

#### Objectif du cours



- Le but de ce cours est l'introduction aux réseaux de neurones, nous allons aborder ensemble les principaux composants d'un réseau de neurone simple, ainsi que les différents types de réseaux suivant la problématique à résoudre.
- Le cours se décompose en une partie théorique, des lectures (Drive into deep learning), et Des TP sous *python* afin de mieux assimiler les différents concepts.
- Le cours se focalise essentiellement sur les aspects suivants :
  - ☐ Introduction à AI et ML
  - ☐ Introduction aux modèles réseaux de neurones

#### **Evaluation**



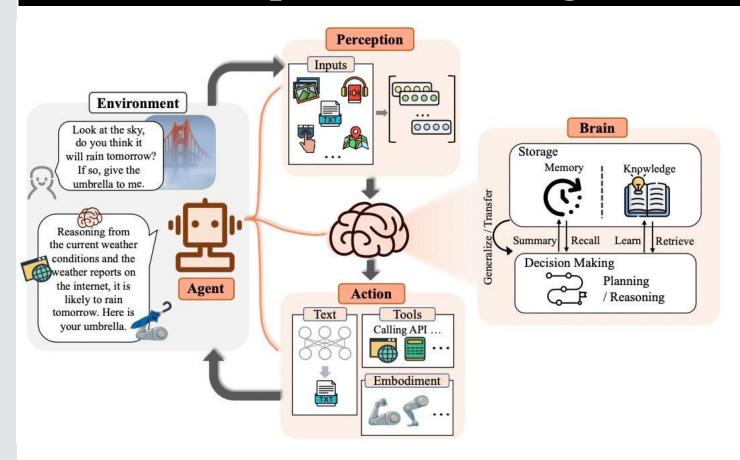
- 2 Q&A (50%)
- 2 TP noté (50%)





L'expérience de la chambre chinoise par John Searle



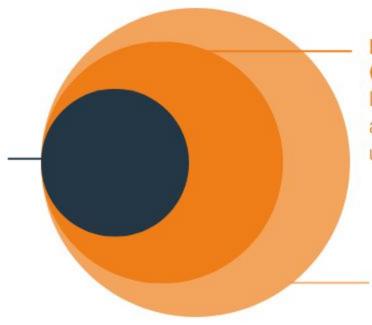


Créer un Agent qui peut avoir un comportement intelligente





Les systèmes apprennent par les réseaux neuronaux sans intervention de l'homme.



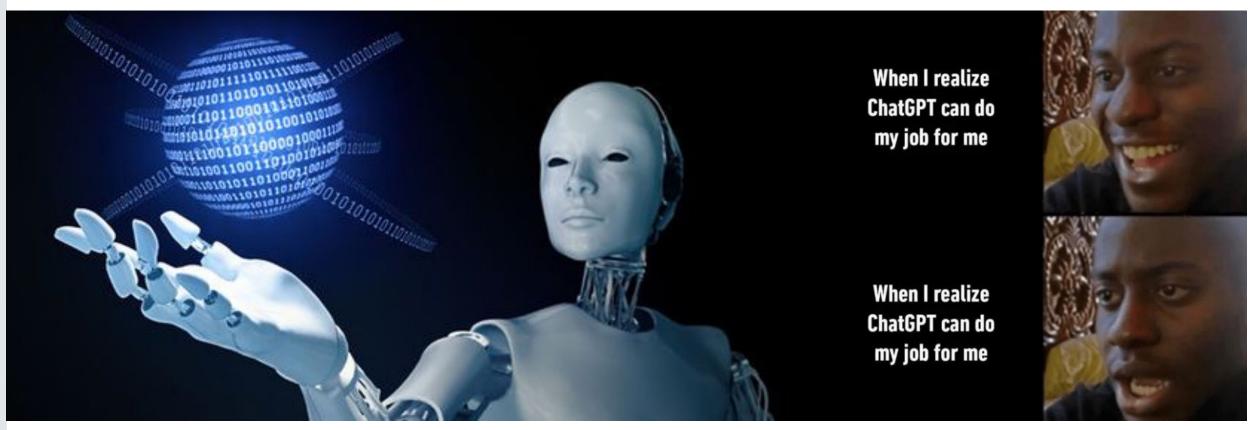
## Le machine learning (apprentissage automatique)

Les systèmes sont programmés pour apprendre à réaliser des tâches en étudiant un ensemble de vastes données

#### L'Intelligence Artificielle (IA) ou (AI) pour Artificial Intelligence

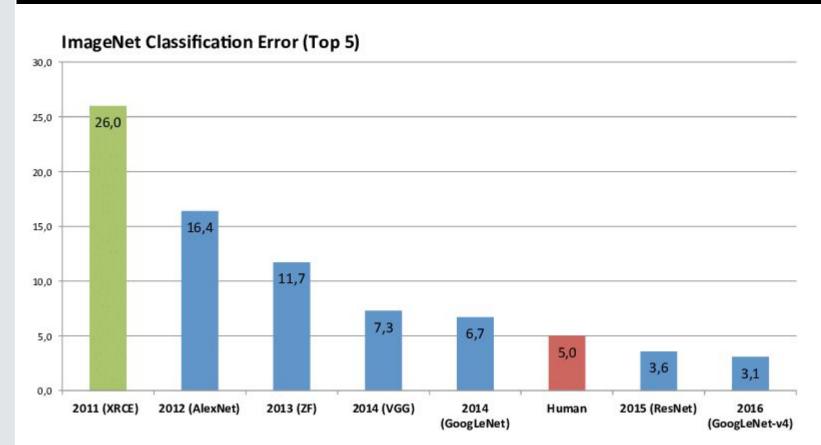
Les systèmes reproduisent le fonctionnement de l'intelligence humaine.





