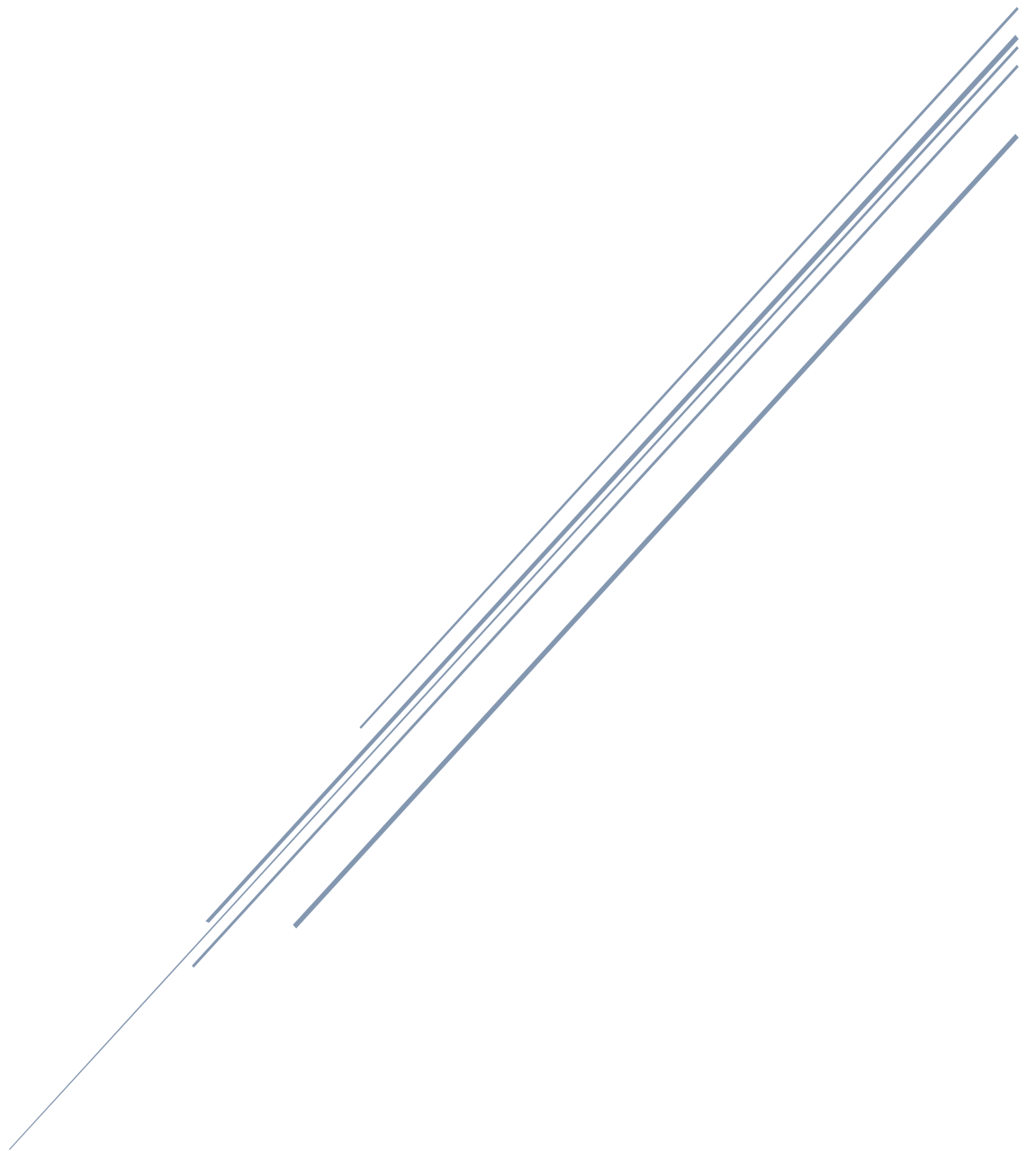


VEILLE DE SPECIALITE

Comment créer une application mobile multiplateforme permettant de savoir avec une casi exactitude si le parking du CESI est surchargé ? Le tout, en limitant les coûts.



IA, image recognition, Application mobile, Python

CESI

Sommaire

Une IA, qu'est-ce donc ?	2
I / Qu'est-ce que l'on veut dire par IA ? les différents types d'IA.....	2
II / Il est facile de me confondre !	3
III / L'IA du futur et de maintenant	4
Le machine learning.....	6
I / Supervised learning	6
II / Dataset	7
III / Le modèle et ses paramètres	8
Les mathématiques derrière l'IA !	8
I / La fonction Coût.....	8
II / Algorithme de minimisation et descente de gradient	9
III / Calcul matriciel	10
IV / Addition & soustraction de matrices	10
V / Multiplication matrices	11
Le deep learning	12
I / Fully connected neuronal network.....	13
II / Convolutional Neural Network.....	14
III / Classification supervisée	16
Détection d'objets	17
I / Les filtres passe-haut, passe-bas et data wrangling	17
II / Traitement d'une image par ordinateur.	19
Les tenseurs (tensors) :	19
Développement d'une application mobile multiplateforme (technologies Web)	21
Amazon Web Service (gestion des données + liaison application).....	22
I / Qu'est-ce qu'AWS ?	22
II / Avantages et inconvénients	22
Proof of concept, maquettage à taille réduite + tests de l'IA	24
I / Mieux vaut VM que guérir	24
II / Application sur une plaque non conforme, ou ancienne.....	25
III / Déparasitage et test concluant.....	26
Conclusion	27
Sitographie :	28

Une IA, qu'est-ce donc ?

I / Qu'est-ce que l'on veut dire par IA ? les différents types d'IA

L'IA pour intelligence artificielle est un procédé qui vise à reproduire, ou du moins à se rapprocher le plus possible de la réflexion humaine. La réflexion humaine se construit par nos expériences passées et les conclusions que l'on en tire. Maintenant pourquoi ne pourrais-t-on pas appliquer cela à une machine ?

Tout simplement puisqu'une machine ne ressent rien, elle répond tout simplement à une suite d'instructions rédigées par un programmeur.

Repartons à la base de notre schéma de pensée, d'où proviennent les émotions, actions/ réactions qui rythment notre quotidien ? Eh bien de notre expérience passée, et des conclusions que nous avons pu établir. C'est grâce notamment à la neuroscience que les tout premiers réseaux de neurones artificiels ont pu être développés.

Depuis la naissance de "L'Intelligence Artificielle" à l'atelier de Dartmouth en 1956. Les fondateurs de l'IA l'ont alors définie comme la poursuite de la fabrication de machines aussi intelligente que nous.

