# 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.

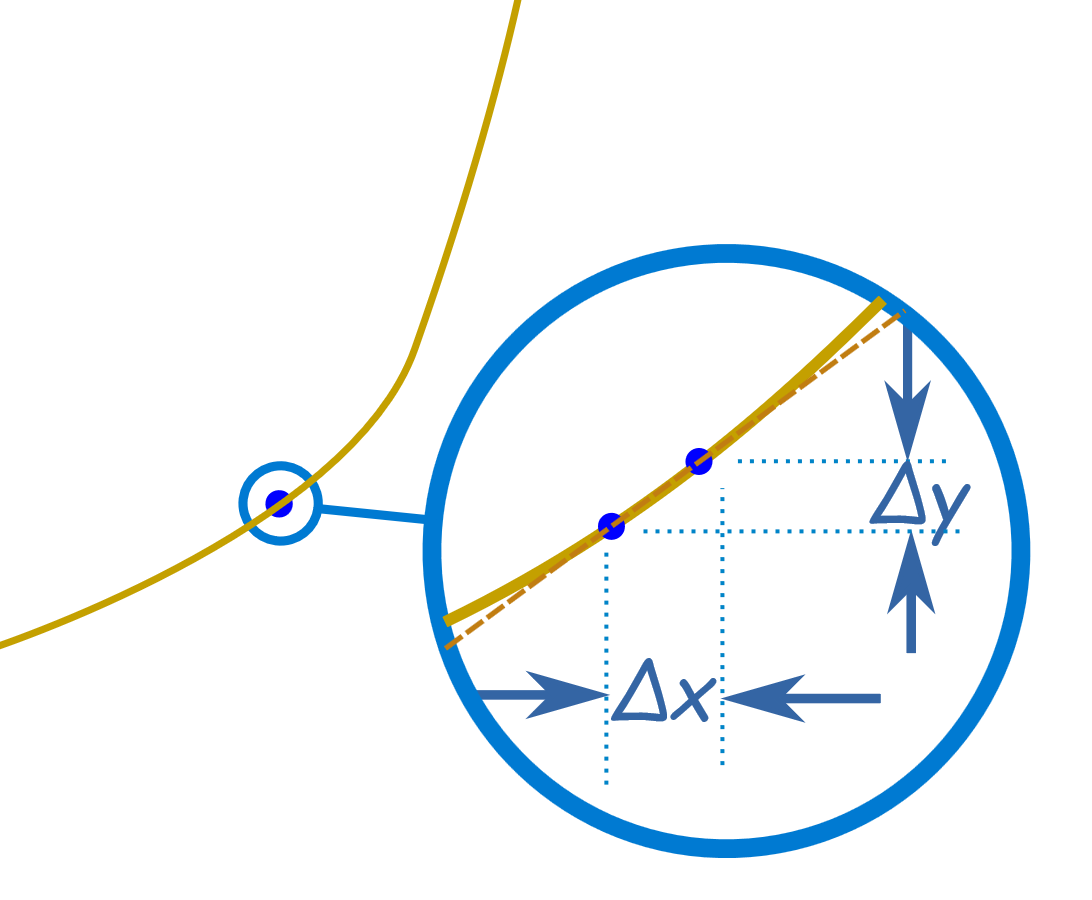
L’algèbre linéaire est la branche des mathématiques qui s'intéresse à l'étude des espaces vectoriels (ou espaces linéaires), de leurs éléments les vecteurs, des transformations linéaires et des systèmes d'équations linéaires (théorie des matrices). (Algèbre linéaire - Définition, 2024)

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.

Plut 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 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 les quelles 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 forcement 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 pris 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 parler 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.

