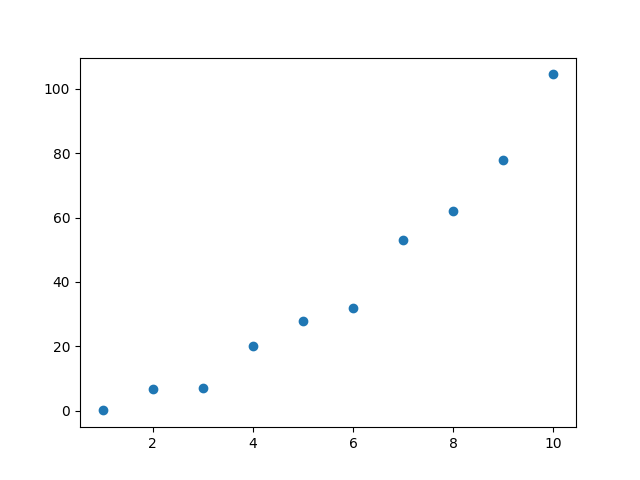
Il suffira de faire les calculer comme nous l’avons fait avec la régression linéaire pour mettre à jour les trois (3) poids. Apres avoir fait tous les calculs, nous pouvons avoir un tableau comme celui-ci

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X0 | X1 | X2 | W0 | W1 | W2 | Y |  | OU |
| 1 | 0 | 0 | -2.2121 | 5.41528 | 5.41528 | -2.2121 | 0.099 | 0 |
| 1 | 0 | 1 | 3.20318 | 0.961 | 1 |
| 1 | 1 | 0 | 3.20318 | 0.961 | 1 |
| 1 | 1 | 1 | 8.61846 | 0.999 | 1 |

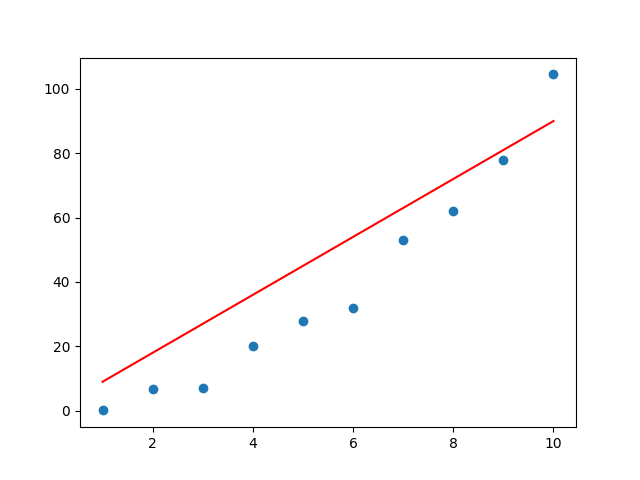
##### La régression polynomiale

Ci-après un tableau de valeur et sa représentation graphique.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Y | 0.1 | 6.9 | 7.2 | 20 | 28 | 32 | 53 | 62 | 78 | 104.5 |



