

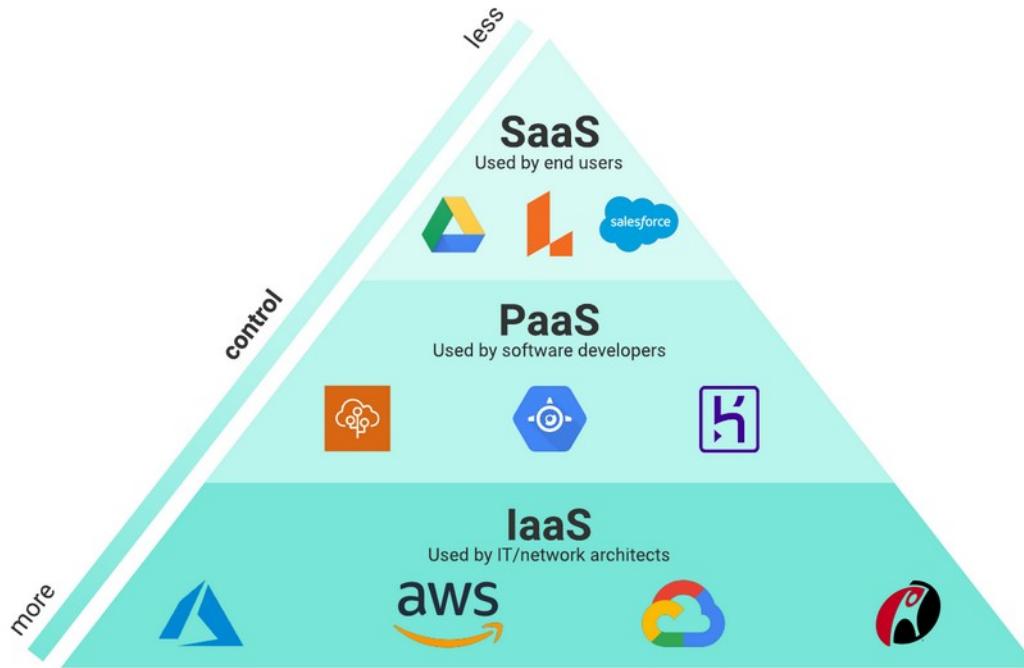
# Cloud Computing



# Une définition

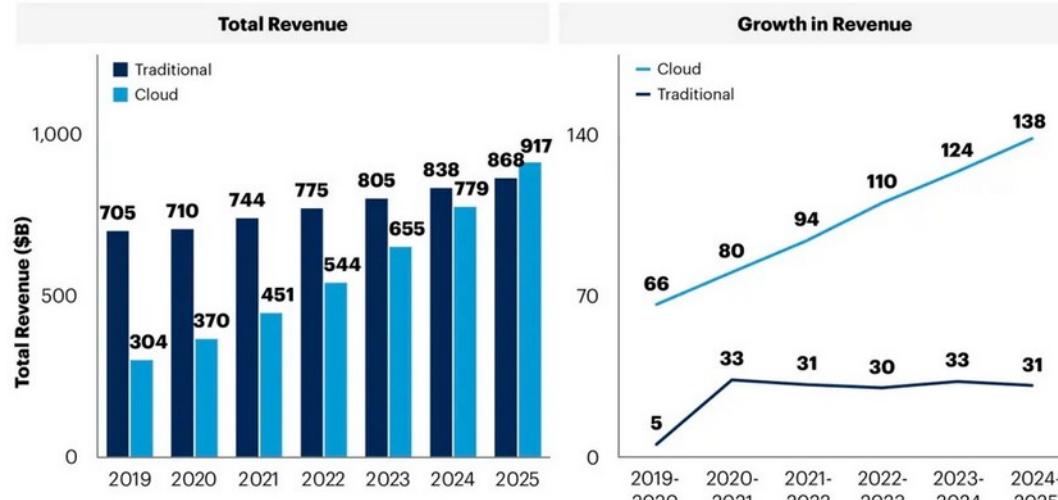
Le cloud computing, ou l'informatique en nuage, désigne un modèle permettant l'accès à des ressources informatiques, telles que des serveurs, des bases de données, des réseaux, des logiciels, et bien d'autres, via Internet. Plutôt que de posséder et de maintenir physiquement ces ressources en interne, les utilisateurs peuvent les louer ou les utiliser à la demande auprès de fournisseurs de services cloud. Cela permet une flexibilité accrue, une évolutivité facile et des économies potentielles de coûts, car les utilisateurs ne paient que pour les ressources qu'ils consomment réellement.

# Pyramides des services



# Cloud computing PDM

Figure 1: Sizing Cloud Shift, Worldwide, 2019 – 2025



Source: Gartner  
758067\_C

Gartner

# Pyramides des services

Worldwide Public Cloud Services Revenue and Year-over-Year Growth, Calendar Year 2020 (revenues in US\$ billions)

Segment	2020 Revenue	Market Share	2019 Revenue	Market Share	Year-over-Year Growth
IaaS	\$67.2	21.5%	\$50.2	19.9%	33.9%
SaaS – System Infrastructure Software	\$49.2	15.7%	\$40.2	16.0%	22.4%
PaaS	\$47.6	15.2%	\$36.1	14.4%	31.8%
SaaS – Applications	\$148.4	47.5%	\$125.2	49.7%	18.6%
<b>Total</b>	<b>\$312.4</b>	<b>100%</b>	<b>\$251.7</b>	<b>100%</b>	<b>24.1%</b>

