

IA Géosciences - Deep learning: **Introduction**

Romain Wenger (Laboratoire Image Ville
Environnement)

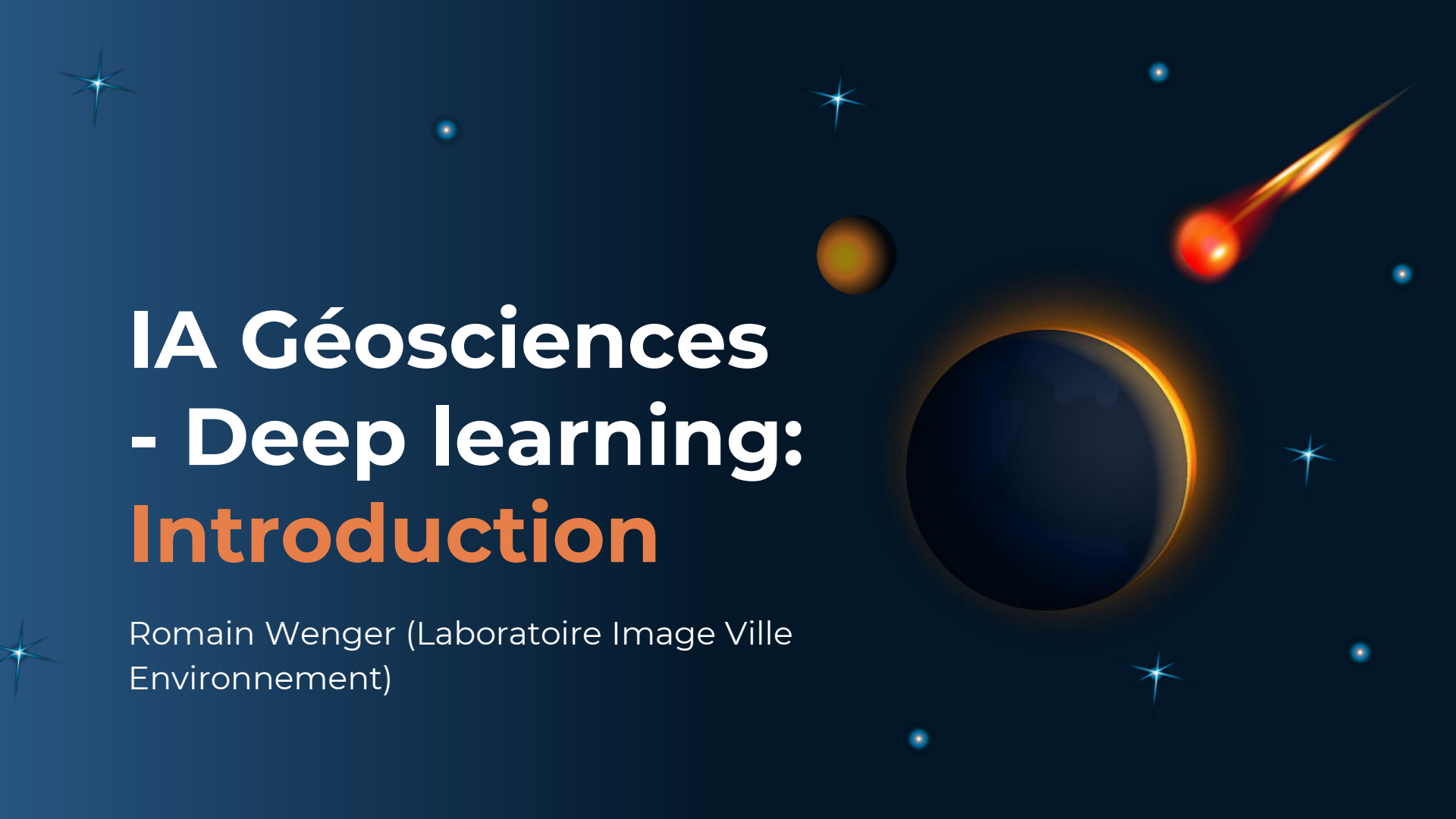


Table des matières

01

La révolution deep learning

Nouvelles découvertes
et accomplissements

02

Historique

Histoire du deep
learning : de 1930 aux
années 2020

03

Apprentissage supervisé

L'abondance des
données d'entraînement
et l'apprentissage
supervisé

Table des matières

01

La révolution deep learning

Nouvelles découvertes
et accomplissements

02

Historique

Histoire du deep
learning : de 1930 aux
années 2020

03

Apprentissage supervisé

L'abondance des
données d'entraînement
et l'apprentissage
supervisé

AlphaGO a battu Hui le champion du monde de GO en 2016

- Programme IA conçu par *Google DeepMind*
- Battu une 2eme fois en 2017
- Premier accomplissement historique de l'IA



Source : MIT Technology Review: Brenden Lake

Traduction automatique (Google Traduction)

- **Artificial Neural Network (ANN)** pour traduire
- Mise en place en novembre 2016
- **160M à 380M** de paramètres



