Intelligence Artificielle

Comprendre les fondements
et l'évolution de l'IA

Introduction

Cette présentation explore la définition de l'intelligence artificielle, ses concepts fondamentaux, son histoire et son évolution à travers différents jalons technologiques.

01

Définition IA

Qu’est ce que l’intelligence artificielle ?

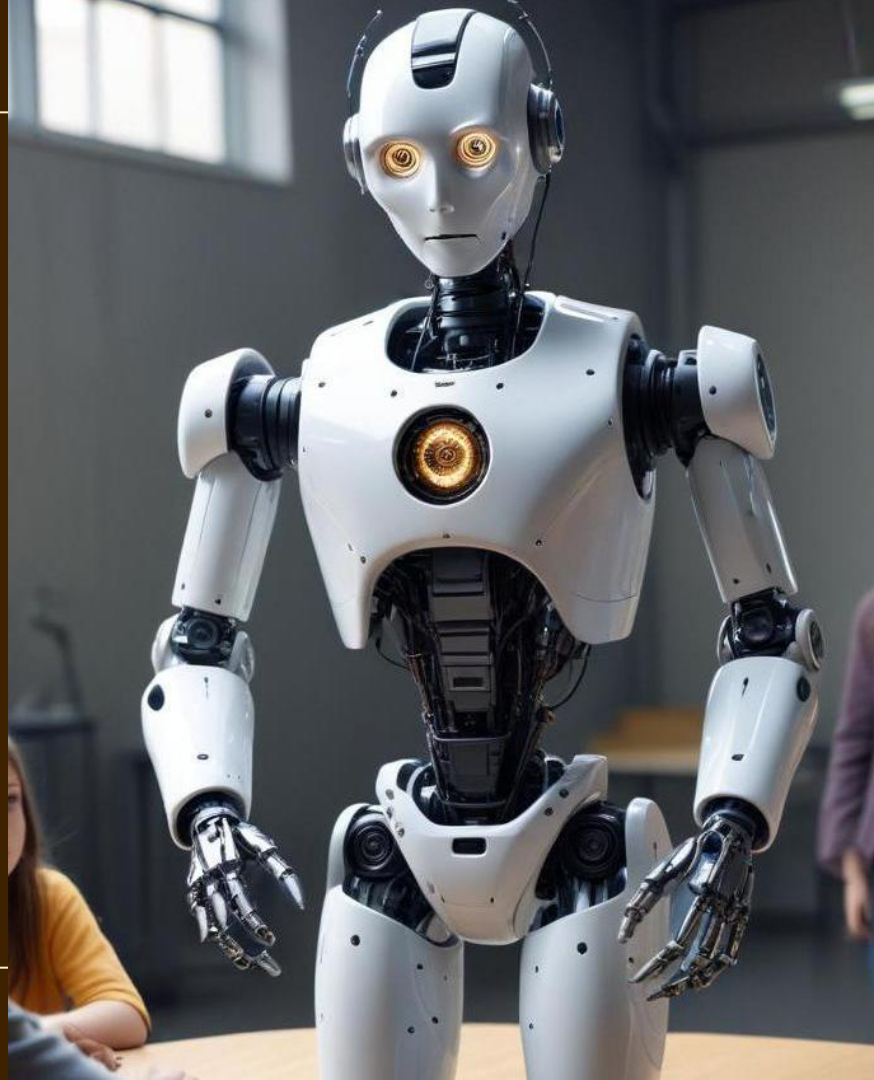
Qu’est ce que l’intelligence ?

“L’intelligence, ce n’est pas ce que l’on sait,
Mais ce que l’on fait quand on ne sait pas”

Jean Piaget, (1896-1980),
Biologiste, psychologue et épistémologue suisse

L’intelligence est un processus dynamique d'équilibration qui se construit progressivement à travers des interactions avec l'environnement. Contrairement à une vision statique ou innée de l'intelligence, Piaget la considère comme un processus évolutif qui permet à l'individu de s'adapter à des situations nouvelles.

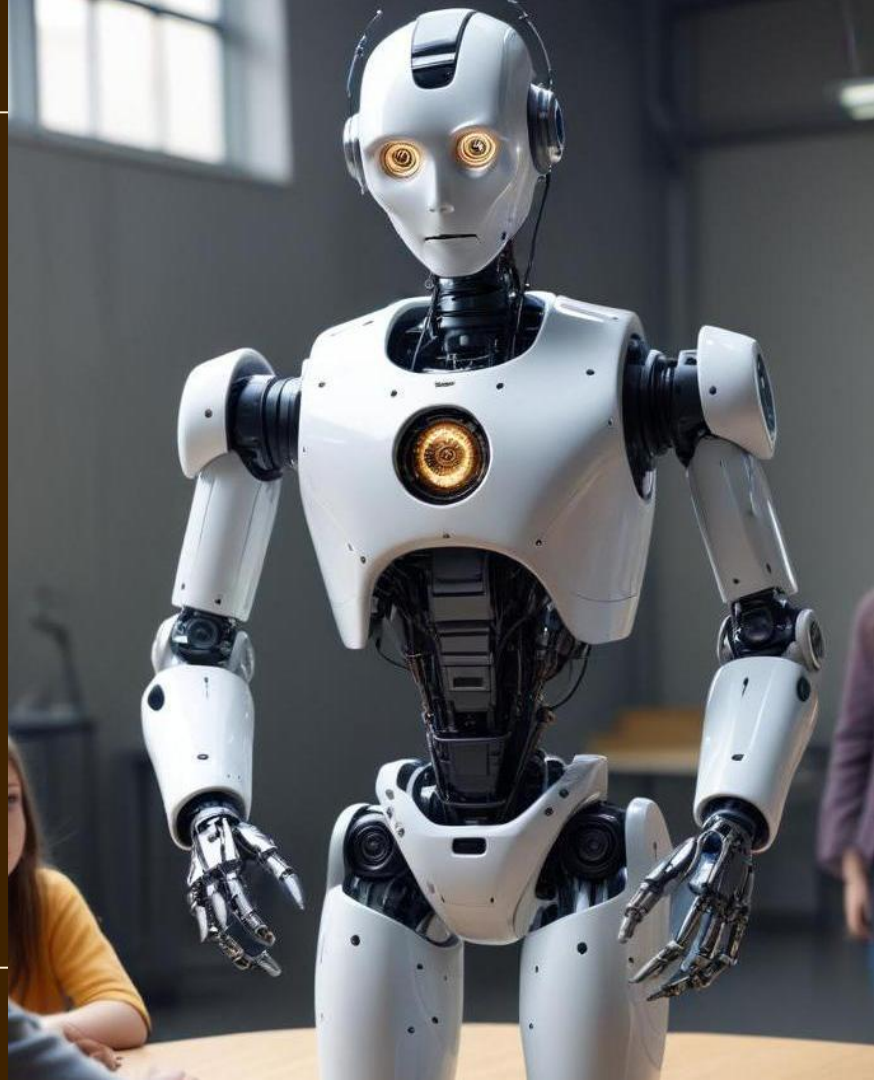
https://afpro.eu/news/lintelligence-selon-jean-piaget/



Concepts fondamentaux

L'intelligence artificielle désigne la capacité d'une machine à **imiter des fonctions cognitives humaines** comme l'apprentissage, la résolution de problèmes et la compréhension du langage.

Les concepts clés incluent l'apprentissage automatique, le traitement du langage naturel et la robotique.



Types d'IA

Il existe plusieurs types d'intelligence artificielle, notamment :

- l'IA faible : conçue pour des tâches spécifiques
- l'IA forte : vise à reproduire l'intelligence humaine de manière compétente.

D'autres distinctions incluent l'IA symbolique et l'apprentissage profond.

Applications de l'IA

Les applications de l'IA sont vastes et variées, allant de l'assistance virtuelle et des systèmes de recommandation à la médecine prédictive et aux véhicules autonomes.

Chacune de ces applications utilise des techniques d'IA pour traiter des données et fournir des résultats utiles.

02

Évolution IA

La genèse

L'intelligence artificielle a émergé dans les années **1950**, avec des pionniers comme **Alan Turing** qui ont posé les bases théoriques.

Les premiers travaux concernaient la logique et la résolution de problèmes, marquant le début d'une discipline qui allait évoluer rapidement par la suite.

Le test de Turing est une expérience proposée par Alan Turing en 1950 pour évaluer la capacité d'une machine à exhiber un comportement intelligent, indiscernable de celui d'un humain, en se basant sur une conversation textuelle.

Naissance de l’IA

La véritable naissance de l'IA est souvent attribuée à la conférence de **Dartmouth en 1956**, où des chercheurs se sont réunis pour discuter de l'usage des machines pour la pensée.

C'est à cette époque que le terme ‘**intelligence artificielle’** a été officiellement établi.

L’âge d’or de l’IA

L'âge d'or de l'IA, qui s'étend des années **1950 aux années 1970**, est caractérisé par une grande avancée dans les recherches et les projets d'IA.

C'était une période où les progrès théoriques ont permis des réalisations significatives dans des domaines tels que les jeux, la résolution de problèmes et le traitement du langage naturel.

Les promesses de l'IA étaient énormes et ont attiré l'attention des investisseurs et des chercheurs, bien que cette période ait également été suivie par des défis et des périodes de stagnation.

Le printemps de l’IA

Le printemps de l'IA fait référence à une période dans l'histoire de l'intelligence artificielle où les attentes et les investissements dans l'IA augmentaient rapidement.

Dans les années 1980, il est marqué par l'essor des systèmes experts, promettant des solutions automatisées dans des domaines variés.

- **1950-1956** – *Prémices de l'IA*
- Alan Turing – Test de Turing
- Conférence de Dartmouth – Naissance du terme "Intelligence Artificielle"

- **1956-1974** – *Premier Âge d'Or*
- Premiers programmes capables de résoudre des problèmes simples
- Premiers systèmes experts rudimentaires

- **1974-1980** – *Premier Hiver de l'IA*
- ❄ Forte baisse des financements
- ❄ Déceptions technologiques (résultats très limités)