Source: IDC Worldwide Semiannual Public Cloud Services Tracker, 2H20

## Avantages

- Pas d'investissements dans le matériel (Serveur+peering)
- Moins d'immobilisation financière
- Pas de mise à jour OS ou applications tierces
- Scaling facilité
- Pay as you go

## Inconvénients

- Pas nécessairement moins chère que le on premise
- Données sensible hébergées par un tiers
- Migration compliquée et onéreuse
- Nécessité d'une connexion internet pour accéder aux services

# Usages Cloud Computing

- Hébergement serverless
- Content Delivery Networks
- Calculs intensifs à la demande
- IA
- Stockage des données
- Opérations spéciales (black friday, St Valentin)

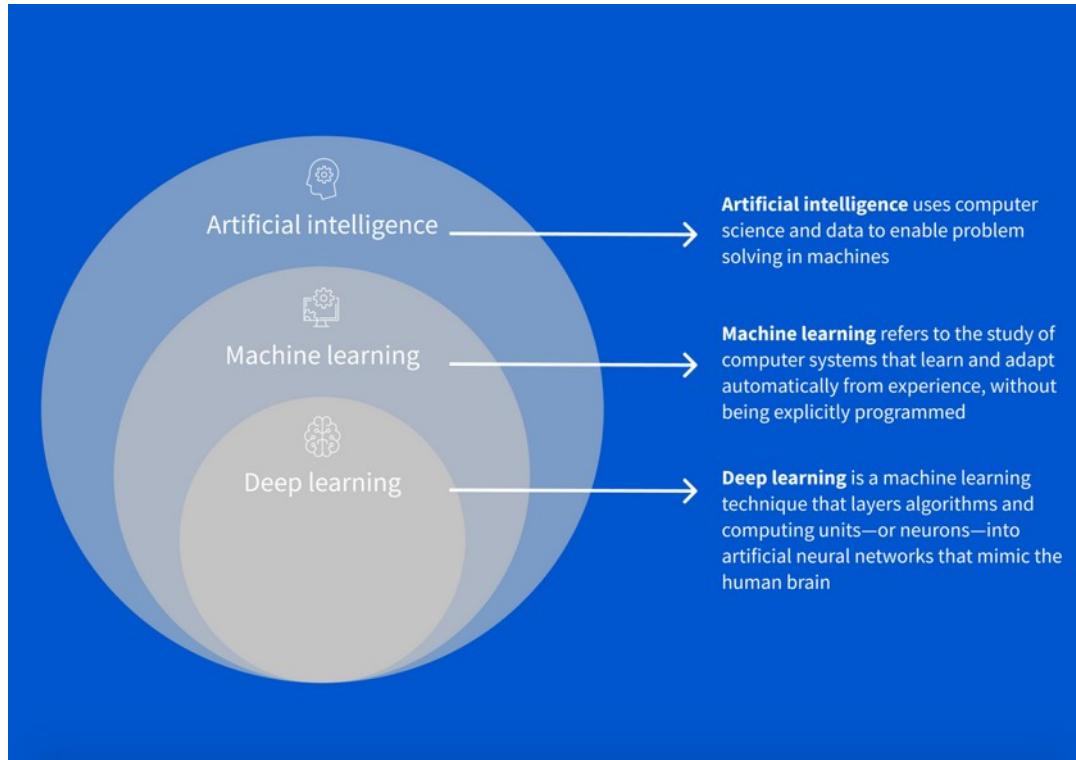
# Hébergement et déploiement

- Ansible (Python)
- Capistrano (Ruby)
- Kubernetes

# IAM : Identity & Access Management

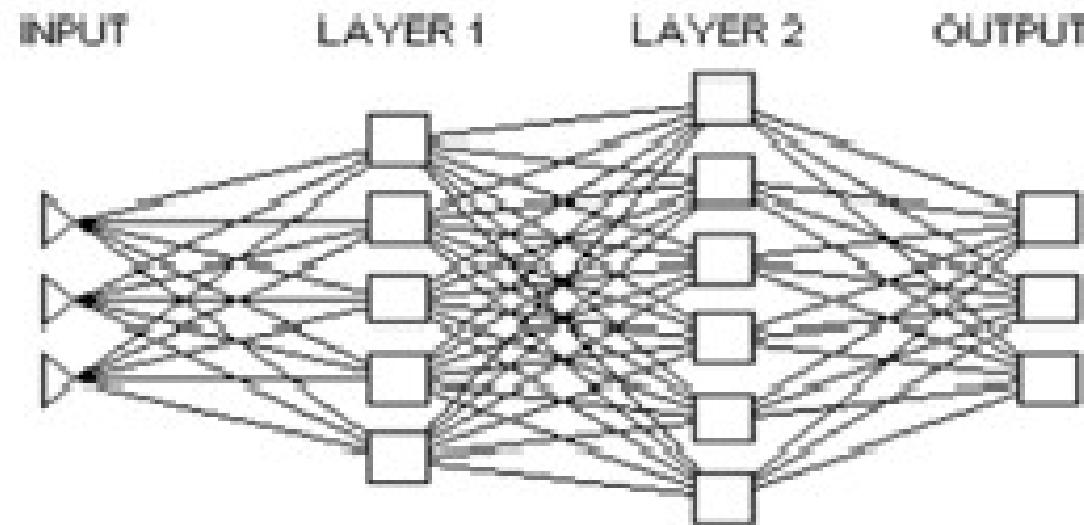
- Identity & Access Management
- MFA et OTP (One Time Password)
  - TOTP
  - HOTP
  - SMS ou mail OTP