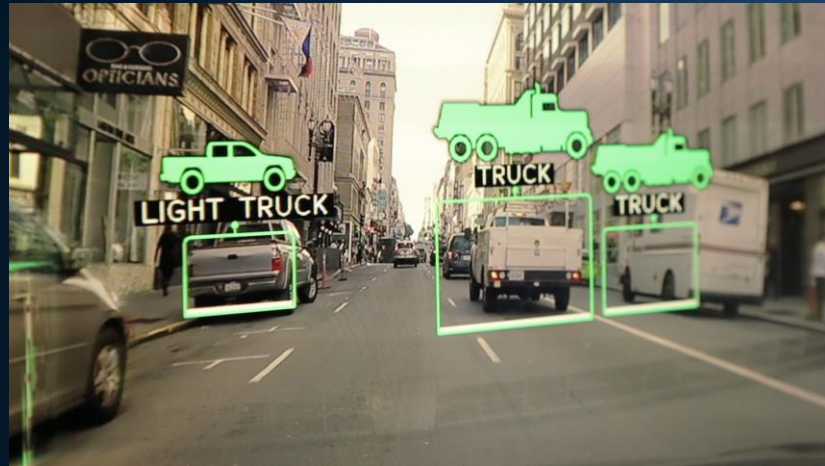
Google's Neural Machine Translation

...reduced translation errors by an average of 60% when compared to the prior Google Translate technology

Source : Dilmanj: Poorya Zeynalzadeh

Véhicules autonomes (Uber, Tesla ...)

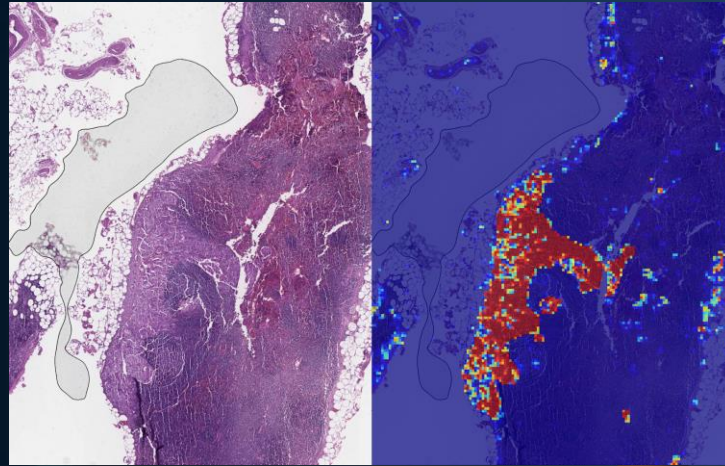
- Détection des objets par deep learning
- Réseaux de neurones embarqués



Source : Verne Global: Nick Dale

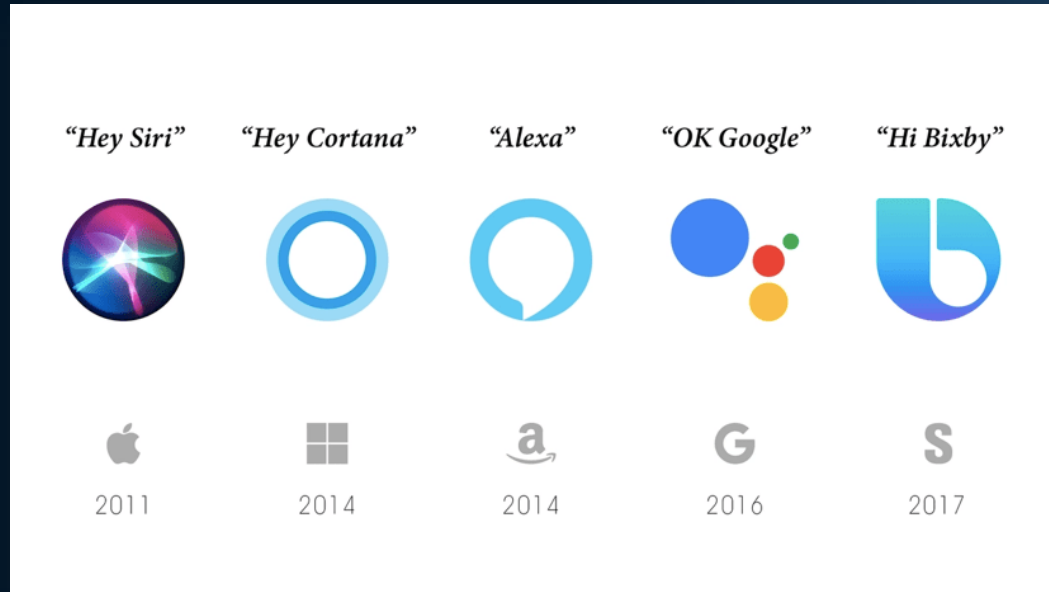
Détection des tumeurs (Google's *Breast Cancer detector*)

- **Identification** des tumeurs sur des coupes histopathologiques
- **Assister** les radiologues dans la prise de décision
- Prise en charge accélérée des pathologies



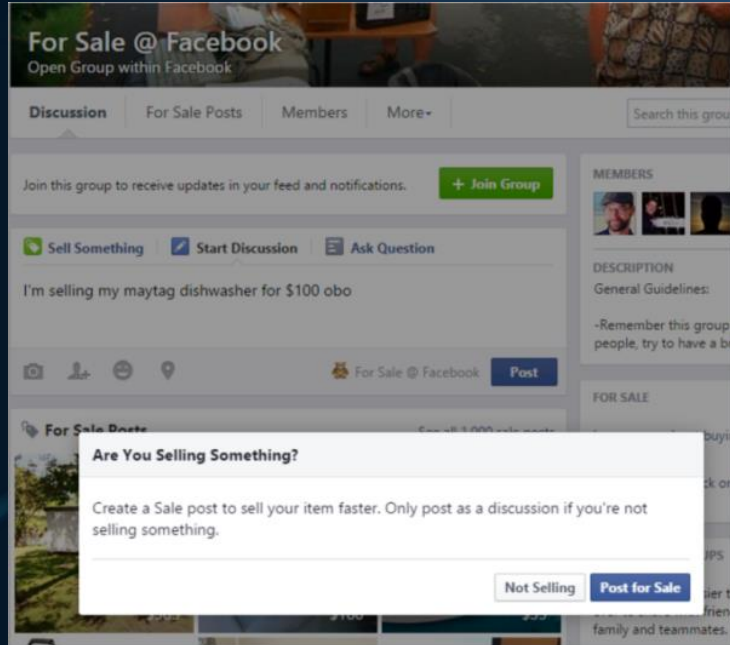
Source : Venture Beat: Kyle Wiggers

Reconnaissance de la parole (Siri, Google Assistant ...)



