

# Pour aller plus loin

Différentes architectures pour le Machine Learning

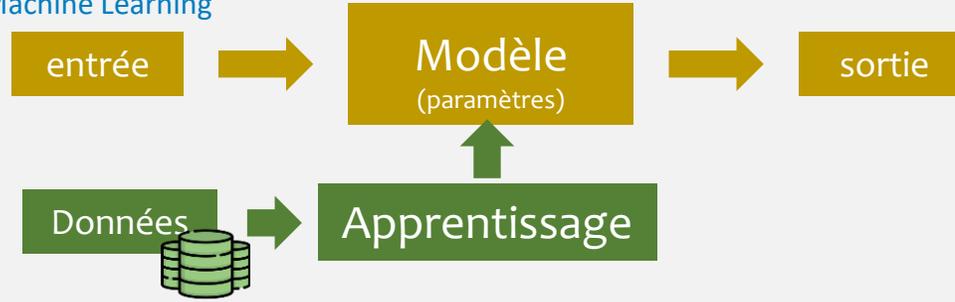


# Introduction à l'IA : Résumé de la séance 1

Approche Cognitiviste  
(« ancienne »)

Systèmes experts,  
calcul formel, etc.

Approche Connexioniste :  
Machine Learning



Quel sera le futur ?

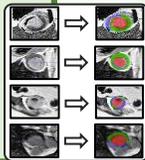
Hybridations,  
Robotique,  
Autre ?



## 3 Grandes méthodes d'apprentissages :

### Apprentissage Supervisé

Données =  
exemples d'entrées + sorties



### Apprentissage Non-supervisé

Données =  
uniquement exemples d'entrées



### Apprentissage Par Renforcement

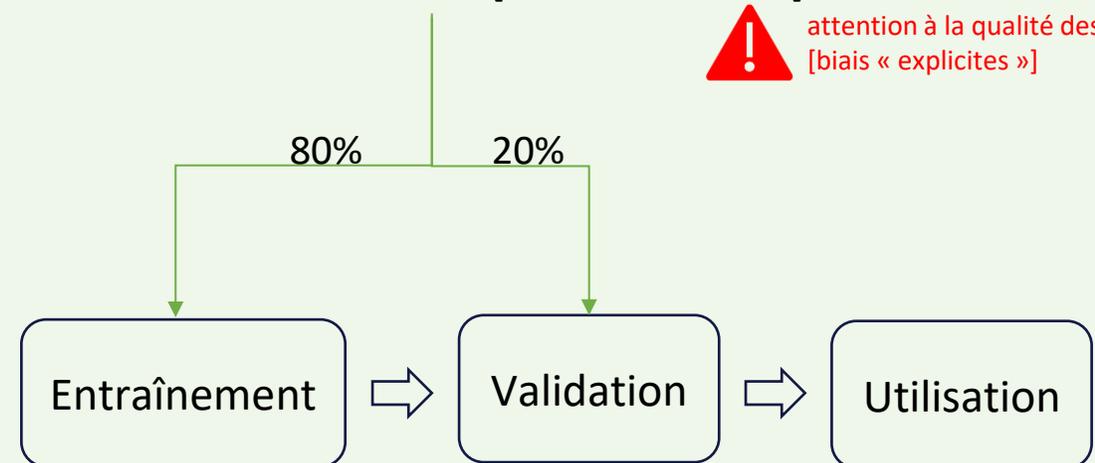
Données =  
entrées + (tentatives de) sorties +  
récompenses reçues



Combinaisons...

## L'apprentissage supervisé :

Données d'entraînement [entrées + sorties]



attention à la qualité des données  
[biais « explicites »]



attention aux erreurs de généralisation  
[biais « implicites »]

# Pour aller plus loin

## Algorithmes classiques

### Apprentissage supervisé

- Notion essentielle : minimisation de la « loss »
- Classification versus Régression
- Un autre algorithme classique : Random Forest

### Apprentissage par renforcement

- Value Based, Policy Based, et Value+Policy Based

### Apprentissage non supervisé

- Clustering
- Réduction de dimensions

## Architectures plus avancées

### Réseaux de neurones convolutionnels (CNN)

Génération d'images avec les **Generative-Adversial Networks (GAN)**

**Modèles de langage** : token embedding, Transformers, spécialisation de ChatGPT

# Pour aller plus loin

## Algorithmes classiques

### Apprentissage supervisé

- Notion essentielle : minimisation de la « loss »
- Classification versus Régression
- Un autre algorithme classique : Random Forest

### Apprentissage par renforcement

- Value Based, Policy Based, et Value+Policy Based

### Apprentissage non supervisé

- Clustering
- Réduction de dimensions

## Architectures plus avancées

Réseaux de neurones convolutionnels (CNN)

Génération d'images avec les Generative-Adversarial Networks (GAN)

Modèles de langage : token embedding, Transformers, spécialisation de ChatGPT

# Apprentissage Supervisé : Minimisation de l'Erreur

AlphaI, robot apprenant  
Paramètres IA Outils Aide



### Apprentissage supervisé, piloté. Mémoire d'expérience : 16

The interface displays three main data plots on the left:

- Données capteurs:** A plot of sensor values over time, showing a step function that drops from 1 to 0 at approximately 0.5 seconds.
- Récompense:** A plot showing a constant reward of 0.
- Erreur:** A plot showing the error starting at approximately 0.65 and decaying towards 0 over 600 steps.

On the right, a neural network diagram illustrates the model's internal state. It features a central input node with a value of 0.677, which is multiplied by various weights to produce outputs. The weights and their corresponding functions are:

- $f(-1.17) = 0.22$  with weights  $-6.86$ ,  $-7.36$ ,  $-7.12$
- $f(1.16) = 0.77$  with weights  $4.32$ ,  $6.87$ ,  $4.32$
- $f(-6.56) = 1.00$  with weights  $27.75$ ,  $-3.98$ ,  $-28.26$
- $f(1.16) = 0.77$  with weights  $4.44$ ,  $3.82$ ,  $3.82$
- $f(-1.17) = 0.22$  with weights  $-7.01$ ,  $6.84$ ,  $6.84$

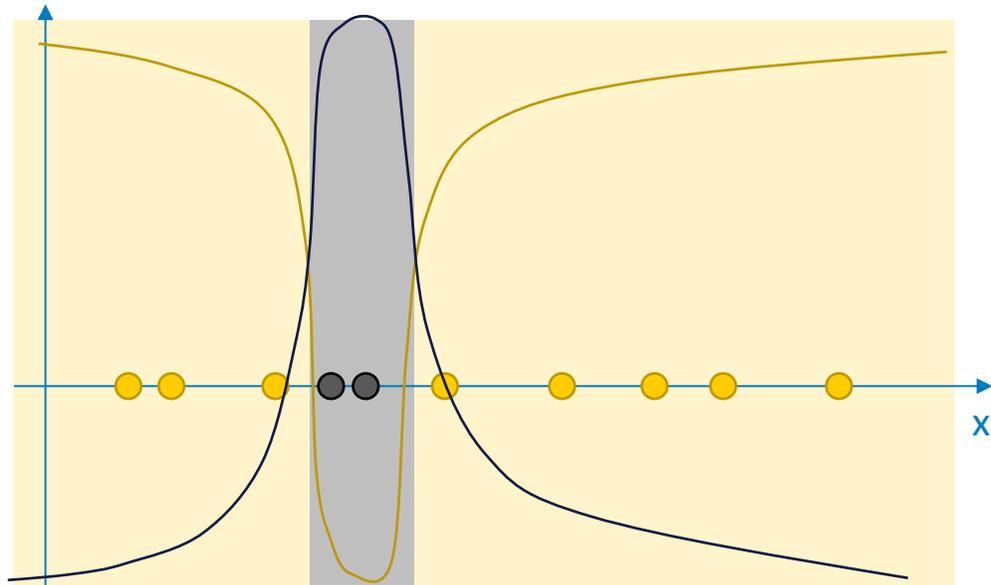
The network outputs are shown as nodes with values: 0.11 (with a yellow arrow pointing up to 227) and 1.41 (with a blue arrow pointing down to 150). A red box highlights a small circle in the bottom right corner.

At the bottom, a control bar includes buttons for: autonome, pas à pas, apprentissage, édition manuelle, oublier action, réinitialiser l'IA, graphes, vitesse : 30, simulation accélérée, and déco...xion.

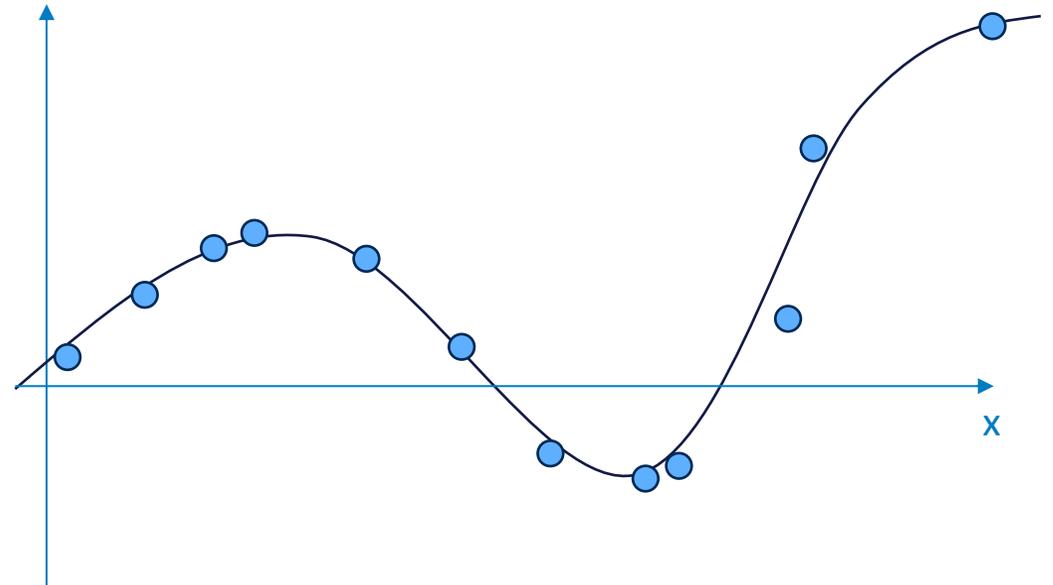
# Apprentissage Supervisé : Classification versus Régression