# IA : Usage majeur du cloud computing

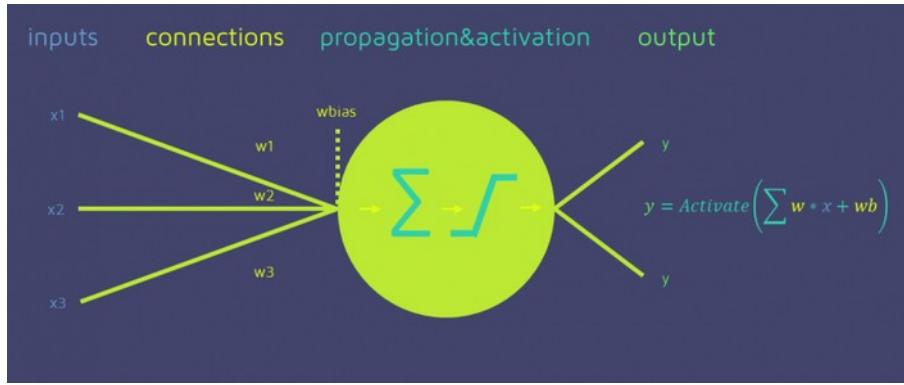


- IA :Utilisation math. bayésienne
- ML :Le programme peut apprendre et s'adapte sans être explicitement programmé. Il suit des règles données (algorithme) :
  - Ex : spotify
- DL :Le DL utilise un grand nombre de données diverses. Les données passent par des couches d'algorithmes (des neurones) et on parle de deep neural networks
  - Ex : ChatGPT

# Réseau de neurones



# Réseau de neurones



- Un neurone est une fonction qui prend en entrée la sortie des couches précédentes
- La fonction d'un neurone  
 $y_i(t) = F_i(f_i[a_i(t - 1), \sigma_i(w_{ij}, x_j(t))])$

$x_j(t)$  Paramètres d'entrée

$w_{ij}$ : Poids du paramètre d'entrée

$\sigma_i(w_{ij}, x_j(t))$  Fonction de propagation. Très souvent :  $h_i(t) = \sum_j w_{ij} \cdot x_j(t)$

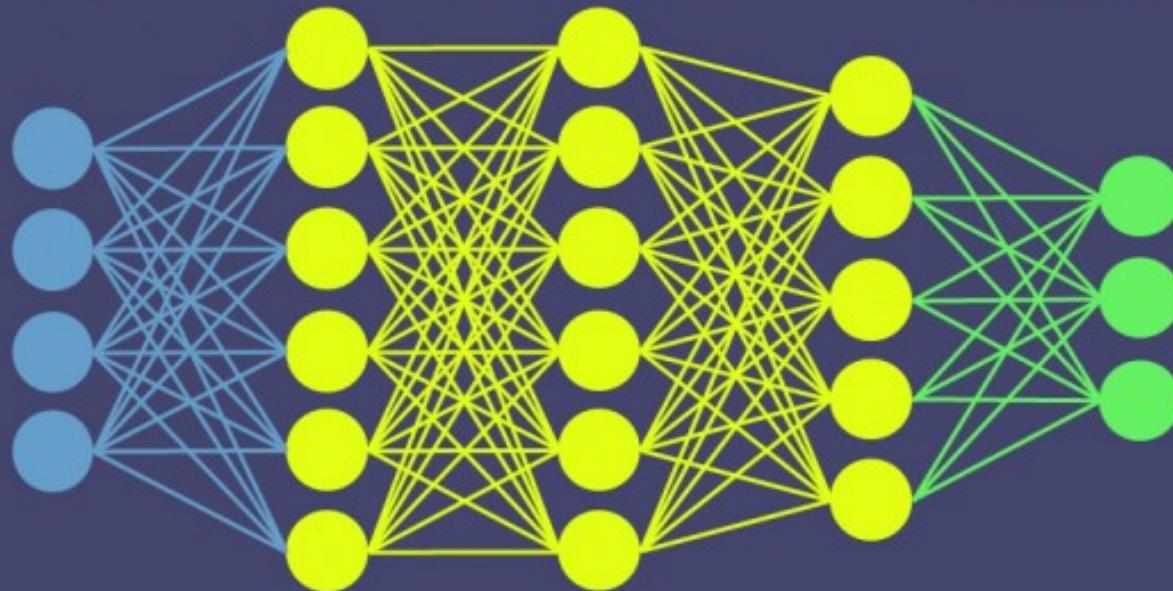
$f_i(a_i(t - 1), h_i(t))$  Fonction d'activation. Permet de réguler le poids d'une entrée précédente pour éviter les valeurs extrême

$F_i(a_i(t))$  Fonction de sortie.