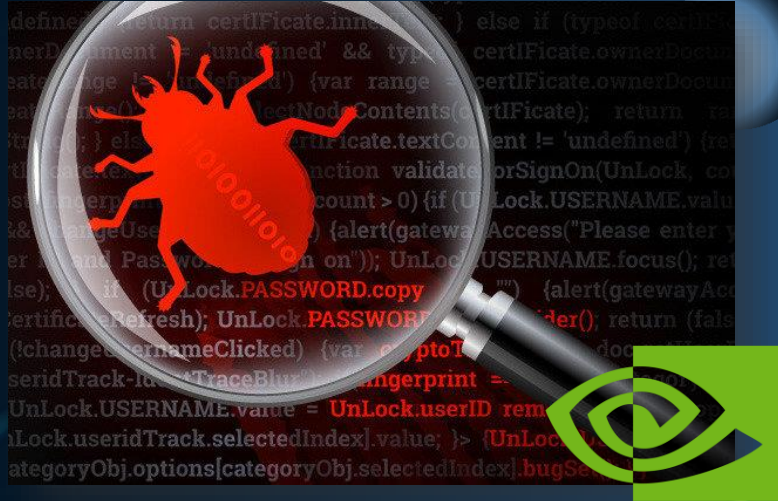
Classification des sentiments (Facebook DeepText)

- **Comprendre** le langage humain
- Proposition **adaptée** aux besoins utilisateurs
- Capable de détecter des photos violentes ou offensantes



Détection des applications malveillantes (Nvidia MalConv)

- Modèle IA pour la detection de **Malware**
- **Grand nombre** de données d'entraînement
- Précision de plus de **98%**



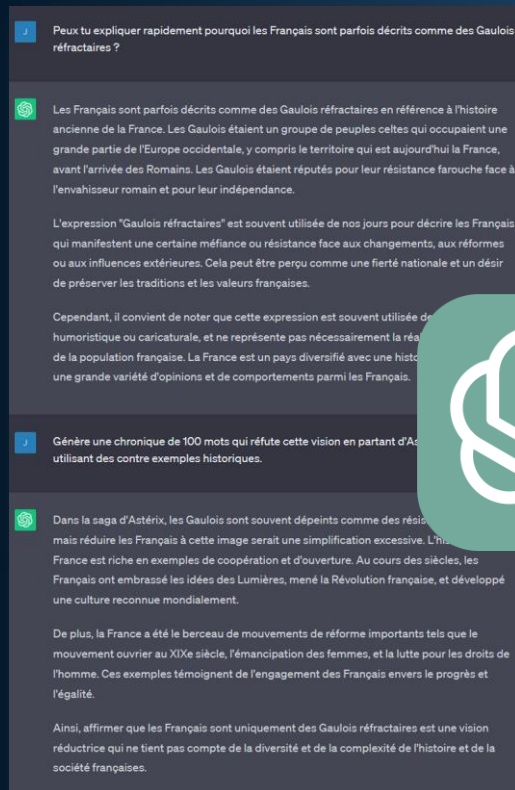
NVIDIA®

... Plus récemment, ChatGPT

- Chatbot **utilisant l'IA et les LLM** (Large Language Model ou Grand Modèles de Langage)
- Apprentissage supervisée et par **renforcement**
- Entraîné sur 10 000 Nvidia V100 -> 10k€ pièce soit **100M€ de GPU**
- **1.76 trillion** de paramètres (pour GPT4)



Et Google traduction en 2017 ?



... Plus récemment, ChatGPT

Vidéo Youtube de Germain Forestier (Professeur en Informatique et IA à l'UHA)



La place de ChatGPT à l'université

The image shows a screenshot of a video presentation. The main part is a white slide with purple and black text. In the top right corner, there is a small inset video of a man with a beard and short hair, wearing a dark shirt, sitting in a blue chair. The slide text is as follows:

Modèles de langage et ChatGPT
Comment vont-ils révolutionner l'Université ?

Pr. Germain Forestier, IRIMAS, Univ. Haute-Alsace
<https://germain-forestier.info/llm.pdf>

September 18, 2023

Table des matières

01

La révolution deep learning

Nouvelles découvertes
et accomplissements

02

Historique

Histoire du deep
learning : de 1930 aux
années 2010

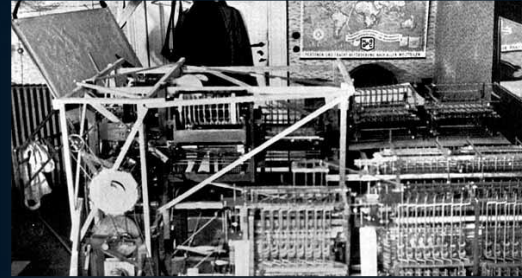
03

Apprentissage supervisé

L'abondance des
données d'entraînement
et l'apprentissage
supervisé

Années 1930s

- Création du premier PC binaire programmable: **Z1**
- Alan Turing se demande :
« **Est-ce-que la machine peut réfléchir ?** »
- Il n'avait pas les machines actuelles pour pouvoir y répondre



A computer would deserve to be called intelligent if it could deceive a human into believing that it was human.