1D

Classification



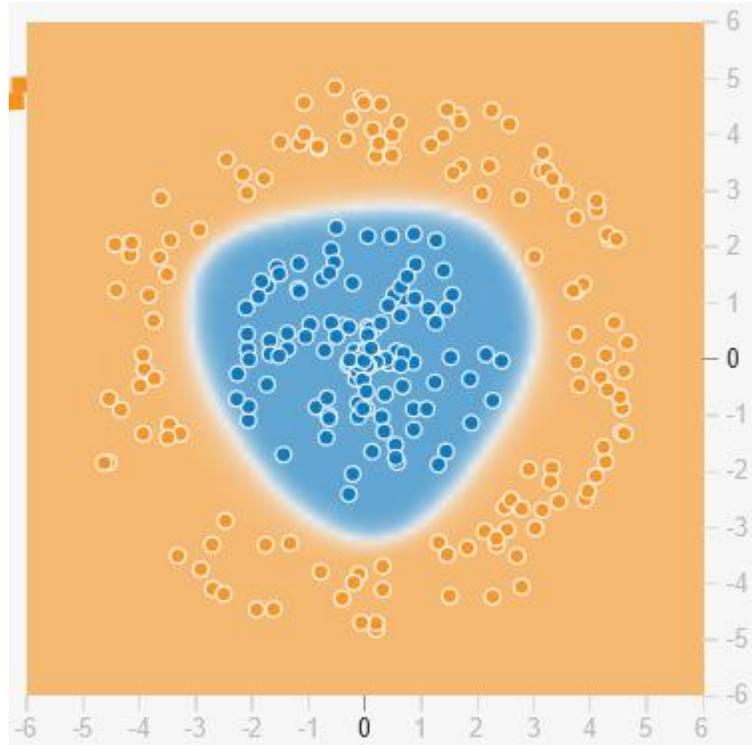
Régression



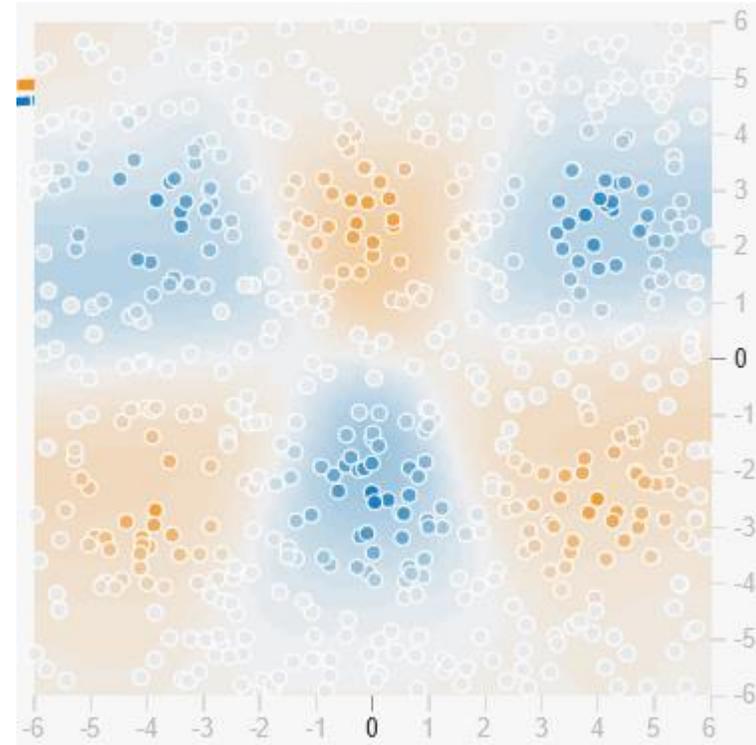
# Apprentissage Supervisé : Classification versus Régression

2D

Classification



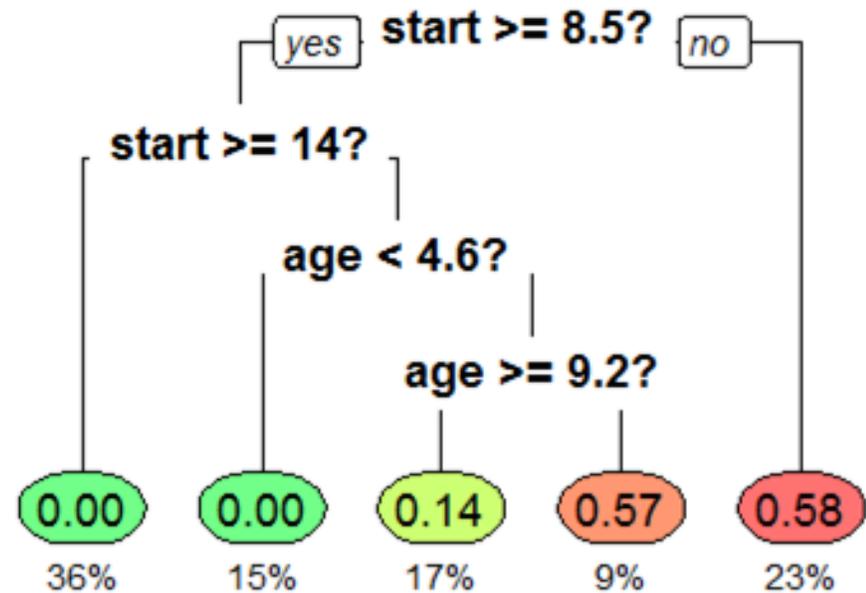
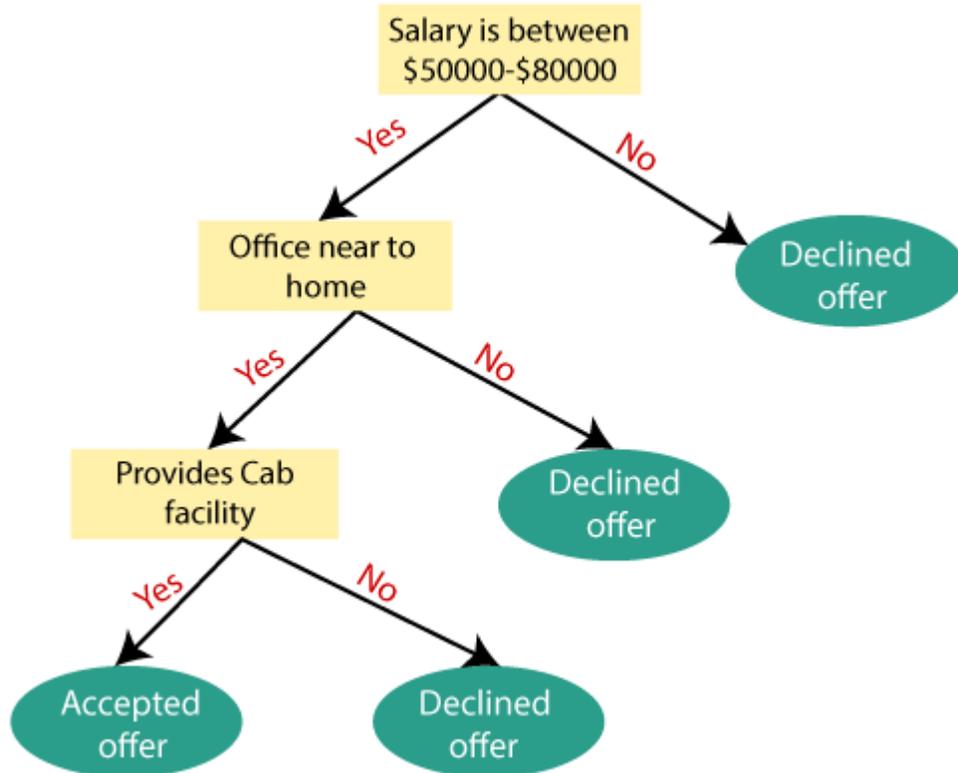
Régression



<https://playground.tensorflow.org>

# Apprentissage Supervisé : Forêt Aléatoire

## Arbres de décision



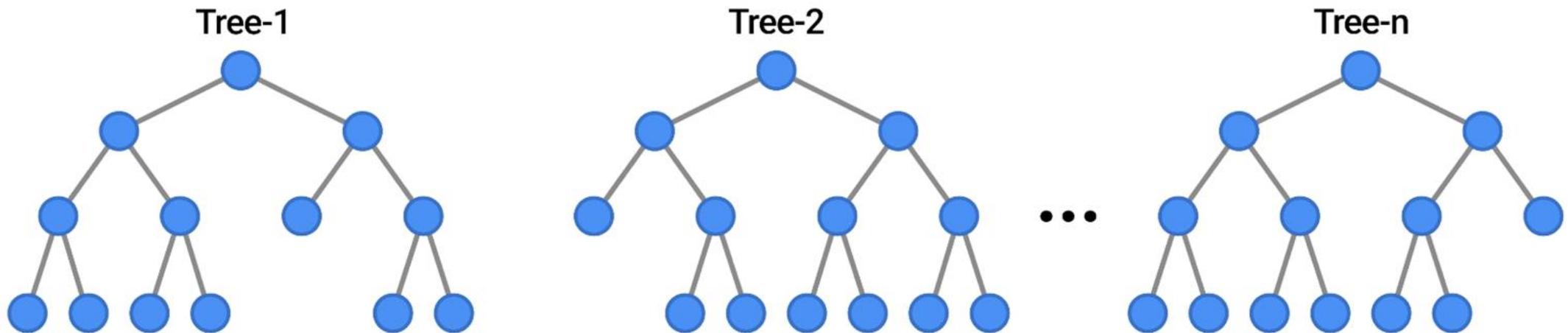
Paramètres = quel critère regarder en premier, en deuxièmes ? Quelle règle de décision ? Quel seuil ?

Entraînement = modifier les paramètres pour trouver les meilleures règles de décisions sur les données d'entraînement

!