Brace yourself, AI is coming



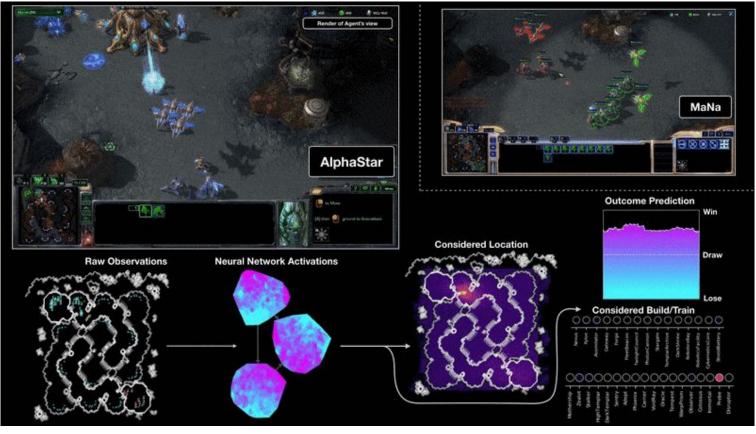


Dans certains domains, un modèle deep learning peut sur performé un humain...

Avant la naissance de ChatGPT







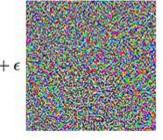


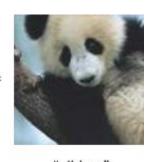


Rank	Model	EM	F1
	Human Performance Stanford University (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble)  Google Al Language https://arxiv.org/abs/1810.04805	87.433	93.160
2 Oct 05, 2018	BERT (single model)  Google Al Language  https://arxiv.org/abs/1810.04805	85.083	91.835
2 Sep 09, 2018	nlnet (ensemble) Microsoft Research Asia	85.356	91.202
2 Sep 26, 2018	<b>nlnet (ensemble)</b> Microsoft Research Asia	85.954	91.677
3 [ Jul 11, 2018 ]	<b>QANet (ensemble)</b> Google Brain & CMU	84.454	90.490
4 [ Jul 08, 2018 ]	<b>r-net (ensemble)</b> Microsoft Research Asia	84.003	90.147
5 Mar 19, 2018	<b>QANet (ensemble)</b> Google Brain & CMU	83.877	89.737





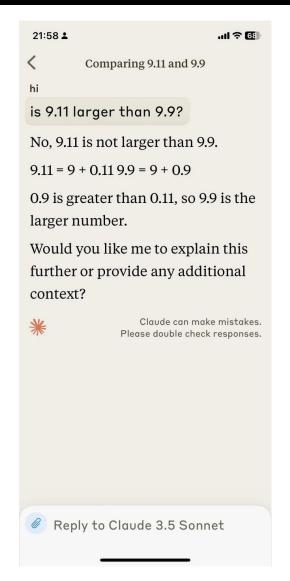




"panda" 57.7% confidence

"gibbon" 99.3% confidence

https://arxiv.org/pdf/1412.6572.pdf



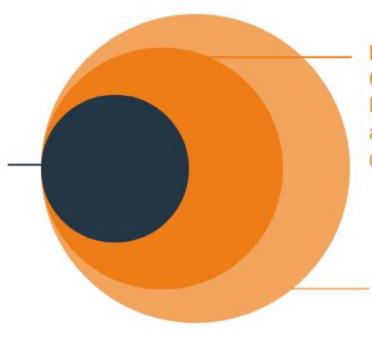
https://www.linkedin.com/pulse/do-ll ms-make-errors-way-troubleshoot-ll m-error-rates-the-next-tech-i70nc/





# Le Deep learning (apprentissage profond)

Les systèmes apprennent par les réseaux neuronaux sans intervention de l'homme.



#### Moyenne

## Le machine learning (apprentissage automatique)

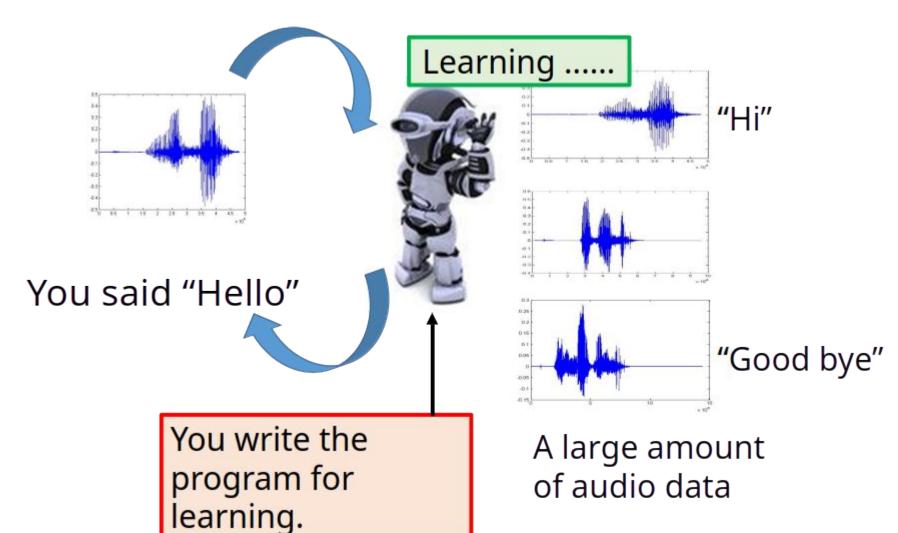
Les systèmes sont programmés pour apprendre à réaliser des tâches en étudiant un ensemble de vastes données