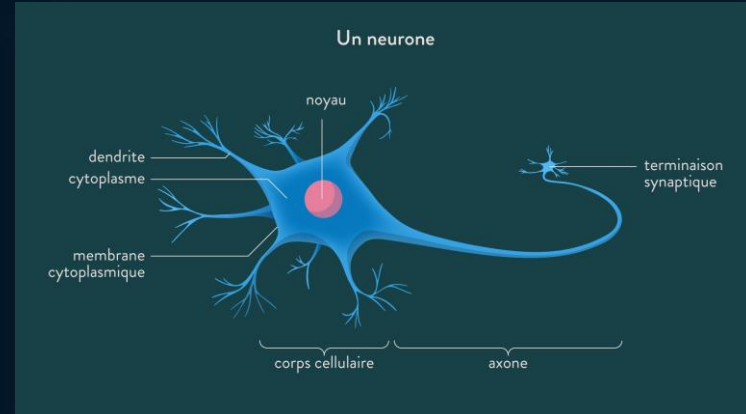
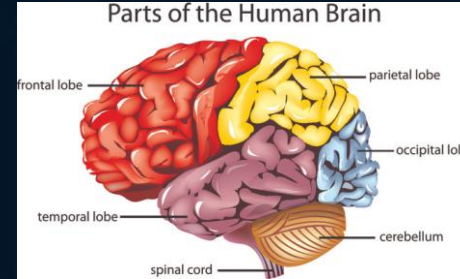
Alan Turing

www.thequotes.in



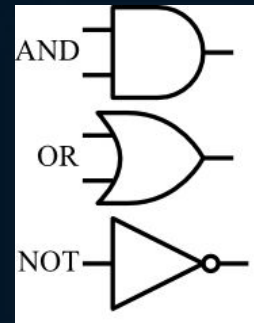
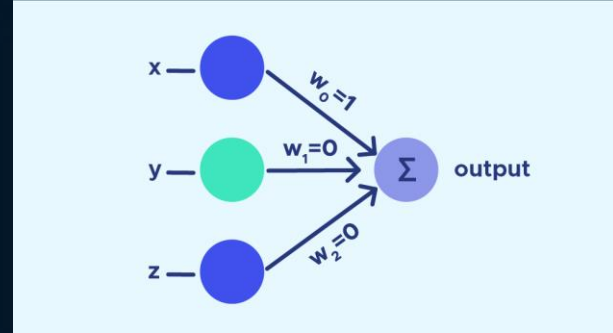
Années 1940s

- La recherche dans les neurosciences a avancée
- On commence à mieux comprendre comment fonctionne le cerveau



Années 1940s

- Des modèles mathématiques simulent le fonctionnement des neurones
- Par exemple pour reproduire les fonctions logiques (AND et OR)
- Toujours pas d'apprentissage : le développeur doit lui-même définir ce que fait chaque neurone

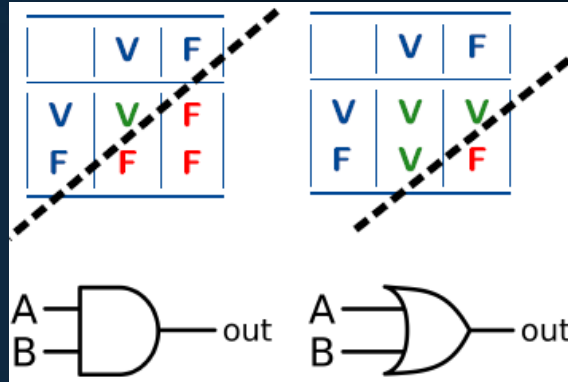






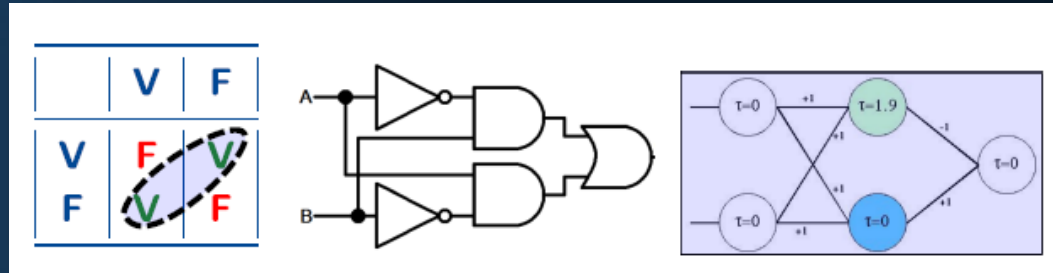
Année 1969

Minsky et Papert ont montré que le Perceptron n'est pas capable d'apprendre des fonctions non linéaires avec les algorithmes d'optimisation limités à l'époque



(Haut) Table de vérité pour AND (gauche) et OR (droite) avec la ligne en pointillés pour la séparation linéaire. (Bas) La fonction logique pour AND (gauche) et OR (droite). La séparation est linéaire, une seule ligne peut séparer les deux classes **Vrai** et **Faux**.