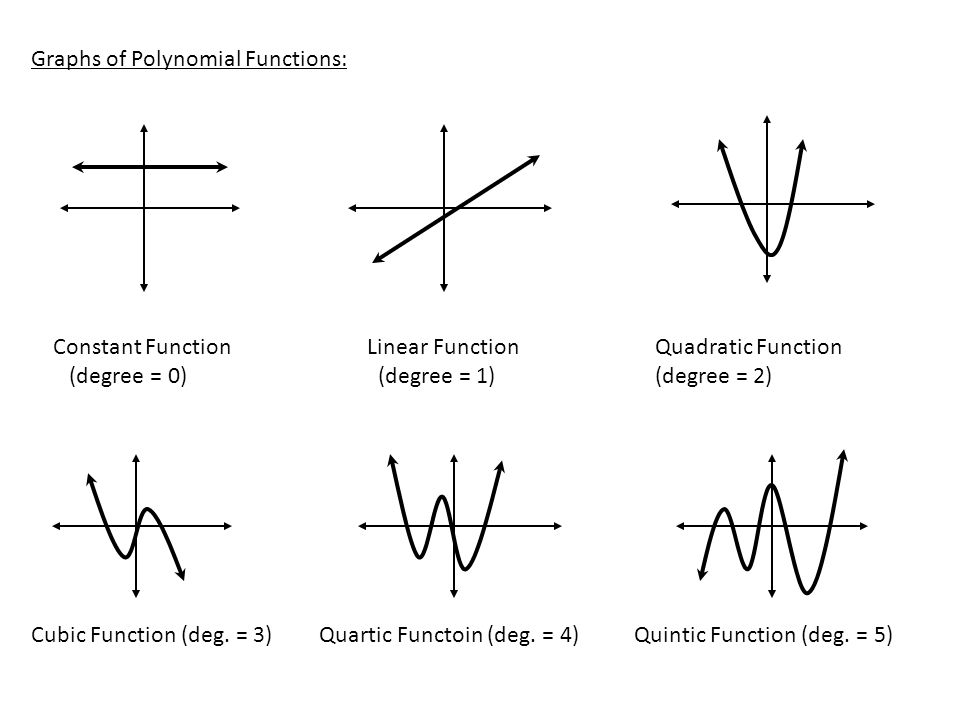
Apres avoir fait passer ces données dans un modèle de régression linaire, le résultat obtenu n’était pas a la hauteur de nos attentes.



Nous voyons que le modèle linéaire ne marche pas sur ces données. Conséquence, nous aurons besoin de quelque chose de plus sophistiquée, et ce quelque chose c’est la régression polynomiale, ce type de régression nous permet représenter une courbe des données qui adapte une forme exponentielle. Les étapes de régression polynomiale restent les même que les autres algorithmes mais ses calculs vont changer.

* Forward propogation

Pour le forward pass du polynomial régression, nous allons utiliser, une fonction quadratique, c’est-à-dire qui admet une puissance dans la variable. On va parler de dégrée de la fonction. Plus le dégrée est élevée plus la fonction pourra être mesure d’aller chercher des variations.



Du fait que nous n’avons pas beaucoup de variation dans le tableau, nous allons utiliser le deuxième degré, ainsi notre se présente comme suit :

A partie de la, on initialise les poids a 1. C’est le moment de préciser que l’initialisation des poids se fait forcement avec des uns (1). En vraie, on utilise dans fonction génération de nombre aléatoire, ici nous les initialisons a 1 par souci de simplicité.

Il y a erreur puisque pour la valeur x = 3, y = 7,2 donc nous allons calculer cette erreur.

* Calculer l’erreur

La fonction d’erreur ne change toujours pas, c’est le MSE.

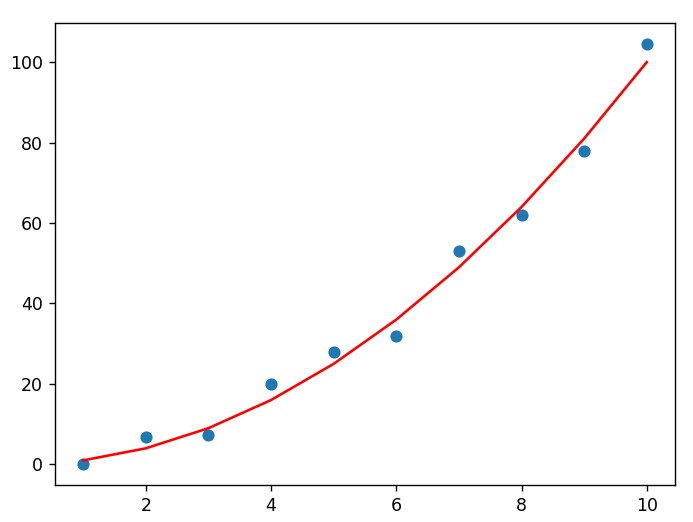
* Backpropagation

Il est observé une erreur de 17,64, nous allons par la suite retro-propagé cette erreur pour mettre à jour les poids.

La valeur de la dérivée partielle pour w1 se présente comme suit :

Nous avons décidé de prendre w2 car il a la dérivée partielle la plus complique à calculer, avec ce calcul établi, nous pouvons passer à l’étape des mises a jour des poids, il fait oublier de prendre un Learning rate.