#### Objective

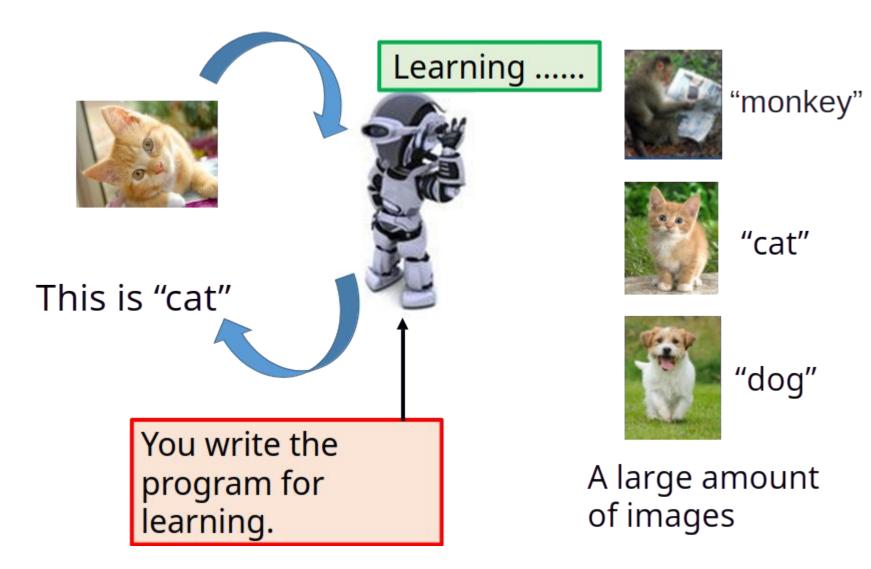
#### L'Intelligence Artificielle (IA) ou (AI) pour Artificial Intelligence

Les systèmes reproduisent le fonctionnement de l'intelligence humaine.











Reconnaissance vocal:

- Reconnaissance image:

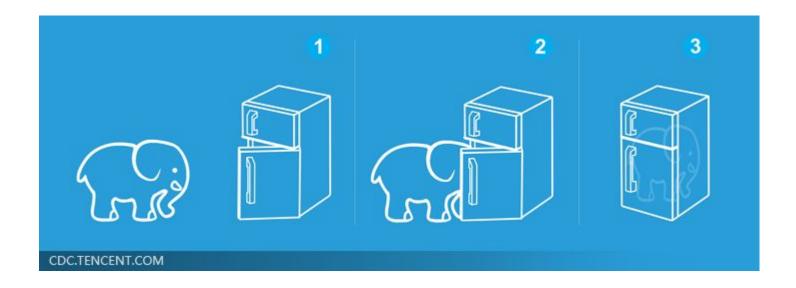
- Jouer le jeux GO:

Chatbot:
 f( "Bonjour, Ca va bien?" ) = "Oui, très bien."

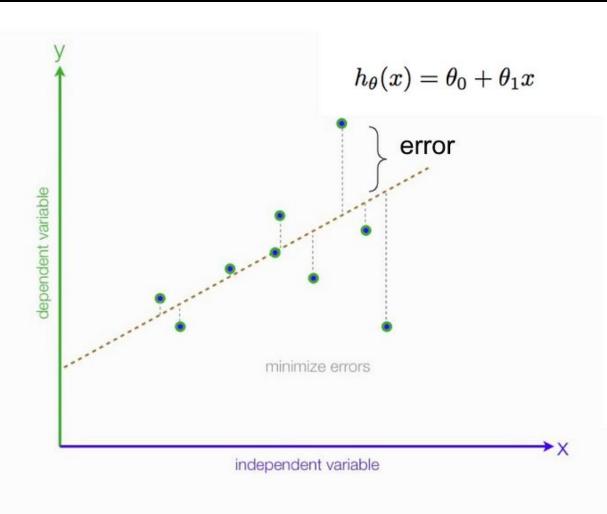
https://ml-playground.com/#



- Une modèle à apprendre
- Des données pour entraîner le modèle
- Une fonction de coût qui évalue l'apprentissage de ce modèle







Hypothesis:

 $h_{\theta}(x) = \theta_0 + \theta_1 x$ 

Parameters:

 $\theta_0, \theta_1$ 

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

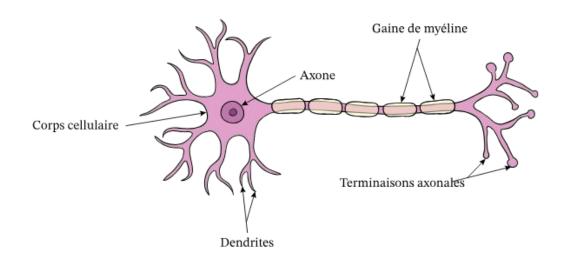
Goal:

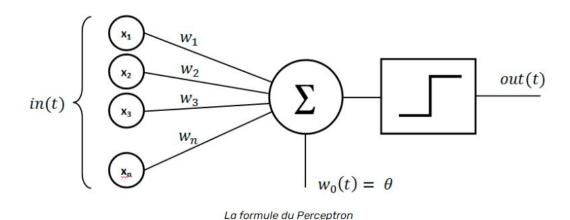
 $\displaystyle \mathop{minimize}_{ heta_0, heta_1} J( heta_0, heta_1)$ 

