

Université Gaston Berger de Saint Louis

UFR : SAT/SEG

Département : Informatique/Gestion

Filière : Méthode Informatiques Appliques a la Gestion (MIAGE)

Année : 2022/2023

Etudiant : Omar Abd Al Wahab DIASSE

Sous la direction de : Pr Jean Marie DEMBELE

**Théories derrières les algorithmes d’IA**

Sujet : Développement de modèles de Machine Learning pour faire une analyse prédictive des finances d’une entreprise

Plan

[Introduction de chapitre 2](#_Toc166766273)

[1 Les prérequis 2](#_Toc166766274)

[1.1 Les mathématiques 3](#_Toc166766275)

[1.1.1 Les statistiques et probabilités 3](#_Toc166766276)

[1.1.2 L’algèbre linéaire 4](#_Toc166766277)

[1.1.3 L’analyse 4](#_Toc166766278)

[1.2 L’informatique 5](#_Toc166766279)

[1.2.1 L’algorithme 6](#_Toc166766280)

[1.2.2 Les structure de données 7](#_Toc166766281)

[1.2.3 Les langages de programmation 7](#_Toc166766282)

[1.3 L’intelligence sociale 8](#_Toc166766283)

[2 Les algorithmes d’intelligence artificielle 8](#_Toc166766284)

[2.1 Machine Learning 8](#_Toc166766285)

[2.1.1 Supervised learning 9](#_Toc166766286)

[2.1.2 Unsupervised learning 26](#_Toc166766287)

[2.2 Deep Learning 31](#_Toc166766288)

[2.2.1 Artificial neuron network (ANN) 31](#_Toc166766289)

[2.2.2 Convolutional neuron network (CNN) 34](#_Toc166766290)

[2.2.3 Recurrent neuron network (RNN) 36](#_Toc166766291)

[2.3 Reinforcement Learning 38](#_Toc166766292)

[Conclusion partielle 39](#_Toc166766293)

[Bibliographie 39](#_Toc166766294)

[Webographie 40](#_Toc166766295)

# Introduction de chapitre

Le domaine de l’IA est en train de suivre un boom littéral. Nous voyons de plus en plus des applications utilisant l’IA pour nous faciliter la vie et aussi amuser la galerie. Mais de l’autre côté, beaucoup d’étudiant et de chercheurs mènent leurs recherches sur ce domaine passionnant. Maintenant, pour tout étudiant, ou ingénieur qui se respecte, avant de produire ou écrire sur l’IA, il faut comprendre l’algorithme derrière. Car oui, aujourd’hui il y a mille et une outils nous permettant de créer des programmes intelligents, mais c’est toujours intéressant de comprendre ce qu’il y a derrière chaque fonction utilisée.

Pour faire simple les algorithmes d’IA ne sont ni plus ni moins que des mathématiques, nous avons déjà dits que le soubassement de l’IA c’est les mathématiques. Nous allons voir la magie derrière les familles de modèles et comment elles fonctionnent.

Avant de commencer il faut préciser que les calculs mathématiques que nous allons voir sont quelque peu complexes et couteux.

Pour ce qui est du plan nous allons voir d’abord les prérequis à savoir les mathématiques, l’informatique et d’autres éléments très intéressants. Après cela, nous allons entrer dans le vif du sujet à savoir comment fonctionnent les modèles intelligents.

# 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 quelques autres prérequis qui ne sont pas nécessaires mais peuvent aider dans notre objectif, et tous ses autres prérequis vont être classés dans le domaine de l’intelligence sociale.

## Les mathématiques

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

Les mathématiques peuvent être compliquées, mais quand on lui trouve une application c’est là que ça devient intéressant et l’une des plus belles applications des mathématiques c’est l’IA. Nous allons vous montrer 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 facilités avec un ordinateur, et c’est là que va être utile l’ordinateur, l’informatique et l’algorithme.

En dépit du fait qu’il y a plusieurs domaines mathématiques qui nous serons utiles 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 s’avérer être plus intéressante.

### 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. La statistique est un domaine 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 appelle les statistiques descriptives. Il y a aussi les statistiques inférentielles qui, comme son nom l’indique, vont nous permettre de faire des inférences c’est-à-dire faire des estimations. Et c’est là que réside le lien entre les probabilités et statistique car les statistiques inférentielles vont avoir besoin des 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 de modèle d’IA.

* **Avant le développement du modèle** : les statistiques nous aident à comprendre les données, car très souvent les données brutes ne sont pas exploitables. Ici, nous vérifions le maximum des valeurs, le minimum, la moyenne, les outliers et l’une partie des plus importantes du « Feature Engineering » la mise à l’échelle etc.
* **Apres le développement du 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 une fois déployer. Nous pouvons aussi faire des graphs comme la matrice de confusion par exemple.

### L’algèbre linéaire

Au fait, il y a trois grandes parties dans le développement d’un réseau de neurones et à titre illustratif, nous pouvons dire qu’il y a le travail a posteriori, le développement du modèle et travail a priori. Pour le modèle il y a deux parties le Feed-forward et le Back-propagation et L’algèbre linéaire vont intervenir dans ces deux parties.

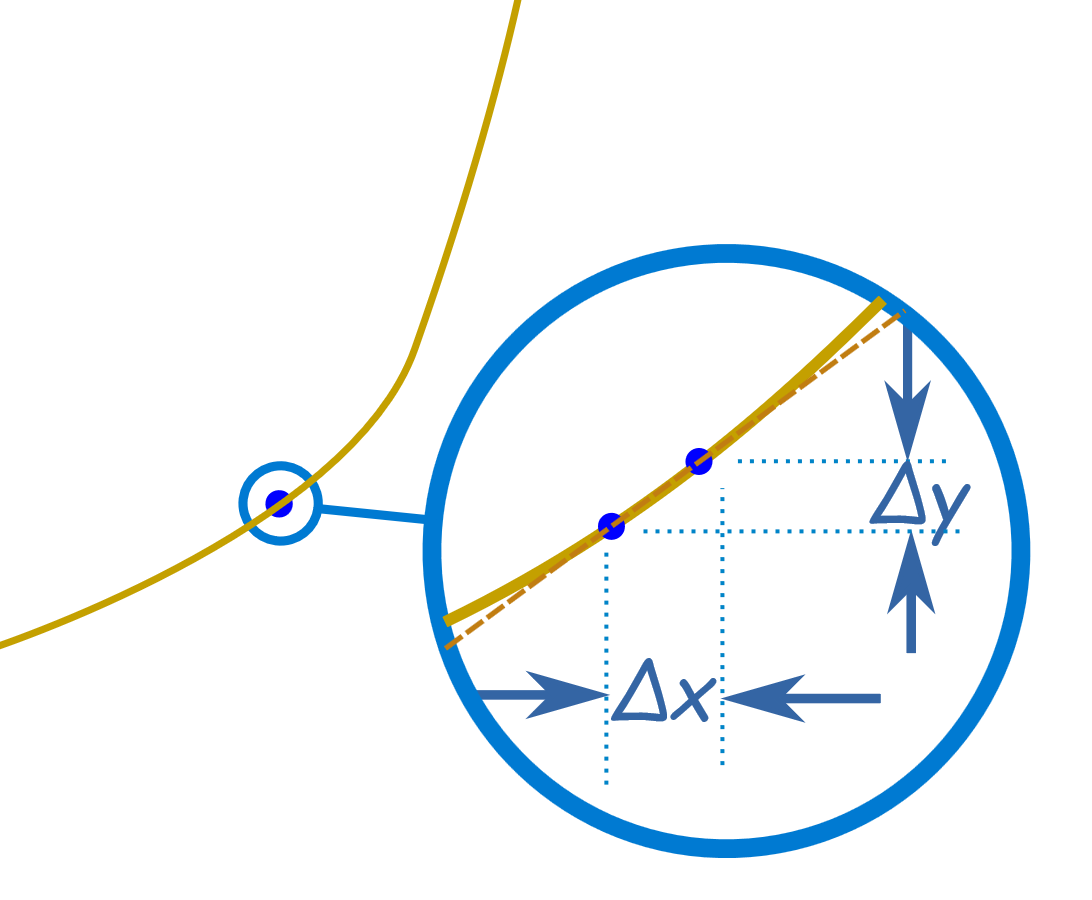
L’algèbre linéaire est la branche des mathématiques qui s'intéresse à l'étude des espaces vectoriels (ou espaces linéaires), de leurs éléments les vecteurs, des transformations linéaires et des systèmes d'équations linéaires (théorie des matrices). (Algèbre linéaire - Définition, 2024)