Année 1969 (suite)



(Gauche) la table de vérité de la fonction XOR qui montre qu'un Perceptron, étant linéaire, ne peut pas séparer (avec une seule ligne droite) les instances vraies des instances fausses. **(Milieu)** La fonction XOR en utilisant plusieurs couches de NOT, AND et OR. (Droite) Le Perceptron multi-couches (MLP), réseau profond, qui modélise la fonction XOR.

XOR : « Si A et B sont différents, alors Q est vrai »

Problème : On n'avait pas un algorithme à l'époque pour apprendre ce MLP (le message principal de Minsky et Papert).

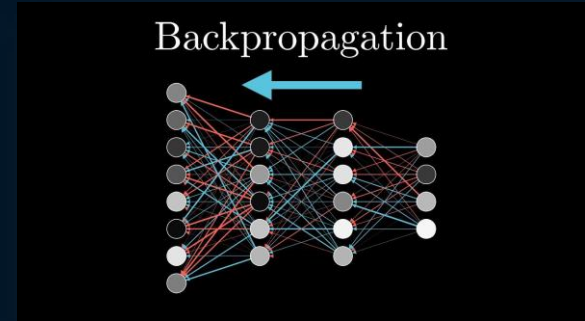
Années 1970s

- Le problème d'apprentissage s'accroît avec Minsky et Papert
- On n'arrivait pas à faire apprendre la machine
- On n'avait ni les **algorithmes** ni les **moyens** à l'époque
- La recherche en IA commence à diminuer et reculer
- On recevait de moins en moins de financement pour l'IA
- Une période connue par **AI-Winter**



Années 1980s

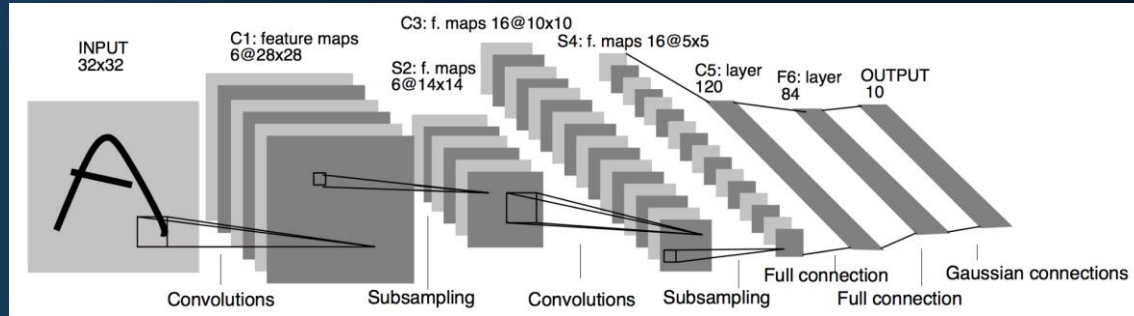
- 1974 Werbos durant sa thèse a proposé l'algo de **retropropagation**
- Ce qui résout le problème découvert par Minsky et Papert
- Cet algorithme est utilisé jusqu'à maintenant en Deep Learning
- A cause de l'AI winter l'algorithme n'a été publié qu'en 1982
- En 1986 Rumelhart, Hinton et Williams ont redécouvert l'algorithme
- Avec cette deuxième publication l'algorithme a gagné en popularité



Paul Werbos – IJCNN 1991

Années 1990s

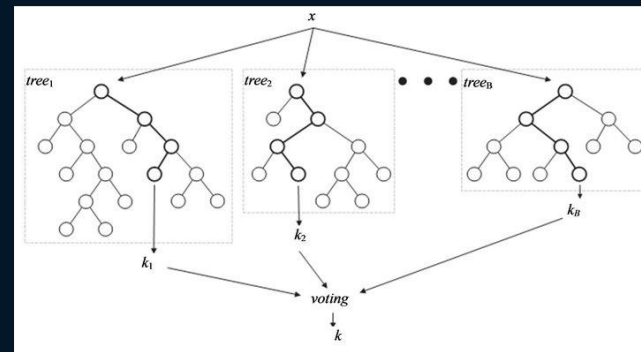
- Proposition des réseaux de neurones convolutifs par **Yann LeCun**
- Les convolutions ont donné aux réseaux la capacité de vision
- Premières applications industrielles du Deep Learning
- **10%** des chèques et codes postaux sont lus par les CNNs



Architecture LeNet-5

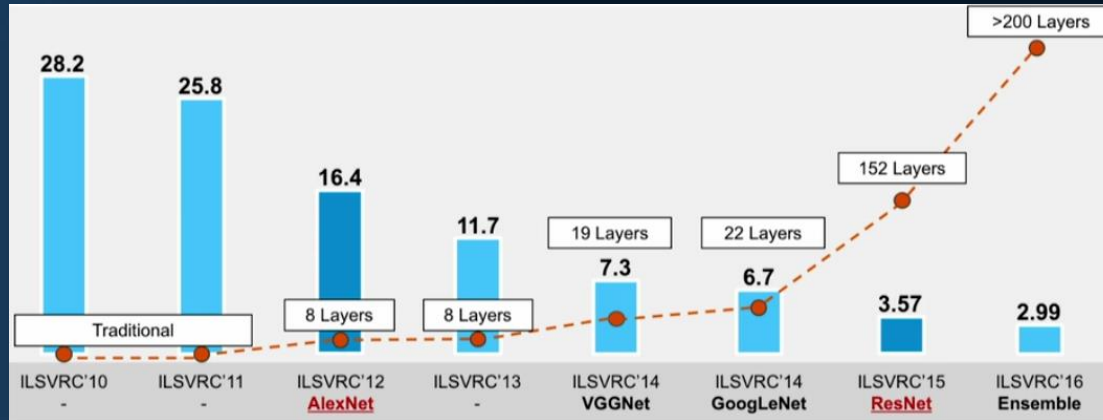
Années 2000s

- Un creux concernant les réseaux de neurones
- On n'arrivait pas à entraîner sur des données complexes
- Les autres algorithmes d'apprentissage ont gagné plus en popularité
 - Ne requièrent pas une puissance de calcul énorme
 - Ne nécessitent pas trop de données pour converger
 - Par exemple les **SVMs** et les **arbres de décision**
- En 2006 développement des **auto-encoders** mais toujours sans applications réelles importantes