#### Introduction réseau de neurones



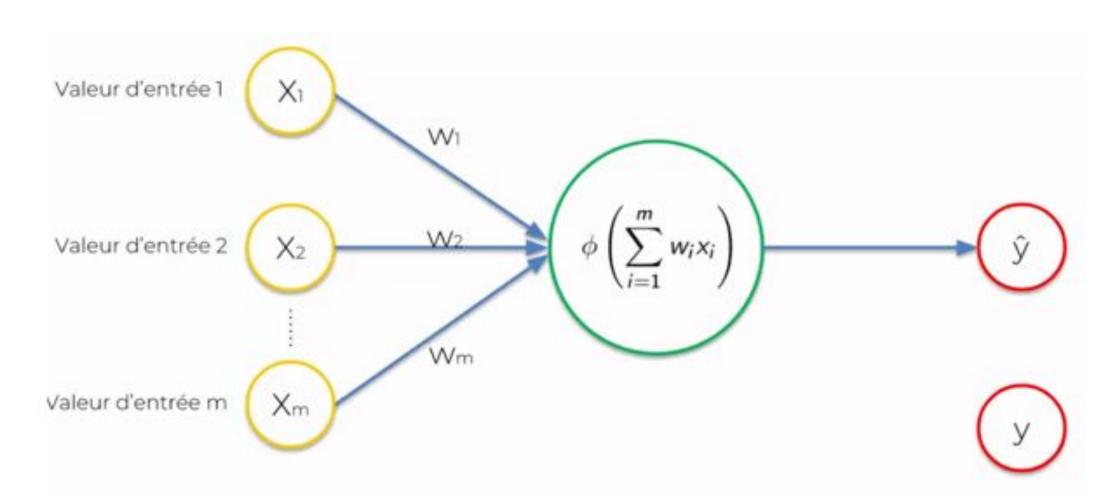
• Les réseaux de neurones (artificiels) sont des programmes informatiques fondés sur un modèle simplifié du cerveau.





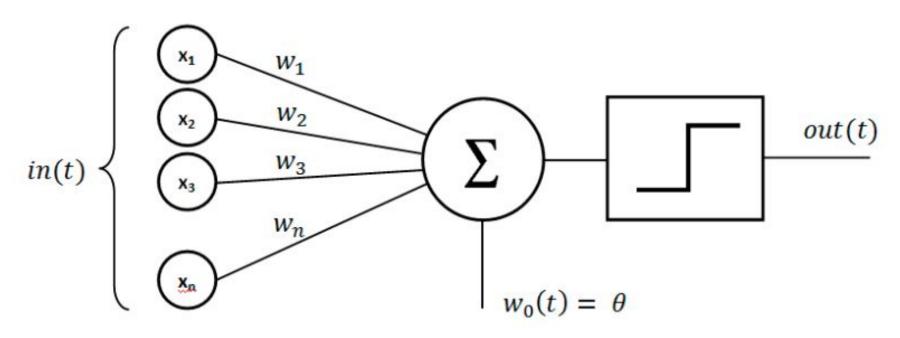
## Réseau de neurone simple





## Perceptron





La formule du Perceptron

# C'est quoi une fonction d'activation?



• La fonction d'activation sert **avant tout** à **modifier** de manière **non-linéaire** les données. Cette non-linéarité permet de **modifier spatialement leur représentation**.

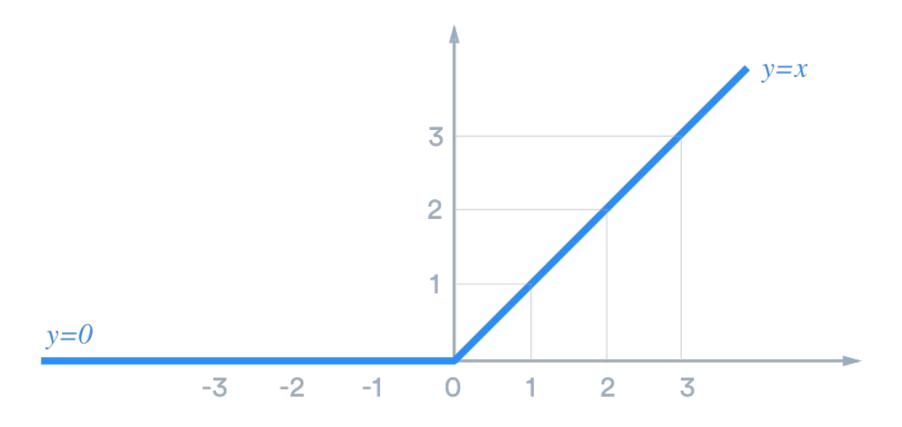
• Dit simplement, la fonction d'activation permet de changer notre manière de voir une donnée.

#### Linear Unit (ReLU)



La fonction Linear Unit (ReLU) est la fonction d'activation la plus simple et la plus utilisée.