Ainsi, la plus grande utilité de l’algèbre linéaire est le calcul de poids, elle va nous permettre d’automatiser les calculs 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 neurones 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 à cause des calculs d’analyse mathématique.

La dérivée d’une fonction nous informe sur la variation de la fonction en un point. Pour être plus claire, elle nous permet de calculer la pente sur n’importe quel point de la fonction. Le calcule de dérivée est très important dans beaucoup de domaines 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 ses 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 d’erreur va nous permettre de rectifier cette erreur.

De manière pratique, on calcule l’erreur en premier lieu, puis on calcule la dérivée de la fonction d’erreur. La manière dont la rectification va se faire, c’est qu’on va donner à chacun des poids une valeur correspondante à sa responsabilité dans l’erreur et c’est ça le Back-propagation ou la rétropropagation en français.

## L’informatique

L’informatique c’est la science de l’automatisation de l’information, d’ailleurs son nom vient de là : une contraction entre information et automatique. Chez les anglo-saxons, ils parlent plutôt de de Computer Science qui se traduit littéralement par science de l’ordinateur.

Plus haut, nous avions attesté que l’IA est une science purement mathématique avec que des calculs que l’on pourrait même faire sur feuille. Dès lors, que représente l’informatique pour l’IA : il a le rôle d’une calculatrice géante capable de faire des super-calculs en un temps record. Ajouté à cela, l’informatique nous permet en outre de présenter les résultats dans une interface graphique pour que n’importe qui 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 voir 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 à 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 et savoir quelles étapes à suivre, sinon beaucoup de frustration nous attend.

Exemple : écrivons un algorithme qui résout un polynôme du second degré :

* 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 x2 = (-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 peut nous permettre d’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 n’avions pas directement la façon optimale de gestion de données.

Ces structures peuvent partir d’un simple tableau dans un langage de programmation jusqu’à atteindre les graphs (structure de données complexe et très puissante). Comme nous l’avons dit et redit l’IA travaille sur des données. Citons 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 aussi appelés langages informatiques sont les syntaxes qui traduisent les algorithmes d’une manière compréhensible à 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 sa 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 tout 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 mais qui ne sont pas forcément liés à notre science mais qui apportent une plus-value importante, nous allons les classer dans l’intelligence sociale.

* Repérage de problèmes

Celui ou celle qui prétend à travailler dans l’IA doit à tout prix être en mesure de repérer les problèmes auxquels est confrontée 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 de chercher bien loin, des problèmes sont trouvables partout, il suffit juste de bien observer son environnement.

* La résolution de problèmes

Apres avoir descellé de potentiels problèmes, il va falloir naturellement proposer des solutions. Il y a plusieurs modèles et algorithmes d’IA qui excellent dans différant domaines, modèles et algorithmes que nous allons voir dans la partie suivante. Il faut noter aussi que c’est le problème qui définit le modèle utilisé mais pas l’inverse.

* La pédagogie

Pourquoi la pédagogie ? même si 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 importantes de ce travail de mémoire. Nous avons parlé de l’IA dans ce document mais cette fois nous allons voir comment elle fonctionne en parcourant différant des plus importants algorithmes d’IA, ceux qui sont vraiment utilisés 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 tête, mais en regardant notre environnement et en imitant nos parents, nous apprenons. Ce processus peut être répliqué sur un ordinateur, c’est le 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 données va être étiquetée, 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.

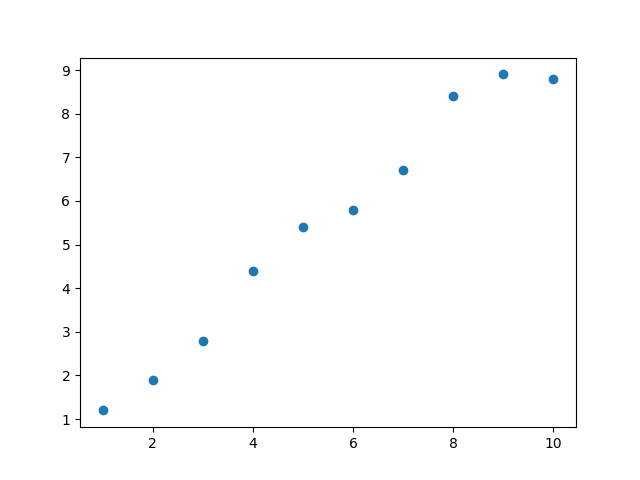
#### La régression

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

##### La régression linéaire

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, qu’allons-nous faire. Il y a la méthode des moindres carrés, élaboré par le légendaire Carl Friedrich Gauss, qui est une méthode purement statistique mais nous 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 devient . Le but du jeu est de trouver les w0 et w1 qui vont au mieux correspondre à nos points. Dans un premier temps on va les donner 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 ce fait, il faut calculer l’erreur, nous allons utiliser 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 essai.

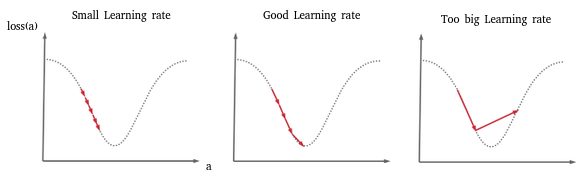
Attention ! il ne faut pas oublier que de la même manière que l’on a calculer pour x = 1, il faut aussi le calculer pour tous les autres x et ainsi avoir toutes les erreurs pour pouvoir appliquer la formule générale.

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 où nous allons utiliser l’algorithme de la descente des gradients.

* Back-propagation (rétropropagation qui met à jour les poids)

Maintenant que nous avons l’erreur nous pouvons enfin mettre à jour nos poids w0 et w1. Cela veut dire que chacun va prendre une part de l’erreur qui est égale à sa responsabilité de cette dernière et se rectifier lui-même. Pour se faire nous allons calculer la dérivée de toutes les fonctions qui nous ont mené 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 être lent et si le pas est trop grand, nous allons dépasser le point qui minimise l’erreur, ce pas c’est le « lr » dans les deux fonctions cela signifie « Learning Rate ».