# Apprentissage Supervisé : Forêt Aléatoire

Entraîner plusieurs arbres de décision simultanément, qui prennent des décisions collectivement



# Pour aller plus loin

## Algorithmes classiques

### Apprentissage supervisé

- Notion essentielle : minimisation de la « loss »
- Classification versus Régression
- Un autre algorithme classique : Random Forest

### Apprentissage par renforcement

- Value Based, Policy Based, et Value+Policy Based

### Apprentissage non supervisé

- Clustering
- Réduction de dimensions

## Architectures plus avancées

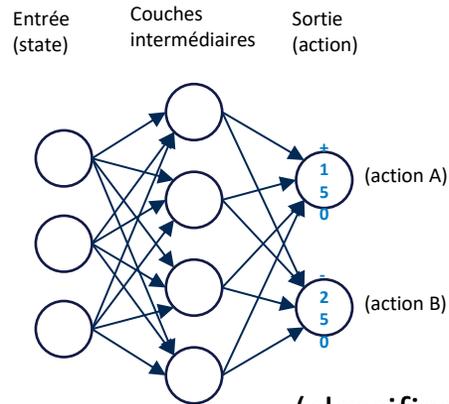
Réseaux de neurones convolutionnels (CNN)

Génération d'images avec les Generative-Adversarial Networks (GAN)

Modèles de langage : token embedding, Transformers, spécialisation de ChatGPT

# Apprentissage par Renforcement : Policy-based vs. Value-based

## Policy-based

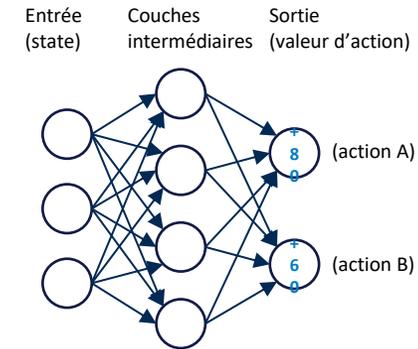


(classification ! peu importe les valeurs de sortie exactes)

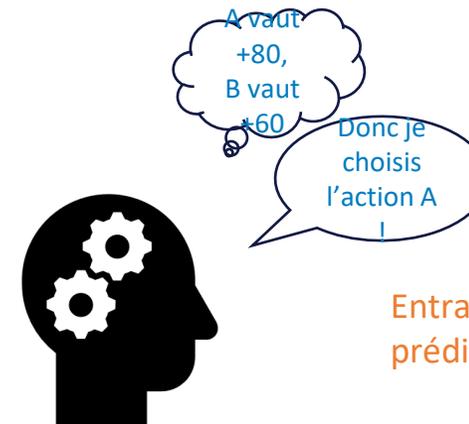


Entraînement : modifier aléatoirement les paramètres et regarder si cela améliore ou pas les récompenses reçues

## Value-based

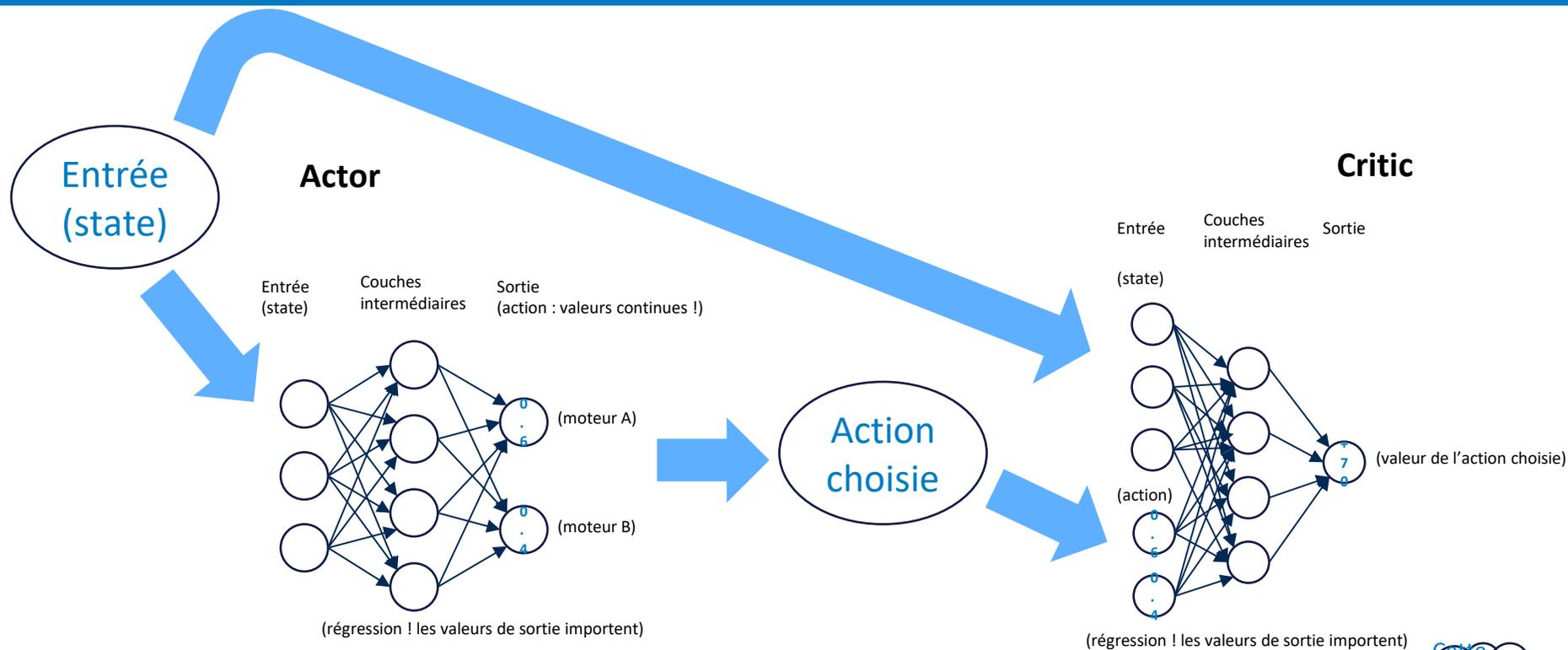


(régression ! les valeurs de sortie important)



Entraînement : apprendre à prédire les récompenses futures

# Apprentissage par Renforcement : Actor Critic = Policy & Value - based



Entraînement :

- Les rewards et nouveaux états reçus servent à entraîner **Critic**
- Critic sert à entraîner **Actor**

# Pour aller plus loin

## Algorithmes classiques

### Apprentissage supervisé

- Notion essentielle : minimisation de la « loss »
- Classification versus Régression
- Un autre algorithme classique : Random Forest

### Apprentissage par renforcement

- Value Based, Policy Based, et Value+Policy Based

### Apprentissage non supervisé

- Clustering
- Réduction de dimensions

## Architectures plus avancées

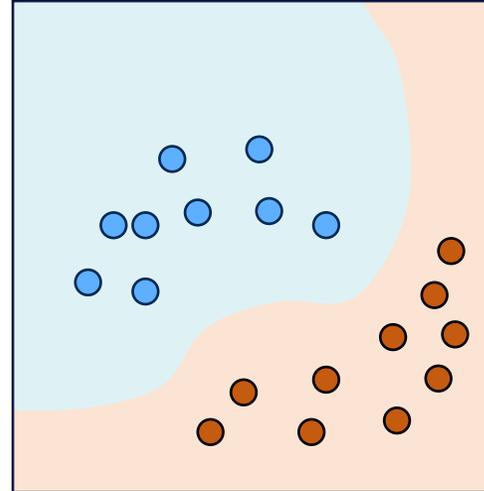
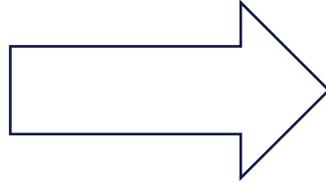
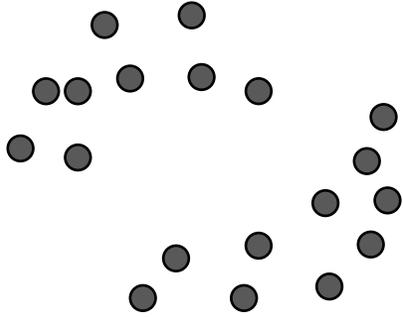
Réseaux de neurones convolutionnels (CNN)

Génération d'images avec les Generative-Adversarial Networks (GAN)

Génération d'images avec Stable Diffusion

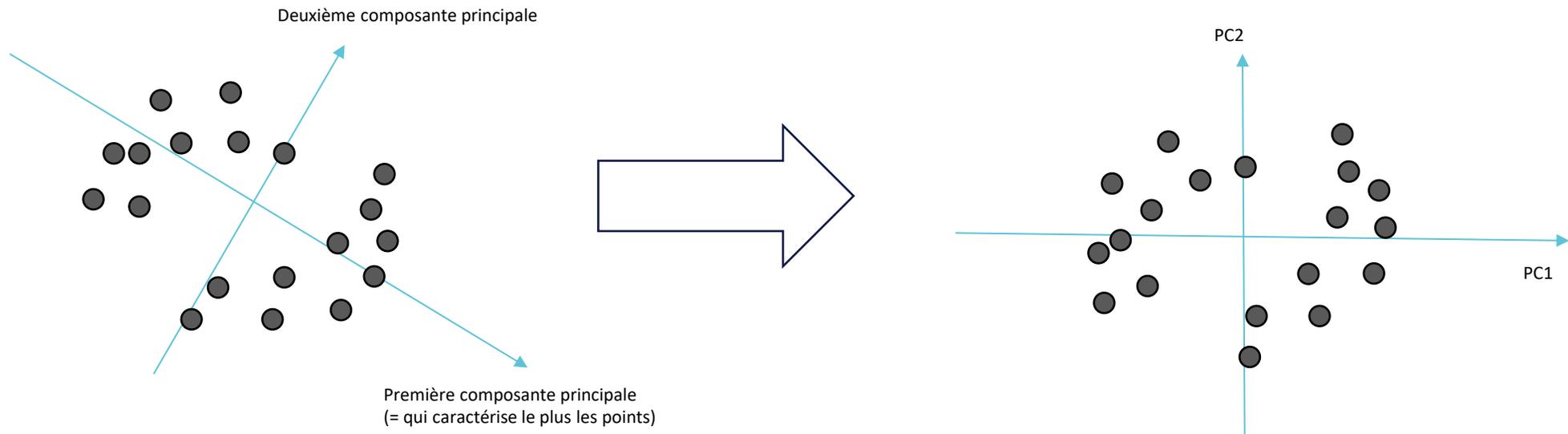
Modèles de langage : token embedding, Transformers, spécialisation de ChatGPT