1956 Dartmouth Conference: The Founding Fathers of AI



John McCarthy



Marvin Minsky



Claude Shannon



Ray Solomonoff



Alan Newell



Herbert Simon



Arthur Samuel



Oliver Selfridge

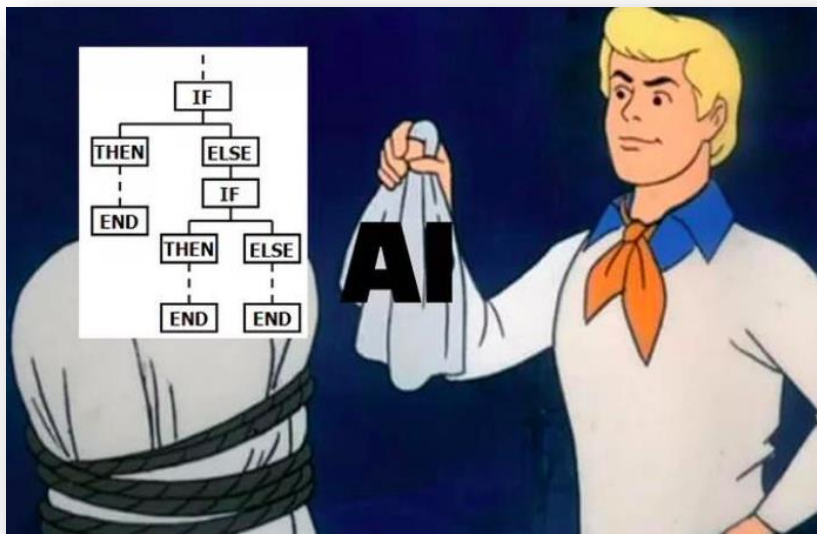


Nathaniel Rochester



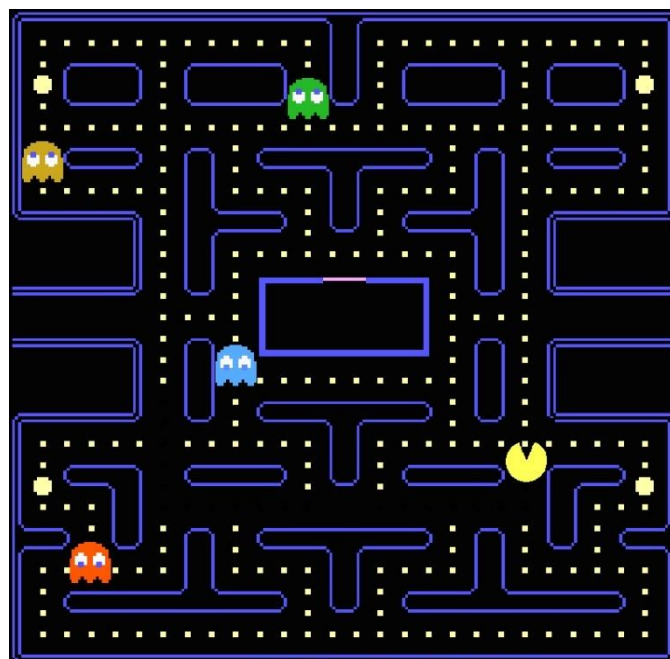
Trenchard More

Il / Il est facile de me confondre !



Aujourd'hui on a tous envie de dire que chaque petit programme qui nous paraît intelligent utilise une technologie d'intelligence artificielle tel que le deep learning ou le machine learning.

Mais ça n'a rien à voir, en programmation, en règle générale nous sommes obligés d'expliquer, de donner chaque instruction étape par étape à la machine afin qu'elle sache quoi faire... Une machine n'a pas d'intuition.



Voyons donc un exemple comme Pac-Man. Dans ce jeu nous avons donc notre personnage contrôlable et 4 fantômes qui cherchent à nous manger. Dans le monde du jeu vidéo en règle générale tout le monde parle d'IA à partir du moment où un autre personnage que celui qui est contrôlé par le joueur prend part au jeu, cela va donc de Pac-Man jusqu'à Assassin's Creed.

Revenons à nos fantômes, on a donc Blinky le rouge qui est programmé pour nous traquer, Pinky le rose qui cherchera toujours à se placer devant nous, Inky le bleu qui cherche à tendre des embuscades et Clyde le fantôme orange qui adopte un déplacement aléatoire. Ici rien n'est engendré par de l'IA chacun des fantômes est programmé pour répondre à son unique tâche et rien d'autre, pourtant beaucoup a tendance à considérer cela comme de l'IA.

III / L'IA du futur et de maintenant

Reactive AI réagi de la même manière à un évènement bien défini. Elle ne peut pas apprendre ou garder en mémoire des informations pour en tirer des conclusions pour une future application. (Par exemple les recommandations Netflix).

Limited memory AI, absorber des données basées sur des expériences au cours du temps pour en tirer des conclusions affectant ces performances. Cette IA utilise ce que l'on appelle le « machine learning » et c'est celle qui est le plus utilisée de nos jours. On peut entraîner son modèle afin d'améliorer son taux de succès ou bien mettre à jour les données. Cette dernière peut donc interpréter son environnement et ajuster son paramétrage en fonction des données que l'on lui donne pour faire travailler ces modèles.

Theory of mind AI sera le terme que l'on utilisera le jour où on aura développé une IA capable de faire des choix équivalents aux choix que peut faire un humain.

La machine aura alors la capacité de comprendre et de se rappeler des émotions. L'IA serait alors capable de reconnaître et de répliquer les émotions.

Le robot Humanoïde Sophia mis au point par Hanson Robotics en 2015 a réussi à interpréter les émotions de ses interlocuteurs et à y répondre de façon appropriée. On reste encore loin de ce que « theory of mind » évoque mais cela reste un pas important et non négligeable dans cette direction. Le jour où l'on ne pourra plus faire la distinction entre une machine et un robot on pourra éventuellement dire que l'on a atteint ce stade de conscience immédiate.



Self Aware AI à ce moment-là une IA ne sera pas seulement capable d'interagir comme le ferait un humain avec son environnement, mais elle sera également capable d'avoir conscience d'elle-même ce que l'on appelle conscience réfléchie chez l'Homme.

Ce qui fait de nous des humains est que l'on partage ces deux types de consciences (immédiate et réfléchie) contrairement aux machines à l'heure actuelle. Une citation qui reflète bien cette évolution d'une IA, d'après Kant : « Avant il se sentait, maintenant il se pense ». Ici Kant parle bel et bien d'un Homme, mais si une AI atteignait ce stade alors qu'elle serait la différence avec l'Homme ?

Le machine learning

Maintenant que nous avons vu les différents types d'IA nous allons pouvoir rentrer dans le détail sur comment le machine learning fonctionne concrètement. Le machine learning offre par exemple la possibilité à un utilisateur de pouvoir se faire proposer du contenu personnalisé, un contenu que le modèle comprend comme généralement plus apprécié par l'utilisateur. La programmation habituelle ne permet pas de faire cela ou bien il faudrait créer un programme immense comprenant chaque possibilité et chaque particularité que possède un individu ce qui rendrait la tâche titanesque. Avec le ML, on donne la capacité à une machine d'apprendre sans avoir à la programmer de façon explicite.

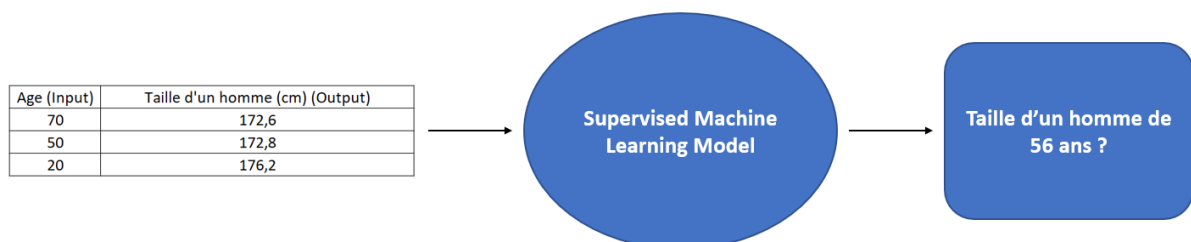
Le ML concerne tout le monde, il révolutionne tous les domaines. On parle du monde agricole jusqu'à la détection de cellules cancéreuses sur une IRM. Aujourd'hui, on le retrouve partout dans notre quotidien.

Le machine learning vise donc à entraîner un modèle avec des données, mais tout d'abord, quels types de modèle existent-ils aujourd'hui ?

I / Supervised learning

De nos jours, il existe deux grandes catégories de machine learning, supervised learning (apprentissage supervisé) et unsupervised learning (apprentissage non-supervisé).

Le but du ML (Machine Learning) supervised est de construire un modèle précis afin de créer des fonctionnalités prédictives, ici c'est l'Homme qui joue le rôle de professeur en fournissant à la machine des données. On donne à la machine des exemples qu'elle doit étudier pour créer un modèle, ces mêmes exemples constituent un « dataset » ou jeu de données.