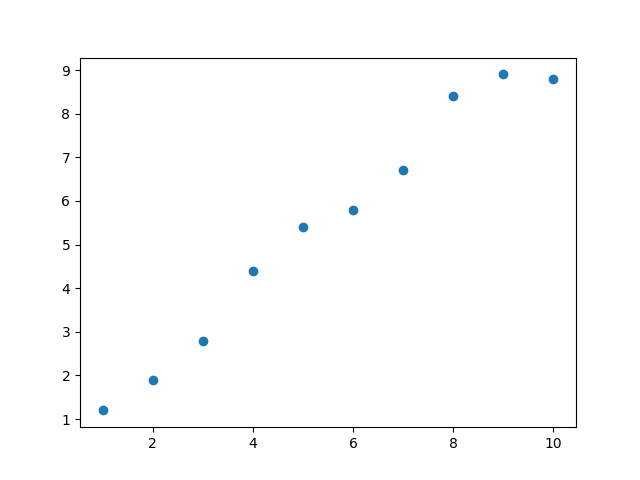
#### La régression

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.

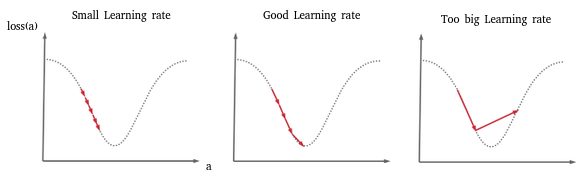
Attention ! il ne faut oublier que de la même manière que l’on a calculer pour x = 1, il faut aussi le calculer pour tous les autres x et ainsi avoir toutes les erreurs pour pouvoir l’appliquer formule générale.

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 a 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 ».



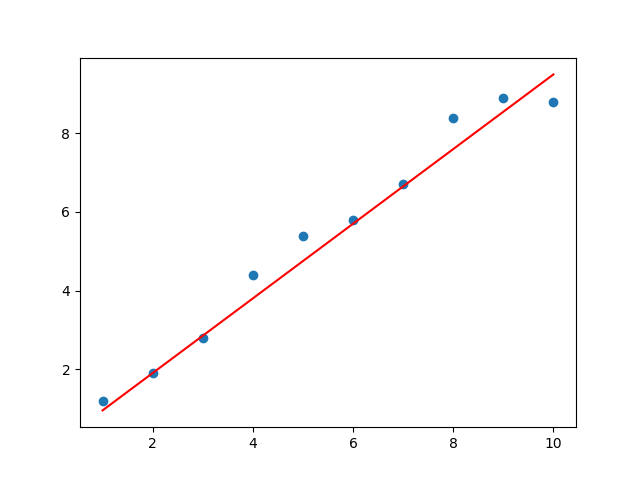
Représente quant a elle, la dérivée de la fonction MSE par rapport au poids concerné, c’est ce qu’on appelle une dérivée partielle.

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

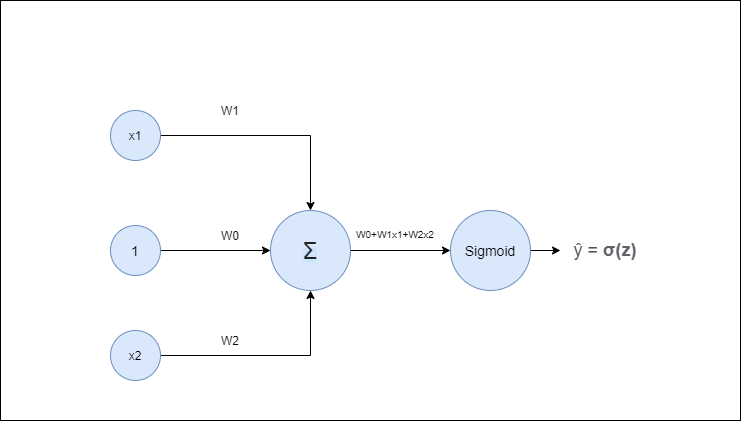
La régression logistique, contrairement à celle dite linéaire, n’a pas pour vocation prédire une future. Sa prédiction est de type binaire : oui ou non, bon ou mauvais, 0 ou 1 etc. Ce genre va s’avérer être très important dans beaucoup de domaines, nous l’utilisons dans nos vies de tous les jours sans 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