input layer

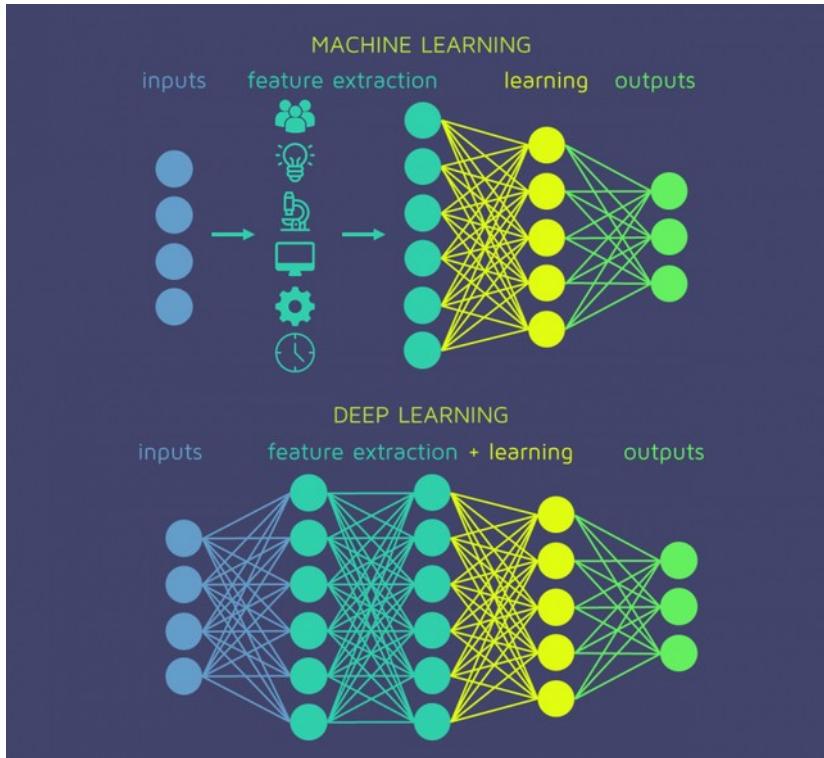
hidden layers

output layer



- Couche d'entrée : Détermine le nombre de paramètres gérés par le modèle (ex : plusieurs millions pour ChatGPT)
- Couches intermédiaires : Détermine la profondeur de l'apprentissage du modèle
- Couche de sortie : Le résultat du modèle

# Génération d'un modèle



- Machine Learning : C'est un humain qui indique au modèle les caractéristiques de ce que l'on souhaite modéliser
- Deep Learning : A partir d'un très grand nombre de données en entrée, Le modèle détermine lui-même les caractéristiques

- Lorsqu'on utilise le modèle, on utilise les poids  $w_{ij}$ : qui ont été calculés pendant la phase d'apprentissage

## Sous apprentissage (under-fitting)

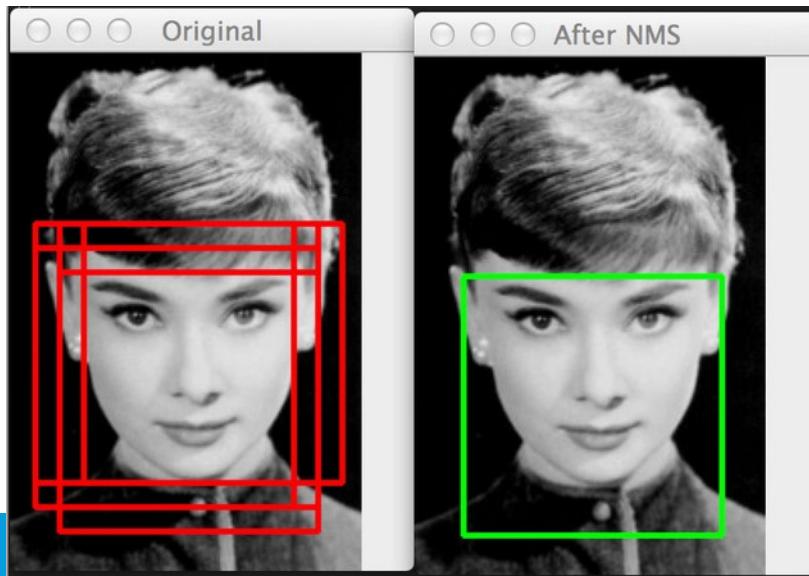
- Le modèle généralise mal les nouvelles données (erreurs de détection)
- Il faut ajouter de nouvelles données d'entraînements

## Sur apprentissage (over-fitting)

- Le modèle ne généralise plus (il ne détecte plus des données similaires)
- Diversifier les données ou limiter les données d'entraînements

# Application : Détection d'objets avec R-CNN

- R-CNN (Region-based Convolutional Neural Network)
- L'algorithme génère d'abord un ensemble de "régions candidates". Ce sont des régions contenant potentiellement des objets



- Exemples d'algorithmes
  - Non Maximum Suppression
  - Bounding Box Regression

# R-CNN

- L'algorithme de réseaux de neurone proprement dit est ensuite appliqué sur chacune des régions
- Le réseau de neurones convolutif (déjà entraîné) va définir à quel classe appartient la région
  - L'entraînement est fait à partir d'un grand nombre d'images taguées comme **imageNet** (160 Go)
  - Une convolution est une opération mathématique utilisée dans le traitement du signal et le traitement d'images