Comme une machine ne comprend pas notre langage, afin de pouvoir interpréter un dataset, elle doit trouver la relation en les données d'entraînement. Prenons x pour l'âge et y pour la taille alors l'ordinateur cherchera la relation :

$f(x) = y$ ou a sera déterminé à l'aide du dataset d'entraînement.

A retenir :

L'apprentissage supervisé vise à connaître la valeur d'une variable continue c'est-à-dire une variable qui peut prendre une infinité de valeurs.

II / Dataset

L'une des notions fondamentales de l'apprentissage supervisé est le dataset. Qui contient deux types de variables, d'un côté on a la « target variable »

Target y	Features		
	x1	x2	x3
Prix (10 ³ \$)	Kilométrage	Qualité	Modèle
11	100 000	5	Peugeot 208
10	125 000	4	Dacia Duster
8	90 000	4	Citroen
6	95 000	3	Renault Clio
2	200 000	3	Kangou 4 places

Par convention on note :

m : nombre d'exemples

n : nombre de features

Pour représenter l'emplacement d'une case feature dans le tableau, on utilisera la convention suivante :

$$x_{feature}^{(exemple)}$$

Un exemple est le numéro de ligne et la feature est une donnée se trouvant dans le dataset.

III / Le modèle et ses paramètres

C'est le développeur qui décide quel modèle la machine doit utiliser et c'est la machine qui doit apprendre les paramètres de ce modèle.

Un modèle peut suivre différents types de fonction :

Affine : $F(x) = ax + b$

Polynomiale de degré 2 : $F(x) = ax^2 + bx + c$ et tous ses autres degrés.

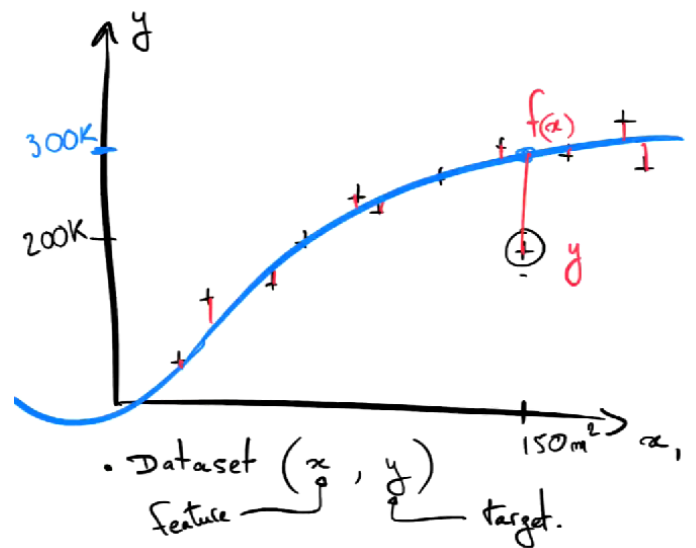
Les mathématiques derrière l'IA !

I / La fonction Coût

Lorsqu'on assemble les erreurs de notre dataset on obtient alors la fonction Coût.

Avoir un bon modèle, c'est avoir un modèle qui nous donne de petites erreurs.

La fonction coût nous permet de visualiser la proportion d'erreur de notre modèle par rapport à notre dataset.



II / Algorithme de minimisation et descente de gradient

Ensuite vient la notion d'algorithme de minimisation qui vise à trouver quels sont les paramètres qui minimisent nos erreurs.

L'algorithme le plus connu est la descente de gradient. Cet algorithme permet de trouver le minimum de n'importe quelle fonction convexe. On s'en sert en ML supervisé pour trouver le minimum de la fonction Coût puisque comme nous l'avons vu, trouver le minimum de la fonction Coût revient à trouver le meilleur modèle pour notre problème de ML.

Autrement dit, si vous obtenez un point très proche de la valeur minimale (ou maximale), le gradient est pratiquement nul car la fonction change peu à ce point.

Gradient descendant :

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

α : représente le learning rate, cela détermine le pas entre chaque itération en se déplaçant le long de la « fonction loss »

a_{i+1} : valeur de sortie (output)

a_i : valeur d'entrer (input)

La « fonction loss » est le nom anglophone pour la fonction Coût. On implémente cette formule dans une loop et alors on obtient un algorithme itératif.

Si on choisit une « mauvaise » valeur pour α , alors il se peut que dans un cas l'algorithme oscille autour de la valeur correcte lors de la descente de gradient, ou bien dans un autre cas il se peut que l'algorithme mette un temps infini pour atteindre l'objectif.

En général, il faut essayer des valeurs au tâtons afin de trouver notre paramètre α .

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

Maintenant que nous avons vu les notions, avant même de pouvoir implémenter notre formule dans la machine, il nous faut trouver la valeur de nos dérivés.

Il faut donc calculer le dérivé de J par rapport à « a » :

$$J(a, b) = \frac{1}{2m} \sum (ax + b - y)^2$$

Attention $G \cdot F$

$$\frac{\partial J}{\partial a} = \frac{1}{m} \sum x(ax + b - y)$$

$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum (ax + b - y)$$

III / Calcul matriciel

Une matrice est un tableau de nombres, elles permettent d'appliquer des opérations communes sur un groupe de données. Voilà qui nous plaît bien, nous, informaticiens !

En machine learning nous pourrons donc manipuler toutes nos données simultanément à l'aide de nos matrices plutôt que de les manipuler point par point.

Une matrice est de dimension ($m \times n$) lorsqu'elle respecte m lignes et n colonnes.

La transposée d'une matrice, il s'agit donc de prendre une matrice et de la retourner sur elle-même (flip ligne et colonnes).

Exemple de transposé matriciel :

$${}^t \begin{pmatrix} 14 & 6 & 6 & 3 \\ 9 & 10 & 6 & 9 \\ 30 & 4 & -9 & -7 \\ 5 & -2 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 14 & 9 & 30 & 5 \\ 6 & 10 & 4 & -2 \\ 6 & 6 & -9 & 1 \\ 3 & 9 & -7 & 0 \end{pmatrix}$$

IV / Addition & soustraction de matrices

Ces opérations mathématiques ne s'appliquent que sur des matrices de dimensions égales et leur résultat donne une matrice de même dimension qu'initialement.

Exemple d'addition matriciel :

$$A + B = \begin{pmatrix} a & b \\ c & d \end{pmatrix} + \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a + e & b + f \\ c + g & d + h \end{pmatrix} = C$$

Exemple de soustraction matriciel :

$$A - B = \begin{pmatrix} a & b \\ c & d \end{pmatrix} - \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a - e & b - f \\ c - g & d - h \end{pmatrix} = C$$

V / Multiplication matrices

Le produit de la matrice A et de la matrice B n'est possible que si le nombre de colonnes de la matrice A est égal au nombre de lignes de la matrice B.

Exemple de multiplication de matrices :

$$a_{11} \times b_{11} + a_{12} \times b_{21} + a_{13} \times b_{31} = c_{11}$$
$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

Maintenant, comme nous ne souhaitons pas réaliser tous ces calculs à la main, une librairie Python nommée numpy propose des fonctions réalisant ces mêmes calculs matriciels.

```
import numpy as np
```

Création d'une matrice :

```
matrice = np.array([4, 7, 9])
```

Produit matriciel :

```
ProduitMatriciel = numpy.dot()
```

Transposé de matrice :

```
ProduitMatriciel = numpy.dot(MatriceA, MatriceB)
```

Connaitre les dimensions de ces matrices permet d'éviter énormément de problèmes lors de l'exécution de code.

Lorsque notre modèle a été implémenté, on peut alors calculer le coefficient de détermination. Plus ce dernier est proche de 1 plus cela veut dire que notre modèle est précis.