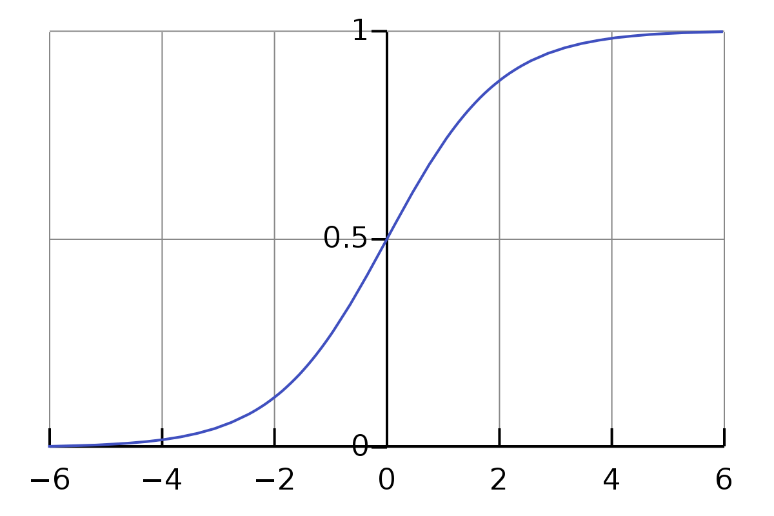
Voici a quoi va ressemble notre réseau de neurone, on va un autre input en plus x1 et x2, c’est le biais qui va toujours être égale a 1, son utilité est d’éviter que certains ne meurt durant l’entrainement si x1 = 0 et x2 = 0.

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 donne qu’on introduit une nouvelle fonction, celle d’action, nous allons de facto nous retrouver avec trois membres dans le calcul de dérivée partielle.

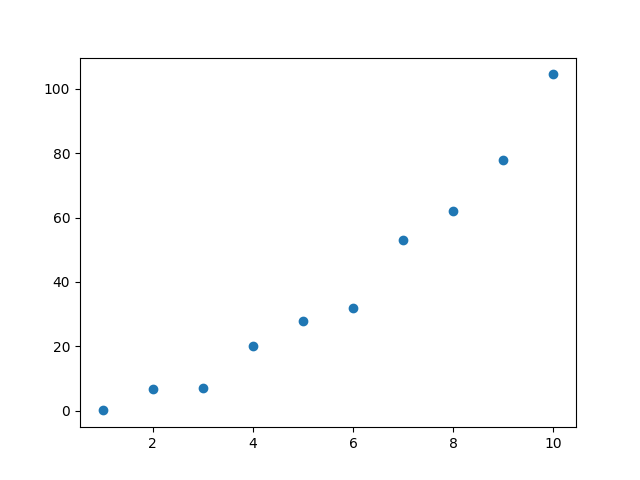
Il suffira de faire les calculer comme nous l’avons fait avec la régression linéaire pour mettre à jour les trois (3) poids. Apres avoir fait tous les calculs, nous pouvons avoir un tableau comme celui-ci