# Apprentissage non supervisé : Clustering

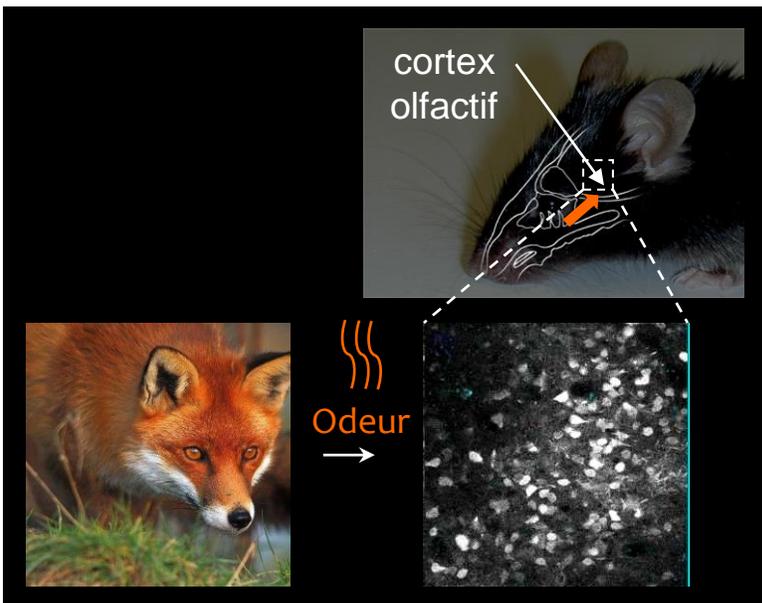


# Apprentissage non supervisé : Réduction de dimensions

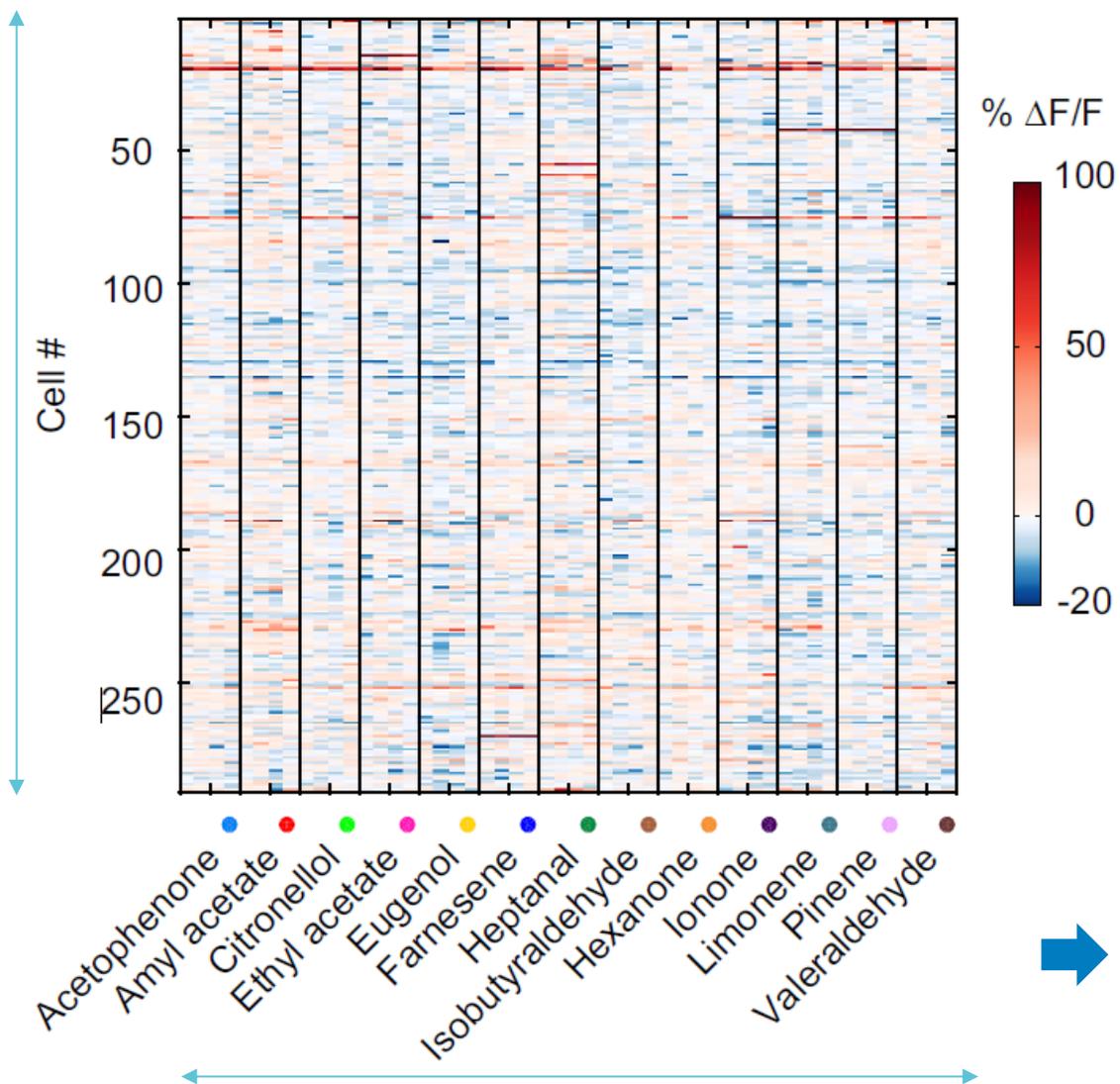
## Principal Component Analysis (PCA)



# Apprentissage non supervisé : Réduction de Dimensions



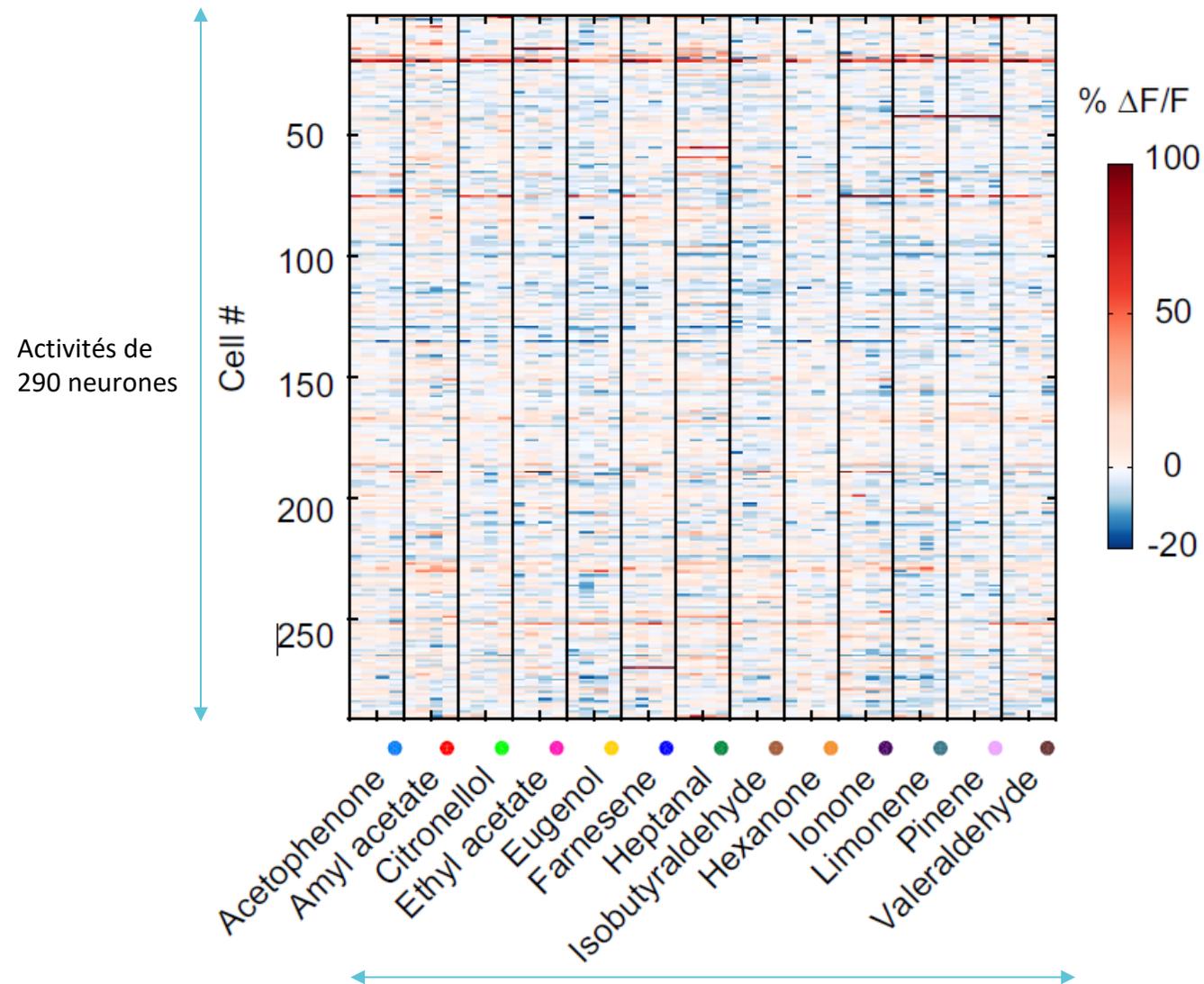
Activités de  
290 neurones



En réponse à 52 stimulations olfactives  
(4 répétitions à chaque fois de 13 odeurs différentes)

➡ Comment mesurer et visualiser la proximité des réponses à différentes odeurs ?