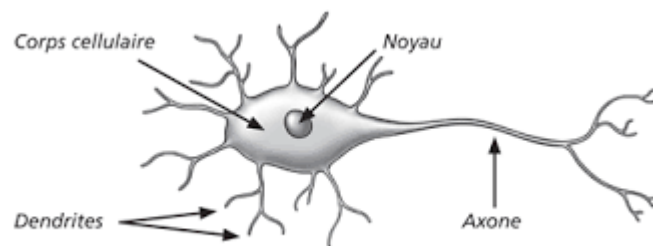
Le deep learning

La différence fondamentale entre la machine learning et le deep learning est que l'Homme n'a pas à paramétrer quoi que ce soit en utilisant la méthode de deep learning. On fournit simplement les données à la machine et c'est elle-même qui définira les paramètres adéquates à l'exploitation du modèle pour atteindre son but.

Le deep learning offre la possibilité de paramétrer l'IA d'une façon dont l'humain n'aurait pas forcément déjà imaginé que ça soit probant. Aujourd'hui, il est très facile notamment à l'aide Python de créer ses premiers modèles de machine learning, mais ce qui fait la différence entre un bon modèle et un mauvais, c'est dans un premier temps les données (fournir de bonnes données) et le paramétrage.

De plus, un modèle surentrainé avec un panel beaucoup trop large de données peut mener, à terme, à causer un « bruit » capable d'influencer les résultats d'un modèle.

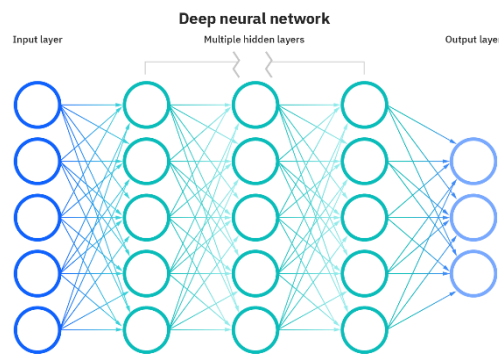
Comme une bonne partie des inventions, les réseaux de neurones virtuels ont été imaginés en s'inspirant de la nature.



Le neurone réagit à une stimulation, chacun des neurones est lui-même relié à un ou plusieurs autres neurones. En fonction du stimuli et de la région du cerveau où se trouve ce dernier, il fera le choix de transmettre ou non l'information via les axones à l'aide d'un stimuli nerveux.

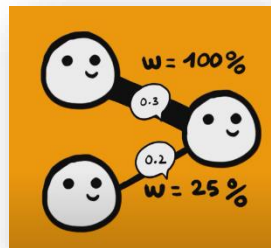
Les dendrites seraient nos inputs si l'on parle d'un point de vue machine et les axones seraient les liaisons entre chaque neurone auxquelles on peut indiquer un poids plus important.

I / Fully connected neuronal network



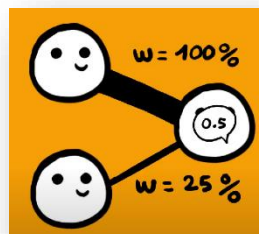
D'un point de vue machine l'information circule de neurone en neurone, entre chaque neurone on applique le poids de la liaison à la donnée. On commence à parler de deep learning à partir du moment où notre réseau de neurones possède un minimum de hidden layers.

$$Information_{re\acute{c}ue} = w * Information_{transmise}$$



ou w est weight, le poids de la liaison de 0 à 1

Pendant la phase d'entraînement des neurones, on obtiendra des résultats plutôt mauvais. Mais après chaque passage à travers le réseau neuronal, chacun des neurones et notamment le poids des liaisons est changé pour obtenir les résultats les plus probants.



Le dernier neurone reçoit la somme des neurones adjacents qui lui sont liés.

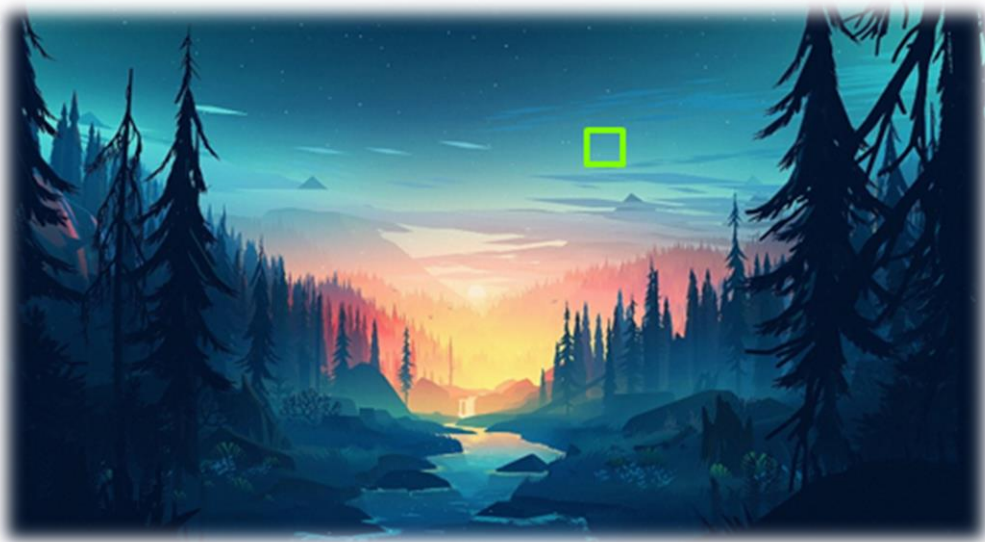
On fait la différence entre la valeur attendue et la valeur ressortie par le réseau de neurones, cette différence nous permet de calculer la fameuse fonction Coût que nous avons vu auparavant. A l'aide de cette fonction, on sera capable de savoir quel(s) poids on(t) le plus participé à cette erreur.

Plus le résultat en output est proche de 1, plus le réseau neuronal produit une prédiction précise.

II / Convolutional Neural Network

Il existe de nombreux types de réseaux neuronaux permettant de traiter différents types de données mais aujourd'hui celui auquel on va s'intéresser est le CNN (Convolutional Neural Network). Ce dernier permet de traiter les images, reconnaître des visages, des formes etc...

L'architecture de ConvNet (CNN) est similaire à celle du modèle de connexion neuronale du cerveau humain et s'inspire des tissus du cortex visuel. Les neurones individuels ne répondent qu'aux stimuli dans une zone limitée du champ visuel, les champs dits récepteurs. Les collections de ces champs se chevauchent pour couvrir tout le cortex visuel.



Dans la même idée, on ne va pas attribuer un neurone artificiel à un seul pixel mais à une zone, ou plus précisément une matrice de pixels. Tout cela dans le but d'éviter d'avoir besoin de faire 36 millions de calculs de poids en liaison neuronale dès la première couche cachée.

Image size = $1920 \times 1080 \times 3$

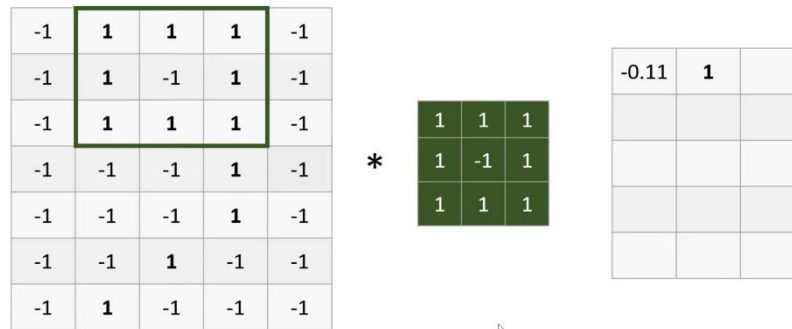
Première couche neuronale = $1920 \times 1080 \times 3 \simeq 6.2208$ millions

Couche neuronale cachée = Disons que l'on garde 4 millions

Poids entre l'Input et la première couche cachée = $6.22 \times 4 \simeq 24,9$ millions de calculs à réaliser.