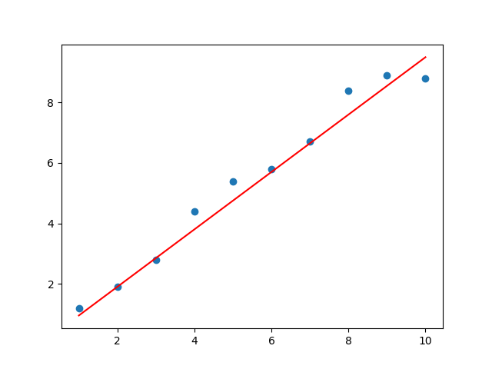
Représente quant à elle, la dérivée de la fonction MSE par rapport au poids concerné, c’est ce qu’on appelle une dérivée partielle.

Exemple :

Donc

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

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



##### La régression logistique

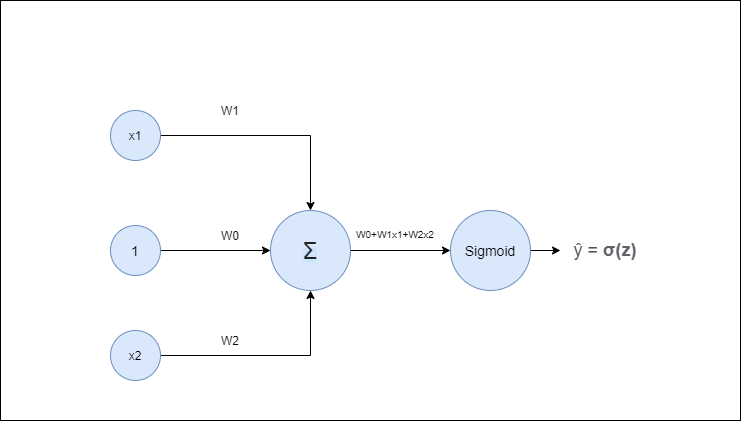
La régression logistique, contrairement à celle dite linéaire, n’a pas pour vocation de prédire une valeur future. Sa prédiction est du type binaire : oui ou non, bon ou mauvais, 0 ou 1 etc. Ceci va s’avérer être très important dans beaucoup de domaines, nous l’utilisons dans nos vies de tous les jours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



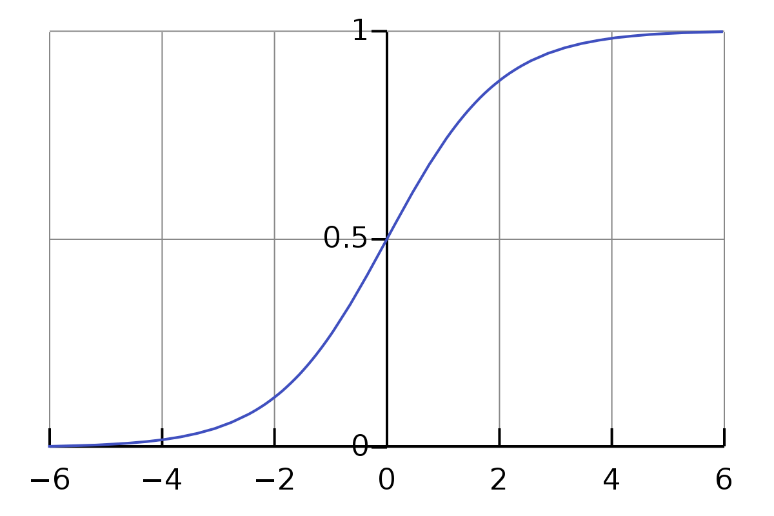
Voici à quoi va ressembler notre réseau de neurones, on va ajouter un autre input en plus x1 et x2, c’est le biais qui va toujours être égale à 1, son utilité est d’éviter que certains neurones ne meurent durant l’entrainement si x1 = 0 et x2 = 0.

Commençons par initialiser

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

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

Nous allons 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 à jour trois poids à savoir w0, w1, w2.

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

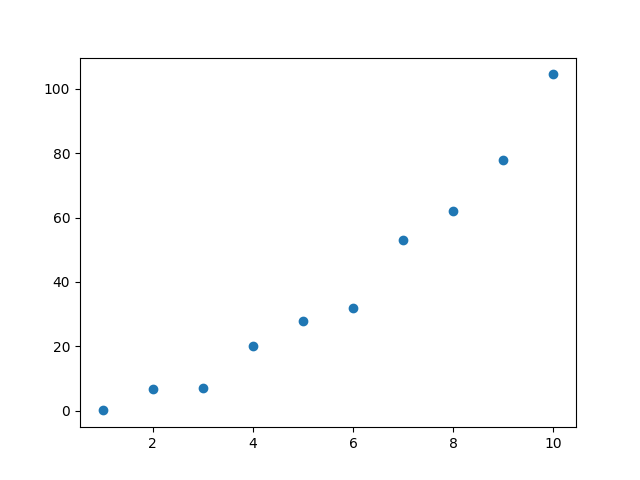
Il suffira de faire les calculs comme nous l’avons fait avec la régression linéaire pour mettre à jour les trois (3) poids. Apres avoir fait tous les calculs, nous allons avoir un tableau comme celui-ci