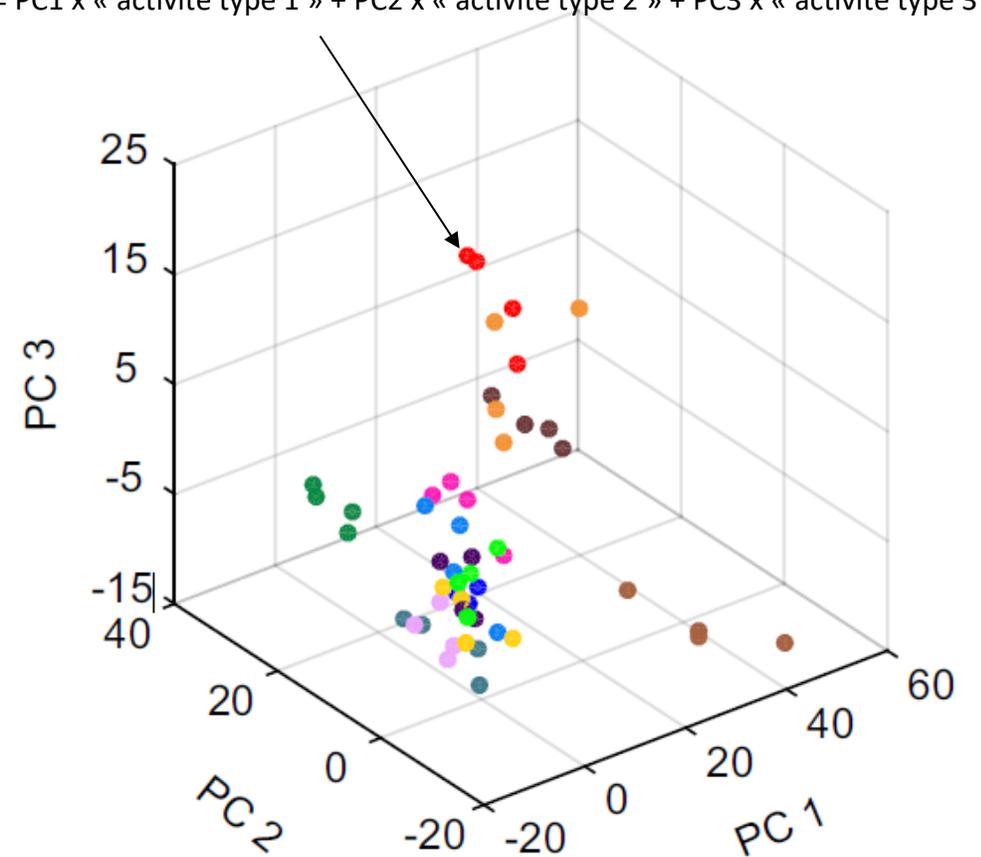
# Apprentissage non supervisé : Réduction de Dimensions



En réponse à 52 stimulations olfactives  
(4 répétitions à chaque fois de 13 odeurs différentes)

## Principal Component Analysis (PCA)

Activité des 290 neurones pour répétition 1 de Amyl acetate  
= PC1 x « activité type 1 » + PC2 x « activité type 2 » + PC3 x « activité type 3 »



# Pour aller plus loin

## Algorithmes classiques

### Apprentissage supervisé

- Notion essentielle : minimisation de la « loss »
- Classification versus Régression
- Un autre algorithme classique : Random Forest

### Apprentissage par renforcement

- Value Based, Policy Based, et Value+Policy Based

### Apprentissage non supervisé

- Clustering
- Réduction de dimensions

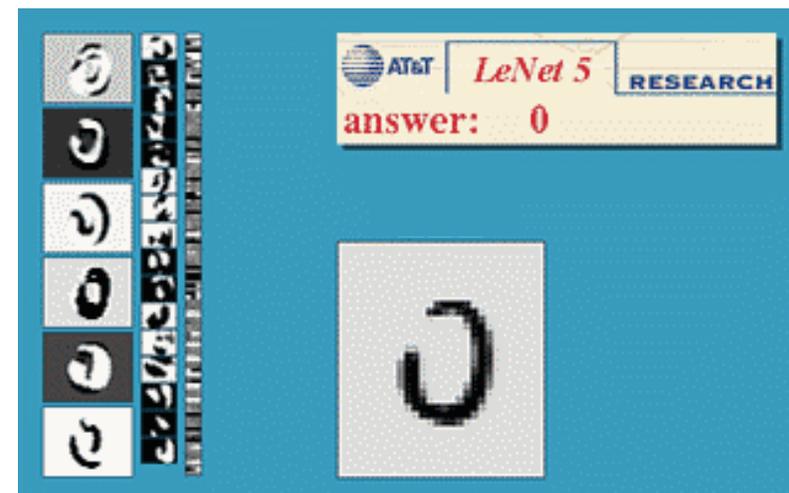
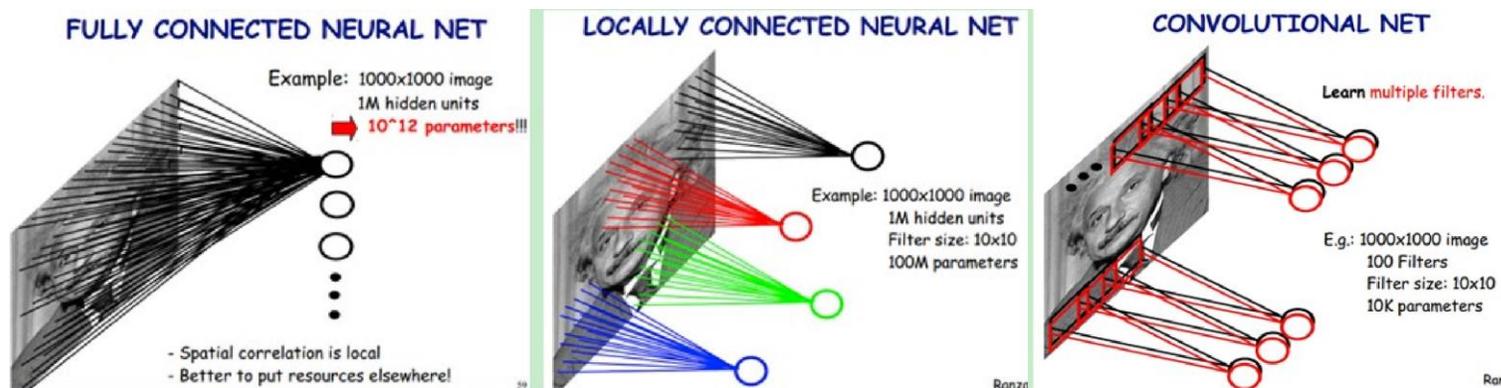
## Architectures plus avancées

Réseaux de neurones convolutionnels (CNN)

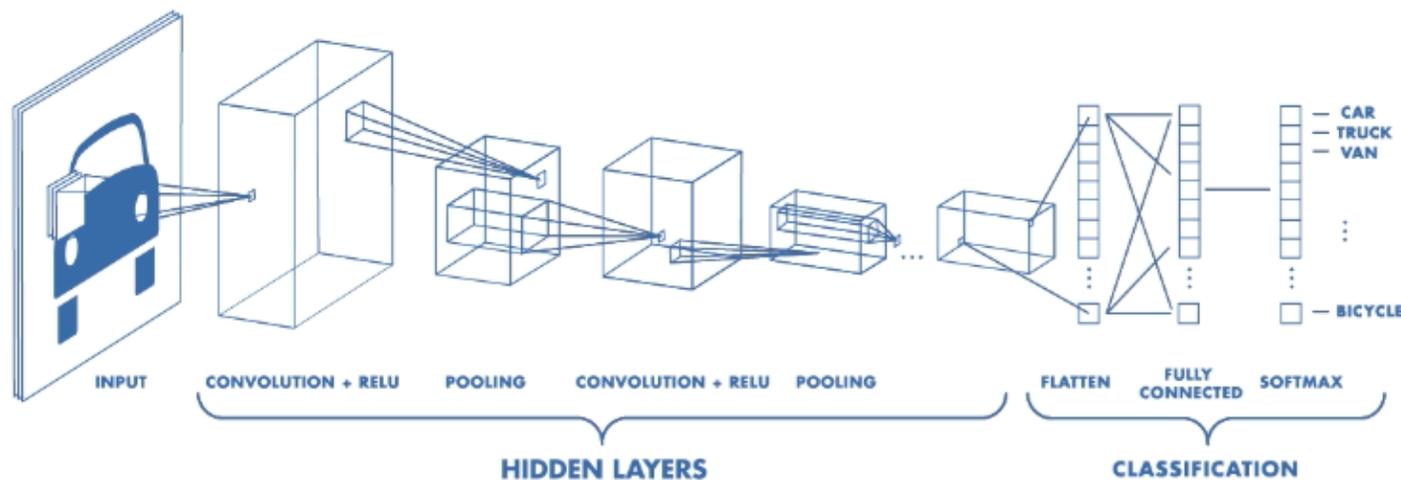
Génération d'images avec les Generative-Adversial Networks (GAN)

Modèles de langage : token embedding, Transformers, spécialisation de ChatGPT

# Reconnaissance d'image : Réseaux de Neurones Convolutionnels (Convolutional Neural Networks, CNN)



<http://yann.lecun.com/exdb/lenet>



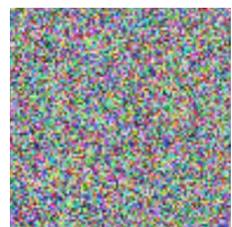
<https://cs.stanford.edu/people/karpathy/cvnnvnetjs/demo/cifar10.html>

# Génération d'Images

## Generative Adversarial Networks (GAN)

Generating fake data!