Et cela avec une seule couche neuronale cachée, alors je vous laisse imaginer si l'on multiplie le nombre de couches (ce qui est toujours le cas dans un modèle performant).

Dans le but d'économiser et aussi d'améliorer la précision de nos réponses, on traitera non pas pixel par pixel une image mais une matrice de ces mêmes pixels.

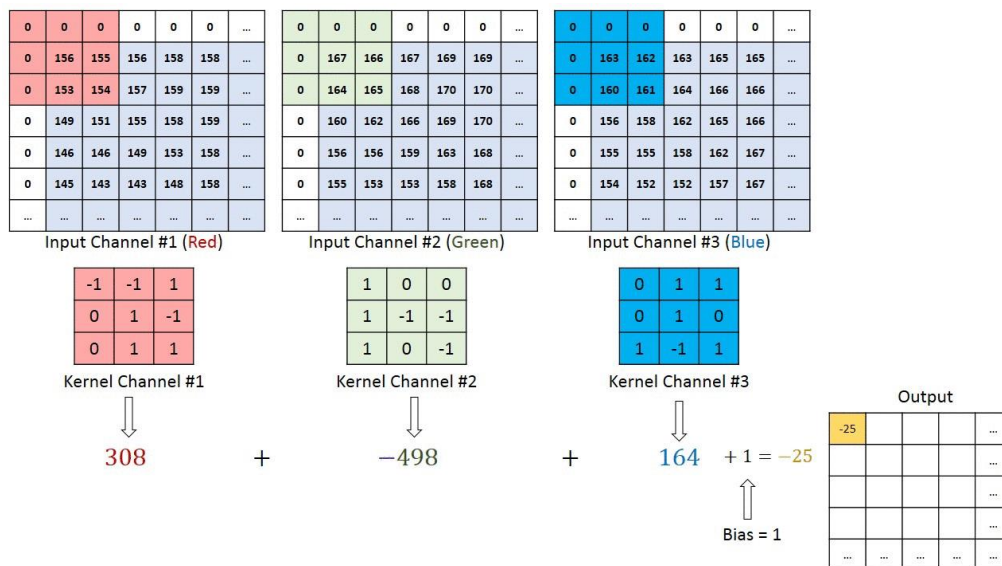


On peut choisir la taille de cette matrice, la seule contrainte est qu'elle doit être multipliable par notre filtre (donc deux matrices de mêmes dimensions).

$$-1+1+1-1-1-1-1+1+1 = -1 \rightarrow -1/9 = -0.11$$

Enfin on fait la somme du résultat obtenu par notre filtre que l'on multiplie par : $\frac{1}{\text{nombreElementMatrice}}$

Le résultat est stocké dans la « feature map »

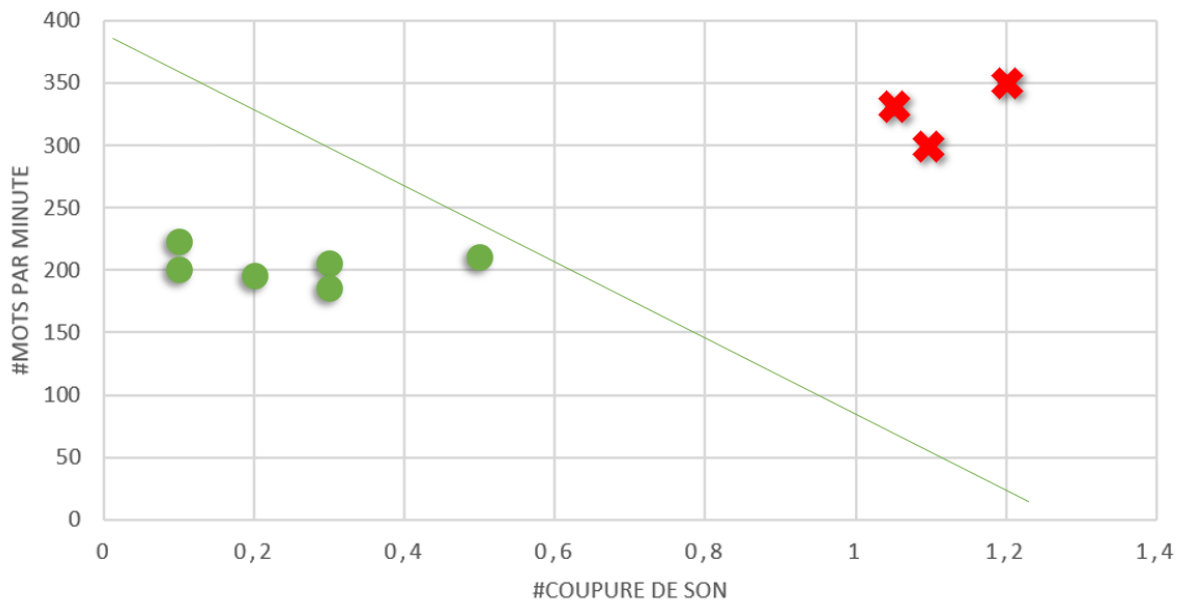


III / Classification supervisée

Le problème de classification vise à déterminer la valeur d'une variable discrète. Ici on prendra l'exemple de quelqu'un qui souhaite développer une IA qui coupe la télévision lorsqu'il y a des pubs.

$$y \begin{cases} 0 \rightarrow \text{pas de publicité} \\ 1 \rightarrow \text{une publicité passe} \end{cases}$$

On admet en vert aucune publicité et en rouge la présence d'une publicité. De plus on considère que le temps qui sépare deux publicités est d'environ 1 seconde.



La machine crée ce que l'on appelle une frontière de décision. Ce qui permet à l'IA, selon où se trouve le point sur le graphique de déterminer si selon son modèle d'entraînement une publicité passe à la télévision ou non. On remarque que le nombre de mots par minute est bien inférieur lorsqu'un programme télévisé passe plutôt qu'une publicité.

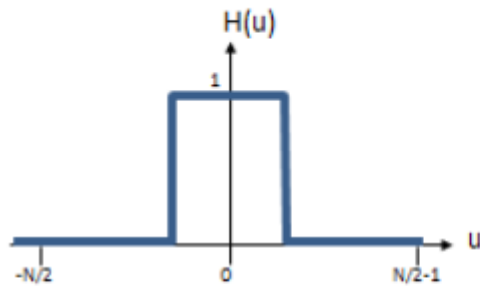
Dans l'objectif de rendre l'IA la plus précise possible et ainsi éviter qu'elle ne coupe la télévision alors que votre programme préféré est en cours de diffusion, on peut alors appliquer une incertitude autour de la frontière de décision dans laquelle les points ne seront ni catégorisés dans l'un, ni dans l'autre.

Détection d'objets

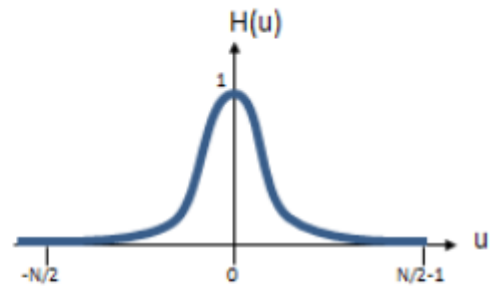
I / Les filtres passe-haut, passe-bas et data wrangling

Le filtrage de fréquence d'un point de vue physique permet de transformer et de mapper des données d'une forme brute dans le but de les rendre plus appropriées avant même de les analyser.

Le filtre passe-bas consiste à éliminer les fréquences hautes et à conserver les basses fréquences.