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

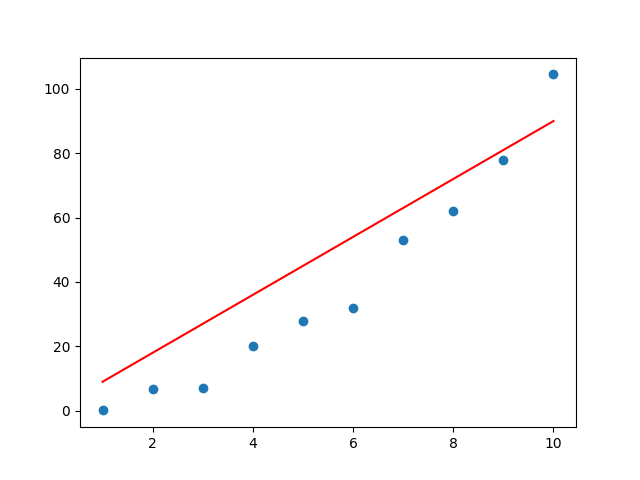
##### La régression polynomiale

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 |



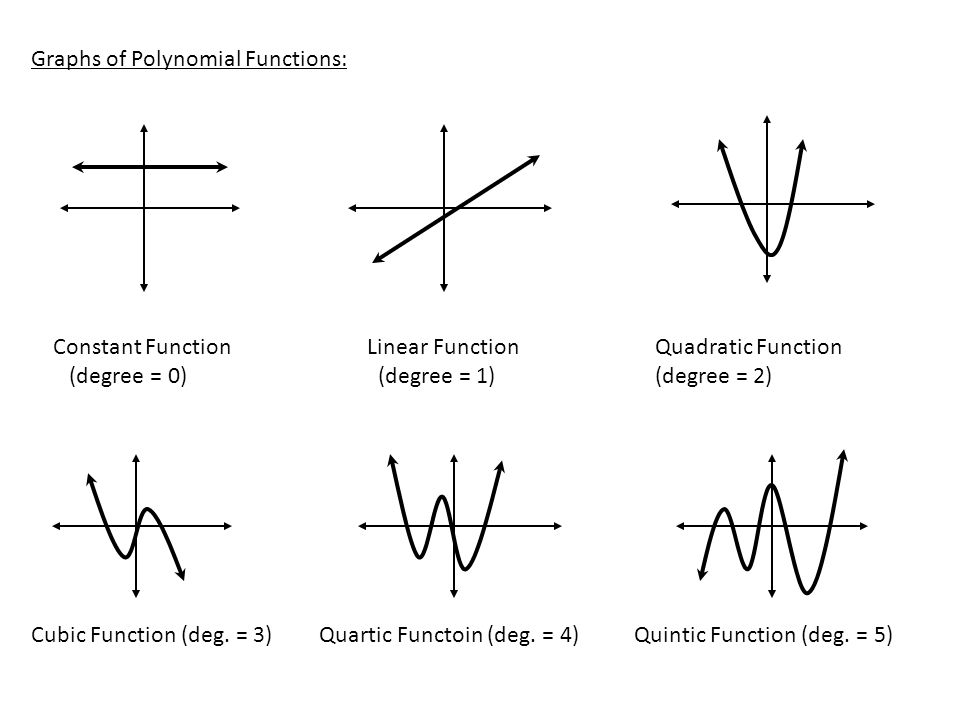
Apres avoir fait passer ces données dans un modèle de régression linéaire, le résultat obtenu n’était pas à la hauteur de nos attentes.



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 de représenter une courbe de donnée qui adapte une forme exponentielle. Les étapes de régression polynomiale restent les mêmes que les autres algorithmes mais ses calculs vont changer.

* Forward-propogation

Pour le Forward-pass de la régression polynomiale, nous allons utiliser, une fonction quadratique, c’est-à-dire qui admet une puissance dans la variable. On va parler de degré de la fonction. Plus le degré est élevé plus la fonction pourra être en 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 formule se présente comme suit :

A partir de là, on initialise les poids à 1. C’est le moment de préciser que l’initialisation des poids ne se fait pas forcément avec des uns (1). En vraie, on utilise une fonction de génération de nombres aléatoires, ici nous les initialisons à 1 par souci de simplicité.

Il y a une 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.

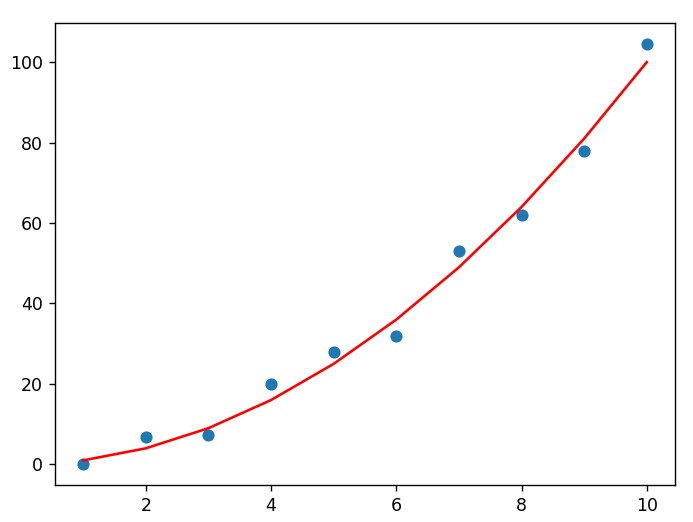
* Backpropagation

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 compliqué à calculer, avec ce calcul établi, nous pouvons passer à l’étape des mises à jour des poids, ne pas oublier de prendre un Learning Rate.

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



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

#### La classification

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

La classification a pour objectif de déterminer les éléments qui différencient les données dans une base de données, ainsi ranger chacune dans sa classe de prédilection et aussi mais surtout prédire les classes pour des données non encore 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 un modèle mathématique qui permet de classer des données en utilisant un séparateur. Ce séparateur peut être une ligne dans un espace 2D ou un plan dans un espace 3D, il est possible d’avoir autant de dimension que nécessaire mais le séparateur va avoir 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 un 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 allons introduire une fonction de linéarisation appelée kernel (cette fonction ajoute une dimension aux données pour les rendre linéairement séparables). 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 classe 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 pour tout modèle de machine Learning d’ailleurs, c’est minimiser l’erreur, en SVM il y a une fonction qui peut minimiser cette erreur.

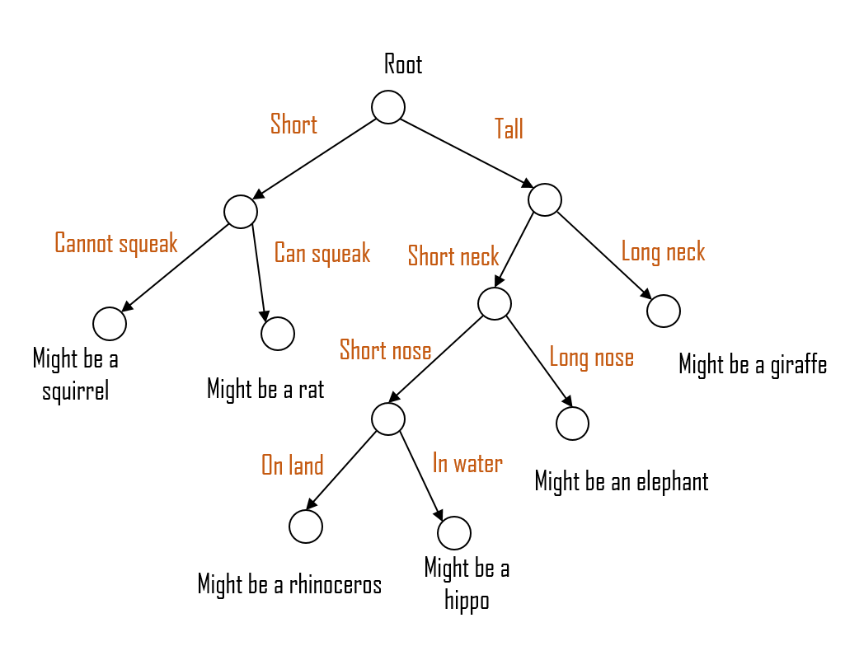
Il faut faire de telle sorte que l’erreur soit inférieure ou égale à 0, et nous savons que :

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



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

Pour se faire prenons un exemple concret : ce tableau suivant nous informe si l’individu est sénégalais ou pas à partir 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 sont équitablement représentées dans un attribut, on dit que le nœud est impur, conséquence l’entropie est maximale (égale ou proche de 1), si une seule classe est 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 de tous les attributs, quel est celui qui nous renseigne le plus si l’individu est sénégalais ou pas.