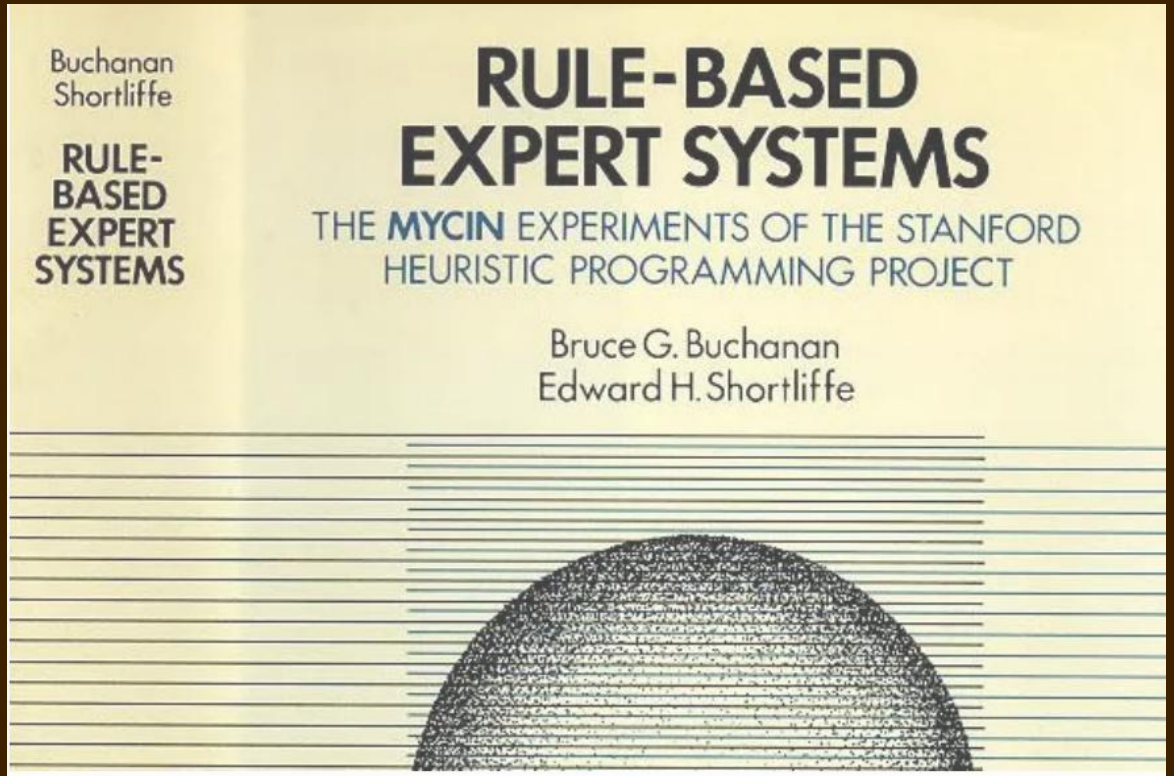
- **1980-1987** – *Printemps de l'IA (Systèmes Experts)*
- Développement massif des systèmes experts en entreprise
- Nouveaux projets nationaux (USA, Japon)

- **1987-1993** – *Second Hiver de l'IA*
- ❄ Systèmes experts jugés trop rigides et coûteux
- ❄ Désertification du marché IA

- **1993-2010** – *Renouveau discret*
- Montée du Machine Learning
- 1997 : Deep Blue bat Kasparov
- 2006 : Renaissance du Deep Learning (Geoffrey Hinton)

- **2010-aujourd'hui** – *Explosion de l'IA Moderne*
- Succès du Big Data et du Deep Learning
- Apparition des assistants vocaux, traduction automatique
- 2022 : ChatGPT – IA générative pour tous

Les systèmes experts



Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, 1984 BUCHANAN AND SHORTLIFFE

03

Perceptron

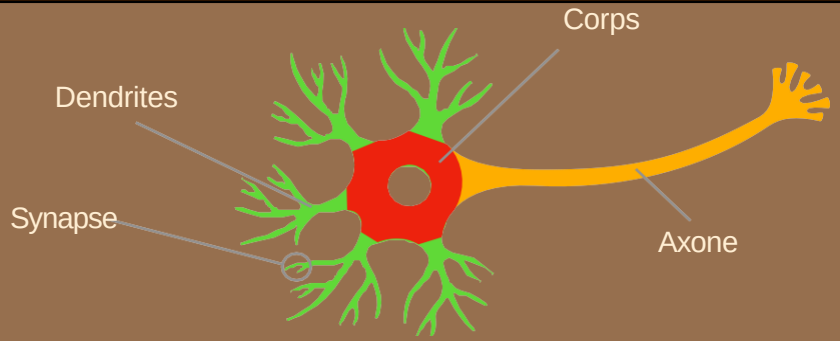
Le Perceptron initial

Le perceptron initial, développé par **Frank Rosenblatt en 1958**, est un algorithme d'apprentissage supervisé utilisé pour la classification binaire.

Ce modèle de neurone artificiel pouvait apprendre à classer des données en ajustant les poids de ses entrées selon les résultats de l'apprentissage.

Alors qu'il a montré des capacités limitées, il a ouvert la voie à des recherches ultérieures en réseaux de neurones.

Le Perceptron initial



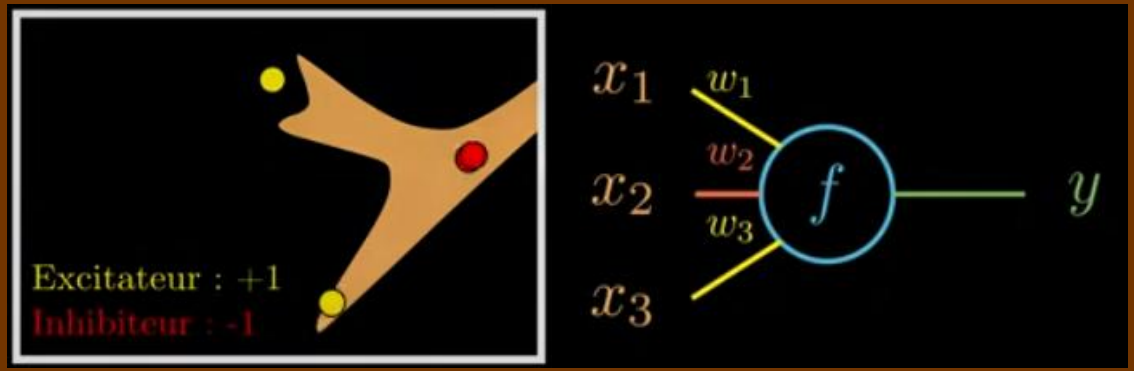
Les neurones, ces acteurs fondamentaux de notre système nerveux, sont en quelque sorte les messagers de notre cerveau. Imaginez-les comme des petites usines de traitement d'informations, connectées les unes aux autres dans un réseau complexe. Chaque neurone reçoit des messages par le biais de ses dendrites, des sortes de tentacules spécialisés dans la réception de signaux provenant d'autres neurones. Ces signaux peuvent être de deux types :

- **excitateurs**, comme une petite poussée (+1) pour encourager le neurone à réagir,
- **inhibiteurs**, qui ont plutôt tendance à freiner son activité (-1).

Maintenant, imaginez un neurone comme un petit comptable : il reçoit tous ces signaux excitateurs et inhibiteurs et les additionne. Si la somme dépasse un certain seuil, alors c'est le moment où le **neurone s'active** !

Il génère alors un signal électrique, une sorte d'impulsion nerveuse, qui se propage le long de son axone, une longue fibre nerveuse, pour communiquer avec d'autres neurones. Et ainsi, le message est transmis.

Le Perceptron initial



Un neurone artificiel est calqué sur le neurone biologique.

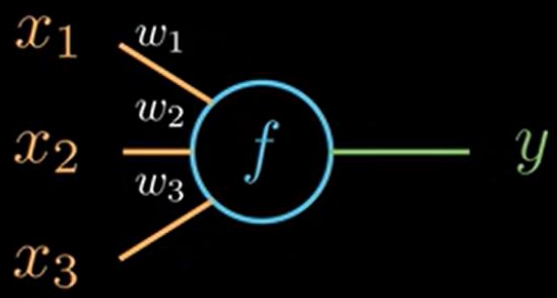
Son corps cellulaire correspond à la fonction f de transfert, tandis que ses dendrites représentent les entrées (x_1, x_2 et x_3) et son axone, la sortie y.

$$f = w_1x_1 + w_2x_2 + w_3x_3$$

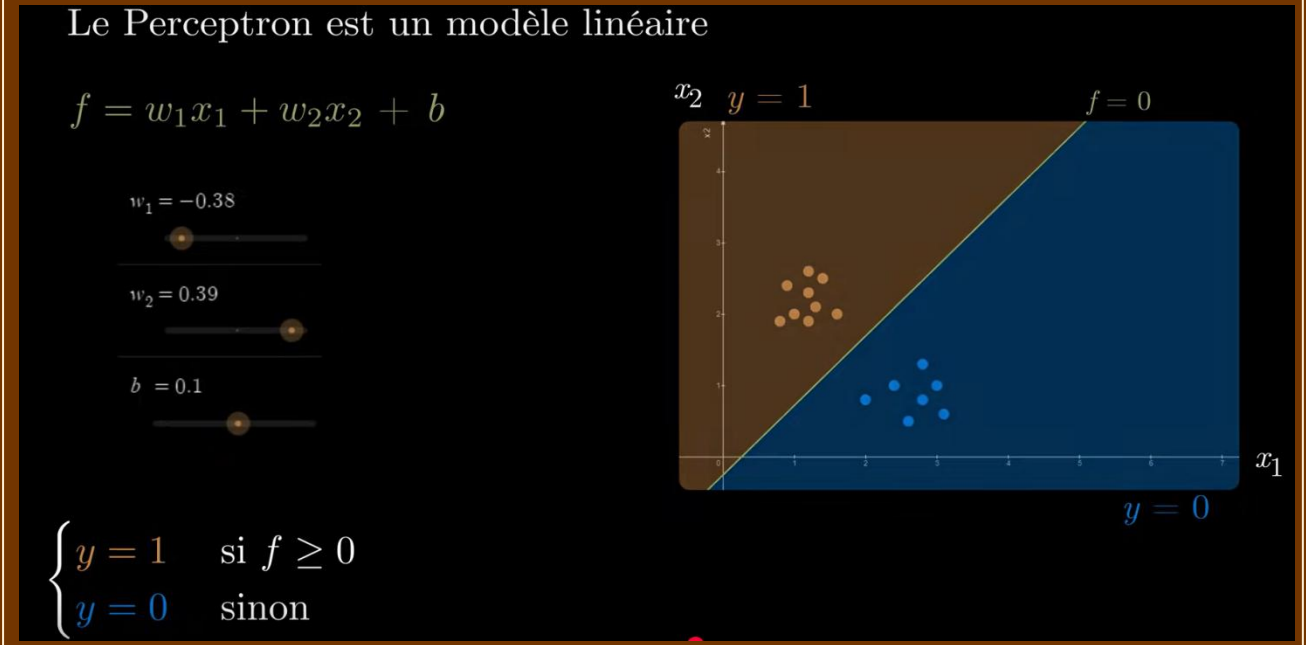
Le Perceptron initial

Agrégation $f = w_1 x_1 + w_2 x_2 + w_3 x_3$

Activation $\begin{cases} y = 1 & \text{si } f \geq 0 \\ y = 0 & \text{sinon} \end{cases}$



Le Perceptron initial



Perceptron multicouche

Le perceptron multicouche (MLP) est une avancée significative par rapport au perceptron simple, permettant l'apprentissage de données non linéaires.

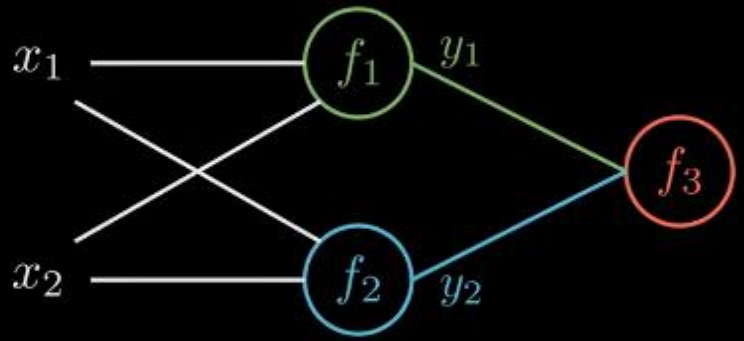
Composé de plusieurs couches de neurones, chaque couche peut apprendre des abstractions de plus en plus complexes des données.

Cette architecture est la base pour de nombreux systèmes modernes d'intelligence artificielle.