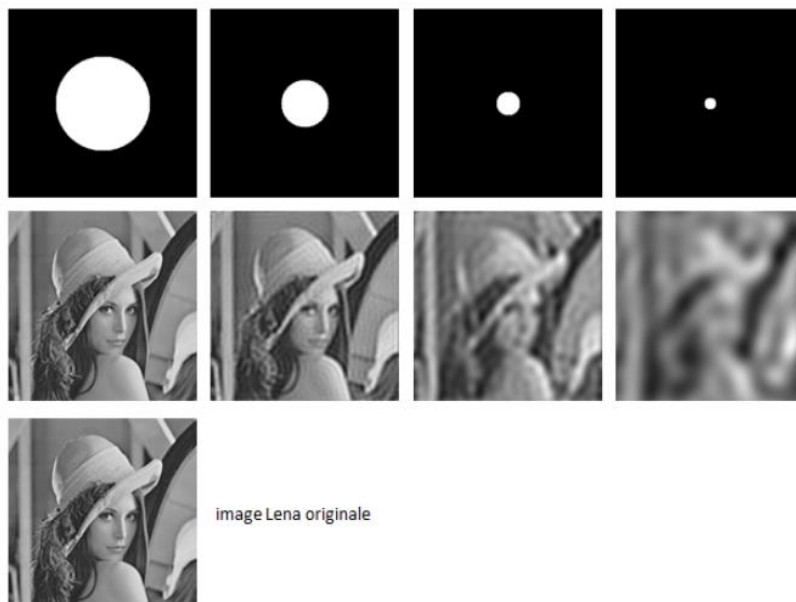
filtre passe-bas idéal



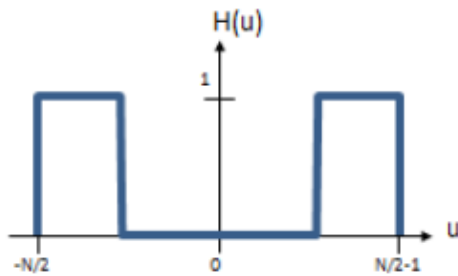
filtre passe-bas gaussien

Exemples de filtres passe-bas 1D

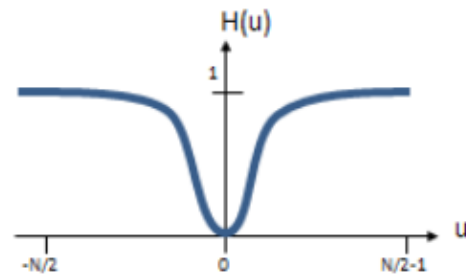
En traitement d'images le fait d'appliquer un filtre passe-bas implique le floutage de l'image.



Quant au filtre passe-haut, celui-ci a la propriété de ne laisser passer que les hautes fréquences et supprimer les basses fréquences.



filtre passe-haut idéal



filtre passe-haut gaussien

Exemples de filtres passe-haut 1D.

En traitement d'images, le fait d'appliquer un filtre passe-haut fait ressortir les traits caractéristiques de l'image. Tel un dessin à colorier soit même on retrouve les « bordures » du visage.



Le filtre renvoie une valeur proche de zéro dans les zones unies ou dégradées.

Pour résumé :

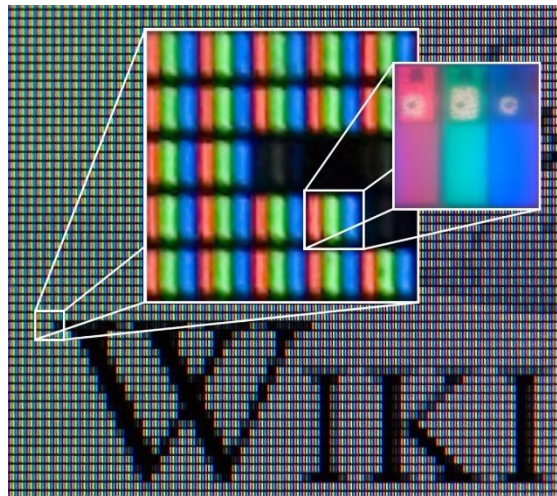
- Les hautes fréquences spatiales correspondent par exemple à des valeurs de gris très variables sur quelques pixels dans une zone très structurée de l'image.
- Les basses fréquences spatiales correspondent à des changements lents des niveaux de gris, par exemple dans la zone de gradient.

Le fait d'appliquer un filtre en aval avant même de traiter l'image informatiquement a pour effet direct de diminuer la consommation de ressources pour traiter l'image.

Nous reverrons cela plus tard lors du développement d'une IA permettant la lecture de plaques d'immatriculation et comment nous aurions pu diminuer l'impact sur les ressources de l'ordinateur de la part du programme.

II / Traitement d'une image par ordinateur.

Nous avons vu les matrices d'un point de vue mathématique mais concrètement comment allons-nous pouvoir les utiliser pour traiter notre image ?



Pour cela il faut revenir à la base, une image de 1920 par 1080 pixels (picture element) représente déjà 2 073 600 de pixels. Chaque pixel est composé d'une matrice de 3 x 1, cela veut donc dire que chaque pixel contient 3 informations, la proportion de Rouge, Bleu et Vert.

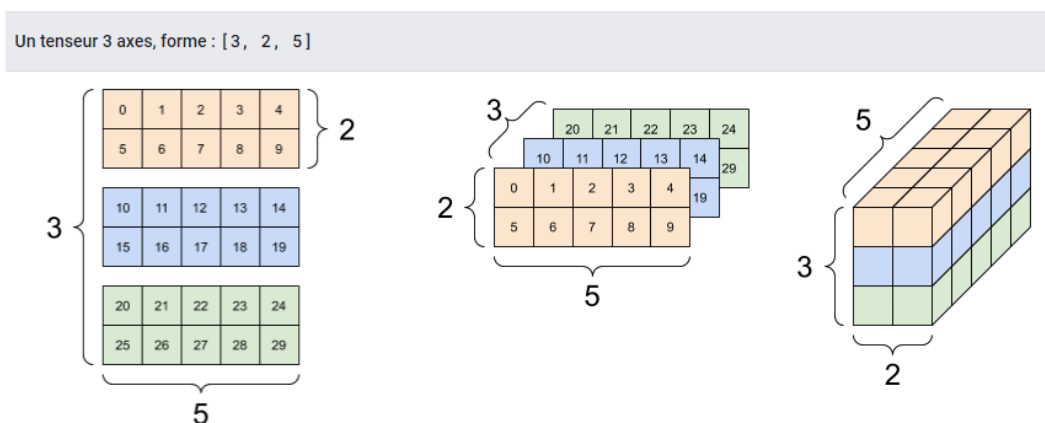
Où alors il faudrait créer une matrice pour stocker les données uniquement le Rouge, une autre pour le Bleu et une dernière pour le Vert.

Les tenseurs (tensors) :

Qu'est-ce qu'un tenseur ?

Eh bien c'est un tableau multidimensionnel ou autrement dit une matrice multidimensionnelle. Pile ce dont on a besoin pour pouvoir stocker nos données RGB sans avoir à créer trois matrices différentes. *Un tenseur de rang 2 n'est jamais qu'une matrice.*

Afin de pouvoir traiter ces données nous auront besoin d'un tenseur sur 3 axes qui prendra la forme suivante :



Si on cherche à obtenir la *shape* de notre tenseur, on obtiendrait :

tf.shape(tensor) = [rang, ligne, colonne]

Le paramètre rang nous permet de stocker 3 informations dans notre exemple, mais en soit, on pourrait en stocker une multitude (en fonction du besoin). Pour stocker notre valeur RGB, 3 suffiront. Nous avons alors ligne et colonne pour connaître l'emplacement de notre pixel et la dimension pour stocker ces valeurs.