Toutefois, qu’en est de notre problème initiale, après l’avoir fait passer dans un modèle de régression polynomiale, nous avons trouvé la courbe suivante.



C’est magnifique, le model est parvenu à trouver une corrélation a la presque perfection. Si nous avions plus variation de la courbe, il nous suffirait d’augmenter le degré et un plus de patience et c’est bon.

#### La classification

La classification est problème est qui est là depuis longtemps dans le domaine de l’intelligence artificielle. Les académiciens ont fait beaucoup de recherches sur le sujet et ont nous ont proposer un certain nombre de méthodes.

La classification a pour objectif de déterminer les éléments qui différentient les données dans une base données, ainsi ranger chacune dans sa classe de prédilection et aussi mais surtout prédire les classes pour données non-observées.

Différents algorithmes sont aujourd’hui là pour nous permettre de régler les problèmes de classification mais nous allons en voir trois (3).

##### Support Vector Machine (SVM)

Le SVM est modèle mathématique qui permet de classer des données en utilisant un séparateur. Ce séparateur peut une ligne dans un espace 2D ou un plan dans espace 3D, il est possible d’avoir autant de dimension que nécessaire mais le séparateur va une dimension n-1 par aux données. Comme tout algorithme, il y a un certain nombre d’étapes à suivre pour pouvoir réussir à implémenter un SVM.

* Le séparateur

Pour le problème linéairement séparable dans un espace 2D, nous avons besoin d’un séparateur (une droite) qui doit être aussi loin du point le plus proche d’une classe que du point le plus proche de l’autre classe. Ce séparateur se présente comme suit.

Si les données se présentent d’une manière non linéairement séparable, nous introduire une fonction de linéarisation appelée kernel (cette fonction ajoute une dimension aux données pour les rendre linéairement séparable). Dans ce cas on aura :

De là, nous nous retrouvons avec deux classes C1 et C2.

* Calcul de distance

Nous devons maintenant calculer la distance d des droites parallèles qui représentent les frontières.

* Calcul de l’erreur pour chaque point

Ici, la fonction de l’erreur va nous permettre de savoir dans quelle un point pris en particulier se trouve-il.

Exemples :

* Si mx + b = -1, y = -1, e = 1 – (-1) (-1) = 0 : pas erreur
* Si mx + b = -1, y = 1, e = 1 – (1) (-1) = 2 : il y a une erreur
* L’apprentissage

Le but de l’apprentissage en SVM, et tout modèle de machine Learning d’ailleurs, c’est minimiser l’erreur, en SVM il y a une fonction qui peut minimiser cette erreur.

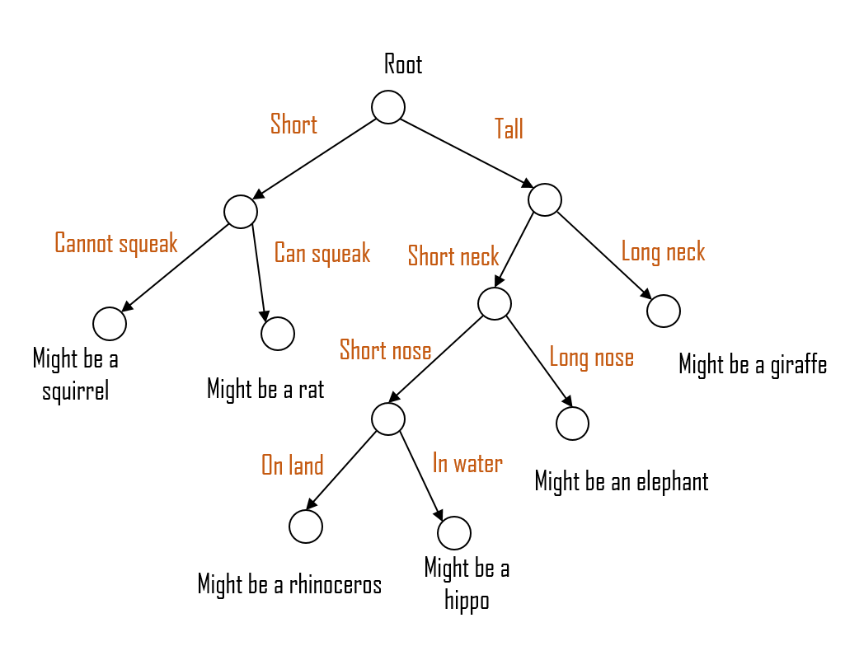
Il faut faire de tel sorte que l’erreur soir inferieur ou égale a 0, et nous savons de que :

Donc

Et nous y voilà, toutes les informations sur cette partie sur le SVM, nous viennent de l’article de S. Suthaharan : Support Vector Machine, chapitre 9.