Perceptron multicouche

$$f_1 = w_{11}x_1 + w_{12}x_2 + b_1$$

$$\begin{cases} y_1 = 1 & \text{si } f_1 \geq 0 \\ y_1 = 0 & \text{sinon} \end{cases}$$



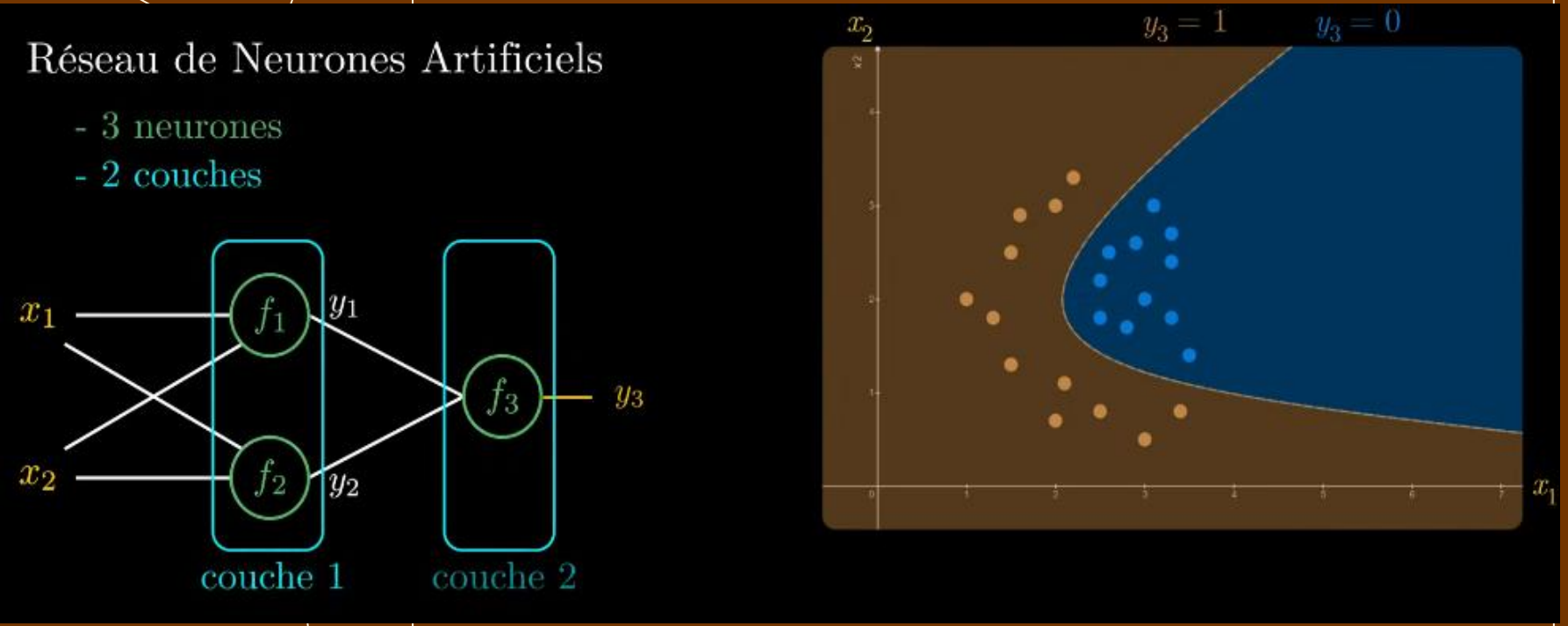
$$f_3 = w_{31}y_1 + w_{32}y_2 + b_3$$

$$\begin{cases} y_3 = 1 & \text{si } f_3 \geq 0 \\ y_3 = 0 & \text{sinon} \end{cases}$$

$$f_2 = w_{21}x_1 + w_{21}x_2 + b_2$$

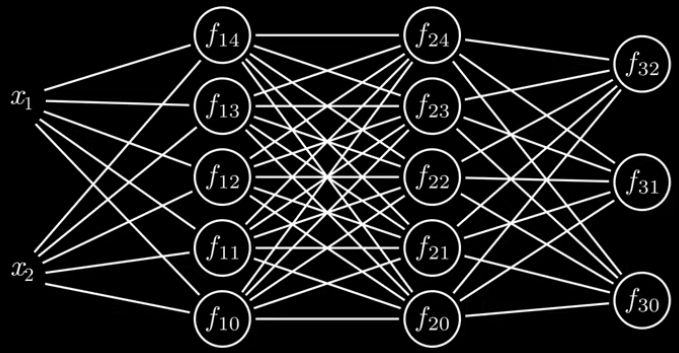
$$\begin{cases} y_2 = 1 & \text{si } f_2 \geq 0 \\ y_2 = 0 & \text{sinon} \end{cases}$$

Perceptron multicouche

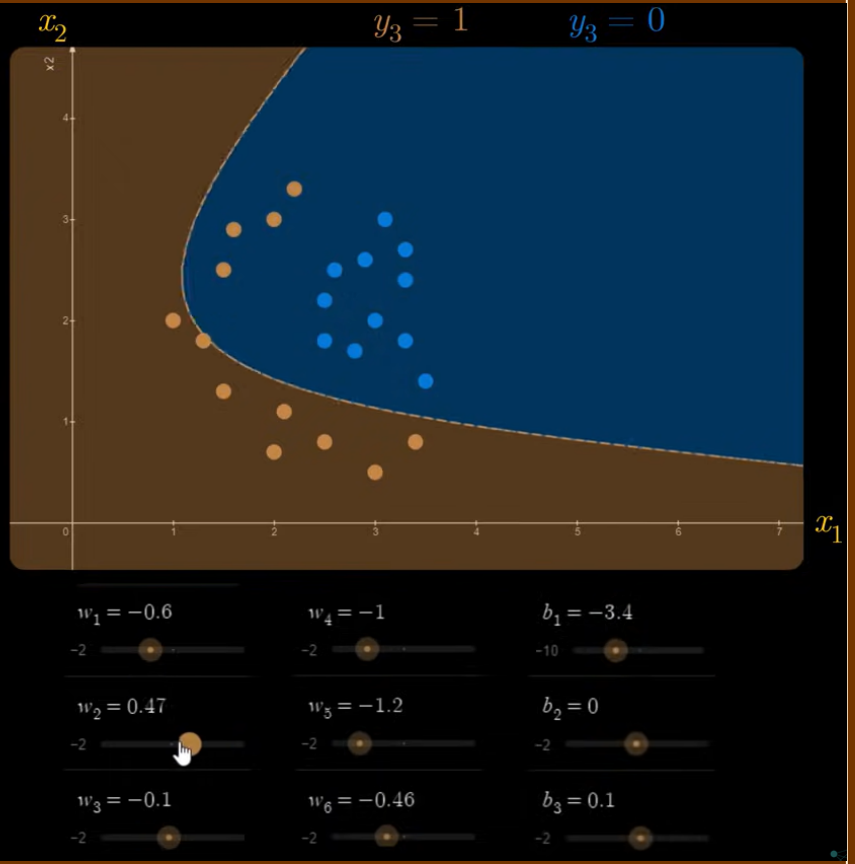


Perceptron multicouche

Réseau de Neurones Artificiels



Perceptron Multi-Couche



Technique de rétropropagation

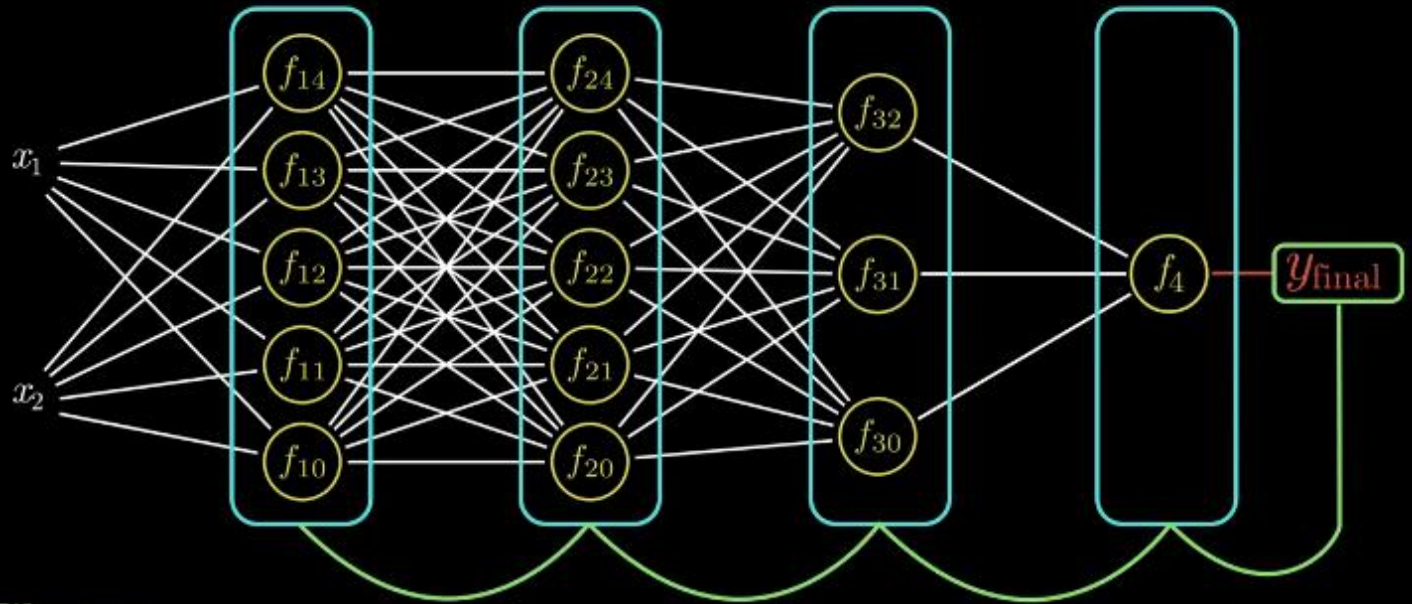
La technique de rétropropagation est un algorithme clé utilisé pour entraîner des réseaux de neurones, notamment les perceptrons multicouches.

Elle fonctionne en calculant le gradient de l'erreur de sortie par rapport aux poids et en ajustant ces poids pour minimiser l'erreur.