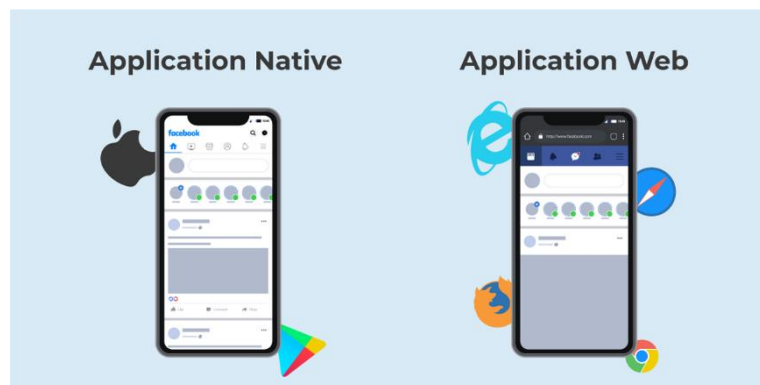
Si l'on traitait une image en couleur, on se retrouverait donc avec une matrice de 1920 x 1080 dans laquelle chacun des emplacements serait lui-même une matrice dimensions 3 x 1.

Une fois l'image passée en noir et blanc on réduit donc considérablement la puissance de calcul nécessaire.

Développement d'une application mobile multiplateforme (technologies Web)

Afin de développer une application mobile multiplateforme, nous avons plusieurs possibilités qui s'offrent à nous. La première serait de coder l'appli pour Android en utilisant Java puis de tout recommencer pour IOS sur Xcode et comme langage Objective-C.

C'est alors que les développeurs se sont posé la question suivante : « Comment gagner du temps ? je n'ai pas envie de tout refaire deux fois ! ». Mais quel est le moteur de rendu que possède Safari et tous les autres smartphones, c'est le moteur de rendu web !



C'est donc en HTML, CSS, Javascript et PHP que l'on pourrait coder notre application. Voyons d'un côté les avantages et de l'autre les inconvénients du développement d'applications mobiles.

Dans un premier temps les technologies web sont relativement faciles à manier, et à maintenir puisque les mises à jour des moteurs de recherche se font naturellement sur tous les appareils.

Si le besoin se faisait ressentir il est bien moins coûteux d'embaucher un développeur web plutôt qu'un développeur natif (se faisant plus rare). De plus il n'y aura besoin du développement que d'une seule application, pour toutes les plateformes (même ordinateurs).

Enfin cela évite les décalages dans les updates en fonction des plateformes, par exemple avoir la dernière mise à jour 2 semaines après sur iPhone à cause d'une nouvelle politique chez Apple. La web app ne peut pas accéder aux applications natives du mobile comme le GPS, appareil photo etc.

Maintenant cela n'a pas que des bons points, en effet une application native sera bien plus performante puisque développée pour une plateforme bien précise, le tout en utilisant un langage natif à la plateforme. De plus cette dernière pourra être accessible hors connexion, et aura un meilleur référencement sur les plateformes comme l'App Store et le Play Store

Amazon Web Service (gestion des données + liaison application)

I / Qu'est-ce qu'AWS ?



Amazon Web Services est une division d'Amazon, un groupe de commerce électronique américain spécialisé dans les services de cloud computing à la demande pour les entreprises et les particuliers.

Le stockage de données :

Le stockage des données est l'une des principales préoccupations du cloud. En conséquence, AWS propose une variété de services dans cette catégorie.

Base de données :

AWS propose aussi l'hébergement et la gestion de bases de données dans le cloud. Amazon Relational Database Service est compatible avec Oracle, SQL, ServerMySQL, MariaDB et bien d'autres.

II / Avantages et inconvénients

AWS permet de disposer d'un espace en ligne, sûr puisque que les données de ses clients sont clonées dans différents datacenters (dépend de la souscription).

Il s'agit d'une offre économique pour les entreprises, en effet ces dernières n'auront pas besoin de dépenser d'argent pour entretenir ou mettre à jour leur data center puisque c'est AWS qui s'en occupe. Le coût final est bien moindre que celui d'un serveur privé.

En quelques clics il est possible de s'exporter à l'international. Besoin de nouveaux accès à des serveurs plus proches de vos clients, c'est possible sans avoir à installer ou acheter quelconque matériel.

Le problème majeur c'est que les utilisateurs perdent le contrôle sur leurs données, ces dernières ne sont plus hébergées sur votre propriété mais bien sur un serveur appartenant à Amazon que vous louez.

AWS Global Infrastructure



Aujourd'hui AWS c'est 24 régions pour un total de 77 zones de disponibilité. AWS couvre 245 pays et tend à continuer à s'exporter.

Proof of concept, maquettage à taille réduite + tests de l'IA

On utilisera Python pour développer notre IA reconnaissant et traitant les plaques d'immatriculation. Vous devez vous demander mais pourquoi une plaque d'immatriculation alors que l'on veut simplement savoir si une place est libre sur un parking.

La plaque d'immatriculation est un moyen unique comme une empreinte digitale elle correspond à une seule entité c'est pourquoi nous allons l'utiliser pour comparer les plaques entrantes et sortantes du parking.

Lorsque le système aura fait la correspondance entre une plaque qu'il a vu rentré, et une plaque qu'il voit sortir alors on en déduit qu'une place s'est libérée ! Et vis versa, le programme incrémente où décrémente le compteur de véhicule.

C'est bien beau mais est-ce réalisable ?

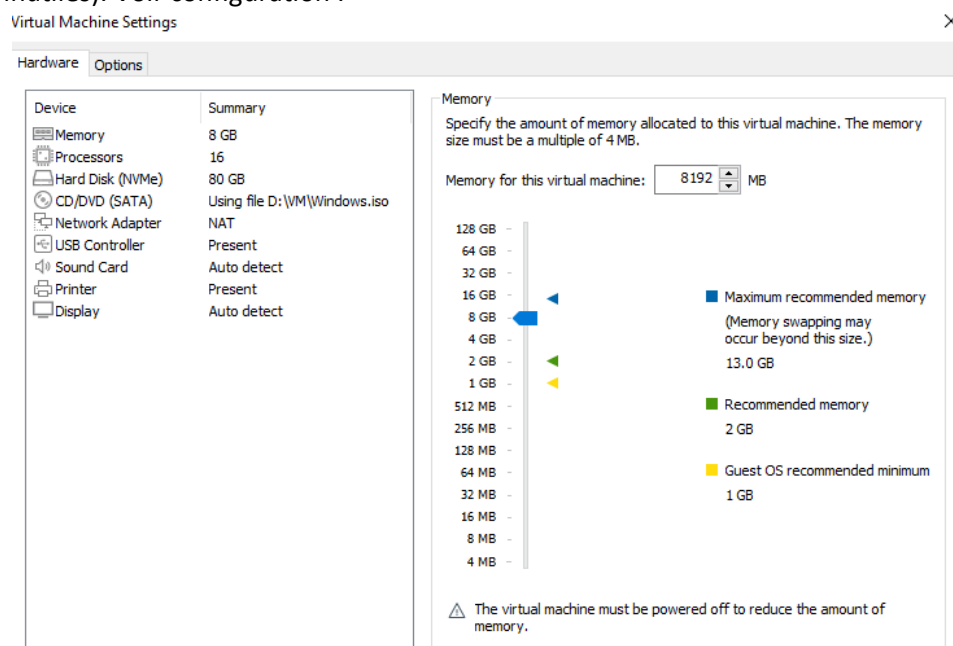
Tout d'abord il faudra nous munir de quelques outils.

Bibliothèques Python à installer :

- easyOCR
- matplotlib
- numpy
- imutils
- PIL
- Pytesseract

I / Mieux vaut VM que guérir

Pour plus de sûreté, on fera nos tests sous une VM de votre choix, le mien s'est tourné vers WIN 10 mais la prochaine fois, j'utiliserai Linux puisque le disque dur se remplit trop vite sur Windows (trop de choses inutiles). Voir configuration :



II / Application sur une plaque non conforme, ou ancienne

Nous partons donc d'une photo de voiture possiblement un grand angle. Nous devons centrer et croper l'image sur la plaque d'immatriculation pour que la Tesseract (l'IA permettant de lire un texte d'une image) ne soit pas perturbée dans son traitement de l'image.