##### L’arbre de décision

L’arbre de décision ou "decision tree" en anglais aussi méthode de classification avec concept qui lui est bien particulier. Comme son l’indique elle prend des décisions en se basant sur l’attribut des données. D’abord l’arbre vérifie l’attribut de plus indicatif et prends la direction d’une de ses valeurs, puis le deuxième attribut la significatif et prends la direction d’une de ses valeurs, ainsi de suite jusqu’à classer un nouvel enregistrement.



En revanche, le fait distinguer un attribut de plus significatif ne fait pas arbitrairement, si non ce ne serai pas une intelligence artificielle. Il y a un certain nombre de calcul (oui encore des maths) à faire pour trouver la bonne structure de l’arbre et nous allons les voir tout de suite.

Pour de faire prenons un exemple concret : nous tableau suivant nous informe sur l’individu est sénégalais ou pas à partie de trois (3) attributs, nous allons faire un arbre de décision.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numéro | Plat | Teint | Taille | Si sénégalais |
| 1 | Riz | Sombre | Grande | Oui |
| 2 | Attiéké | Claire | Petite | Non |
| 3 | Mafé | Sombre | Grande | Non |
| 4 | Riz | Sombre | Grande | Oui |
| 5 | Attiéké | Sombre | Petite | Non |
| 6 | Mafé | Claire | Grande | Oui |
| 7 | Riz | Sombre | Grande | Oui |

* Entropie

L’entropie nous renseigne sur la pureté d’un attribut, si deux classes ont équitablement représente dans un attribut, on dit le que nœud est impure, conséquence l’entropie est maximale (égale ou proche de 1), si une seule classe représentée le nœud est pure et l’entropie est minimale (égale ou porche de 0).

* Gain d’information (GI)

La première des choses à faire c’est de calculer le gain d’information c’est-à-dire pour chaque attribut quel est celui qui nous renseigne le plus si l’individu est sénégalais ou pas.

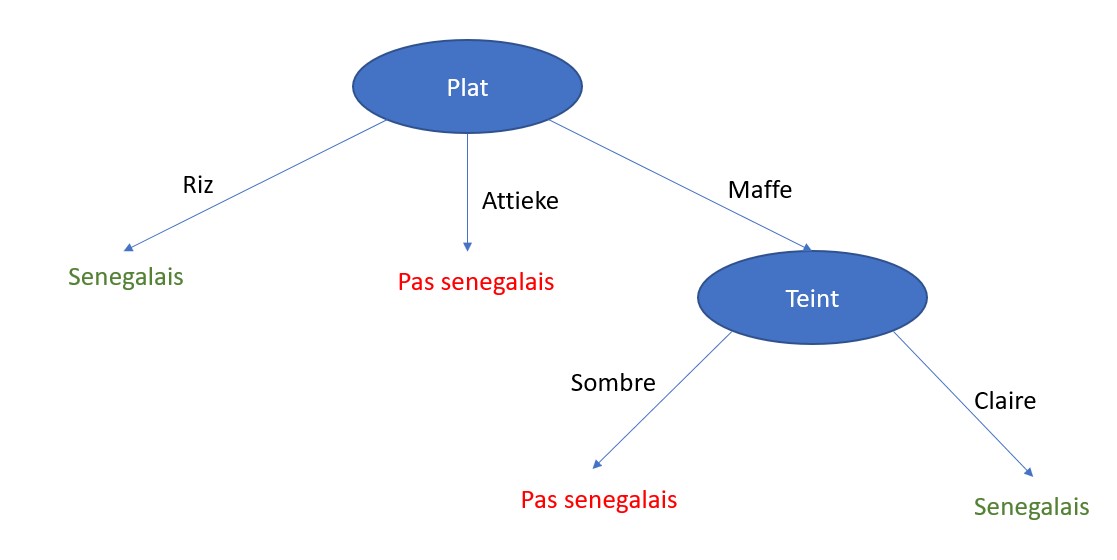
Pour ce qui est de notre exemple :