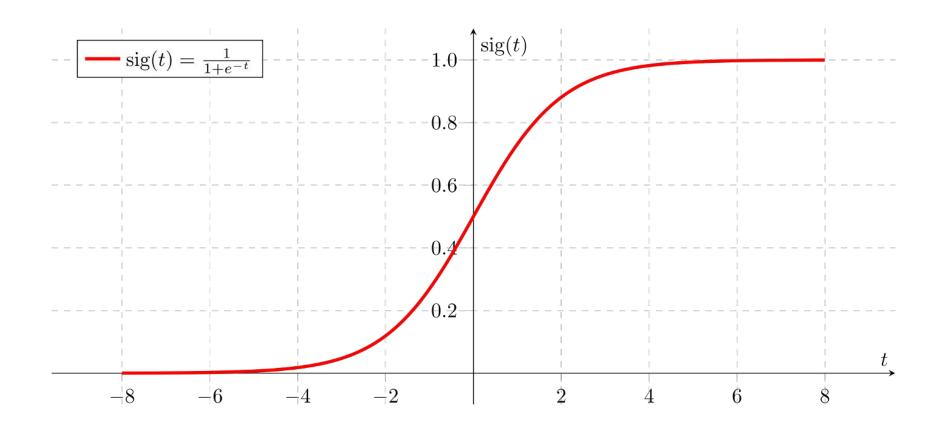
Elle donne x si x est supérieur à 0, 0 sinon. Autrement dit, c'est le maximum entre x et 0:



#### Sigmoïde



La fonction **Sigmoïde** donne une valeur entre 0 et 1, une probabilité. Elle est donc très utilisée pour les **classification binaire**, lorsqu'un modèle doit déterminer seulement deux labels.

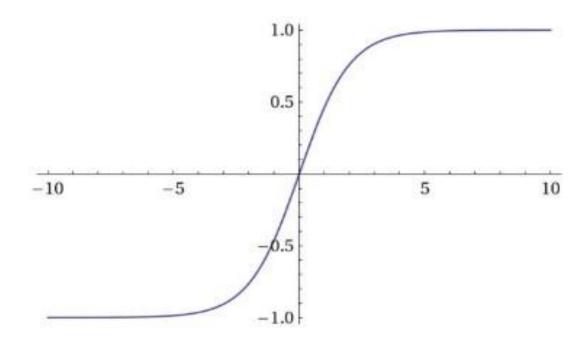


#### **Softmax**



La fonction **Softmax** permet elle de transformer un vecteur réel en **vecteur de probabilité**. On l'utilise souvent dans la couche finale d'un modèle de classification, notamment pour les **problèmes multiclasse**.

#### Softmax Activation Function

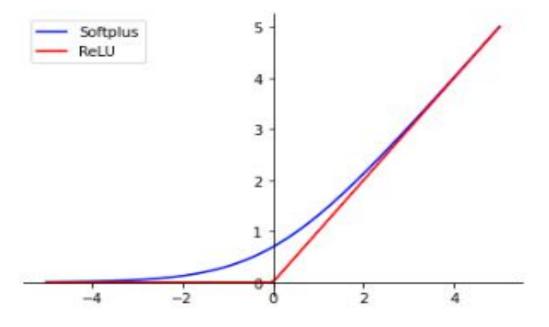


#### Softplus



La fonction **Softplus** est une **approximation 'lisse**' de la fonction **ReLU**. Cet aspect 'lisse' (ou soft) implique que la fonction est **différentiable**.

En fait, cette fonction est intéressante par sa dérivée. Quand on dérive **Softplus**, on obtient la **fonction logistique** f(x) = 1/(1+exp(-x)). On rappelle que la dérivée est utilisée lors de la **Back-propagation** pour mettre à jour les poids



Fonction Softplus

#### **Autres fonctions**



**Softsign**: La fonction **Softsign** est utile pour **normaliser nos données** car elle permet d'avoir un **résultat entre -1 et 1** et garde en mémoire le **signe des données (positif ou négatif)**. Autrement dit, les données sont **recentrées sur zéro** et **bornées par -1 et 1**.

Tanh: La fonction tanh est simplement la fonction de la tangente hyperbolique.

**ELU**: La fonction **Exponential Linear Unit** (**ELU**) est une **amélioration de ReLU** car elle permet d'avoir des **valeurs lisses** lorsque x < 0.

SELU: La fonction Scaled Exponential Linear Unit (SELU) est une optimisation de ELU.

## Quelle fonction à utiliser?



Type de problème	Fonction d'activation de la dernière couche
Classification Binaire	sigmoid
Classification multiclasse, à label unique	softmax
Classification multiclasse, multilabel	sigmoid
Régression	Linear