# R-CNN

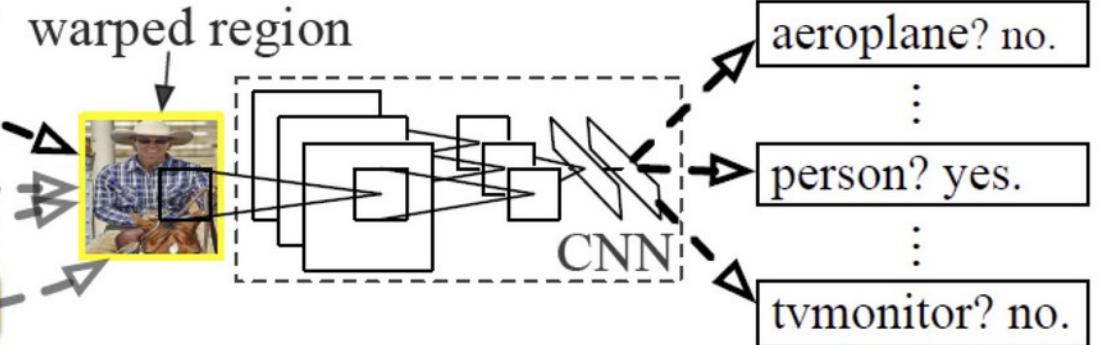
## R-CNN: *Regions with CNN features*



1. Input image



2. Extract region proposals (~2k)

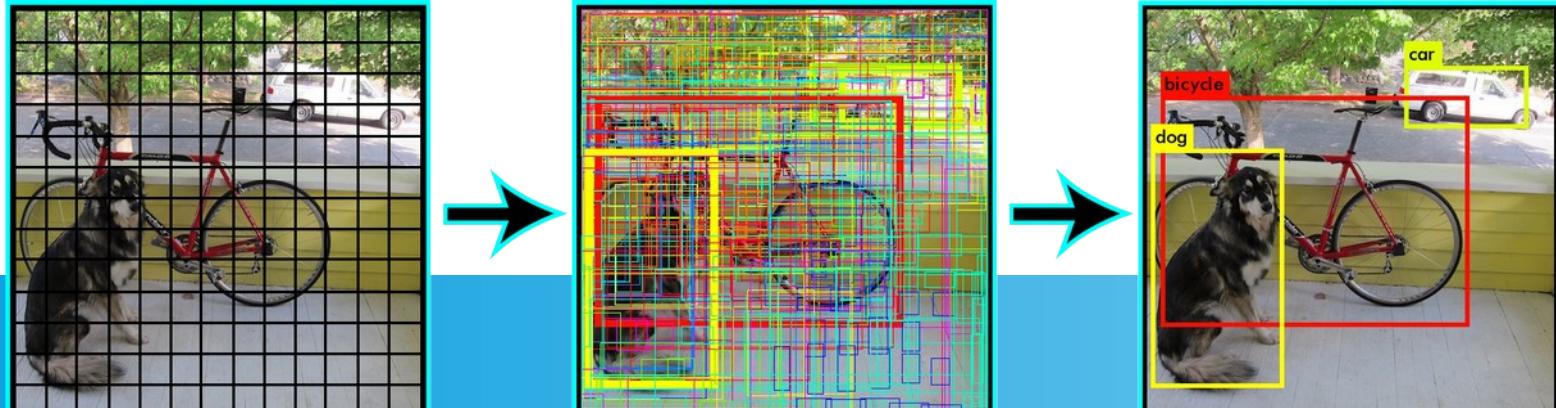


3. Compute CNN features

4. Classify regions

# Exemple : YoloV2

- Yolo est un algorithme de détection d'objets en temps réel. Il est environ 1000 fois plus rapide que R-CNN
- L'optimisation se fait au niveau de la sélection des régions
  - Un premier réseau de neurones cherche la caractéristique la plus probable sur une région de l'image totale (pas de découpe)
  - Pour la caractéristique la plus probable, l'algorithme recherche la boîte englobante et lui attribue ladite caractéristique



# YoloV2 : Utilisation

- YoloV2 est utilisée 19 couches de convolution (basé sur l'algorithme darknet-19)
- Exemple Vidéo Bond
- Utilisation d'un modèle pré-entraîné

```
git clone https://github.com/pjreddie/darknet  
cd darknet  
make
```

- Sous WSL2, vérifier que cc et gcc sont installés
  - gcc --version
  - Si besoin, les installer sudo apt build-essential

# YoloV2

- Télécharger les pondérations du modèle (les données d'entraînement)

```
wget https://pjreddie.com/media/files/yolov2.weights
```

- Lancer la détection d'objets sur une image

```
./darknet detect cfg/yolov2.cfg yolov2.weights data/dog.jpg
```

- Voir le résultat
  - Afficher l'image predictions.jpg qui a été générée
  - Par défaut, cette version ne conserve que les objets avec un pourcentage de prédiction de plus de 25 %