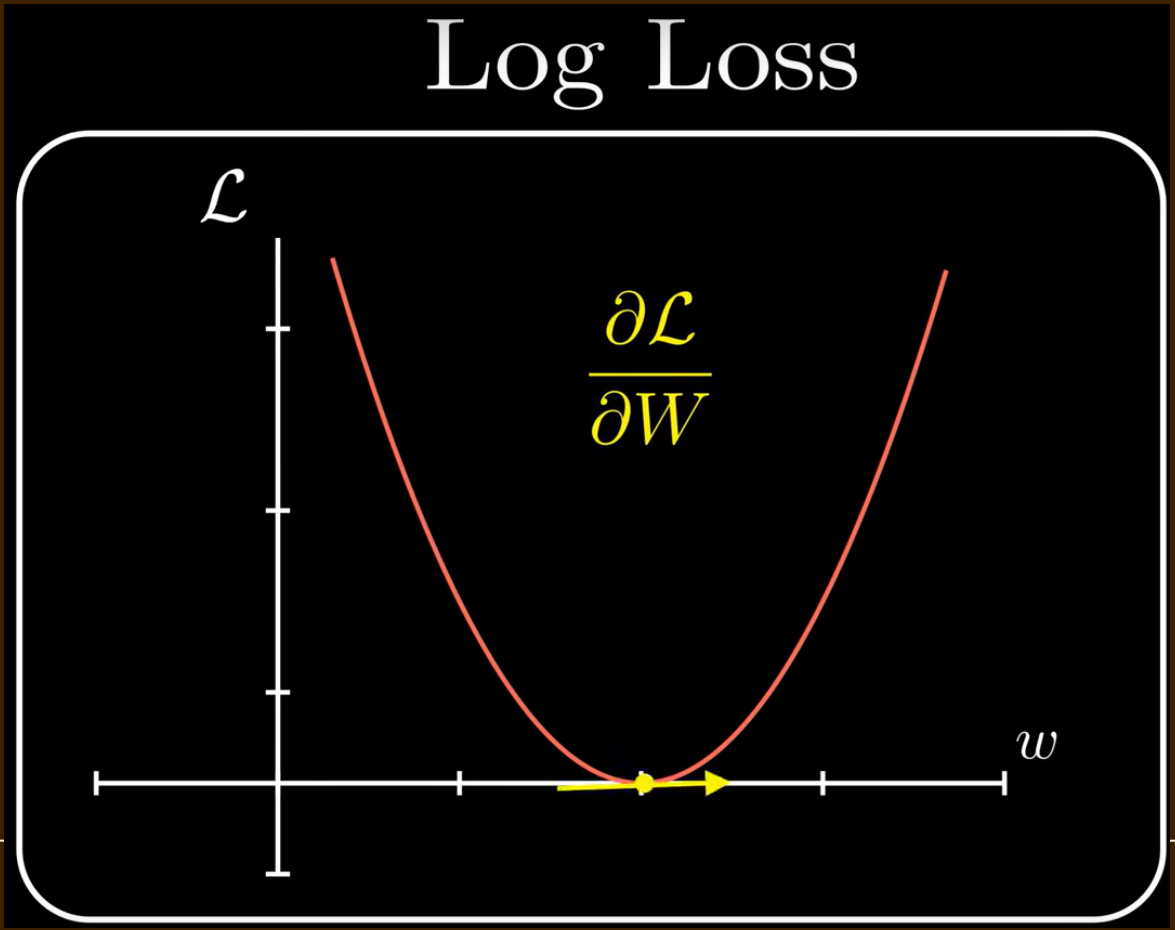
Ce processus itératif a été crucial pour améliorer la précision des modèles d'IA.

Technique de rétropropagation

Consiste à déterminer comment la sortie du réseau varie
en fonction des paramètres (W, b) présents dans chaque couche.

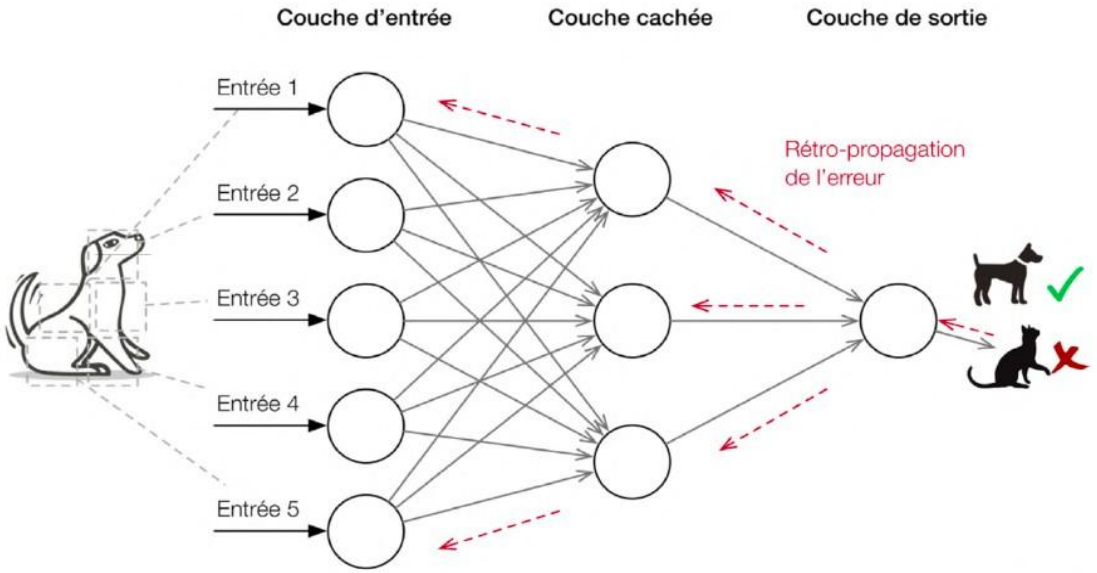


Technique de rétropropagation



Technique de rétropropagation

Figure 5. Fonctionnement d'un réseau de neurones simple

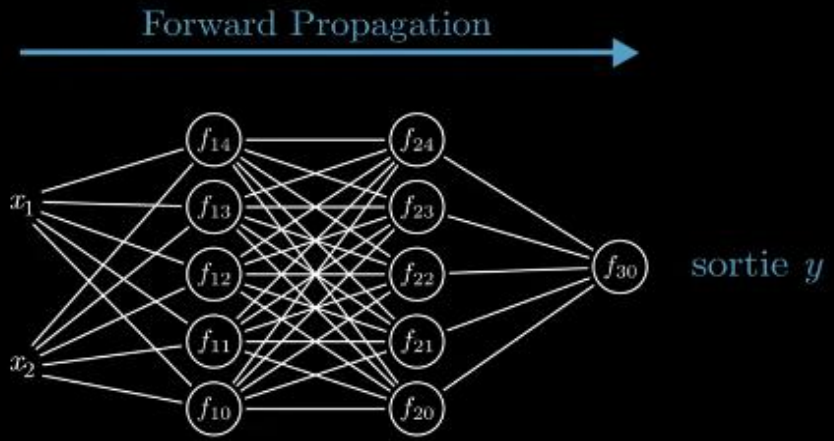


Source : Dominique Cardon, Jean-Philippe Cointet, Antoine Mazières

« L'algorithme fonctionne en prenant la dérivée de la fonction de perte du réseau et en « propage » l'erreur pour corriger les coefficients dans les couches basses du réseau – dans un esprit proche des machines cybernétiques, l'erreur en sortie est « propagée » vers les entrées. »

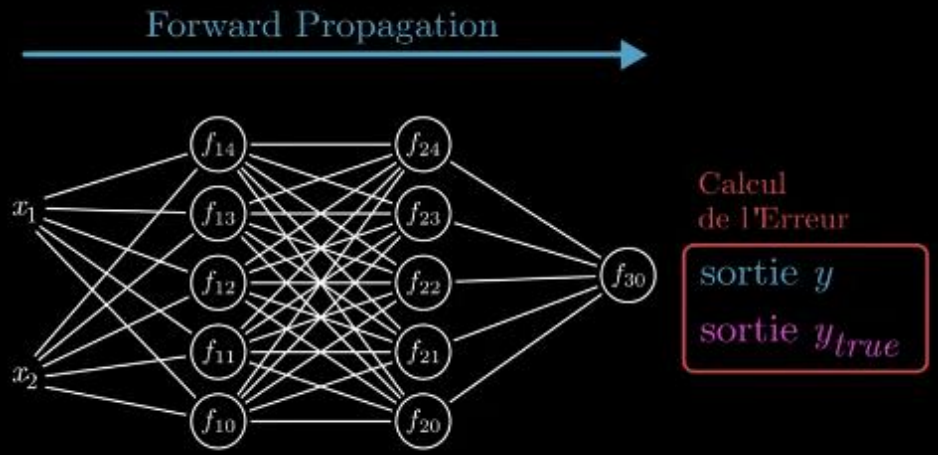
Développer un réseau de neurones artificiels

1. Forward Propagation
2.
3.
4.



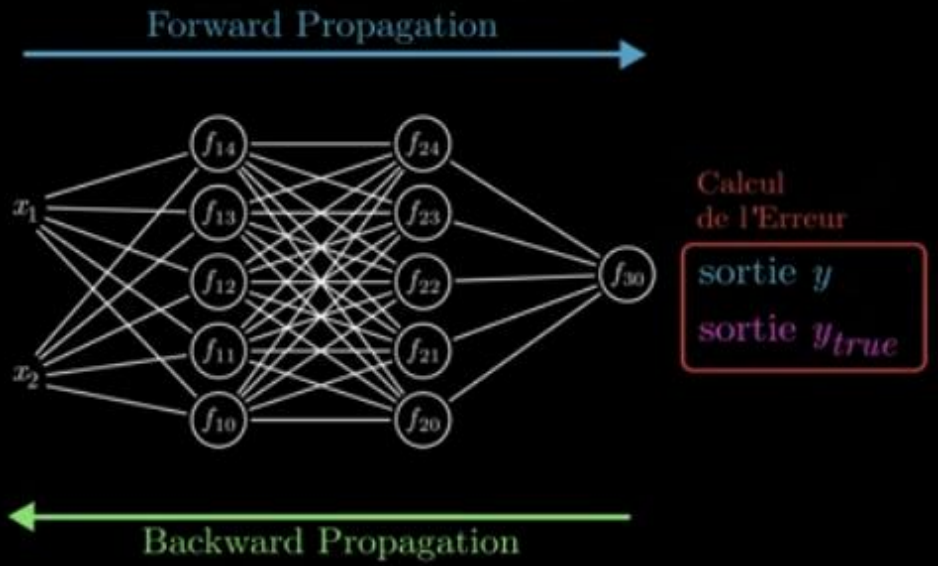
Développer un réseau de neurones artificiels

1. Forward Propagation
2. Cost Function
3.
4.



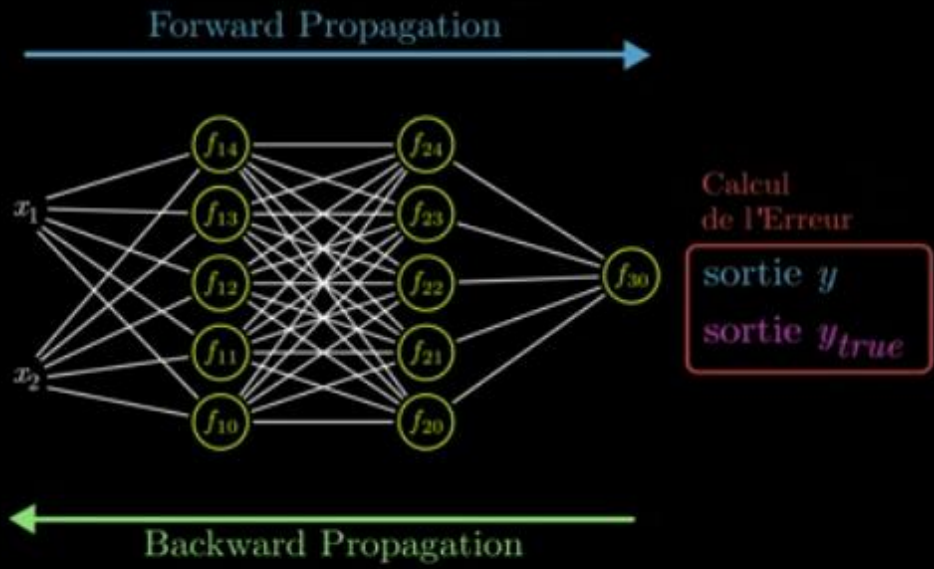
Développer un réseau de neurones artificiels