Années 2010s

- En 2012 un réseau profond a révolutionné la vision par ordinateur
- **AlexNet** a montré une performance exceptionnelle dans **ImageNet**
- A noter qu'AlexNet ne diffère pas trop des CNNs de 1998
- Depuis cette année plein d'idées et d'applications ont émergé



Taux d'erreurs pour ImageNet

Années 2020s

- Avancées majeures de l'IA générative
 - Large Language Models – LLM (GPT, Galatica, Minerva ...)



➡ *Comment fonctionnent ces grands modèles de langage ?*

- Generative Art AI (DALL-E, Midjourney, Stable Diffusion ...)
- Text to speech (VALL-E, AudioLM ...)
- Text to video (Phenaki, Soundify ...)



➡ *Quels sont les dangers de ces IA ?*



Années 2020s

- Deepfakes



- Fausses œuvres d'art

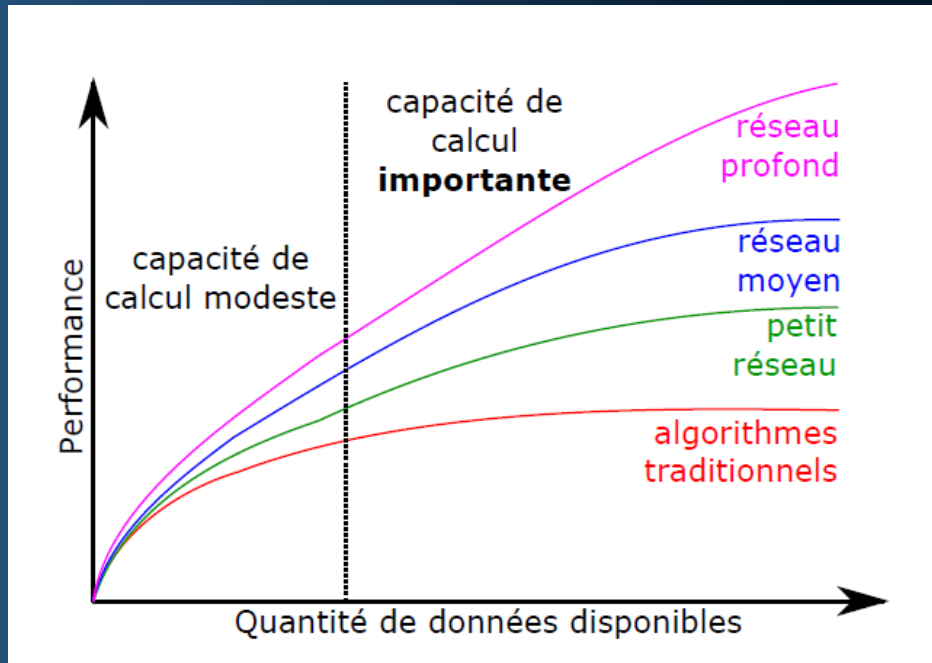


En 2020, Midjourney a gagné le 1^{er} prix lors d'un concours basé d'art digital



Théâtre d'opéra spatial

Pourquoi maintenant ?



- Quantité des données disponible
- Capacité de calcul disponible
- Algorithmes (plus profonds)

Table des matières

01

La révolution deep learning

Nouvelles découvertes
et accomplissements

02

Historique

Histoire du deep
learning : de 1930 aux
années 2020

03

Apprentissage supervisé

L'abondance des
données d'entraînement
et l'apprentissage
supervisé

Apprentissage

- Deux types d'apprentissage :
 - **Supervisé** (on connaît à l'avance ce qu'on prédit)
 - **Non supervisé** (on ne connaît pas à l'avance ce qu'on prédit)
- On se focalise sur l'apprentissage supervisé
- Deux tâches pour l'apprentissage supervisé :
 - **Classification** (catégorisation des images en chien, chat, voiture, etc.)
 - **Régression** (prédiction de la température, prix d'une maison, etc.)
- Exemple : prédiction du prix des maisons à partir du nombre de pièces