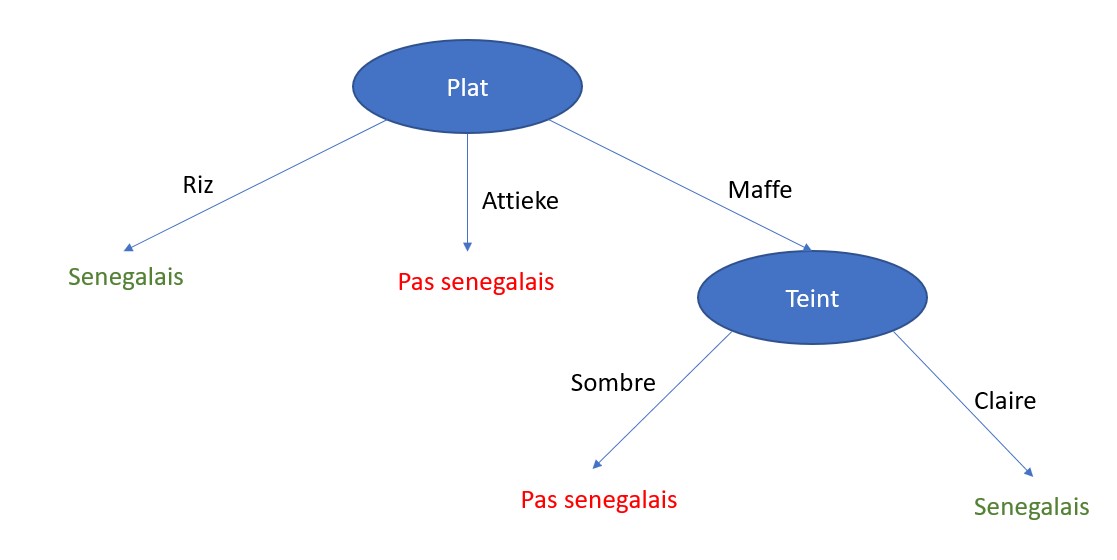
Pour ce qui est de notre exemple :

Calculons l’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 dès lors, il devient l’attribut de plus indicatif et va être à la racine de l’arbre. Notre arbre ressemble à cela après avoir fait tous les calculs :



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

##### Naive Bayes

Le modèle de Naïve Bayes (NB) est un algorithme de ML qui nous vient des statistiques et probabilités. Selon les cas, il peut être 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 cible.

* Probabilité des variables cibles

Tout d’abord il faut calculer la probabilité de toutes les variables cibles afin de savoir nos chances de tomber sur l’un ou l’autre (il est possible d’utiliser le NB dans une 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 peu mais on a presque tout le travail qui est fait, en pratique il y aura beaucoup de calculs à faire. Maintenant nous pouvons classer un nouvel individu en calculant sa probabilité de se trouver dans une classe ou une autre, ensuite nous allons normaliser les probabilités et classer dans celle qui a la plus grande valeur.

Pour normaliser les probabilités :

Exemple : prenons le même exemple du 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 tout 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’individu est 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 avec 100% de certitude 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 supervisé, c’est qu’ici nous n’aurons pas d’output pour les inputs. Dans ce cas de figure nous aurons seulement des données d’entrée mais on ne sait pas comment réagir en conséquence. C’est le modèle qui va à lui seul trouver une représentation générale qui correspond le plus aux données qui lui sont présentées.

Pour ce faire il y a ce qu’on appelle le 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 décrivent les données.

#### Clustering

Le clustering est une méthode d’apprentissage non supervisé dans lequel le but est de rassembler les individus qui se ressemblent le plus. Le principe est simple, nous avons des données mais qui ne sont pas étiquetées, 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 (le k de k-means représente le nombre de classe ou cluster).

* Définir le nombre de cluster

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

* Le centre de gravité

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

* Calcul de distance

Maintenant, nous allons calculer toutes les distances de tous les individus par rapport à tous les centres de gravité de chaque cluster. Nous allons nous apercevoir que certains individus sont mal classés, car ils sont plus proches d’un autre cluster que celui où ils sont, il suffit de les mettre à jour. Cette étape va être répéter autant de fois que nécessaire pour avoir des clusters les plus représentatifs des 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 choisis et tous les autres vont se classer par rapport au point le plus proche de ses trois. Nous nous retrouvons avec 3 clusters qui 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 un espace 2D.

Apres calcul, nous avons trouvé 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 fait 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’à faire la même chose, à savoir recalculer les centres de gravité, recalculer les distances et mettre à jour les clusters. Si nous le faisons assez de fois, nous avons 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 permettent de trouver la corrélation entre une donnée et les autres. Ses règles permettent de répondre à des questions comme : dans quelle mesure B et C vont apparaitre sachant que A est apparu ? Ces calculs vont se faire avec un ensemble de sous-ensemble, nous allons parler ici de itemset pour désigner les sous ensemble. Les règles d’associations sont très fréquentes dans les marchés et supermarché pour desceller les produits qui sont souvent acheter en un 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 a acheté en 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 tâche 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 dans la transaction sur le nombre de transaction.

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

* L’élagage

Nous voyons que le produit 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 fréquents est donc {A, B, C, E, AC, AE}, avec cela nous pouvons enfin calculer les règles intéressantes. Une règle est dite intéressante si son support est supérieur ou égal au minsup et sa confiance supérieur ou égale à minconf (la confiance minimum).

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

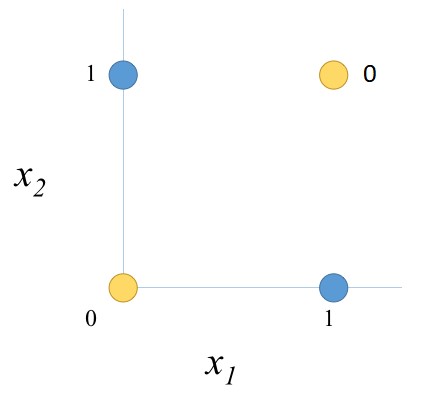
, conf = 2/3, sup = 2/5 : cette règle n’est pas intéressante car son support > minsup et sa confiance < minconf.

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

Conclusion : nous avons pour {AC}, si on achète A il n’est pas forcé que C soit acheté, mais si on achète C, il y a de forte chance que l’on achète 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 produits C.