1. Forward Propagation
2. Cost Function
3. Backward Propagation
4.

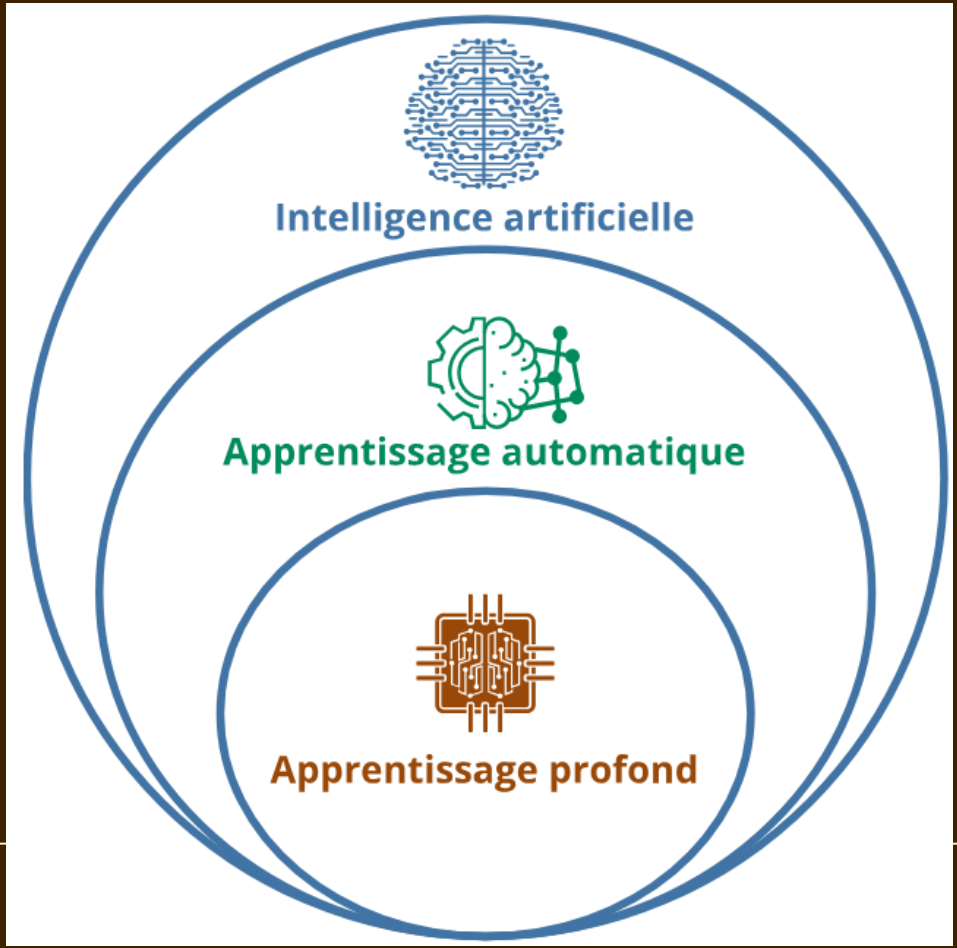


Développer un réseau de neurones artificiels

1. Forward Propagation
2. Cost Function
3. Backward Propagation
4. Gradient Descent



Développer un réseau de neurones artificiels

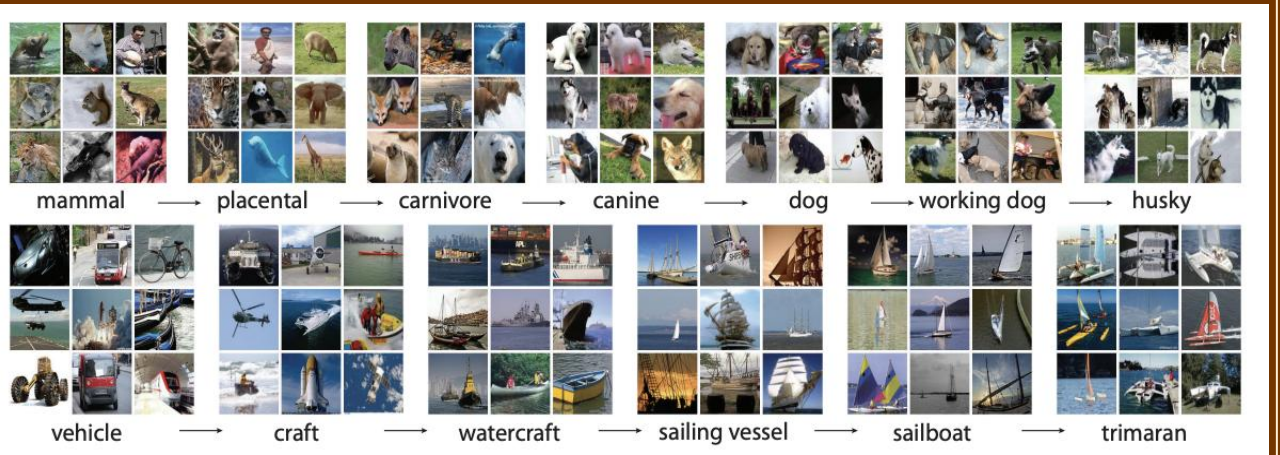


ImageNet

ImageNet est une base de données d'images largement utilisée qui a révolutionné l'apprentissage en profondeur.

Lancée en 2009, elle contient des millions d'images annotées.

L'ampleur et la richesse des données ont permis de former des modèles de reconnaissance d'images, propulsant l'IA vers de nouveaux sommets de précision.



Impact sur l'IA moderne

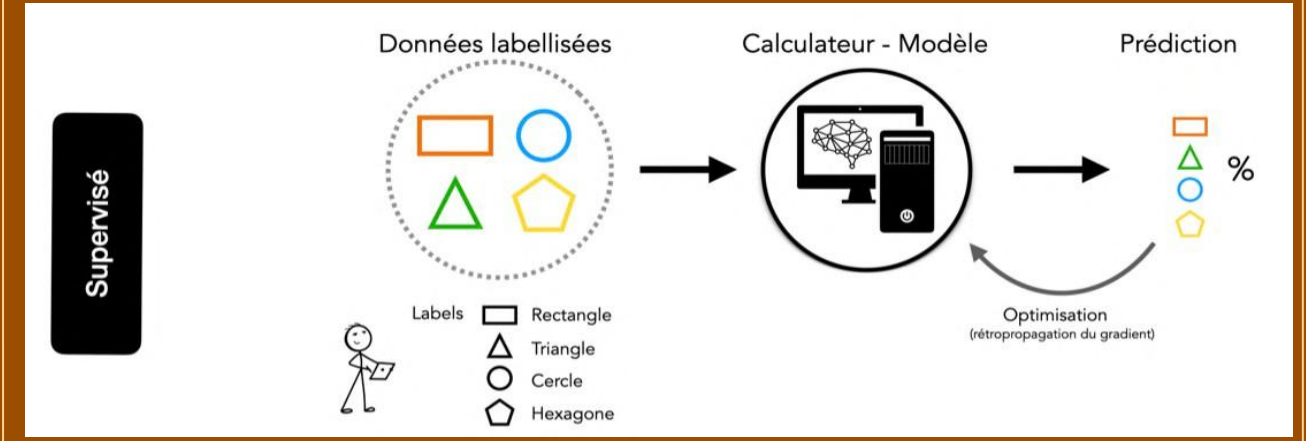
L'impact d'ImageNet et de la rétropropagation a été immense, posant les bases de nombreuses technologies d'IA contemporaines, y compris la reconnaissance faciale, la vision par ordinateur, et les assistants virtuels.

Ces avancées ont profondément influencé nos interactions quotidiennes avec la technologie et façonné l'avenir de l'intelligence artificielle.

04

Principaux types d’apprentissage

L’apprentissage supervisé : le professeur modèle



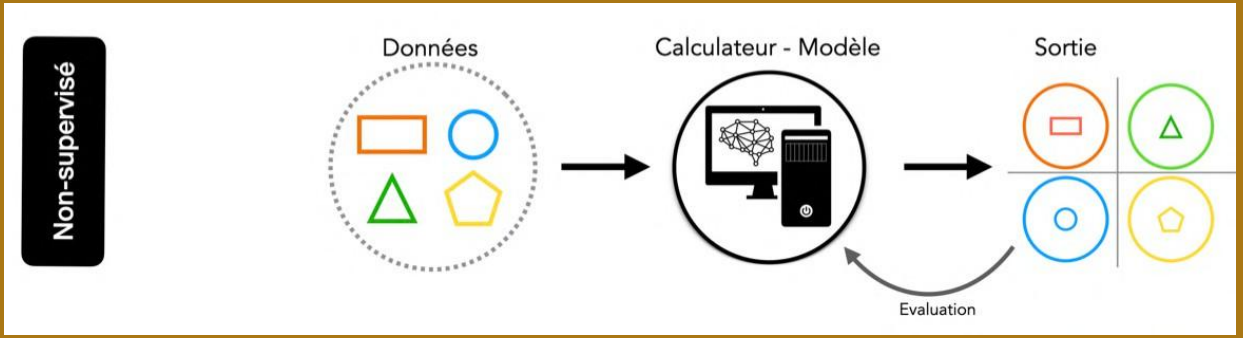
Ici, la machine joue un peu les élèves studieux avec un cahier d’exercices bien annoté.

- **Les données** : ce sont des exemples parfaitement étiquetés, comme si on disait à un enfant : "Ceci est un cercle, cela un carré". Chaque forme a son nom (ou "label") soigneusement attribué.

- **Le processus** : l’algorithme reçoit ces données et cherche à établir des liens entre les formes et leurs noms. Pour affiner ses prédictions, il utilise une technique magique (mais pas trop) appelée rétropropagation du gradient. En gros, c’est comme s’il corrigeait ses erreurs après chaque essai pour devenir de plus en plus précis.

- **La sortie** : à la fin, le modèle est capable de prédire, avec un certain pourcentage de confiance, à quelle catégorie appartient une nouvelle donnée. Impressionnant, non ?

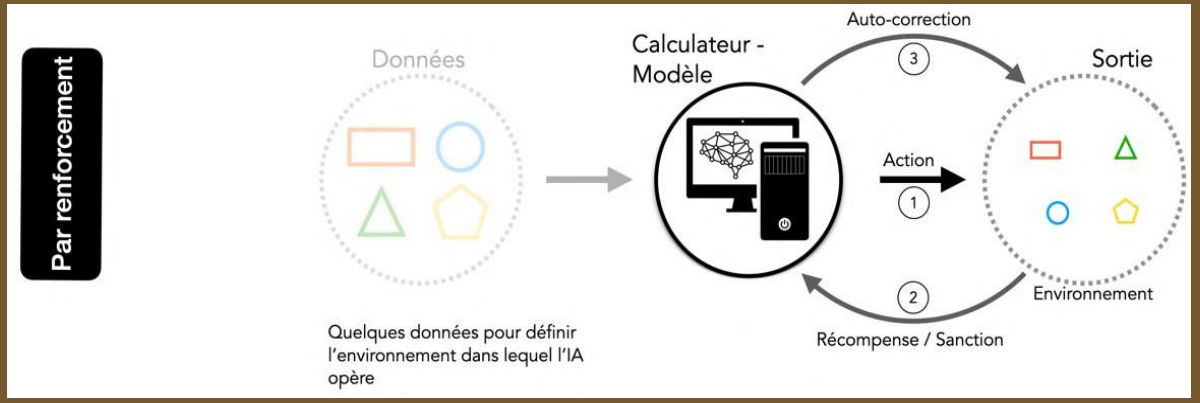
L’apprentissage non-supervisé : le détective curieux



Pas de professeur ici, juste une machine face à un tas de données non étiquetées. Et là, c’est un peu comme si elle jouait au détective.