Calculons entropie générale

Gain d’information de l’attribut plat

Si nous répétons les calculs avec les attributs nous allons trouver que

Donc l’attribut plat a le plus grand gain d’information des lors, il devient l’attribut de plus indicatif et va être à la reine de l’arbre. Notre arbre ressemble à cela après avoir fait tous les calculs :



Nous constatons qu’il n’est nécessaire que tous les attributs soit présentes pour prendre une décision ici l’attribut n’intervient pas.

##### Naive Bayes

Le modèle de Naïve Bayes (NB) est algorithme de ML qui nous vient des statistiques et probabilité. Selon les cas il peut très puissant avec un mécanisme simple de calcul de probabilité. Il fonctionne en calculant les probabilités de toutes les valeurs d’attributs avec la variable cibles.

* Probabilité des variables cibles

Tout d’abord il faut calculer la probabilité de tous les variables cibles afin de savoir nos changes de tomber sur l’un ou l’autre (il est possible d’utiliser le NB dans multi-classe classification aussi).

* La probabilité conditionnelle des valeurs d’attribut

Pour chaque valeur d’attribut, il nous faut calculer sa probabilité conditionnelle par rapport aux valeurs cibles.

Cela semble peut mais on presque tout le travail, mais pratique il y aura beaucoup à faire. Maintenant nous classer un nouveau indivis en calculer sa probabilité de travail dans une classe ou une autre, ensuite nous allons normaliser les probabilités et classer dans celle qui la grande valeur.

Pour normaliser les probabilités :

Exemple : prenons le même exemple tu tableau qui essaie de prédire si l’individu est sénégalais ou pas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numéro | Plat | Teint | Taille | Si sénégalais |
| 1 | Riz | Sombre | Grande | Oui |
| 2 | Attiéké | Claire | Petite | Non |
| 3 | Mafé | Sombre | Grande | Non |
| 4 | Riz | Sombre | Grande | Oui |
| 5 | Attiéké | Sombre | Petite | Non |
| 6 | Mafé | Claire | Grande | Oui |
| 7 | Riz | Sombre | Grande | Oui |

Probabilité des valeurs cibles

Les probabilités des valeurs d’attributs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Plat | Oui | Non |  | Teint | Oui | Non |
| Riz |  |  | Sombre |  |  |
| Attiéké |  |  | Claire |  |  |
| Maffé |  |  |  | | |
|  | | | | | | |
| Taille | Oui | Non |  | | | |
| Grande |  |  |
| Petite |  |  |

Avec ce tableau nous avons tous ce qu’il nous faut pour classer un nouvel individu. D’ailleurs c’est ce que nous allons faire, classons I1 (Plat = riz, Teint = sombre, Taille = Grande) et I2 (Plat = Attiéké, Teint = claire, Taille = Petite).

I1 :

Le modèle nous dit que l’individus un sénégalais a 100% car la normalisation des probabilités va renvoyer 1 pour sénégalais et 0 pour non sénégalais.

Si on le fait pour l’individu 2, nous allons trouver (avec les mêmes calculs bien entendu), nous allons trouver que I2 n’est pas du tout sénégalais a 100% aussi. Cet exemple est simple mais dans la vie une le modèle va rarement répondre avec une confiance de 100%.

### Unsupervised learning

Pour ce qui est de l’apprentissage non supervisée, c’est qu’ici nous n’aurons pas output pour les inputs. Dans ce cas de figure nous aurons seulement des données d’entrées mais on ne sait comment réagir en conséquence. C’est le modèle qui va à lui seul de voir une représentation générale qui correspond le plus aux données qui lui sont présentées.

Pour ce faire il y a ce qu’on appelle de clustering : c’est un modèle dans lequel nous allons essayer de regrouper en cluster les individus qui se ressemble le plus en utilisant plusieurs variables qui décrive les données.