## Deep Learning

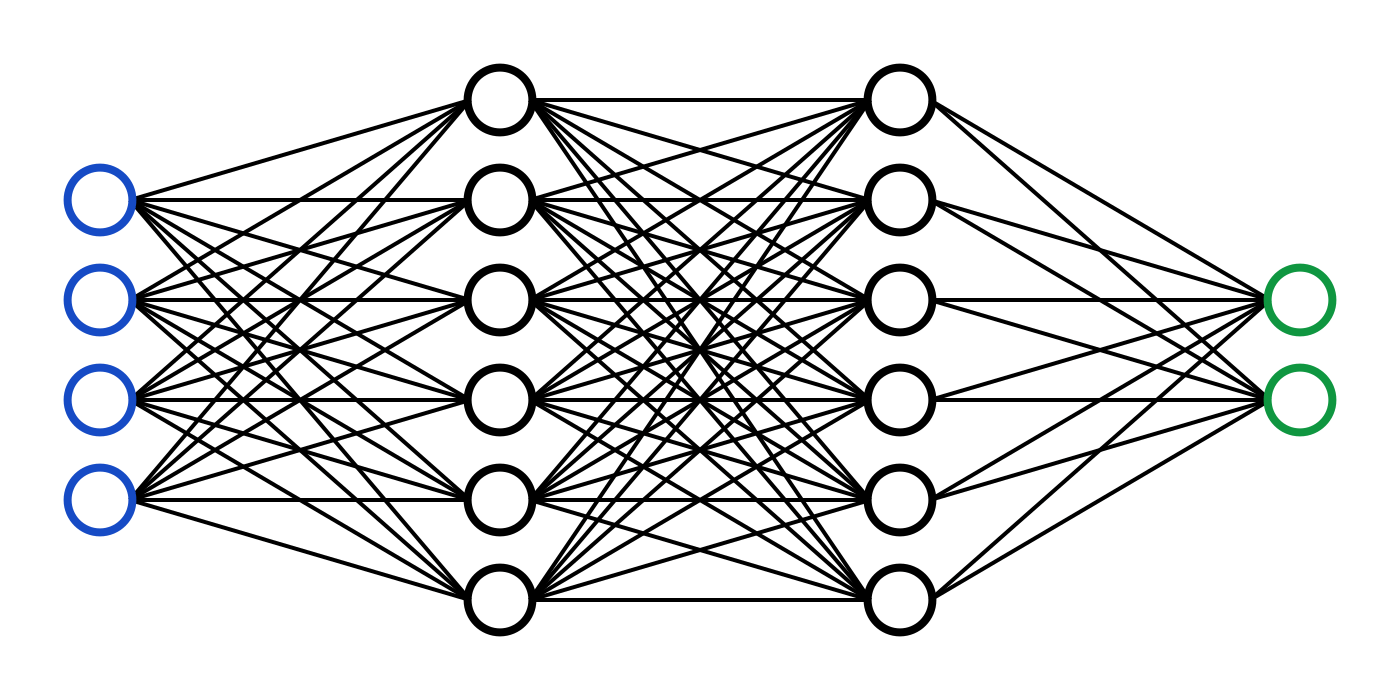
Frank Rosenblatt a créé le perceptron qui nous a permis de résoudre des problèmes, notamment le OU et le ET logique, mais quand ils l’ont essaiés pour le XOR, ils se sont rendu compte que le perceptron ne convergeait pas. Le problème était simple, un perceptron traçait des séparateurs linéaires, or ce 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 qui fut un casse-tête pour les chercheurs pendant le longtemps, jusqu’à qu’ils découvrent les solutions qui vont être présentées ici.

### Artificial neuron network (ANN)

Si vous vous rappelez la partie portant sur la régression logistique, vous avez déjà quelques notions sur les ANN. Là-bas nous faisions un apprentissage, c’est une couche d’entrée et la couche de sortie, mais ici il sera question d’une couche d’entrée, une ou plusieurs couches cachées et la sortie. Plus il y a de couche cachée, plus c’est profond : apprentissage profond : Deep Learning. D’ailleurs nous avons très probablement tous déjà vu la représentation d’un réseau de neurones profonds.



* Architecture du réseau

En pratique, on ne peut pas savoir a priori l’architecture d’un réseau, ce que les scientifiques c’est d’expérimenter jusqu’à trouver le réseau qui le marché de mieux mais consomme le moins d’énergie. Mais nous supposons que ce travail est déjà fait et l’architecture trouvée est :

* + La couche d’entrée avec 3 neurones (il ne faut pas oublier le biais)
  + Une couche de sortie avec trois neurones
  + La couche de sortie avec un neurone
* Feed-forward

Une fois l’architecture définie, on passe à l’essai en initialiser les poids au hasard, nous aurons deux ensembles de poids, ceux qui relient les entrées au cachées, et ceux qui relient les cachées à la sortie.

X : la matrice des entrées

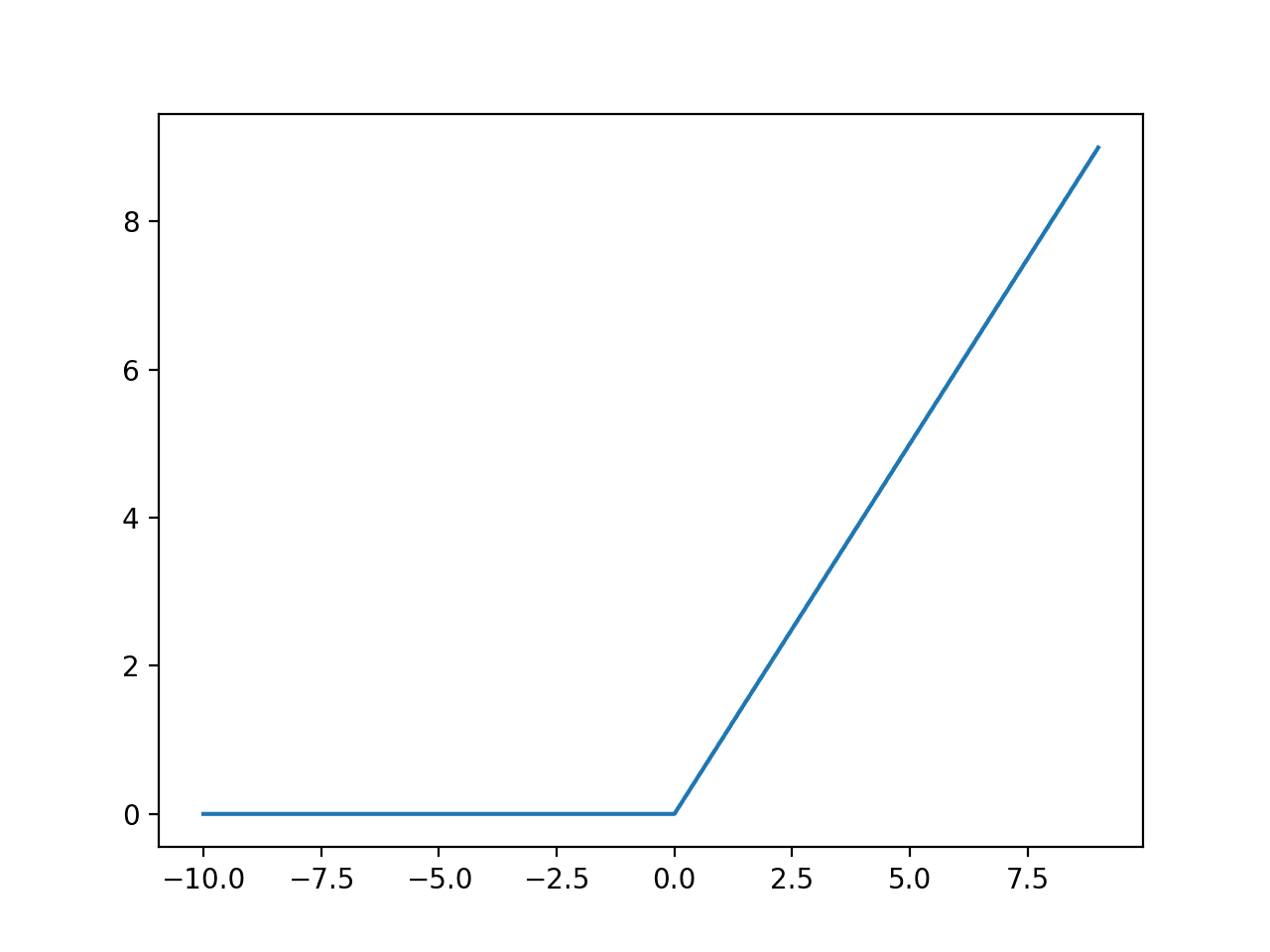
Wxh: la matrice qui relie les entrées et la couche cachée

H : matrice de la couche cachée

Why: la matrice qui relie la couche cachée et les sorties

H : matrice de la couche cachée

Il ne faut pas oublier pour les couches il y a une fonction d’activation, en règle générale, la fonction "relu" est utilisée pour les couches cachées et sigmoid pour les sorties.