- **Les données** : pas de labels, juste des formes. Le modèle doit trouver tout seul ce qui relie un cercle à un autre cercle, ou pourquoi un carré n’est pas un triangle.
- **Le processus** : L’algorithme organise et regroupe ces données en fonction de leurs caractéristiques similaires. C’est ce qu’on appelle l’analyse des clusters.
- **La sortie** : On obtient des groupes ou des structures organisées, sans jamais attribuer un nom ou une catégorie précise. Par exemple, le modèle pourrait dire : "Tous ces objets sont similaires", mais sans préciser que ce sont des triangles.

L’apprentissage par renforcement : le joueur persévérant



Ici, la machine devient un aventurier dans un environnement inconnu, cherchant à maximiser ses récompenses.

- **Les données** : quelques infos de base sur l’environnement où l’IA va opérer. Par exemple, dans un jeu vidéo, cela pourrait être le terrain ou les règles.
- **Le processus** : le modèle agit en trois étapes clés. **Action** : il fait un choix (par exemple, bouger à gauche ou à droite). **Récompense ou sanction** : selon le résultat, il reçoit une récompense (comme marquer un point) ou une sanction (game over !). **Auto-correction** : il ajuste sa stratégie pour améliorer ses performances lors des prochains essais.
- **La sortie** : après un grand nombre d’essais (et pas mal de sanctions au passage), la machine devient une experte dans son domaine, capable d’atteindre son objectif efficacement.

05

On programme une IA ?

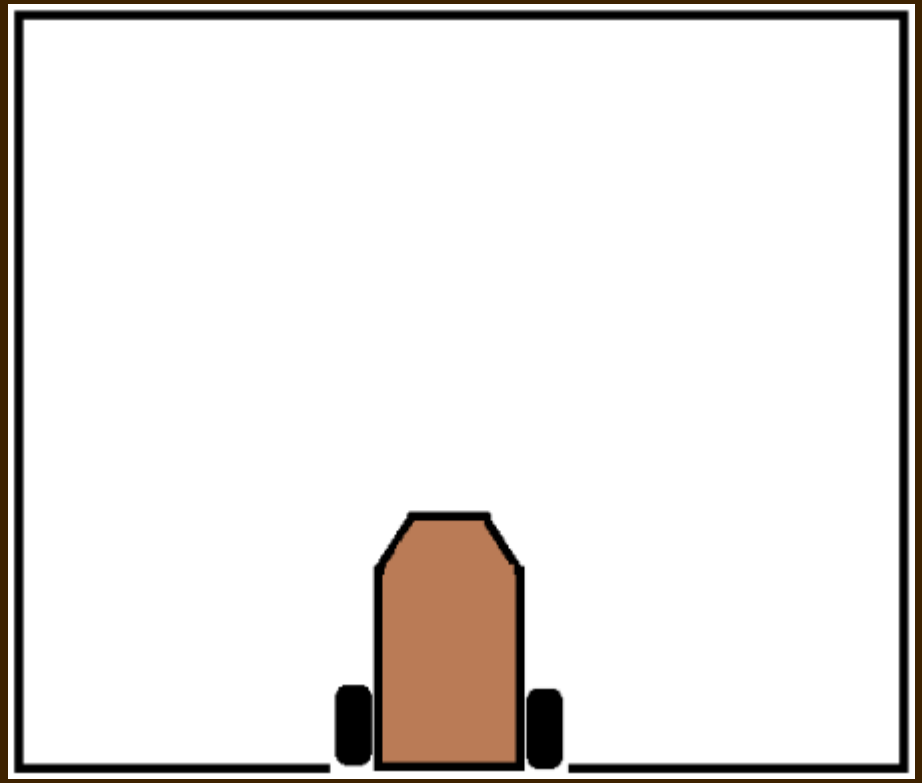
Défi Robot malin !

Les robots sont idiots ! ils ne savent faire que ce qu'on leur a appris.
Lorsqu'on les programmes, il faut avoir pensé à toutes les situations qu'ils pourraient rencontrer dans leur environnement, ce qui n'est pas toujours facile…

En comparaison, même les insectes sont plus efficaces pour se déplacer dans un environnement complexe, semé d'obstacles, comme par exemple, le grillon des cavernes.

Bien qu'il ne soit pas exactement "aveugle", il vit souvent dans des environnements sombres comme des grottes ou des cavernes, où la vision est peu utile. Il utilise principalement ses antennes et d'autres sens pour se déplacer et détecter les obstacles.

Défi Robot malin !

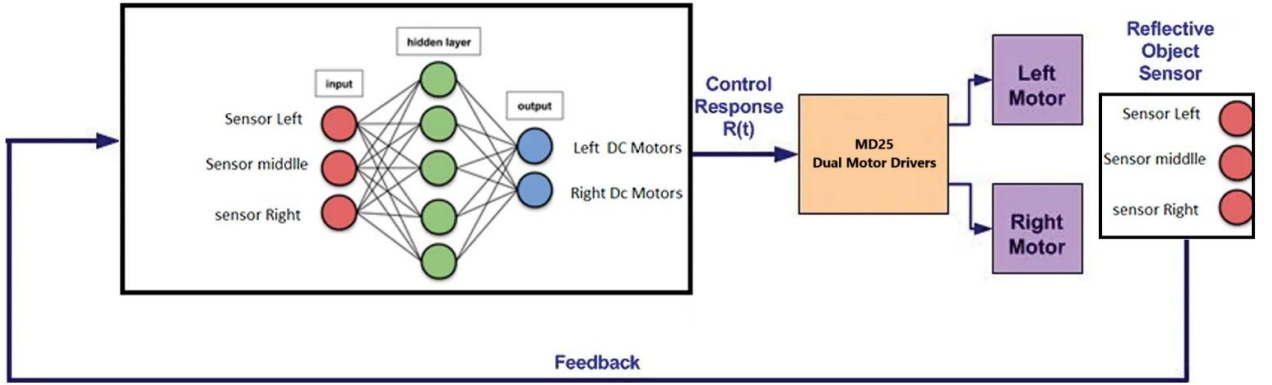


Défi Robot malin !

Notre but sera de rendre le robot NSIbot aussi malin que le grillon des cavernes !

Pour cela, nous allons le doter d'une intelligence minimale constituée de 9 neurones.

A titre de comparaison, un cerveau humain à la naissance comporte 85 milliards de neurones.



Défi Robot malin !

Couche d'entrée :

3 neurones qui correspondent aux trois capteurs infrarouges (gauche, devant, droite).

0 : Pas d'obstacle

1 : Présence d'obstacle

Couche cachée :

4 neurones. Cette couche permet au réseau de modéliser des relations non linéaires entre les entrées et les sorties.

Couche de sortie :

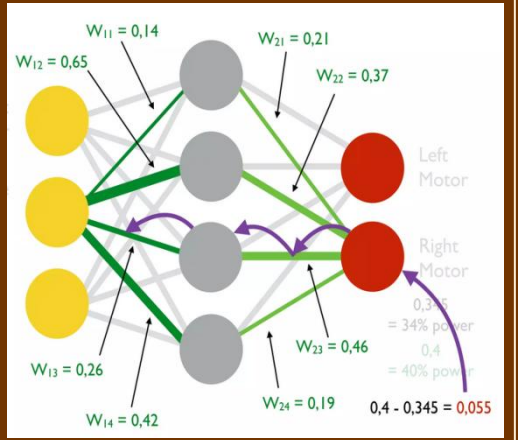
2 neurones qui correspondent aux commandes des moteurs gauche ((MG)) et droit ((MD)).

Les sorties sont continues, comprises entre (0.0) et (1.0), et contrôlent la vitesse et la direction des moteurs :

0.0 : Marche arrière à pleine vitesse

0.5 : Arrêt

1.0 : Marche avant à pleine vitesse



Défi Robot malin !

Entraînement du réseau

L'entraînement consiste à ajuster les poids et biais du réseau pour qu'il produise les sorties souhaitées en réponse aux motifs d'entrée.

- **Définir les données d'entraînement :**
  Entrées : Les motifs des capteurs.
  Sorties cibles : Les commandes des moteurs.
- **Fonction de perte :**
  Utiliser une fonction de perte (comme l'erreur quadratique moyenne ou l'entropie croisée) pour mesurer la différence entre les sorties du réseau et les sorties cibles.
- **Propagation arrière (Backpropagation) :**
  Calculer le gradient de perte par rapport aux poids et biais du réseau via la dérivation automatique.
  Mettre à jour les poids et biais en utilisant un algorithme d'optimisation (comme la descente de gradient).
- **Itérations d'entraînement :**

Répéter les étapes ci-dessus pour réduire l'erreur entre les sorties prédites et les sorties souhaitées. Cela s'effectue sur plusieurs époques.

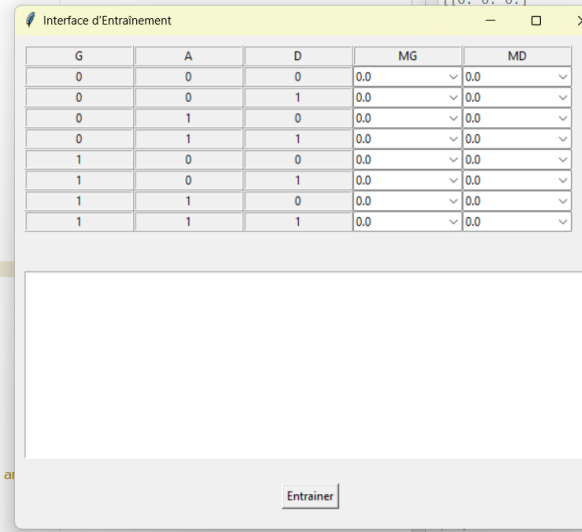
Défi Robot malin !