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

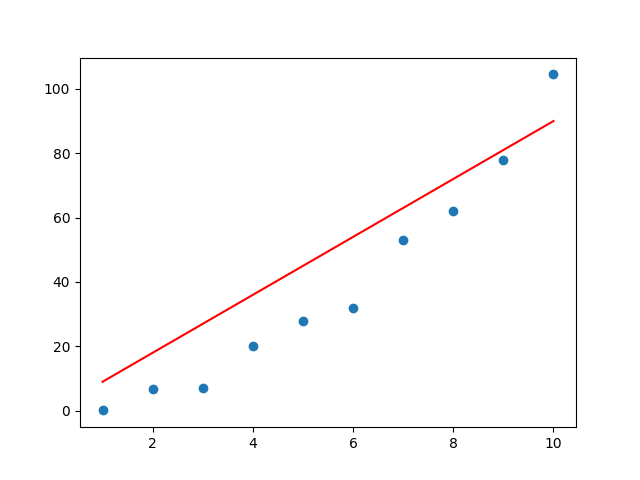
##### La régression polynomiale

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 |



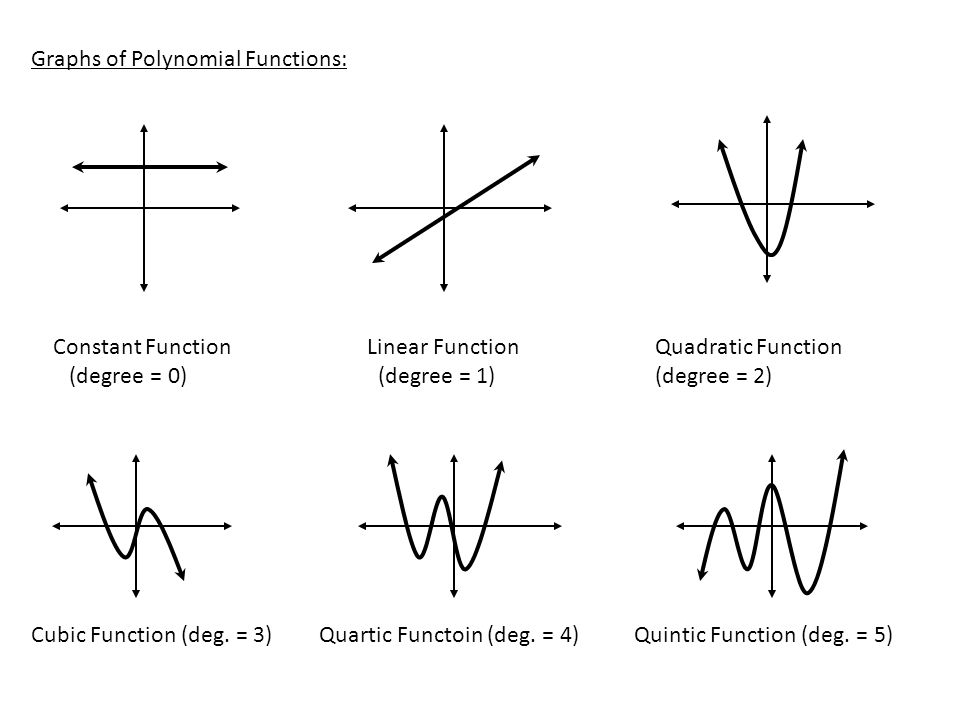
Apres avoir fait passer ces données dans un model de régression linaire, le résultat obtenu n’était pas a la hauteur de nos attentes.



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 etre 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.

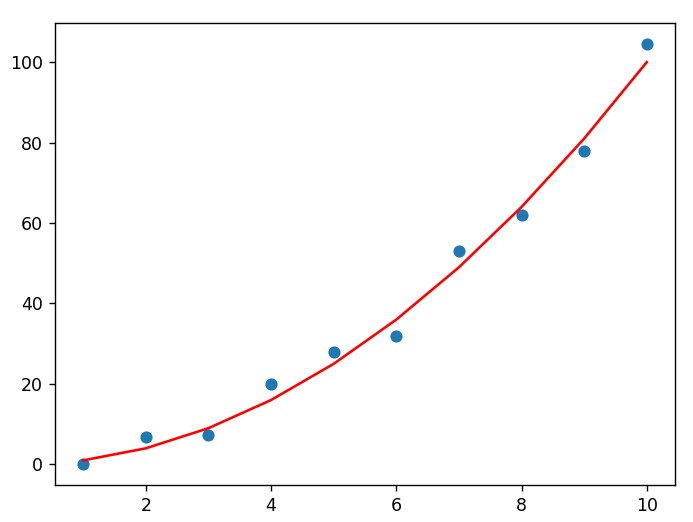
* Backpropagation

Il est observé une erreur de 17,64, nous allons par la suite retropropage cette erreur pour mettre a jour les poids.

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

Nous avons décidé de prendre w2 car c’est il a la dérivée partielle la plus complique à calculer, avec ce calcul établi, nous pouvons passer a l’étape des mise 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

##### Support Vector Machine (SVM)

##### L’arbre de décision

##### Naive bayes

### 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

#### Règles d’associations

#### Reduction de dimensions

## Deep Learning

### Artificial neuron network (ANN)

### Convolutional neuron network (CNN)

### Recurrent neuron network (RNN)

## Renforcement Learning

# Conclusion

# Bibliographie

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

# 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