#### artificiels



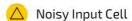
A mostly complete chart of

#### **Neural Networks**

©2016 Fjodor van Veen - asimovinstitute.org







Hidden Cell

Probablistic Hidden Cell

Backfed Input Cell

Spiking Hidden Cell

Output Cell

Match Input Output Cell

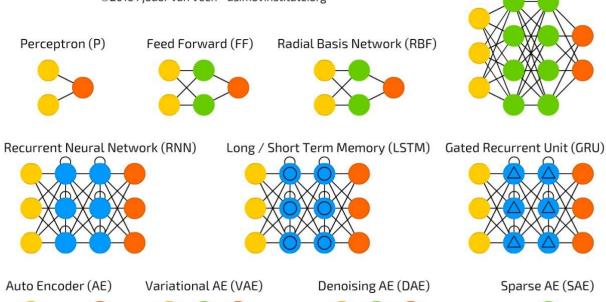
Recurrent Cell

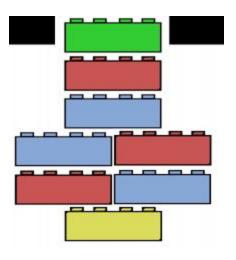
Memory Cell

Different Memory Cell

Kernel

O Convolution or Pool

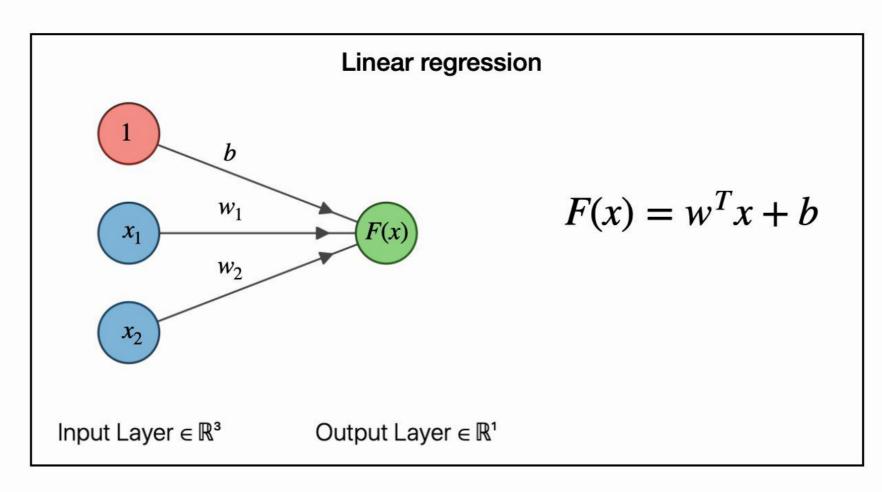




https://www.asimovinstitute.org/neural-network-zoo/

#### Lien avec le modèle de régression



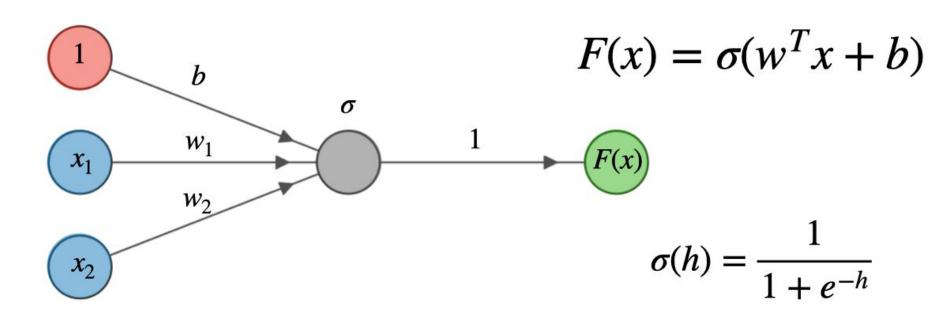


$$F(x) = w_1 \cdot x_1 + w_2 \cdot x_2 + 1 \cdot b$$

#### Perceptron



#### Logistic regression (generalized linear model)



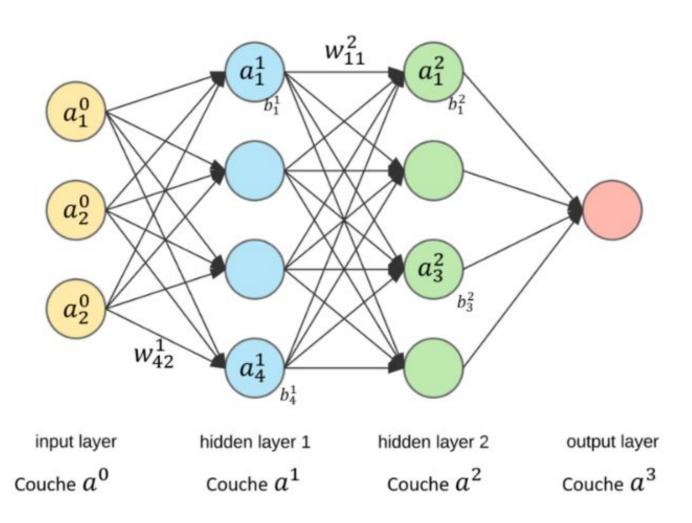
Input Layer ∈ R³

Hidden Layer  $\in \mathbb{R}^1$ 

Output Layer  $\in \mathbb{R}^1$ 

#### artificiels



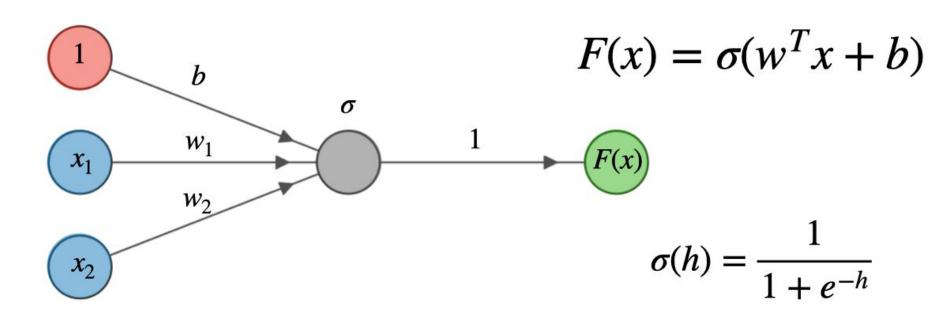


https://www.asimovinstitute.org/neural-network-zoo/

#### Perceptron



#### Logistic regression (generalized linear model)



Input Layer ∈ R³

Hidden Layer  $\in \mathbb{R}^1$ 

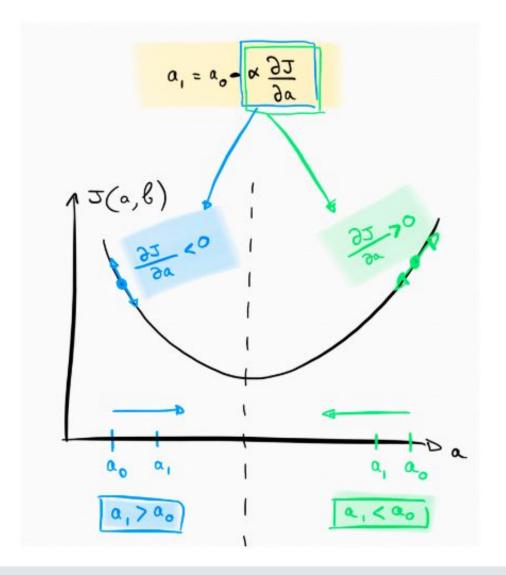
Output Layer  $\in \mathbb{R}^1$ 



#### Descente de gradient

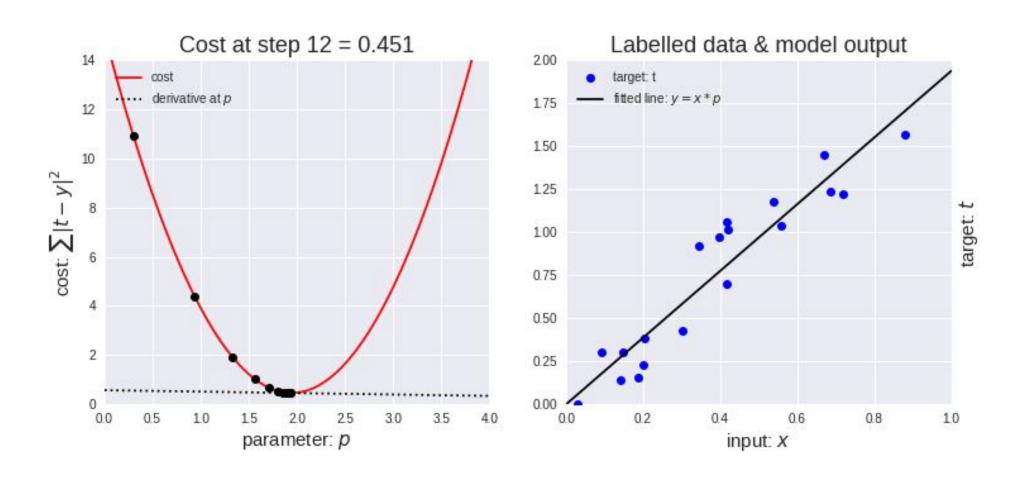


- Étape 1 : Calcul de la dérivée de la Fonction Coût
- Étape 2 : Mise à jour des paramètres du modèle
- Étape 3 : Répéter l'étape 1



#### Descente de gradient





# Les hyperparameters



ML Algorithms	Hyperparameters	Definition	<b>Defined Parameters</b>
SVM	Kernel type	the kernel function	RBF
(support vector	C	the penalty parameter	0.01-100