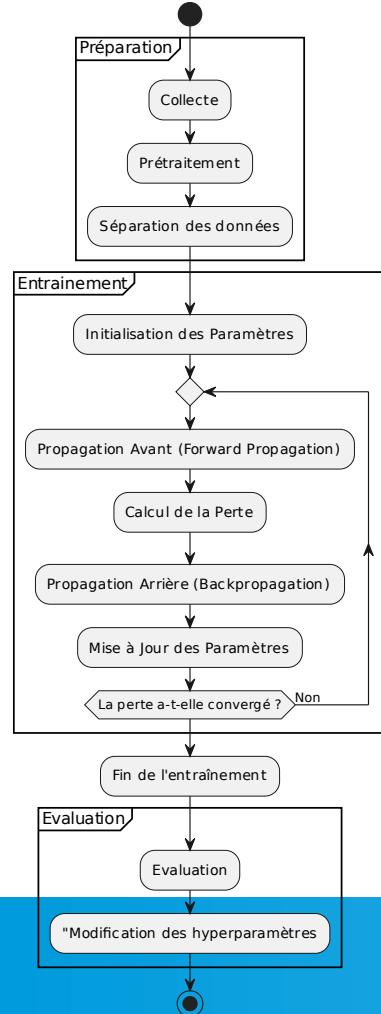
# Entrainement

- L'entraînement consiste à donner des informations pour effectuer des prédictions sur des données inconnues au préalable.
- Le processus consiste en :
  - Collecter les données (dataset représentatif)
  - Nettoyage, valeurs manquantes, normalisation
  - Diviser les données en 2 parties (training set et test set)
  - Choix du modèle (Utiliser un ou plusieurs algorithmes de machine learning)

# Yolo V2

- Il est possible d'entraîner le modèle sur des données spécifiques de l'entreprise
  - Comptage et suivi des produits
  - Inspection de la qualité
  - Contrôle d'accès
  - Déetecter et localiser des pièces ou des composants dans un environnement complexe

# Workflow machine learning



- collecte
  - Nettoyage
  - Faire les datasets d'entrainements et de tests
  - Définir les hyperparamètres
- 
- Initialiser les poids et biais du modèle
  - Propagation avant : Passer les données d'entraînement sur le modèle
  - Calcul de la perte : Permet de calculer l'erreur entre les prédition du modèle et les valeurs réelles (cf page suivante)
  - Propagation arrière : Calculer comment minimiser l'erreur trouver avec les erreurs précédentes (cf page suivante : la descente de gradient)
  - Mettre à jour les poids du modèle
- 
- Valider le modèle avec les données de tests

# Fonction de coût

- La fonction de coût dépend des algorithmes utilisés et des facteurs importants pour une bonne détection
- Exemple pour Yolov2, il y a 3 éléments très importants
  - Perte sur la localisation. Le modèle ne détecte pas bien les boîtes englobantes des objets
  - Perte de la confiance dans l'existence d'un objet dans une boîte englobante (1 si un objet est présent et 0 sinon)
  - Perte de la classification. Le modèle se trompe sur la classification (Le résultat du modèle donne en fait une probabilité pour chacune des classes)

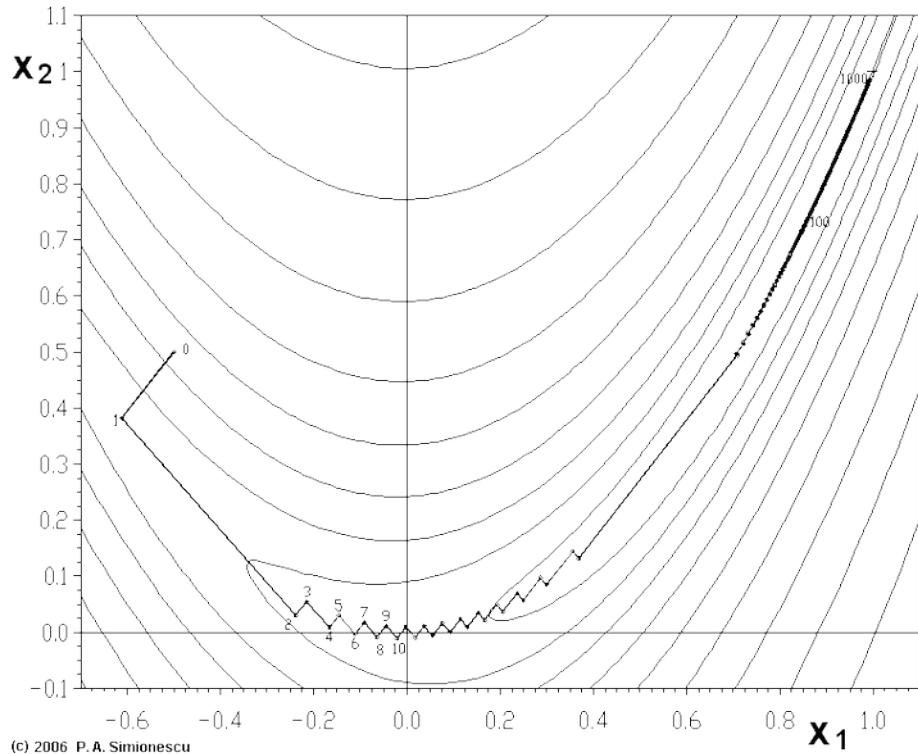
# Fonction de coût : traduction mathématique

$$\begin{aligned} \text{Loss} = & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} (\hat{C}_i - C_i)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{noobj}} (\hat{C}_i - C_i)^2 \\ & + \sum_{i=0}^{S^2} 1_i^{\text{obj}} \sum_{c \in \text{classes}} (\hat{P}_i(c) - P_i(c))^2 \end{aligned}$$

Où :

- $\lambda_{\text{coord}}$  et  $\lambda_{\text{noobj}}$  sont des hyperparamètres qui contrôlent l'importance relative de la perte de localisation et de la perte de confiance pour les boîtes ne contenant pas d'objets respectivement.
- $1_{ij}^{\text{obj}}$  est un indicateur binaire qui vaut 1 si la boîte  $j$  de la cellule  $i$  contient un objet.
- $1_{ij}^{\text{noobj}}$  est un indicateur binaire qui vaut 1 si la boîte  $j$  de la cellule  $i$  ne contient pas d'objet.
- $\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i$  sont les coordonnées prédites de la boîte englobante.
- $x_i, y_i, w_i, h_i$  sont les coordonnées réelles de la boîte englobante.
- $\hat{C}_i$  est la confiance prédictive pour la boîte englobante  $i$ .
- $C_i$  est la confiance réelle (1 si un objet est présent, 0 sinon).