Jeu de données: Boston house prices

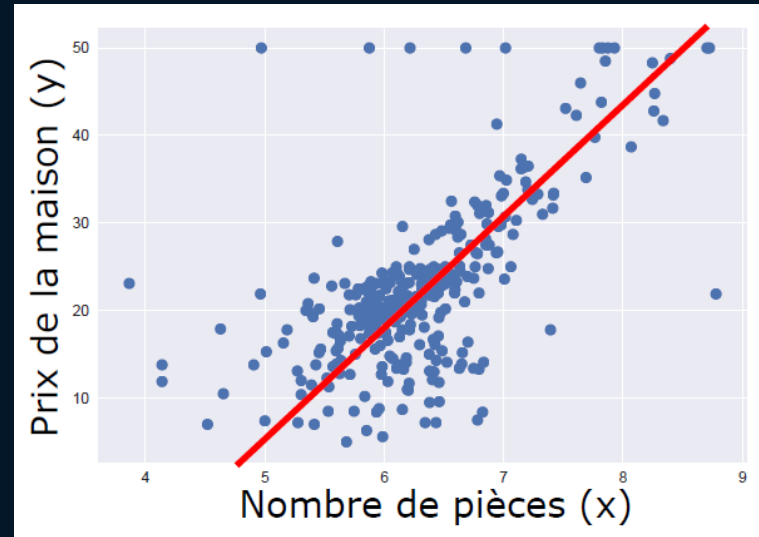
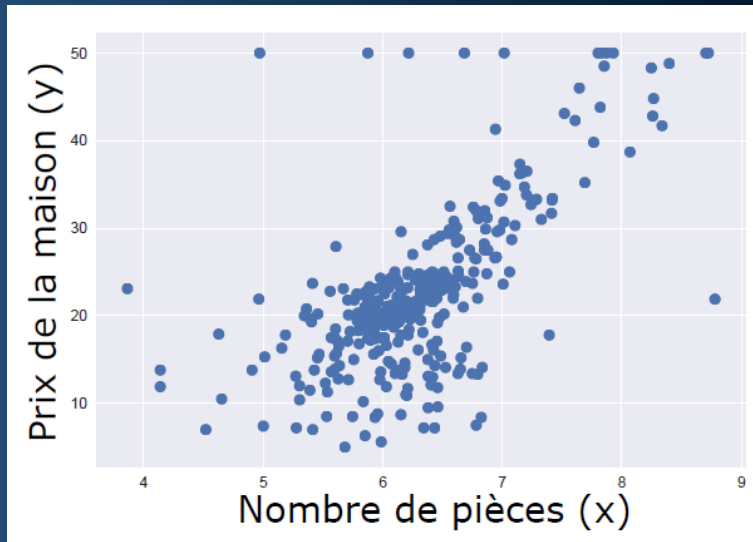
Aperçu des données

Prix	CRIM	ZN	INDUS	CHAS	NOX	RM	...
18.2	0.7258	0.	8.14	0.	0.538	5.727	...
20.3	0.3494	0.	9.9	0.	0.544	5.972	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Données qui nous intéressent

Prix (y)	RM (x) (nombre de pièces)
18.2	5.727
20.3	5.972
⋮	⋮

Tracer la variation



Un modèle linéaire est-il suffisant ?

Un modèle linéaire

- On prédit y (prix) en fonction de x (nombre de pièces)
- Linéaire → **droite** → de la forme $y = w * x + b$
- Simplification : considérer $b = 0 \rightarrow y = w * x$



Comment déterminer w ?

- Plusieurs approches existent :
 - Solution optimale
 - Méthode de descente du gradient

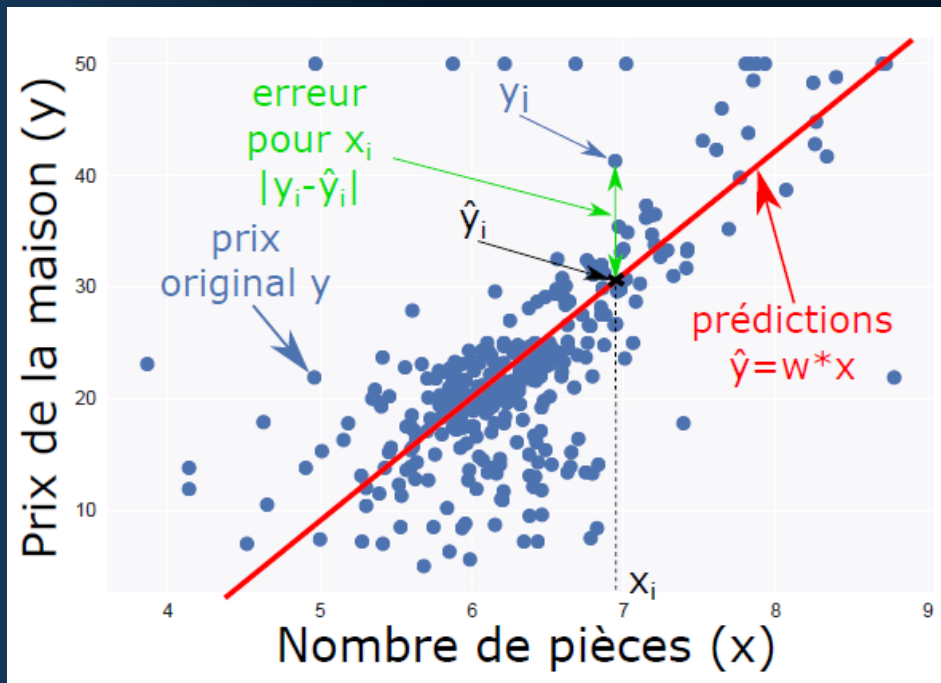


Aujourd'hui, on va utiliser la force brute : On essaye **plusieurs valeurs de w** pour trouver la meilleure



Ne pas l'utiliser : très lent !

Une fonction de « coût » ?