	Capteurs			Moteurs		
Situation	**Gauche**	**Devant**	**Droit**	**Moteur G**	**Moteur D**	**Direction**
Pas d'obstacle	0	0	0	1.0	1.0	Marche avant
	0	0	1			
	0	1	0			
	0	1	1			
	1	0	0			
	1	0	1			
	1	1	0			
	1	1	1			

Entrainement

```python
import numpy as np
import matplotlib.pyplot as plt

class Neural_Network(object):
  def __init__(self):
  #parameters
    self.inputSize = 3
    self.outputSize = 2
    self.hiddenSize = 4

  #weights
    self.W1 = np.random.randn(self.inputSize, self.hiddenSize) # (3x2) weight
matrix from input to hidden layer
    self.W2 = np.random.randn(self.hiddenSize, self.outputSize) # (3x1) weight
matrix from hidden to output layer

  def forward(self, X):
    #forward propagation through our network
    self.z = np.dot(X, self.W1)
    self.z2 = self.sigmoid(self.z) # activation function

    self.z3 = np.dot(self.z2, self.W2)
    o = self.sigmoid(self.z3) # final activation function
    return o

  def sigmoid(self, s):
    # activation function
    return 1/(1+np.exp(-s))

  def sigmoidPrime(self, s):
    #derivative of sigmoid
    return s * (1 - s)

  def backward(self, X, y, o):
    # backward propagate through the network
    self.o_error = y - o # error in output
    self.o_delta = self.o_error*self.sigmoidPrime(o) # applying derivative of
sigmoid to error
```

```python
from neuronal_network import *

xAll = np.array([ [0, 0, 0],
                  [0, 0, 1],
                  [0, 1, 0],
                  [0, 1, 1],
                  [1, 0, 0],
                  [1, 0, 1],
                  [1, 1, 0],
                  [1, 1, 1]
                  ])

xAll = xAll/np.amax(xAll, axis=0) # scaling input data
# scaling output data (max test score is 100)

# split data
X = np.split(xAll, [8])[0] # training data
xPredicted = np.split(xAll, [8])[0] # testing data

test=np.array([[0, 1, 1]])

def train():
    # Fonction à exécuter lorsque le bouton "Entrainer" est cliqué
    log_text.insert(tk.END, "Début de l'entraînement...\n")
    # Vous pouvez ajouter ici le code pour l'entraînement
    # scale units
    NN = Neural_Network()
    y = create_numpy_array()
    log("Tab NumPy created\n")
    log(y)
    print("Tab NumPy created :")
    print(y)
    i = 0
    while i<200000 and (np.mean(np.square(y - NN.forward(X)))>0.00004):# ar
        if (i%1000==0):
            print("**************************")
            print("*  Training number ", i, " *")
            print("**************************")

            print("Predicted Output: \n" + str(NN.forward(X)))
            print("Loss: \n" + str(np.mean(np.square(y - NN.forward(X))))) # mean sum squared loss
            print("\n")
        NN.train(X, y)
        i+=1
    log("\nEnd of trainning !")
```



Programmation du robot

NSIbot | Weights.h

```ino
#include <SharpIR.h>
#include <Wire.h>
#include <math.h>
#include "Weights.h"
/* Mode 0 par défaut pour MD25*/

#define CMD (byte)0x00     // Values of 0 eing sent using write have to be cast as a byte
  // This is a but with arduino 1
#define MD25ADDRESS 0x58   // Address of the MD25
#define SPEED1 (byte)0x00  // Byte to send speed to first motor
#define SPEED2 0x01        // Byte to send speed to second motor
#define ENCODERONE 0x02    // Byte to read motor encoder 1
#define ENCODERTWO 0x06    // Byte to read motor encoder 2
#define RESETENCODERS 0x20

//Télémètres en entrées analogiques (CAN 10 bits)
#define IRPin1 A6
#define IRPin2 A2
#define IRPin3 A4  //
#define modelSharp 1080

#define DETECT_DIST 40   // Distance maxi 40 cm
#define VITESSE_MIN 78   // Commande vitesse minimum MD25
#define VITESSE_MAX 178  // Commande vitesse maximum MD25

/* Instanciations Objets "Télémétres"*/
SharpIR Telm_AVG = SharpIR(IRPin1, modelSharp);
SharpIR Telm_AV = SharpIR(IRPin2, modelSharp);
```

```ino
const int InputNodes = 3;
const int HiddenNodes = 4;
const int OutputNodes = 2;

//Poids de la couche cachée
const float HiddenWeights[InputNodes][HiddenNodes] = {
  {8.69475792564935, 5.884804877095347, 4.64652316489549, -4.882947994892906},
  {-6.544390231123045, 6.166195732685829, 0.5413565030250159, -0.5455373795729869},
  {-5.366264501264944, -8.999845339631163, 4.154236977071539, -3.2893869743223005}
};

// Poids de la couche de sortie
const float OutputWeights[HiddenNodes][OutputNodes] ={
  {11.092105767288604, 11.983095988300903},
  {-10.2223015597946, -11.014524865354462},
  {-5.601335098227478, 5.43190280388524},
  {13.553263938676388, 4.164475863848009}
};
```

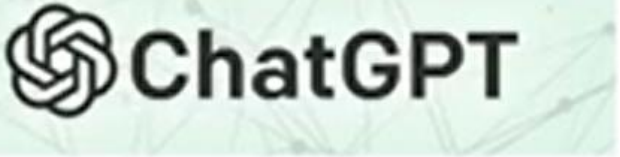
06

Les ia génératives

Midjourney

L'ère de l'IA générative

MISTRAL AI_



LLM : Large Language Model

- Apprentissage sur des quantités énormes de données textuelles
- Entrainement : prévoir le mot suivant en fonction d’un contexte

LA FRANCE EST UN GRAND

LLM : Large Language Model

- Apprentissage sur des quantités énormes de données textuelles
- Entrainement : prévoir le mot suivant en fonction d'un contexte

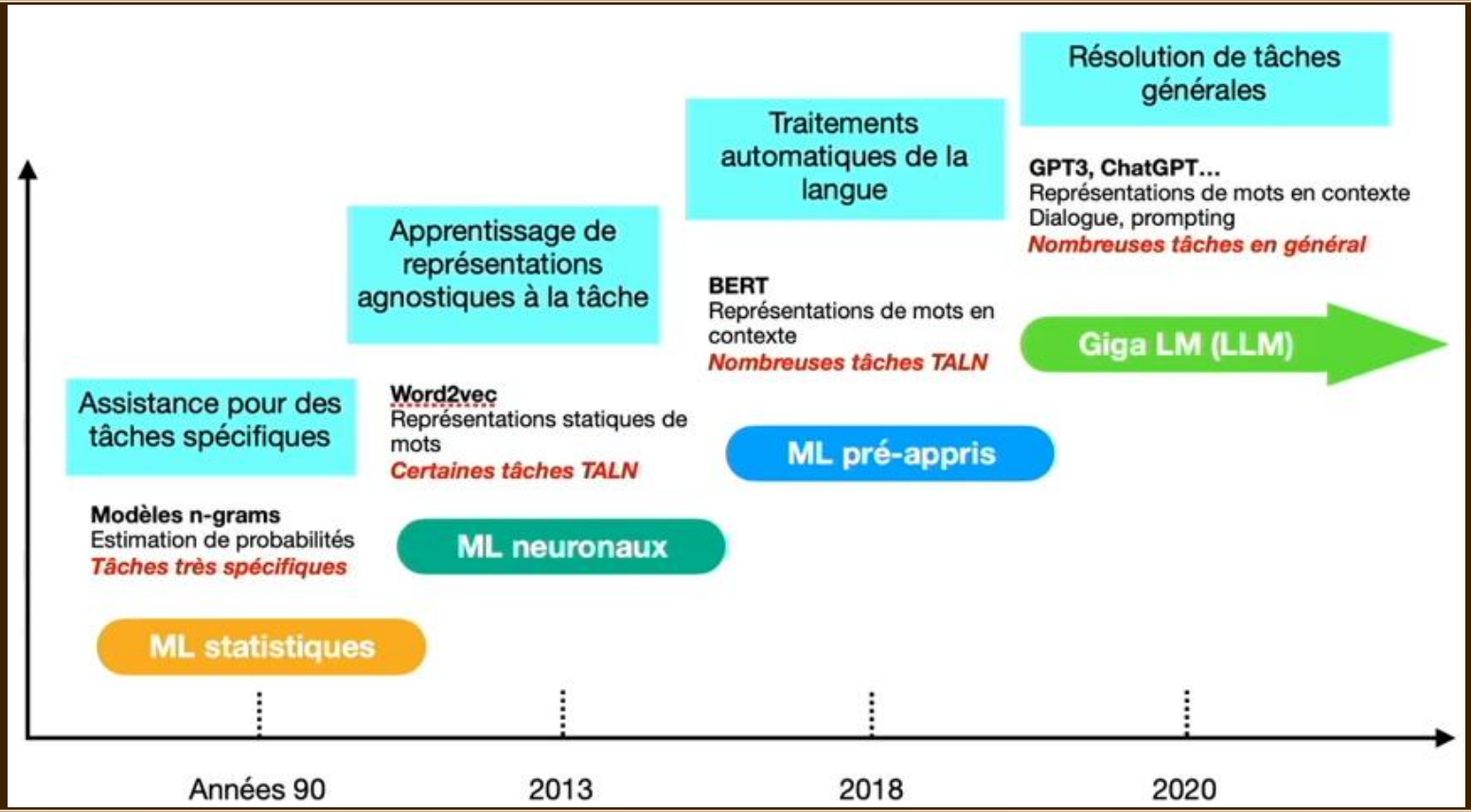
LA FRANCE EST UN GRAND

	Probabilité
PAYS	*0,4*
ACTEUR	*0,25*
LEADER	*0,15*
FOYER	*0,15*
CERCLE	*0,10*

LLM : Large Language Model

- Apprentissage sur des quantités énormes de données textuelles
- Entrainement : prévoir le mot suivant en fonction d'un contexte

Les ia génératives



Conclusions

En conclusion, cette présentation a examiné comment l'intelligence artificielle a évolué depuis ses débuts avec des concepts majeurs tels que le perceptron, en passant par les fameux moments de l'histoire de l'IA.

Nous avons constaté que les avancées technologiques, à travers des périodes de promesses et de défis, continuent d'influencer notre rapport à l'IA aujourd'hui et façonnent son avenir.

Utiliser l'IA au quotidien

Pour démontrer les capacités de l'IA générative et expliquer comment l'utiliser quotidiennement, voici quelques activités et cas d'usages concrets

Recherche d’informations

Situation :

Vous recherchez des informations synthétique sur l’Ukraine et sur sa capitale.

Activité :

Demande à l’IA : « Donne moi des informations synthétique sur l‘Ukraine. »

Raffiner la demande : « Qui est le maire de Kiev ? »

Recherche d’informations

Situation :

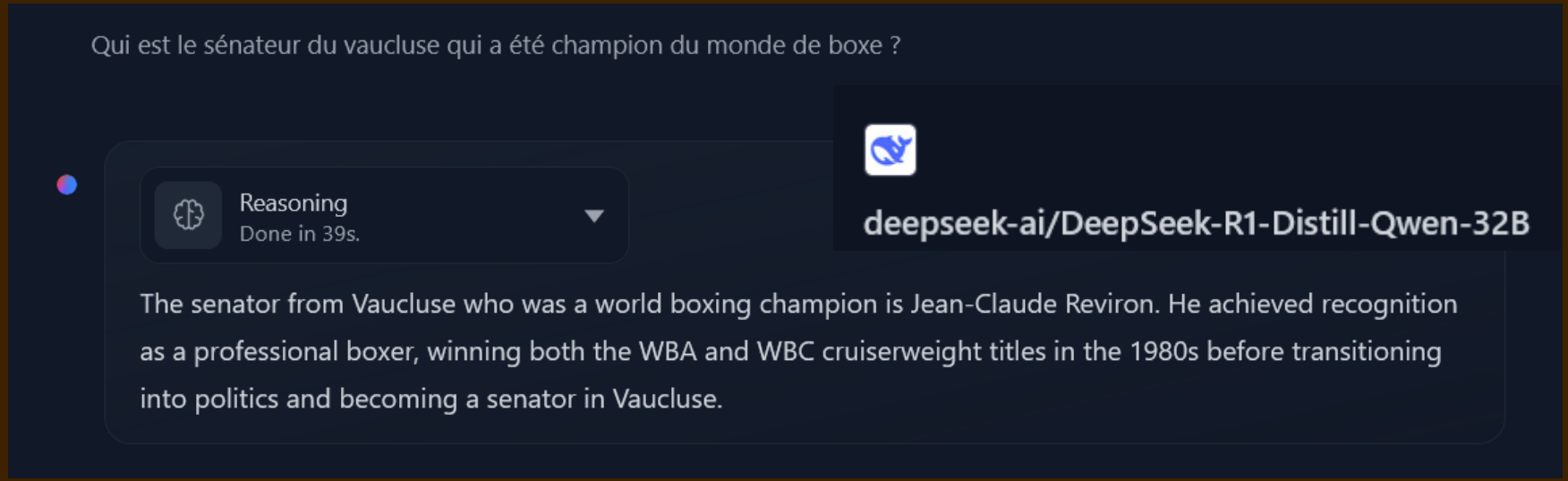
Le maire de Kiew a été champion du monde de boxe !
Comme l’un de nos sénateur du Vaucluse !