machines)	$\sigma$	the bandwidth parameter	0.01-100
Cubist	committees	the number of model trees	1-100
(regression tree)	neighbors	the number of nearest neighbors	0-9
XGBoost (extreme gradient boosting)	booster max_depth min_child_weight colsample_bytree subsample eta	the type of model the depth of tree the minimum sum of weights of all observations the number of variables supplied to a tree the number of samples supplied to a tree learning rate	gbtree 3–10 0–5 0.5–1 0.5–1 0.01–0.5
RF	Mtry	the number of input variables	1-30
(random forest)	Ntree	the number of trees	100-3000
ANN (artificial neural networks)	decay size	learning rate the number of neurons in the hidden layer	0.001-0.05 1-10
DNN	hidden	the number of hidden layers	2-10

## Les hyperparameters



## Classification of Neural Network Hyperparameters

Classification by Network Structure

Directly affect the structure of the neural network

- · Number of hidden layers
- Number of nodes in each layer
- · Type of activation function

Classification by Learning & Optimization

Directly control the training process of the neural network

- Type of optimizer
- Learning rate-α
- · Type of loss function
- Type of model evaluation metric
- Batch size
- Epochs
- Training steps (iterations) per epoch

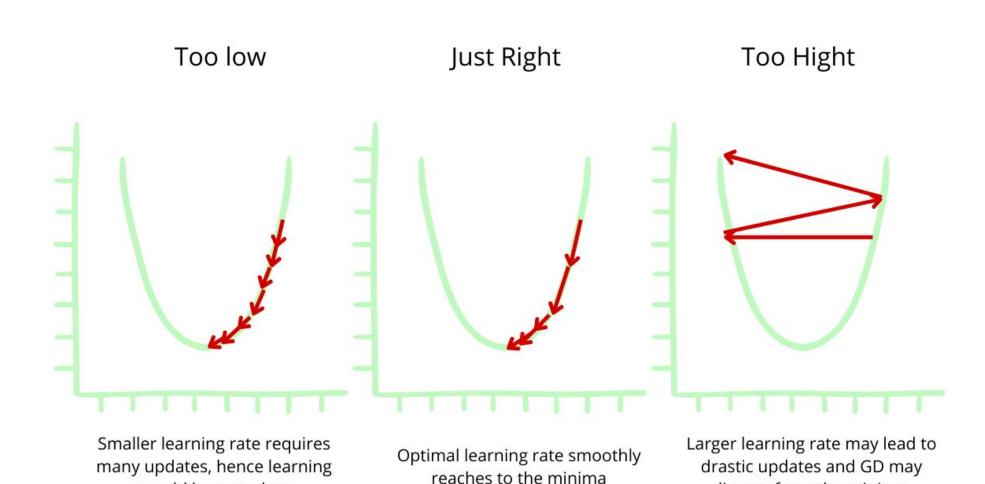
Classification by Regularization Effect

Directly control the overfitting in the neural network

- Lambda-λ in L1 and L2 regularization
- Dropout rate in dropout regularization

## Taux d'apprentissage (Learning rate)





would be very slow

enjoyalgorithms.com

diverge from the minima.