# Descente de gradient



- La descente de gradient consiste à minimiser la différence entre la prédition A et la prédition B

# Application

- Nous allons entraîner un modèle capable de différencier 10 objets différents. On va utiliser les images labelisées et découpées du CIFAR (Canadian Institute For Advanced Research)
- Dataset de 60000 images
- `wget https://pjreddie.com/media/files/cifar.tgz`
- Dans le dossier cifar, générer les fichier train.list et test.list qui contiennent les noms des fichiers png présents dans les dossiers train et test
  - `find `pwd`/train -name \*.png > train.list`
  - `find `pwd`/test -name \*.png > test.list`

# Fichier de configuration pour Yolo

- Créer un fichier cifar.data dans le dossier cfg
- Ce fichier décrit le dataset

- `classes=10`
- `train = data/cifar/train.list`
- `valid = data/cifar/test.list`
- `labels = data/cifar/labels.txt`
- `backup = backup/`
- `top=2`

Nombre de classes du dataset

On conserve les 2 meilleures prédictions

# Définition du réseau

- Darknet permet de construire le modèle souhaité. Il est décrit dans un fichier de configuration
- Ce sont les hyperparamètres du modèle
  - Batch size (le nombre d'images utilisés à chaque itération)
  - Max batches (le nombre maximal de batch)
  - classes (le nombre de classes du modèle)
  - .....
- Le développeur ajustera ces valeurs pour obtenir le modèle le plus précis possible selon le résultat qu'il souhaite obtenir

- L'entraînement peut être long. Il dépend
  - Présence d'un GPU
  - Nombre de données d'entraînements
  - Vitesse d'exécution des algorithmes
- Créer un fichier bash permettant de connaître le temps d'exécution du script
- L'apprentissage se lance avec la commande
  - `./darknet classifier train cfg/cifar.data cfg/cifar_small.cfg`

- Lorsqu'on lance darknet, la progression de l'apprentissage s'affiche à l'écran

```
2, 0.005: 2.392326, 2.330726 avg, 0.099840 rate, 0.976985 seconds, 256 images
3, 0.008: 2.379231, 2.335576 avg, 0.099760 rate, 0.942072 seconds, 384 images
4, 0.010: 2.282773, 2.330296 avg, 0.099680 rate, 0.984935 seconds, 512 images
5, 0.013: 2.283662, 2.325632 avg, 0.099601 rate, 0.961632 seconds, 640 images
```

- Numéro de l'itération
- Progression au sein du batch en cours
- Perte (l'objectif est de minimaliser ce chiffre)
- Perte moyenne (avg)
- Taux d'apprentissage

# Apprentissage supervisé/non supervisé

- Dans le cas d'un apprentissage supervisé, les hyperparamètres donnent clairement les objectifs de l'apprentissage (nb classes, batches,...)
- Dans le cas d'un apprentissage non supervisé, on essaie de découvrir des corrélations non évidentes dans le dataset

Caractéristique	Apprentissage supervisé	Apprentissage non supervisé
<b>Données</b>	Étiquetées (entrées et sorties connues)	Non étiquetées (seulement entrées)
<b>Objectif</b>	Prédiction, classification, régression	Découverte de motifs, segmentation
<b>Exemples d'algorithmes</b>	Régression linéaire, SVM, réseaux neuronaux	K-means, PCA, hiérarchique
<b>Applications</b>	Classification de spam, prévisions de vente	Segmentation de clients, réduction de dimensionnalité