* L’erreur

En Deep Learning il y a plusieurs fonctions d’erreur, il y a le MSE, que nous avons déjà vu, le MAE (mean absolute error), log loss … Mais nous allons encore utiliser le MSE car il fonction très bien pour le problème du XOR. Pour rappel :

* Backpropagation

Une fois que l’on a l’erreur on peut mettre à jour les poids, ici il y aura deux niveaux de mise à jour puisqu’il y deux ensemble de poids qu’il faut mettre à jour. Nous pouvons d’ores et déjà calculer l’erreur pour les neurones cachés.

En résumé, il faut d’abord calculer la matrice relative aux erreurs des couches cachées, ensuite nous mettons à jour les poids des sorties et enfin ceux des couches cachées. C’est comme cela que fonctionnent les réseaux de neurones même dans leurs formes les plus complexes.

### Convolutional neuron network (CNN)

S’il y a un domaine où l’humain a toujours dépassé la machine, c’est la vision reconnaitre des choses, des éléments de la nature et de les classer. Mais depuis quelque temps les scientifiques ont réalisé d’énormes avancés sur le domaine appelé Computer Vision ou vision par ordinateur. Et l’un des premiers algorithmes qui a permis de réaliser cela reste le CNN que l’on va voir tout de suite.

Comment un être humain fait pour reconnaitre les objets qui l’entoure ? nous le faisons tout le temps mais savons-nous comment ce procédé fonctionne dans nos cerveaux. Si nous parvenons à répondre à cette question, il nous sera facile de faire imiter ce procédé par un ordinateur.

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

Ici nous avons deux images de personnes célèbres, il s’agit du grand Cheikh Anta Diop et de Gaston Berger. Nous les avons tout de suite reconnus mais comment ? Diop porte des lunettes et Berger pas, Diop et de teint noire et Berger blanc, la moustache de Cheikh Anta Diop et plus touffue que celle de Gaston Berger. Et donc nous avons vu des différences considérables qui nous en permis de les distinguer. Nous allons voir les étapes que l’ordinateur va prendre pour desceller des éléments de différenciation entre ces deux personnes.

* Les filtres

Le docteur Cheikh Anta Diop portait très souvent des lunettes, alors on peut avoir un filtre pour les lunettes, on va chercher sur les images du docteur quelques choses qui ressemble à des lunettes. Ce filtre va être représenté sous forme de matrices et nous allons boucler dans l’image pour le chercher. Si nous le trouvons, on dit qu’il y a des lunettes. Par contre, porter des lunettes de suffit pas pour dire que c’est Dr. Diop, il nous faut trouver d’autres filtres, il va y avoir autant que nécessaire. Mais comment cela se passe en pratique

|  |  |  |
| --- | --- | --- |
| 0.5 | 1 | 2 |
| 6 | 2.3 | 4.1 |
| 2.8 | 0 | 6 |
| 0.14 | 3.51 | 14 |
| 95.2 | 47. | 5 |
| 2 | 15.96 | 4 |

Considérons ce tableau comme les pixels de notre image, nous pouvons avoir un filtre , il nous suffit de faire une multiplication élément par élément, avec chaque matrice qui a les mêmes dimensions que notre filtre à travers tout le tableau. Ensuite, on divise le résultat par le nombre d’éléments dans le filtre. Si vous regardez bien ce filtre est présent dans le tableau, alors nous aurons :

La valeur est plus grande que la valeur maximale du filtre, cela veut dire qu’on a trouvé ce filtre, il faut faire de même pour tous les autres filtres, on obtient ce qu’on appelle un Feature Map.

* L’activation

Une fois ce travail fait, on peut se retrouver avec un Feature map qui ressemble à cela :

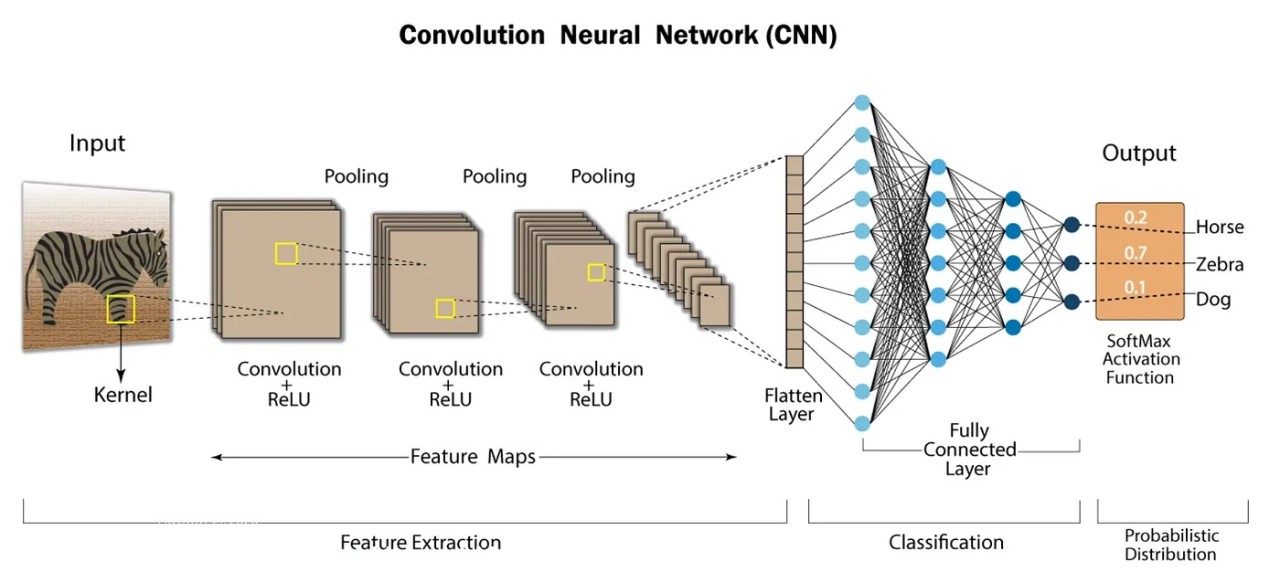
|  |  |
| --- | --- |
| -1 | 14.5 |
| 8 | 4.5 |
| -9 | -5 |

Nous allons le faire dans la fonction "relu" et nous allons obtenir cela

* Le réseau de neurone

Il peut y avoir plusieurs Feature map, on va faire ce que l’on appelle le flatten, c’est passer le N dimensions en 1D, maintenant il ne suffit plus qu’à le faire passer ce résultat flatten dans un ANN pour faire la classification.

Pour résumé, il y a deux étapes dans le CNN, le feature extraction et la classification. Le feature extraction c’est tout ce qui est relatif à la recherche d’informations sur l’image.



### Recurrent neuron network (RNN)

Nous venons juste de parler des ANN et de leurs utilités, mais dans tout domaine, il y a toujours des limites. Le principal reproche que l’on peut faire au ANN, c’est ils n’ont pas de mémoire. Imaginons un jeu de données avec 60 000 inputs, de la première ligne du premier epochs jusqu’à la dernière ligne du dernière epochs, le modèle va oublier tout ce qui s’est passé et se concentre seulement sur les caractéristiques principales. Mais très souvent, il est nécessaire de savoir ce qui s’était passé pour décider de ce que l’on va prédire.