Noise  $\sim N(0,1)$



Generative Model



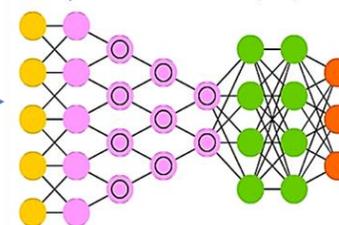
How this works

Real faces



Discriminator

Deep Convolutional Network (DCN)



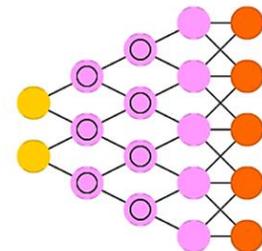
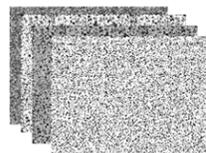
Fake

Real

Generator

Deconvolutional Network (DN)

Random noise

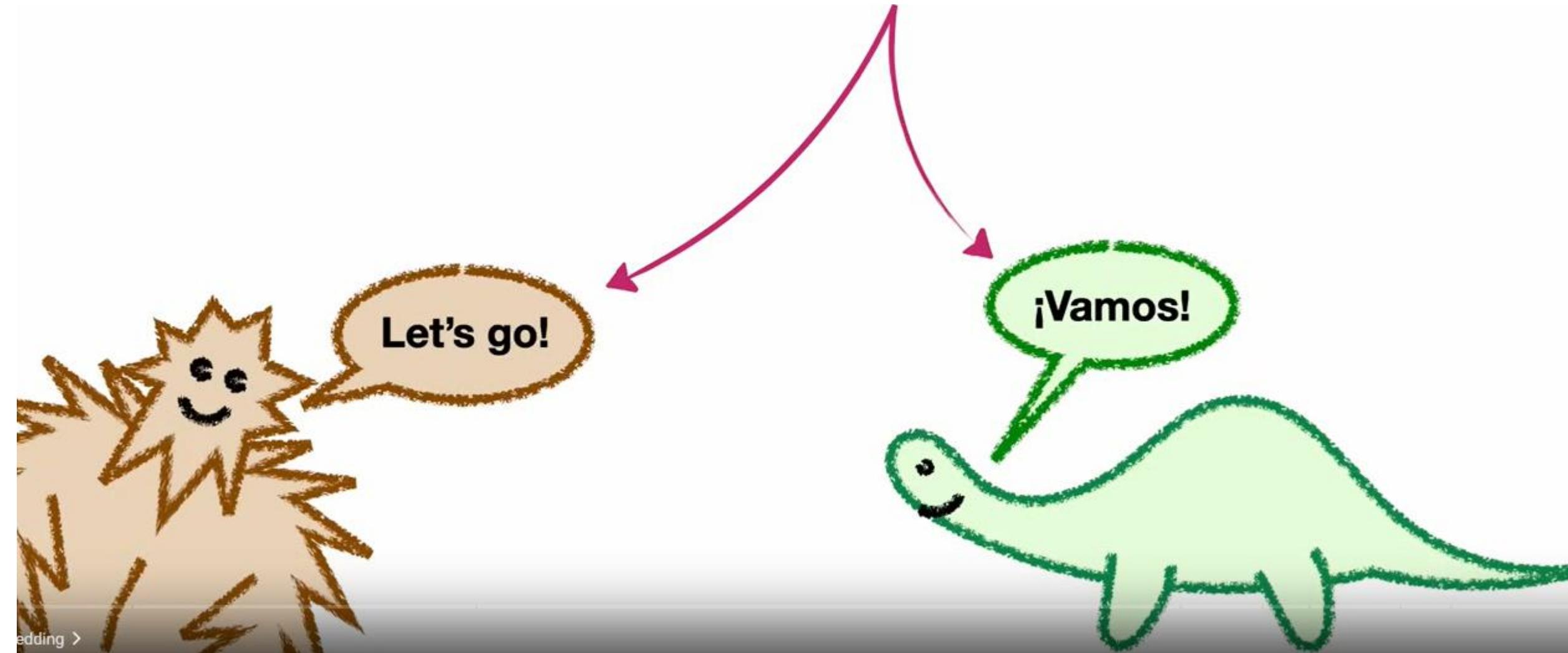


Generated faces



# Language Models avec StatQuest (Josh Starmer)

<https://www.youtube.com/watch?v=zxQyTK8quyY>



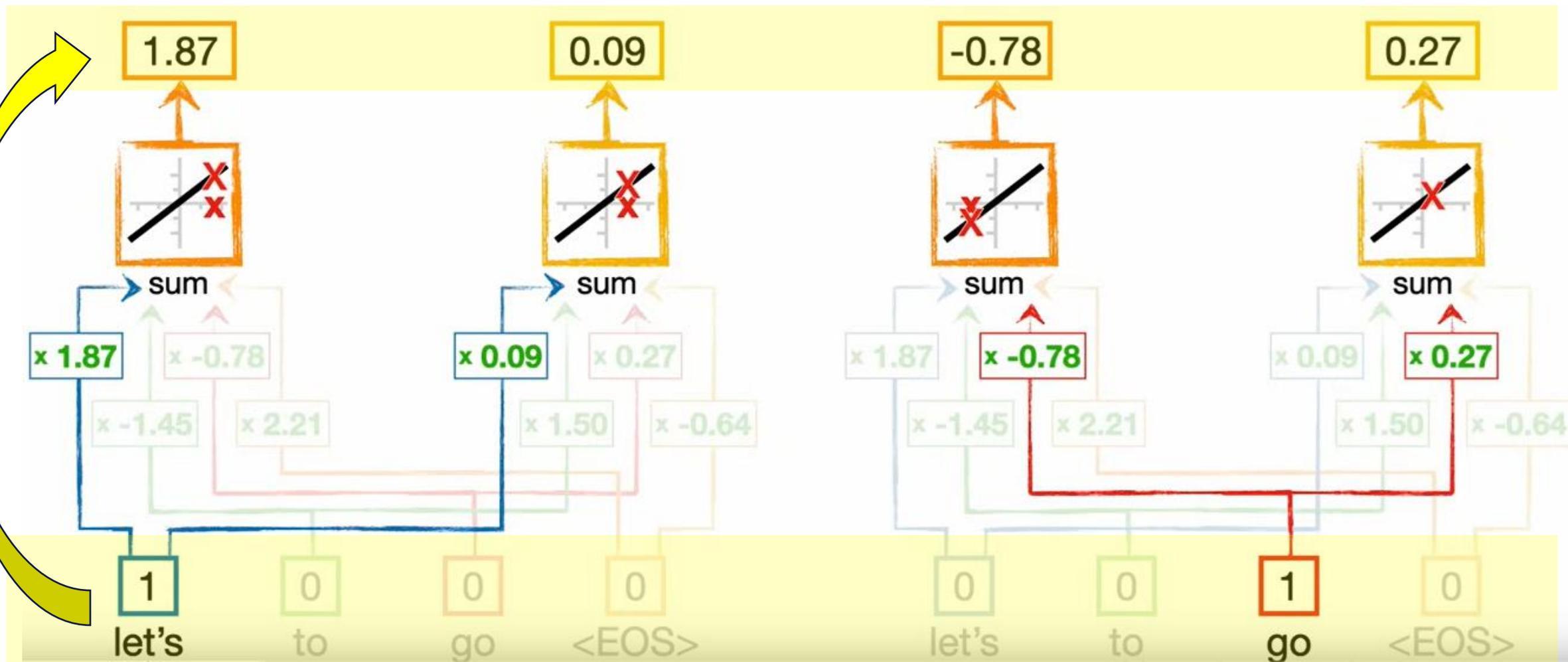
# Language Models : word embedding

<https://www.youtube.com/watch?v=zxQyTK8quyY>

Chaque **mot** est représenté par un ensemble de **nombres**  
Tous les mots de la phrase sont traités en parallèle !

Le mot « **let's** » est représenté par les nombres **(1.87, 0.09)**

Le mot « **go** » est représenté par les nombres **(-0.78, 0.27)**



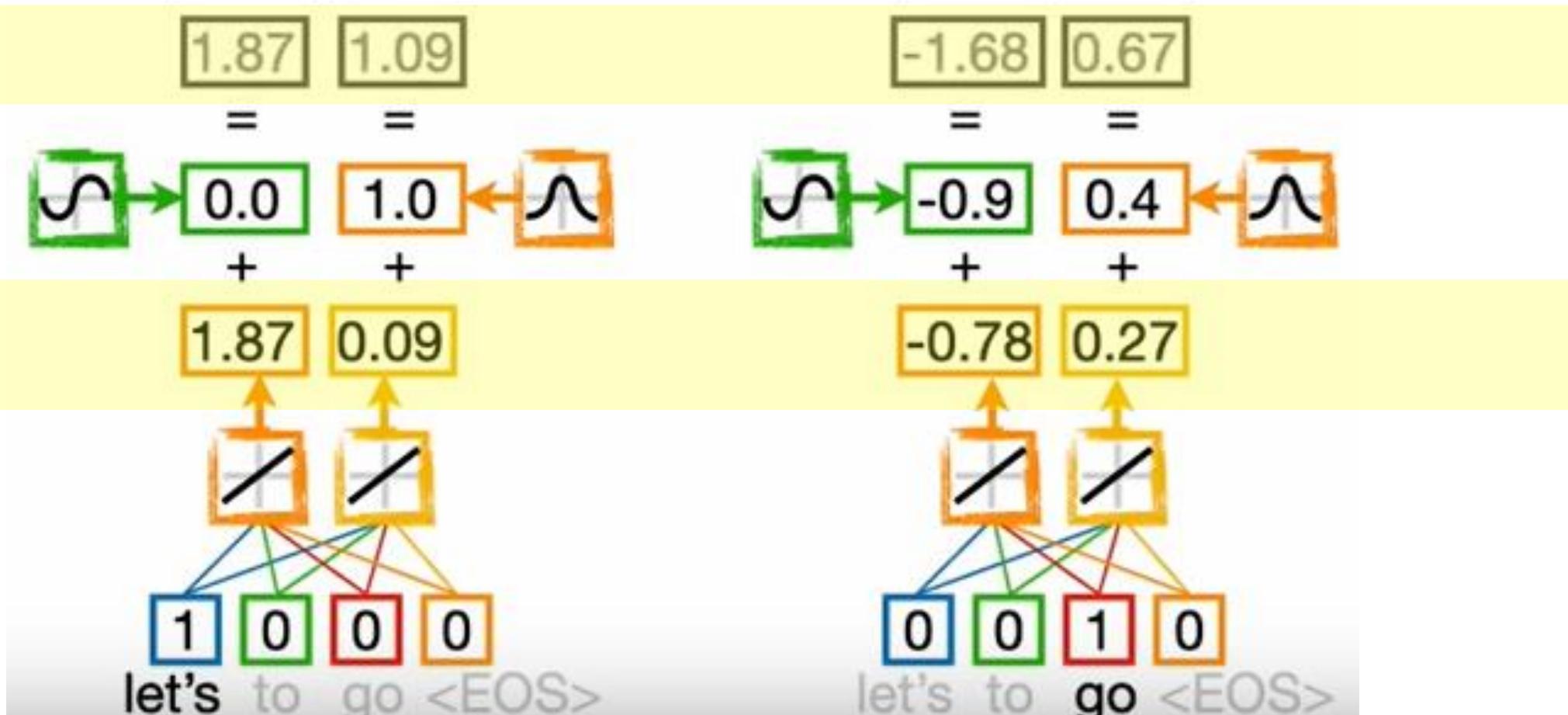
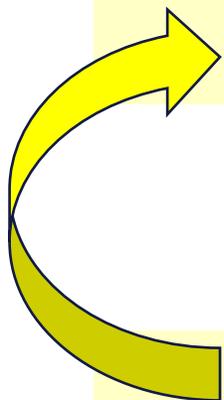
# Langage Models : positionnal embedding