On arrive maintenant au stade de la reconnaissance des formes pour chaque caractère par Tesseract.

Le problème étant que Tesseract semble uniquement pouvoir détecter un texte noir sur fond blanc. Nous devons alors inverser le noir et le blanc, on obtient :



La nuance de noir reste trop faible pour que Tesseract ne distingue les caractères. Pour résoudre ce problème il faudrait donc intensifier la puissance du noir sur l'image.

Pour le moment certaines plaques d'immatriculation ne sont pas lisibles par notre IA. Voyons maintenant un cas où tout fonctionne.

III / Déparasitage et test concluant

Imaginons, notre caméra capture l'image suivante :



Malheureusement un texte parasite s'est glissé sur notre image. Bien heureusement le modèle d'IA d'easyOCR qui nous permet de croper l'image sur une plaque d'immatriculation est correctement entraîné à cet effet.

On remarque qu'après traitement le texte « légal ? » a été totalement mis à l'écart lors du traitement.

PD-909-LV

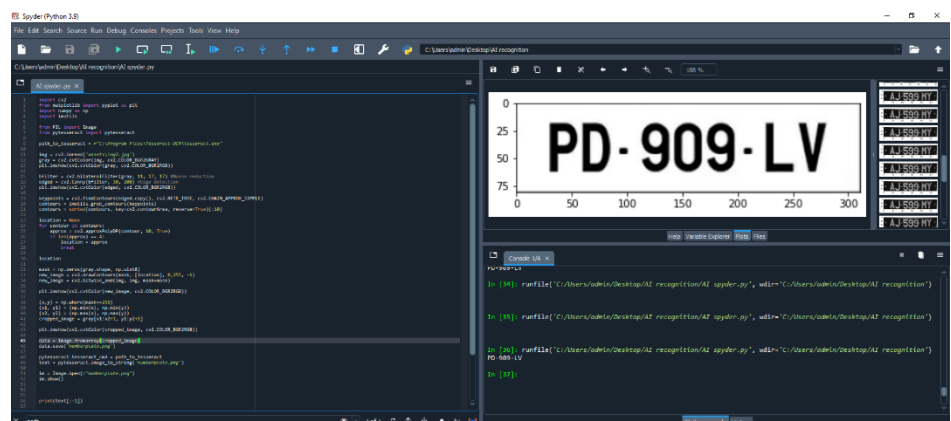
```
In [36]: runfile('C:/Users/admin/Desktop/AI_recognition/AI_spyder.py', wdir='C:/Users/admin/Desktop/AI_recognition')
PD-909-LV
```

De plus le texte a correctement été extrait de l'image.

On arrive donc finalement à un résultat concluant, l'IA arrive à exploiter une photo, croper l'image et extraire les caractères se trouvant sur la plaque d'immatriculation.

Code exploité lors de la réalisation se trouve sur mon Github :

<https://github.com/P3mast/AI-numberplate-recognition>



Conclusion

La reconnaissance d'image nous a permis d'économiser énormément de ressources, au lieu de placer un détecteur sous chaque place de parking nous pouvons comparer la plaque d'immatriculation d'un véhicule sortant par rapport à ceux sur le parking. De cette manière nous pouvons savoir avec précision le nombre de places restantes sur un parking.

Le data wrangling peut-être une méthode que l'on pourra appliquer sur notre dispositif physique pour économiser toujours plus sur le coût du matériel, les besoins énergétiques et la puissance de calcul nécessaire au bon fonctionnement du système.

De plus les services proposés par AWS offriront un moyen rapide et à moindres coûts pour étendre le stockage de l'entreprise.

Enfin nous avons vu comment les techniques web nous permettrons de développer une application multiplateforme et à moindre coût pour pouvoir tenir informer les utilisateurs sur l'évolution du nombre de places sur un parking.

Sitographie :

What do you mean by "AI?" :

https://linas.org/mirrors/nars.wang.googlepages.com/2007.06.27/wang.AI_Definitions.pdf

4 types of artificial intelligence:

https://www.youtube.com/watch?v=Whpcb-gClBY&ab_channel=BernardMarr

Sophia robot:

[https://fr.wikipedia.org/wiki/Sophia_\(robot\)](https://fr.wikipedia.org/wiki/Sophia_(robot))

Conscience et philosophie :

<https://la-philosophie.com/la-conscience-philosophie>

Machine Learnia :

<https://www.youtube.com/channel/UCmpptkXu8ilFe6kfDK5o7VQ>

Gradient descendant :

<https://ledatascientist.com/gradient-descent-lalgorithme-du-gradient/>

Transposée de matrice :

<https://www.bibmath.net/dico/index.php?action=affiche&quoi=.t/transposee.html#:~:text=Soit%20A%20une%20matrice%20de,des%20lignes%20et%20des%20colonnes.>

Opérations sur les matrices :

<https://www.alloprof.qc.ca/fr/eleves/bv/mathematiques/les-operations-sur-les-matrices-m1467>

Tableaux et calcul matriciel avec NumPy :

<https://courspython.com/tableaux-numpy.html>

La confusion avec l'IA :

<https://possibility.teledyneimaging.com/a-farewell-to-ai-confusion/>

Les fantômes de Pac-Man :

<https://www.etaletaculture.fr/geek/les-4-fantomes-de-pac-man-ont-un-secret/>

La qualité des données et projets IA :

<https://www.lemagit.fr/conseil/9-problemes-de-qualite-des-donnees-qui-menacent-vos-projet-IA>

Histoire de l'IA :

<https://www.coe.int/fr/web/artificial-intelligence/history-of-ai#:~:text=La%20paternit%C3%A9%20du%20terme%20%C2%AB%20IA,instant%2C%20accomplies%20de%20fa%C3%A7on%20plus>

La bibliothèque tensorflow :

https://www.tensorflow.org/api_docs/python/tf/shape

Les tenseurs :

<https://ichi.pro/fr/qu-est-ce-que-le-tensor-dans-le-deep-learning-131678050854049>

Bibliothèque easyOCR :

<https://www.analyticsvidhya.com/blog/2021/06/text-detection-from-images-using-easyocr-hands-on-guide/>

Web app ou application native :

<https://www.anthedesign.fr/creation-de-sites-internet/web-app/>

Web app ou application native :

<https://www.app-press.com/blog/web-app-vs-native-app>

Mobile Apps - Web vs. Native vs. Hybrid :

https://www.youtube.com/watch?v=ZikVtdopsfY&t=186s&ab_channel=TraversyMedia

Cloud AWS :

[https://www.lebigdata.fr/amazon-cloud-amazon-web-services#Amazon Web Services quels sont les principaux services AWS](https://www.lebigdata.fr/amazon-cloud-amazon-web-services#Amazon%20Web%20Services%20quels%20sont%20les%20principaux%20services%20AWS)

Comment utiliser AWS :

<https://www.commentcoder.com/aws-amazon-web-services/>

AWS subscription :

<https://aws.amazon.com/fr/>

Bibliothèque tesseract tuto :

https://www.youtube.com/watch?v=62JZImLe6Qw&ab_channel=JV

Documentation PIL :

<https://www.geeksforgeeks.org/python-pil-image-open-method/>

Invert colors of image :

<https://stackoverflow.com/questions/2498875/how-to-invert-colors-of-image-with-pil-python-imaging>

Check whether a string variable is empty or not :

<https://stackoverflow.com/questions/9926446/how-to-check-whether-a-string-variable-is-empty-or-not>

Inverser noir et blanc :

<https://stackoverflow.com/questions/28084908/how-to-invert-black-and-white-with-scikit-image>

Extraire le texte d'une image :

<https://www.geeksforgeeks.org/how-to-extract-text-from-images-with-python/amp/>

Installation Pytesseract :

<https://github.com/madmaze/pytesseract>

Tableau numpy vers une image :

<https://www.geeksforgeeks.org/convert-a-numpy-array-to-an-image/>