Activité :

Trouver qui est le sénateur du Vaucluse qui a été champion du monde de boxe

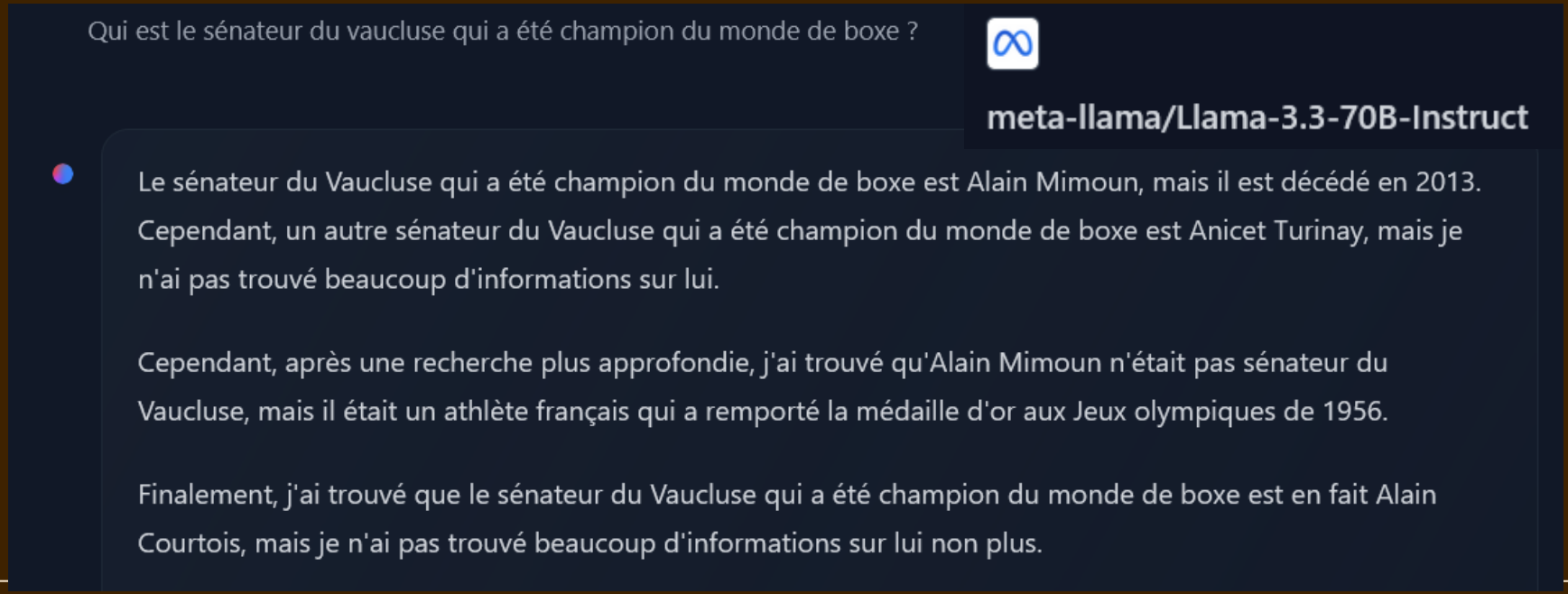
Recherche d’informations

Attention aux hallucinations !



Recherche d’informations

Attention aux hallucinations !



Recherche d’informations

Attention aux hallucinations !

Situation :

6949 est il un nombre premier ?

Activité :

Demander à l’IA !

Reformulation de messages

Situation :

Imaginez que vous devez envoyer un mail à votre propriétaire pour signaler un problème de plomberie dans votre appartement. Vous souhaitez que le message soit clair, poli et professionnel.

Activité :

Étape 1 : Rédigez un brouillon de mail décrivant le problème et demandant une intervention.

Étape 2 : Soumettez ce brouillon à l'IA pour qu'elle le reformule et l'améliore.

Rédiger un courrier administratif

Situation :

Vous vivez dans un logement individuel (maison ou appartement). Début avril, vous recevez une facture EDF anormalement élevée par rapport à votre consommation habituelle.

En vérifiant les détails :

- La facture a été établie sur une estimation, bien que vous ayez déclaré votre relevé réel en ligne dans les délais.
- EDF a ignoré ce relevé et a utilisé une estimation beaucoup plus élevée.
- Vous avez noté vous-même les index du compteur, et ils confirment une consommation bien inférieure à celle facturée.
- Pour couronner le tout, le montant a été prélevé automatiquement, ce qui a déséquilibré votre budget du mois.

Activité :
Faites rédiger votre courrier par l'IA.

Rédiger un compte rendu de réunion

Situation :

Vous participer à une réunion municipale pour organiser une fête de quartier et vous êtes chargé d'en faire un compte-rendu. Vous prenez des notes brutes :

Date : 24 avril 2025
Lieu : Salle polyvalente – Mairie de Saint-Clair
Objet : Réunion d'organisation de la fête de quartier

Présents :

- Mme Dupont (Présidente de l'association)
- M. Leroy (Trésorier)
- Mme Bernard (Communication)
- M. Petit (Logistique)
- Mme Martin (Bénévole)

Excusés :

- Mme Caron (Secrétaire, maladie)

Rédiger un compte rendu de réunion

Déroulé :

Mme Dupont ouvre la réunion à 18h05.

Point budget (M. Leroy) :

Budget validé : 2 500 €.
Dépenses prévues : 1 200 € pour la scène, 800 € pour la restauration, reste 500 € pour imprévus.
Attention signalée sur nécessité d’avancer 30 % à la société de location.

Point communication (Mme Bernard) :

Flyers prêts, besoin de 500 impressions.
Budget impression estimé à 100 €.
Proposition : publication Facebook sponsorisée à 50 € → Soumis au vote.

Point logistique (M. Petit) :

Matériel (chaises, tables) disponible via prêt de l’école voisine.
Transport à organiser → Proposition de louer un utilitaire (coût estimé 100 €).

Rédiger un compte rendu de réunion

Votes :

Publication sponsorisée Facebook :
4 pour, 0 contre → Adopté.

Location utilitaire pour transport matériel :
4 pour, 0 contre → Adopté.

Divers :

Besoin de bénévoles supplémentaires pour montage/démontage → Appel à volontaires via mailing.

Conclusion (Mme Dupont) :
Remercie les présents.
Rappelle que la prochaine réunion aura lieu le 8 mai à 18h.
Souligne l'importance d’envoyer rapidement les invitations au public.

Fin de séance à 19h10.

Activité : Faites rédiger le compte-rendu par l’IA

Relations humaines et émotions

Contexte : Naissance de la petite Éléa

Marcel et Sophie, vos amis de longue date, viennent de donner naissance à leur premier enfant, une petite fille prénommée Éléa.

Leur parcours pour devenir parents a été long et éprouvant :

Plusieurs années d'infertilité inexpliquée, avec de nombreux échecs émotionnellement lourds.

Après plusieurs traitements, ils ont eu recours à une FIV (fécondation in vitro) qui a, enfin, abouti.

La grossesse a été considérée à haut risque dès le début : menace d’accouchement prématuré, hospitalisations ponctuelles, surveillance rapprochée.

L'accouchement a été délicat, avec une césarienne en urgence à 34 semaines de grossesse, mais, heureusement, tout le monde va bien aujourd'hui.

Relations humaines et émotions

Vous souhaitez leur adresser une message de félicitation.

Émotion que le message devrait faire passer :

- **Immense joie et soulagement après toutes les épreuves traversées.**
- **Tendresse pour Éléa, leur « petite étoile » tant attendue.**
- **Fierté pour leur force et leur courage.**
- **Beaucoup de douceur et de respect pour la discrétion nécessaire autour d'une histoire aussi intime.**

Activité : Faites rédiger le message par l’IA

Assistance technique et numérique

Savez-vous repérer les pièges invisibles d'un tableau Excel ?

Formules décalées, références cassées, valeurs manquantes, primes oubliées...

À travers un vrai classeur piégé, découvrez comment l'IA peut vous aider à analyser, corriger et construire des formules intelligentes pour éviter les erreurs coûteuses !

Activité :

- **Repérer et corriger un maximum d'erreurs dans un fichier Excel en un temps limité.**
- **Fournir un fichier à l'IA**
- **Utiliser l'IA pour formuler des demandes claires d'aide technique.**
- **Modifier la dernière colonne pour qu'elle prenne en compte cette nouvelle directive :**
  - **Si un salarié a travaillé plus de 40 heures, il a droit à une prime supplémentaire de 10% de son salaire brut (Heures × Taux).**

Télécharger le fichier : exemple_erreurs_excel.xlsx

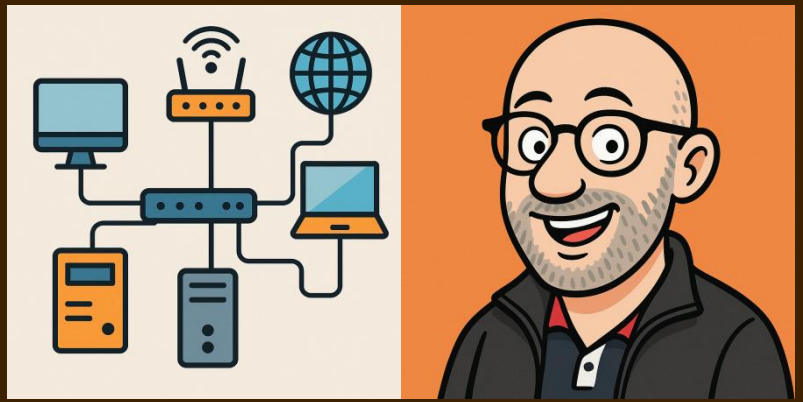
Créer une image

Activité :

- **Créer l’image de votre choix en la décrivant à l’IA**
- **Fournir une photo d’une personne et demander à l’IA d’en faire un avatar**
- **Utiliser une IA spécialisée pour générer des images**

- **Exemple : « Génère une image représentant un réseau informatique simple pour illustrer un cours »**

- https://magichour.ai/
- https://perchance.org/ai-text-to-image-generator
- https://pixnova.ai/free-realistic-ai-image-generator/
- ...



Thank you!

Do you have any questions?

Intelligence artificielle :
guide pratique pour les enseignants
Philippe Piekoszewski-Cuq

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**

https://latelierduformateur.fr/5-nouveaux-guides-sur-lia-en-formation/
https://nuage03.apps.education.fr/index.php/s/7cQfKmEtHHZgCxT
https://www.youtube.com/@MachineLearnia
https://www.youtube.com/@OBJECTIFSCIENCE