<https://www.youtube.com/watch?v=zxQyTK8quyY>

La representation est modifiée en fonction de la **position** du mot dans la phrase

Le mot « **let's** » est maintenant représenté par les nombres **(1.87, 1.09)**

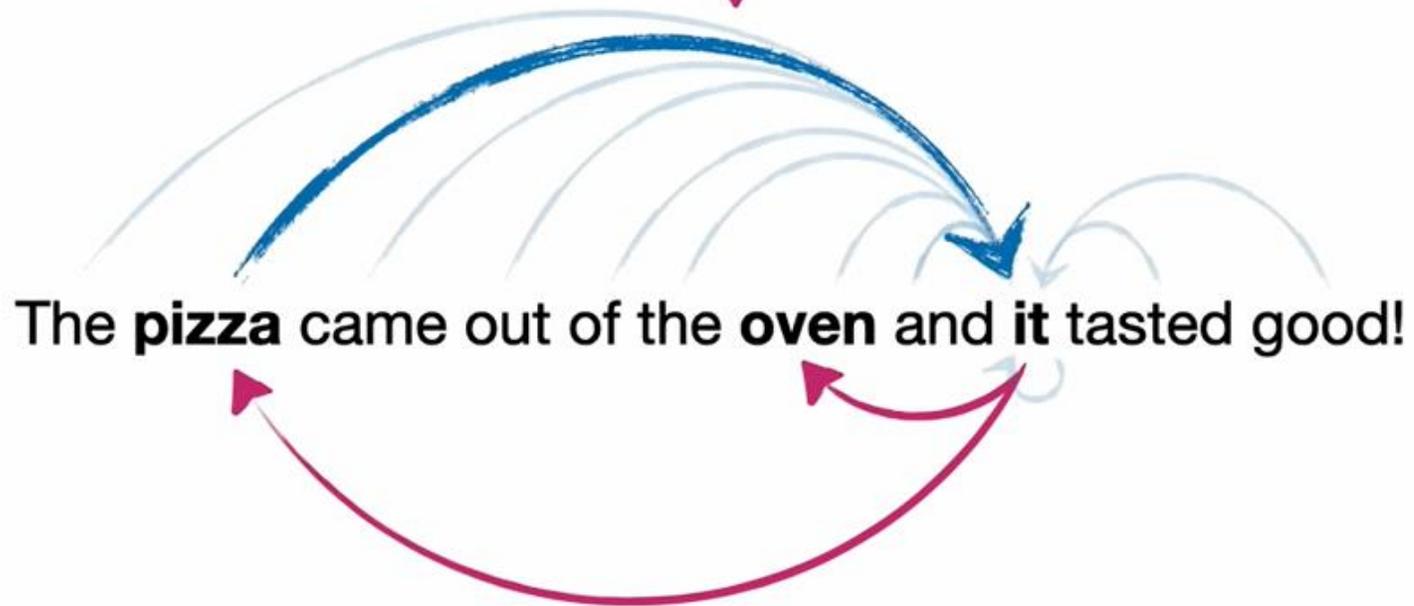
Le mot « **go** » est maintenant représenté par les nombres **(-1.68, 0.67)**



# Language Models : self-attention

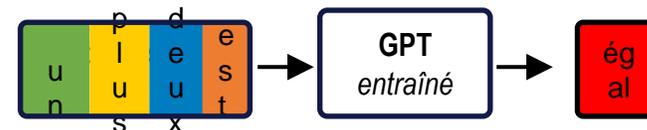
<https://www.youtube.com/watch?v=zxQyTK8quyY>

...the word **it** should have a relatively large similarity value with respect to the word **pizza**, since **it** refers to **pizza** and not **oven**.



# Language Models : ChatGPT

GPT = modèle de langage capable de deviner le mot suivant dans une phrase



Spécialisation par :

- 1 – apprentissage supervisé
- 2 – apprentissage supervisé d'une récompense
- 3 – apprentissage par renforcement

Step 1  
Collect demonstration data  
and train a supervised policy.

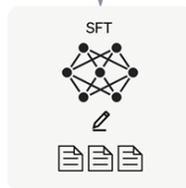
A prompt is  
sampled from our  
prompt dataset.



A labeler  
demonstrates the  
desired output  
behavior.

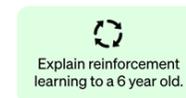


This data is used to  
fine-tune GPT-3.5  
with supervised  
learning.



Step 2  
Collect comparison data and  
train a reward model.

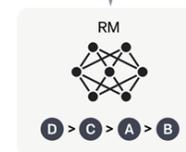
A prompt and  
several model  
outputs are  
sampled.



A labeler ranks the  
outputs from best  
to worst.



This data is used  
to train our  
reward model.



Step 3  
Optimize a policy against the  
reward model using the PPO  
reinforcement learning algorithm.

A new prompt is  
sampled from  
the dataset.



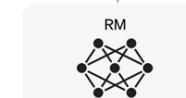
The PPO model is  
initialized from the  
supervised policy.



The policy generates  
an output.



The reward model  
calculates a reward  
for the output.



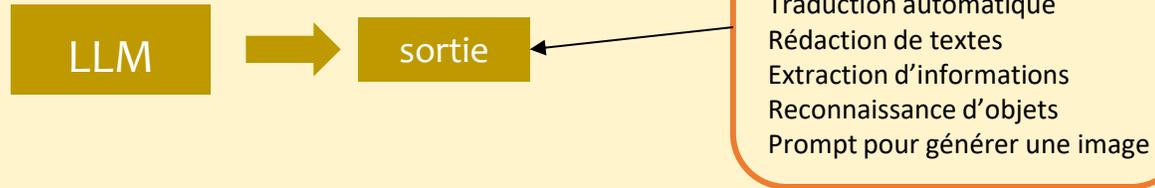
The reward is used  
to update the  
policy using PPO.



ChatGPT = programme spécialisé pour un mode conversationnel

# Que veut dire « GPT » dans « ChatGPT » ?

Les **IA génératives** génèrent des sorties riches  
Parmi elles les **LLM** génèrent du texte



**GPT = Generative Pre-trained Transformer**

# A l'intérieur d'un LLM !

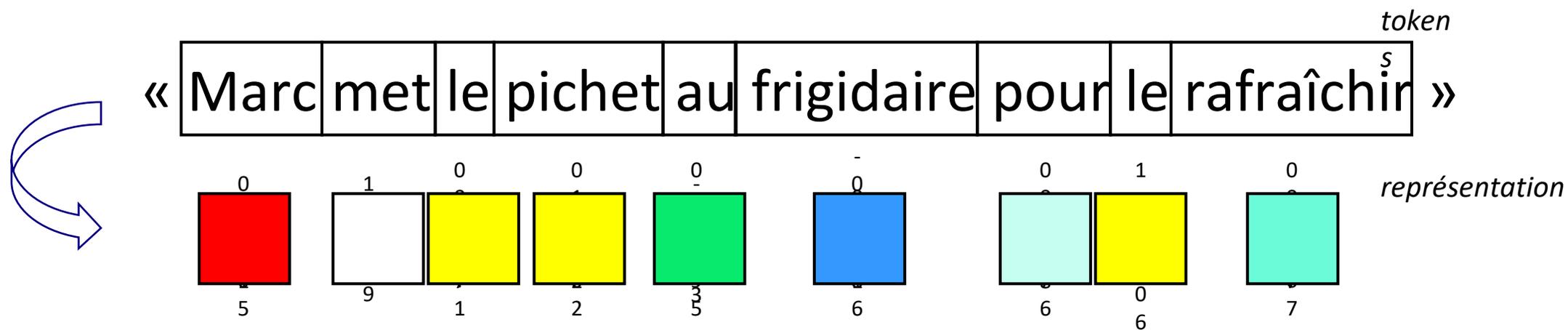
Animation avec le mini-LLM nano-gpt : <https://bbycroft.net/llm>

The screenshot displays the 'LLM Visualization' interface for nano-gpt. The top navigation bar includes a home button, a chapter overview, and model selection options: GPT-2 (small), nano-gpt (selected), GPT-2 (XL), and GPT-3. The main content area is divided into a left sidebar with a 'Table of Contents' (Intro, Introduction, Preliminaries, Components, Embedding, Layer Norm, Self Attention, Projection, MLP, Transformer, Softmax, Output) and a central visualization of the LLM architecture. The architecture diagram shows the flow from input tokens ('How to predict text') through token embeddings, position embeddings, and a series of transformer blocks. Each transformer block contains a multi-head, causal self-attention mechanism, followed by a residual connection and layer normalization, a feed-forward network, another residual connection and layer normalization, and finally a linear layer and softmax output. A right-hand panel shows a detailed view of a 'Réseau de neurones « classique »' (classical neural network) with nodes and weights, and a '« Transformer »' section. The interface also includes a 'Continue' button and a 'Skip' button at the bottom.

# Qu'est-ce qu'un Transformer?

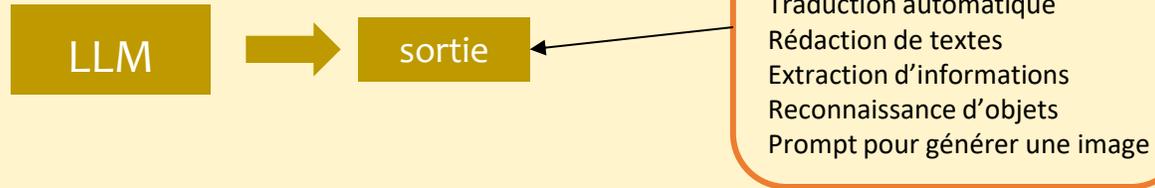
La technique « **Transformers** » transforme des nombres qui représentent du texte, en mettant en valeur les relations entre les mots

Elle a été inventée par les équipes de Google en 2017.