#### Clustering

Le clustering est une méthode d’apprentissage non supervisé dans lequel le but est de ressemble des individus qui se ressemble le plus. Le principe est simple, nous avons des données mais qui ne sont étiquetés, donc c’est au modèle de trouver la représentation la plus fidèle des données. Il y a plusieurs algorithmes de clustering mais nous allons voir le fameux k-means (k le de k-means représente de nombre de classe ou cluster).

* Définir le nombre de cluster

En premier lieu, il définir le nombre de cluster, ce choix peut relever du libre arbitre de l’ingénieur ou peut être définit en fonction de méthodes.

* Le centre de gravite

Pour chaque cluster il faut calculer son centre de gravite et on affecte chaque point de la base données a la classe la plus proche. De là, tous les individus appartiennent à une classe et c’est que le travail commence.

* Calcul de distance

Maintenant, nous allons calculer toutes les distances de tous les individus par rapport a tous centre de gravite de chaque cluster. Nous allons nous apercevoir que certains individus sont mal classe, car ils sont plus proche d’un autre cluster que celui ou ils sont, il suffit de les mettre à jour les classe. Cette étape va être répéter autant de fois que nécessaire pour avoir des clusters les plus représentatifs dans données que possible.

Exemple : prenons le tableau suivant :

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 |
| X | 1 | 1 | 2 | 5 | 5 | 6 | 1 | 1 | 2 |
| Y | 1 | 2 | 1 | 5 | 6 | 5 | 9 | 10 | 9 |

Nous prenons 3 pour la valeur de k, donc nous aurons 3 clusters.

P1, P2, P3 sont choisi et tous les autres vont se classer par rapport au point le plus proche de ses trois. Nous nous retrouvons avec 3 clusters que se présentent comme suit :

C1 = {P1},

C2 = {P2},

C3 = {P3, P4, P5, P6, P7, P8, P9}.

Calculons les centres de gravité Cg, c’est le point qui représente la moyenne des X et moyenne de Y.

Cg(C1) = (1, 1), Cg(C2) = (1, 2), Cg(C3) = (3.42, 6.42)

Calculons les distances d pour de tous les points par rapport au centre de gravité, nous allons utiliser la distance euclidienne dans espace 2D.

Apres calcul, nous avons trouver le tableau suivant des points et leur distance par rapport au centre de gravité.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 |
| C1 | 0.0 | 1.0 | 1.0 | 5.66 | 6.40 | 6.40 | 8.0 | 9.0 | 8.06 |
| C2 | 1.0 | 0.0 | 1.41 | 5.0 | 5.66 | 5.83 | 7.0 | 8.0 | 7.07 |
| C3 | 5.94 | 5.04 | 5.60 | 2.12 | 1.63 | 2.94 | 3.54 | 4.21 | 2.94 |

Le constat qui sera de ce tableau c’est qu’il y aura du mouvement, et nos clusters deviennent :

C1 = {P1, P2},

C2 = {P3},

C3 = {P4, P5, P6, P7, P8, P9}.

Il ne reste plus qu’a faire la même chose, à savoir recalculer des centres de gravité, recalculer les distances et mettre à jour les clusters. Si nous le faisons assez fois, nous des clusters qui pourrons prédire la classe d’un nouvel individu.

#### Règles d’associations

Les règles d’association ou en anglais "association rules mining" sont des méthodes non supervisées qui nous permet la corrélation entre uns donnée et les autres. Ses règles permettent de répondre des questions comme : dans quelle mesure B et C vont apparaitre sachant que A est apparu ? Ces calculs vont se faire un ensemble de sous-ensemble, nous allons parler ici de itemset pour designer les sous ensemble. La règles association sont très fréquent dans les marche et supermarché pour desceller les produits qui sont souvent acheter en temps par les clients. Une fois que nous avons des règles intéressants les dirigeants peuvent prendre de bonne décision.

* Les données

Une transaction T est considérée comme une itemset, c’est tous les produits qu’un client acheté un coup. Nous pouvons avoir quelque chose comme cela.

T1 = {A, B, C},

T2 = {E, F},

T3 = {A, C, E},

T4 = {A, E,}

T5 = {B}.