Une fonction de « coût » : formalisation

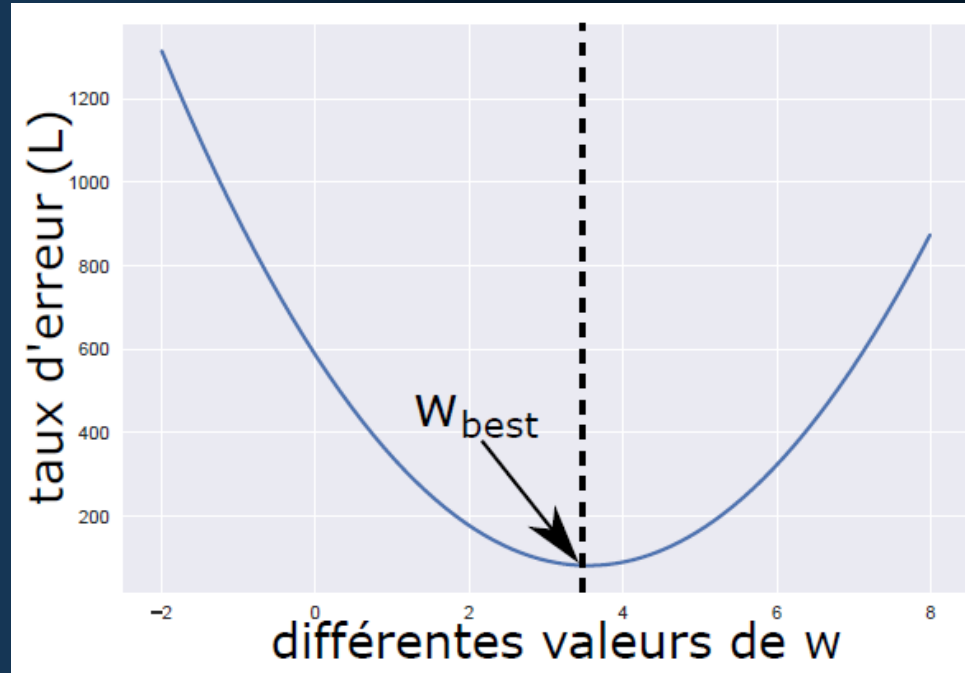
Comment évaluer un choix de w ?

- Notations :
 - y c'est le vrai prix donné de la maison
 - \hat{y} c'est le prix prédit par notre modèle $\hat{y} = w * x$
 - L la fonction de coût du modèle
 - n le nombre d'instances d'entraînement
- Fonction de coût qui mesure l'erreur d'un modèle

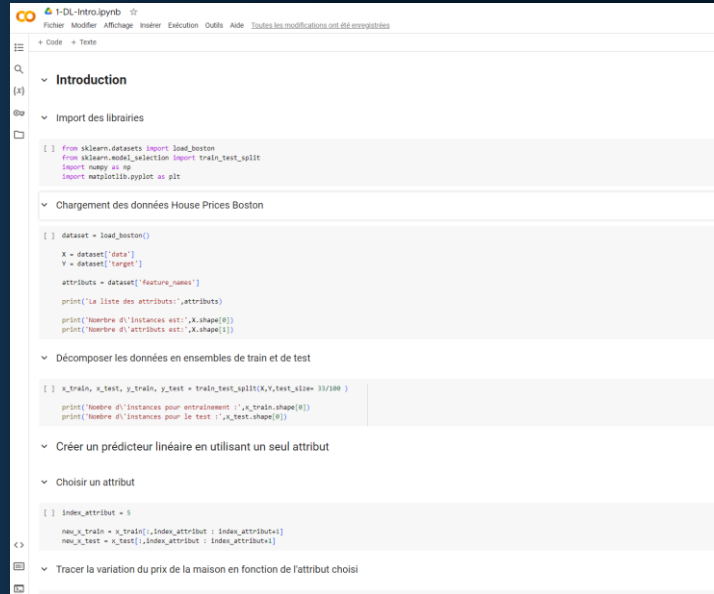
$$L(w) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i(w))^2$$

- Ainsi on essaye les valeurs de w et on choisit celle qui minimise L
- On trouve w_{Best} qui minimise l'erreur $L \Rightarrow$ maximise la précision

Une fonction de « coût » : visualisation



De la théorie vers la pratique



```
1-DL-Intro.ipynb
Fichier Modifier Affichage Insérer Exécution Outils Aide Toutes les modifications ont été enregistrées
+ Code + Texte

Introduction

Import des librairies

[ ] from sklearn.datasets import load_boston
from sklearn.model_selection import train_test_split
import numpy as np
import matplotlib.pyplot as plt

Chargement des données House Prices Boston

[ ] dataset = load_boston()

X = dataset['data']
y = dataset['target']
attributs = dataset['feature_names']

print('La liste des attributs:', attributs)
print('Nombre d\'instances est:', X.shape[0])
print('Nombre d\'attributs est:', X.shape[1])

Décomposer les données en ensembles de train et de test

[ ] x_train, x_test, y_train, y_test = train_test_split(X, y, test_size= 33/100 )

print('Nombre d\'instances pour entraînement:', x_train.shape[0])
print('Nombre d\'instances pour le test:', x_test.shape[0])

Créer un prédicteur linéaire en utilisant un seul attribut

Choisir un attribut

[ ] index_attribut = 5

new_x_train = x_train[:, index_attribut : index_attribut+1]
new_x_test = x_test[:, index_attribut : index_attribut+1]

Tracer la variation du prix de la maison en fonction de l'attribut choisi
```

Télécharger le fichier (https://github.com/r-wenger/cours_ml-m2-OTG/blob/main/IA_geosciences_M2/CM/1_DL_Intro.ipynb) et l'importer sur Google Colab