Exemple : le Sénégal est un pays qui se trouve en Afrique et dont l’ethnie principale est composée de …

Nous voulons prédire ce qui va arriver et nous avons trois propositions : ashantis, masaïs, wolofs. Et bien évidemment c’est les wolofs. Le mot qui nous a permis de décider c’est Sénégal bien sûr, or ce mot se trouve au début de la phrase et donc ce modèle doit avoir une certaine mémoire pour bien prédire.

Les RNN sont le plus souvent utilisé pour le NLP (Natural Proccesing Language) qui nous permet de comprendre les textes sous forme de nombre. Pour mieux comprendre le NLP, prenons cette phrase : « Il est gentil ». L’ordinateur ne comprend pas le texte et on ne peut pas faire des calculs sur du texte, il va falloir trouver une solution si on veut faire passer cette phrase sur un modèle intelligent. C’est cela le travail du NLP : transformer du texte en un langage compréhensible par la machine, faire les calculs, et le retransformer du langage machine en un langage compréhensible par nous humains.

Exemple : pour la phrase « il est gentil », nous pouvons dire la chose suivante {il : 0, est : 1, gentil : 2}, de ce fait notre phrase devient « 0 1 2 » et il est possible de faire nos calculs. Bon ! dans la vraie vie, les scientifiques utilisent des algorithmes bien plus sophistiqués mais c’est juste pour la compréhension.

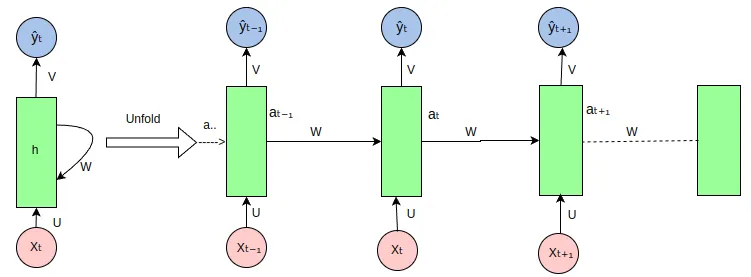
Comment les RNN fonctionnent ? Eh bien, presque de la même manière que les ANN, la seule et unique différence c’est la mémoire.

* La conversion en nombre

Avant de commencer le travail, il faut toujours convertir le texte en nombre, pour être plus précis en vecteur. L’exemple donné où l’on remplaçait les mots par des chiffres s’appelle le label encoding, mais y en a d’autres plus utilisé.

* One hot encoding
* Bag of word
* TF-IDF
* Word embedding
* …
* Feed forward

Une fois que les mots ont été convertis, ils peuvent être passés dans le modèle mot par mot mais en donnant la sortie du mot précédent comme input aussi. C’est cela qui nous permet d’avoir une certaine mémoire. Pour l’input N, on lui ajouter la sortie S-1 pour que le modèle se rappelle ce qui était venue avant.



Voici la structure générale d’un RNN, il y a, en fait, une seule couche et il représente évolution dans le temps. Pour ce qui est de l’erreur et du back-propagation, ce sera la même chose que les ANN que nous avons déjà vue.

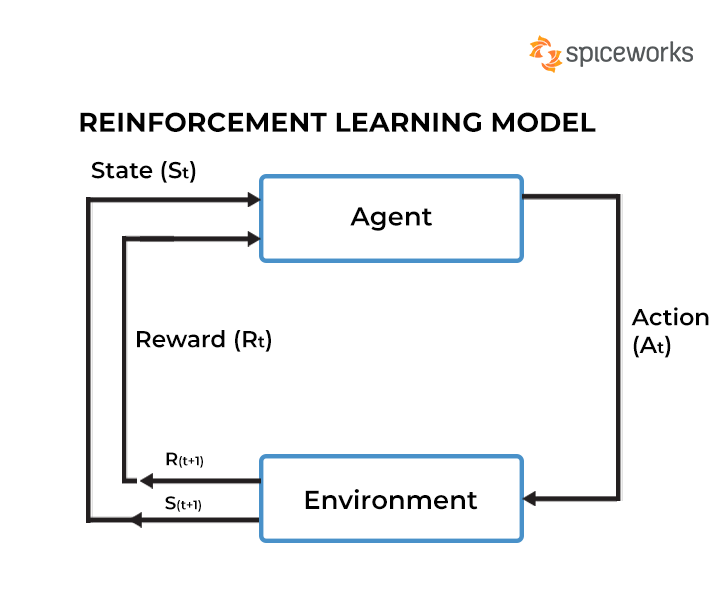
## Reinforcement Learning

Dans beaucoup de domaines de la vie, il est toujours possible d’appliquer une IA, nous le constatons aujourd’hui en 2024. La plupart du temps, le grand challenge pour les data scientistes c’est les données. Imaginez que l’on veuille apprendre une voiture à se déplacer automatiquement, si on utiliser un ANN, quelles seront les données d’entrée, les données de sorties, c’est impossible d’étiqueter l’espace 3D dans lequel nous évoluons. Ceci étant, il devient évident que, un autre algorithme va être nécessaire, c’est le Reinforcement Learning.

“Reinforcement learning is the problem faced by an agent that must learn behavior through trial-and-error interactions with a dynamic environment”. (LP Kaelbling, ML Littman, AW Moore, 1996)

Les applications du Reinforcement Learning :

* Les voitures autonomes
* La robotique
* Les jeux vidéo
* …



Le fonctionnement du Reinforcement Learning est complexe et nous n’allons pas entrer dans les détails (car il ne sera pas utilisé pour le développement de nos modèles), mais il faut comprendre qu’il y a deux choses à retenir, l’agent et l’environnement, récompense et réprimande. D’abord, nous avons deux probabilités celle de faire le bon choix et celle de faire le mauvais choix. L’agent va faire une action au hasard, si c’est le bon il va recevoir une récompense, sinon il va être réprimandé. Ce que cela signifie c’est que s’il fait le bon choix, la probabilité de refaire ses actions va augmenter, si c’est le mauvais la probabilité de refaire ce mauvais choix va diminuer. Répéter ce système autant de fois que nécessaire, nous aurons un agent qui va être capable de se déplacer correctement dans son environnement.

# Conclusion partielle

Nous y voilà, c’est le plus gros du travail dans cette rédaction de mémoire. Comme promis, nous sommes entrés dans les détails des algorithmes et nous espérons que toute personne ayant lu ce chapitre va un tant soit peu comprendre l’IA. Nous avons a priori de cela vu les prérequis pour comprendre les modèles.

La chose la plus importante à retenir dans ce chapitre c’est que l’IA n’est pas facile et demande beaucoup de connaissances dans les mathématiques et l’informatique mais avec le maximum de volonté, de détermination, de discipline et beaucoup de ses bonnes choses, n’importe quelle personne peut le faire.

Avec toutes ses connaissances acquises, nous sommes maintenant fin prêts pour pratiquer tout cela. Nous pouvons à partir de maintenant développer, déployer et intégrer dans une interface graphique nos modèles. Et bonne nouvelle, c’est ce que nous allons faire dans la prochaine partie.

# 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

Rashid, T. (2016). *Make your own neural netwrk.* Createspace Independent Publishing Platform.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. Journal of artificial intelligence research, 4, 237-285.

# 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