# Algorithmes statistiques

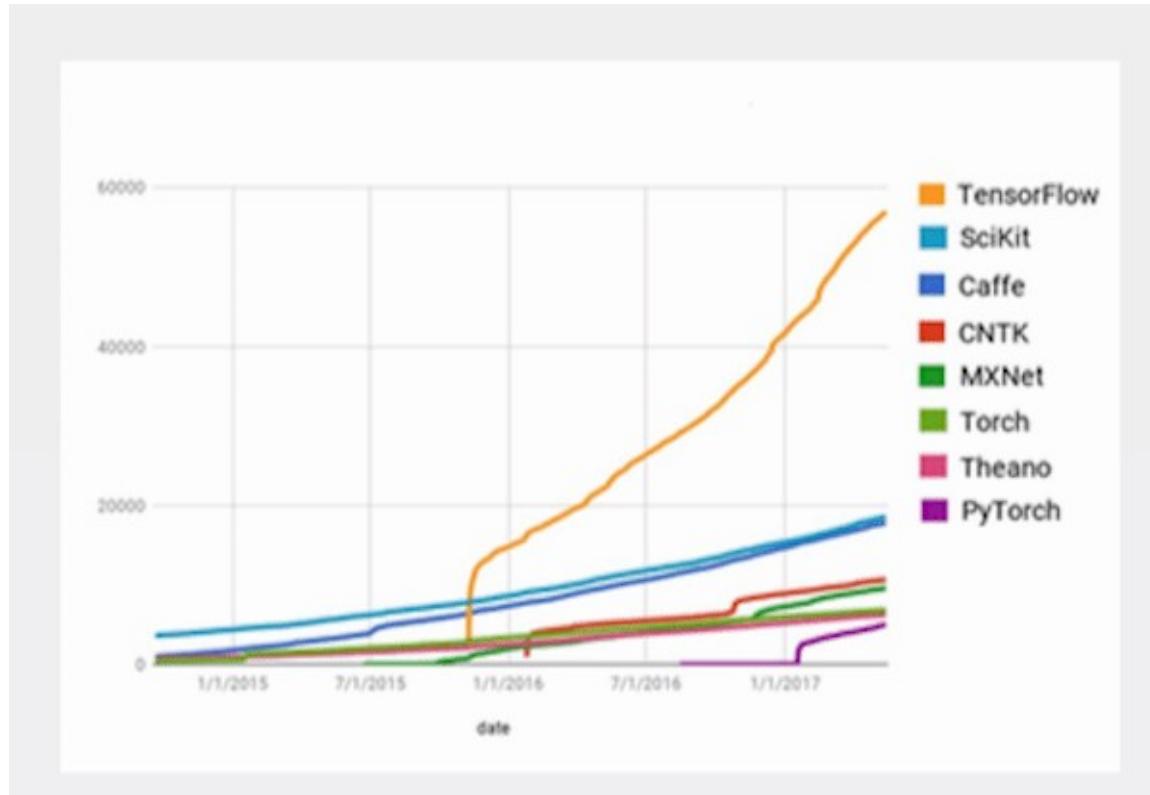
- Supervisés

- Régression linéaire
- Régression polynomiale
- Arbres de décision et forêt aléatoires
- Machines à vecteurs de support (SVM)
- Réseaux de neurones

- Non supervisés

- K-means
- DBSCAN
- Réseaux de neurones auto-encodeurs
- Analyse en composantes principales

# Framework ML



# Analyse de la donnée : kaggle.com

- L'analyse de la donnée est une cible de choix pour le cloud computing
  - Besoin ponctuel de ressources importantes pour l'apprentissage
  - Stockage de données froides
- La data science peut se diviser en 2 catégories
  - Analytics : Analyse et visualisation de la donnée.
  - Compétences
    - Base de la statistique (Bayes)
    - Visualisation de la donnée
  - Predictive : Utilisation et création d'algorithme
    - NLP : Natural Language Processing
    - Deep Learning
    - Reconnaissance d'image

# Projet d'étude de Kaggle Titanic

- L'objectif est de déterminer le modèle (statistique ou machine learning) qui se rapproche le plus de l'observation. En d'autres mots, peut-on trouver un modèle qui explique pourquoi certaines personnes ont été sauvées et d'autres non
- Un projet de data science se déroule souvent avec une partie d'analyse des données
  - EDA : Exploratory Data Analysis
  - Application sur les données de modèles d'analyse existants
  - Recherche éventuelle de nouveaux modèles
  - Validation du modèle ( % de prédiction attendu par le client)

- Recherche de corrélation avec la colonne 'Survived'
- Ajouter une colonne 'IsAlone' à partir des colonnes 'SibSp' et 'Parch'
- Analyse des corrélations
- Supprimer les colonnes sans intérêt pour l'analyse
- Supprimer les colonnes ayant une corrélation forte entre-elles (à l'exception de la colonne 'Survived')
- Supprimer ou compléter les lignes avec des données vides
- Appliquer le modèle sur les données train.csv
- Appliquer le modèle sur toutes les données (train.csv+test.csv)
- Analyser la différence entre les deux

# Bibliothèque Python

impy

andas

ikit-learn (sklearn)

éthodologie (après le nettoyage des données)

Chargement des données

Préparation des données

Visualisation éventuelle

Appliquer un modèle (Ex : model =  
sklearn.linear\_model.LinearRegression())

# Nettoyage des données

- Beaucoup d'algorithmes ne peuvent pas fonctionner avec des valeurs nulles
  - On supprime la ligne ou on affecte une valeur arbitraire (moyenne par exemple)
- Remplacer les valeur textuelle par une donnée chiffrée (Remplacer Vrai par True,...)
- Normaliser les données (Même unité)

# Méthodologie (après le nettoyage des données)

- Chargement des données
- Préparation des données
- Visualisation éventuelle
- Appliquer un modèle (Ex : `model = sklearn.linear_model.LinearRegression()`)
- Entrainer le modèle
  - `model.fit(X, y)`
- Tester le modèle
  - $X = \text{Value}$
  - `model.predict(X)`

- Vérifier la modèle
  - score
- Tester d'autres algorithmes statistiques

# Exercice

- spaceship titanic
- Ecrire le programme permettant d'analyser et de prédire si des passagers ont été transférés dans une autre dimension suite à la collision
- Le programme est commenté
- Vous recherchez les colonnes ayant un intérêt (matrice de corrélation) et vous commentez le résultat (quels sont les données d'intérêts)
- Quel est le meilleur algorithme ? Au delà du score de performance, pourquoi est-ce le meilleur dans le cas du dataset analysé ?