Nous avons ici cinq (5) transactions avec leurs produits associées. Notre tache est de trouver le rapport entre l’achat des produits. Si un client achète un produit, dans quelle mesure il achète un autre produit.

* Le support d’un produit

Le support d’un produit, c’est l’occurrence d’un produit sur dans la transaction sur le nombre de transaction.

C’est le lieu de parler du support minimum (minsup), c’est support qu’un produit ou ensemble de produit doit avoir pour c’est rester dans la recherche de règles. Ce minsup est choisi arbitrairement selon le problème posé. Delà, nous avons :

* L’élagage

Nous voyons le produits F n’a qu’un support de 1/5, donc ne peut pas participer dans une règle intéressante, il va être élagué. Il nous reste {A, B, C, E}.

* La jointure

Cela commence à devenir intéressante, car maintenant nous allons joindre les produits restants et calculer le support des résultats {AB, AC, AE, BC, BE, CE}.

Apres élagage, nous allons nous retrouver avec {AC, AE}.

Une jointure possible et c’est {ACE} qui a un support de 1/5 donc qui va être élagué.

L’ensemble des itemset fréquent est donc {A, B, C, E, AC, AE}, avec cela nous pouvons enfin calculer les règles intéressantes. Une règle est dite intéressante si support est supérieur ou égale à minsup et sans confiance supérieur ou égale a minconf (la confiance minimum).

Pour {AC}, calculons-la les règles intéressante avec un minconf de 4/5.

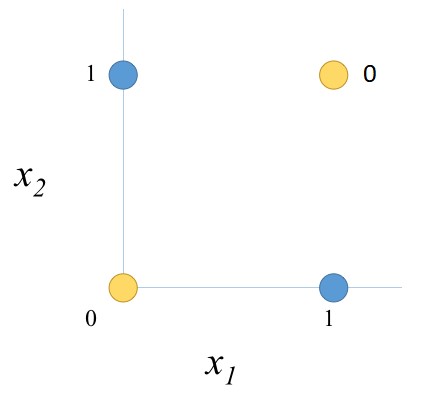
A -> C, conf = 2/3, sup = 2/5 : cette règle n’est pas intéressante car son support > minsup et sa confiance < minconf.

C -> A, conf = 1, sup = 2/5 : cette règle est intéressante car son support > minsup et sa confiance > minconf.

Conclusion : nous avons pour les {AC}, si on achète A il n’est pas forcé que l’achète C, mais si on achète C, il y a de forte chance que l’on A et quand on regarde les transactions, ces conclusions reflètent la réalité. Si je travaille dans cette boutique, je mettrai les produits A à coté des C.

## Deep Learning

Frank Rosenblatt a créé le perceptron qui nous a permis de résoudre de problème, notamment le OU et le ET logique, mais quand ils l’ont essaié pour le XOR, ils se sont rendu compte que le perceptron ne convergeait pas. Le problème était simple, un perceptron traçait des séparateurs linéaires, or ce n’était pas possible pour le problème du XOR.



Allez-y ! essayer de tracer une seule droite qui est capable de séparer les 0 et le 1, une droite ce n’est pas possible. Bienvenue dans le monde du non linéaire, un monde que fut un casse-tête pour les chercheurs pendant le longtemps, jusqu’à qu’ils découvrent les solutions qui vont être présenter.

### Artificial neuron network (ANN)

### Convolutional neuron network (CNN)

### Recurrent neuron network (RNN)

## Reinforcement Learning

# Conclusion

# Bibliographie

Dekking, F. M. (2005). *A Modern Introduction to Probability and Statistics: Understanding why and how*. Springer Science & Business Media.

Suthaharan, S. (2016). Support Vector Machine. Integrated Series in Information Systems, 207–235. doi:10.1007/978-1-4899-7641-3\_9

# Webographie

*Statistiques inférentielles : Définition, types et exemples*. (2024, 5 6). Retrieved from Question Pro: https://www.questionpro.com/blog/fr/statistiques-inferentielles/

*Algèbre linéaire - Définition*. (2024, 5 6). Retrieved from Techno-Science: https://www.techno-science.net/definition/5080.html