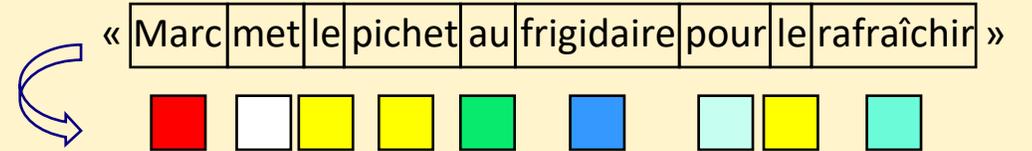


# Que veut dire « GPT » dans « ChatGPT » ?

Les **IA génératives** génèrent des sorties riches  
Parmi elles les **LLM** génèrent du texte



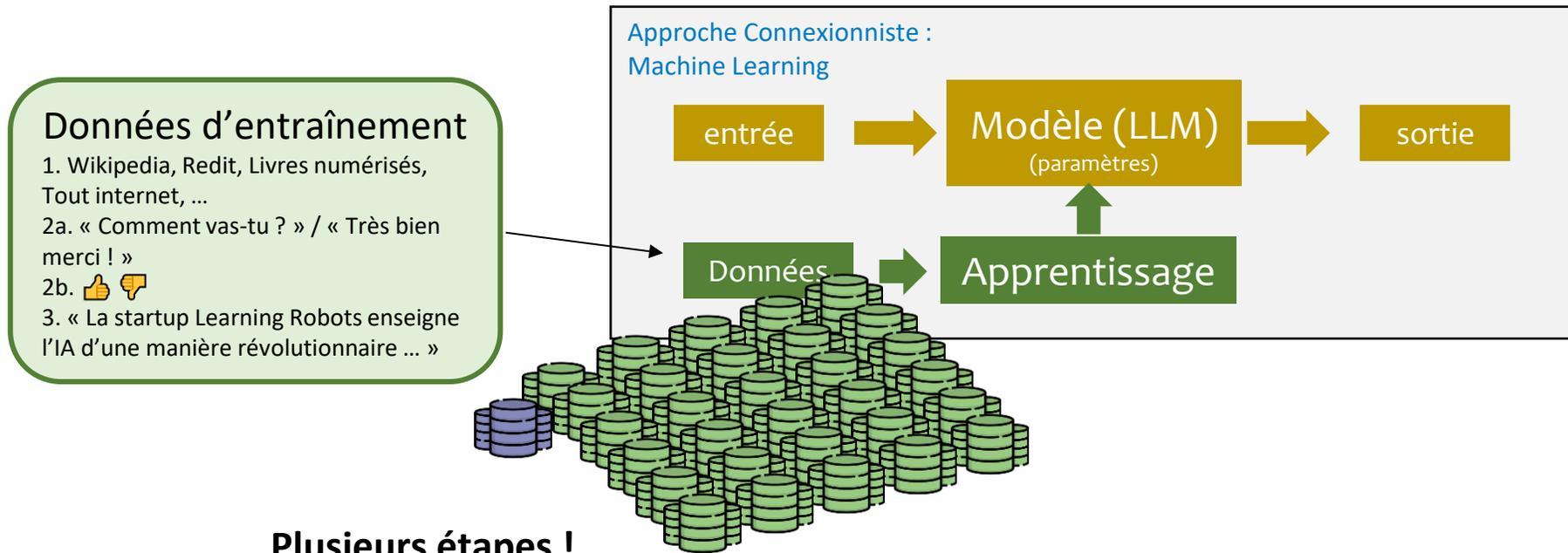
La technique « **Transformers** » transforme du texte en nombres en mettant en valeur les relations entre les mots



**GPT = Generative Pre-trained Transformer**

# Entraînement d'un LLM

## Comment les LLMs utilisent le Machine Learning pour apprendre ?

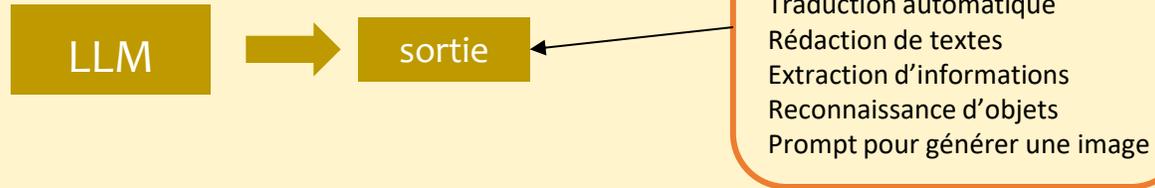


### Plusieurs étapes !

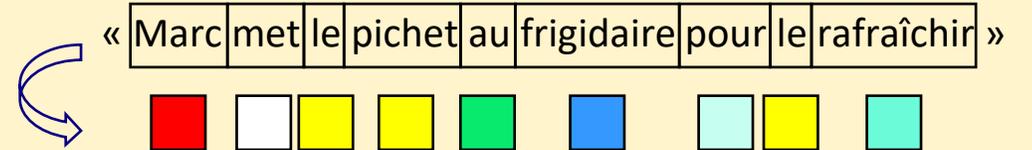
1. **Pré-entraînement du LLM** : Apprentissage Supervisé sur d'énormes corpus textuels.
2. **Spécialisation comme ChatBot** : pour ChatGPT
  - a. Apprentissage Supervisé avec des exemples de questions et réponses
  - b. Apprentissage par Renforcement avec des notes données aux réponses
3. **Spécialisation pour un client** : Apprentissage Supervisé sur des données spécialisées.

# Que veut dire « GPT » dans « ChatGPT » ?

Les **IA génératives** génèrent des sorties riches  
Parmi elles les **LLM** génèrent du texte

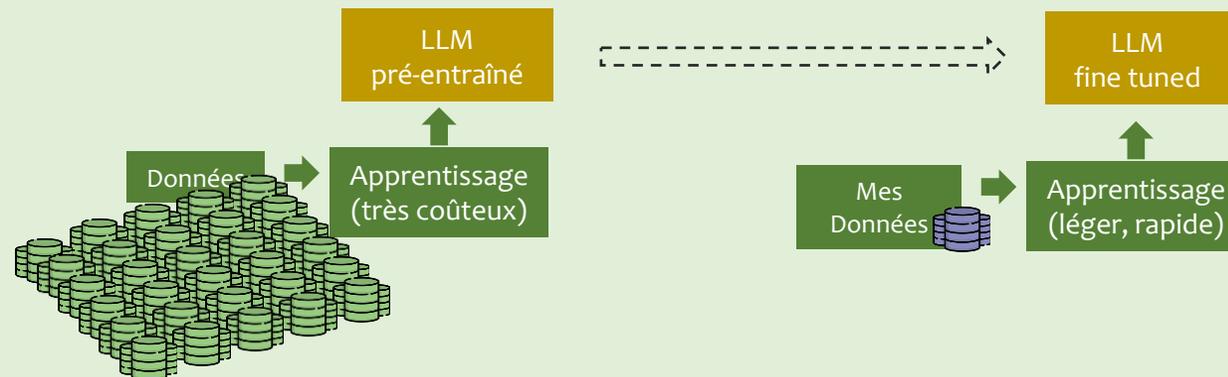


La technique « **Transformers** » transforme du texte en nombres en mettant en valeur les relations entre les mots



## GPT = Generative Pre-trained Transformer

Un LLM est **pré-entraîné** sur des données de texte massives. Il peut ensuite être **spécialisé** (« fine tuning ») pour des tâches plus spécifiques.

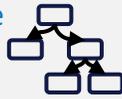


# Résumé des 3 séances du cours « Introduction à l'IA » !

## Séance 1 : les Fondamentaux

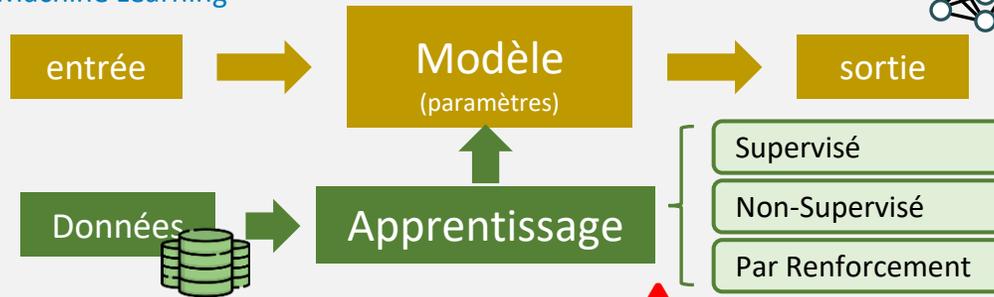


Approche Cognitiviste  
(« ancienne »)



Systèmes experts,  
calcul formel, etc.

Approche Connexionniste :  
Machine Learning

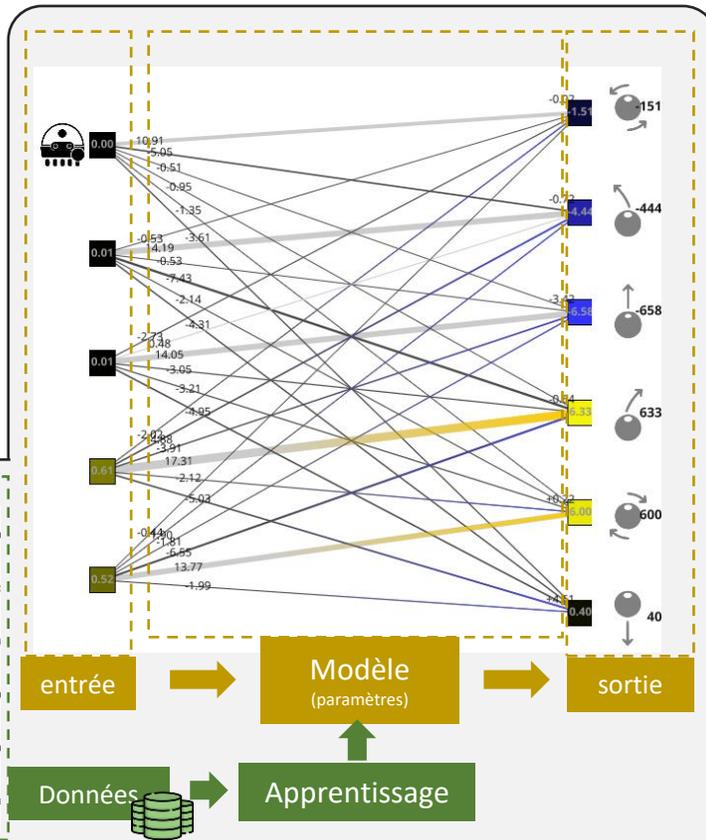


Quel sera le futur ?

Hybridations,  
Robotique,  
Autre ?



## Séance 2 : Entraînement d'une IA simple (Réseau de Neurones artificiels)



attention au biais

## Séance 3 : Entraînement d'une IA complexe (GPT + RAG)