# Fonction d'objectif (Loss Functions)



- Il définit la qualité d'apprentissage
- Il faut définir la fonction en fonction du problème
- Ex: Régression

Mean squared error	$MSE = \frac{1}{n} \sum_{t=1}^{n} e_t^2$
Root mean squared error	$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^{n} e_t^2}$
Mean absolute error	$MAE = \frac{1}{n} \sum_{t=1}^{n}  e_t $
Mean absolute percentage error	$MAPE = \frac{100\%}{n} \sum_{t=1}^{n} \left  \frac{e_t}{y_t} \right $

# Fonction d'objectif (Loss Functions)



Problem Type	Output Type	Final Activation Function	Loss Function
Regression	Numerical value	Linear	Mean Squared Error (MSE)
Classification	Binary outcome	Sigmoid	Binary Cross Entropy
Classification	Single label, multiple classes	Softmax	Cross Entropy
Classification	Multiple labels, multiple classes	Sigmoid	Binary Cross Entropy

$$Entropy = -p_i log_b(p_i)$$

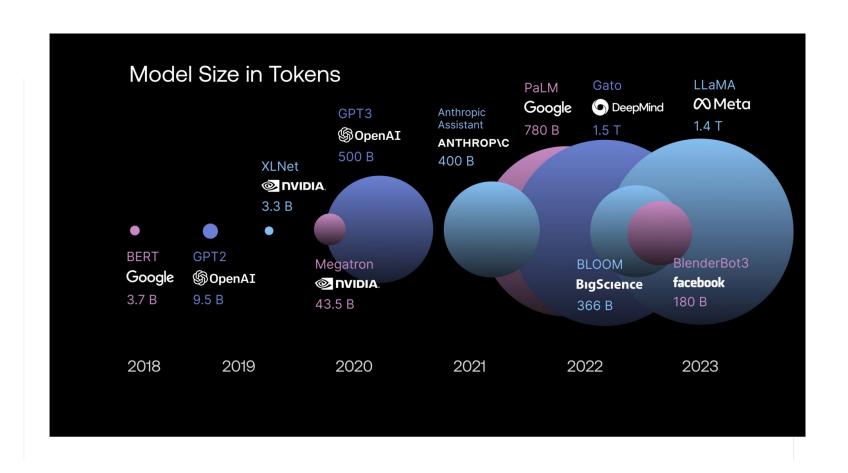
$$CrossEntropy = -\sum_{i=1}^{i=n} Y_i log_b(p_i)$$

# Fonction de coût (Loss Functions)

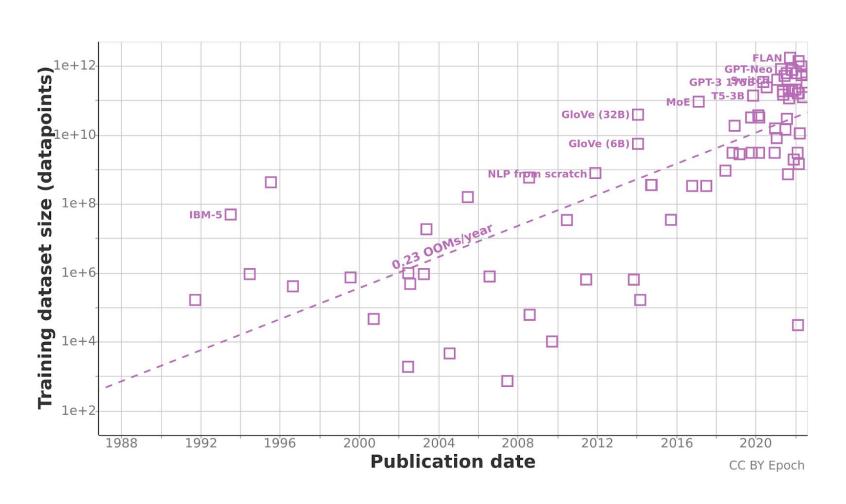


- **Binary Cross-Entropy**: Fonction la plus connue, définissant comme mesure de la différence entre deux probabilité de distribution pour une variable prise aléatoirement.
- Categorical Cross-Entropy: Utilisé pour de la classification multi-classes, mais en souhaitant un seul label en sortie.
- Weighted Cross-Entropy: C'est une variante de la Binary Cross-Entropy. Elle permet de rectifier la faiblesse de la BCE pour les datasets déséquilibrés, en ajoutant des poids selon des coefficients, pour les exemples positifs.
- **Balanced Cross-Entropy**: Elle est similaire à la Weighted Cross-Entropy. A la différence prêt, qu'elle pénalise en plus des exemples positives, les exemples négatifs











Batch Size = 100 Batch Size = 500 Batch Size = 1000

Iterations per Epoch = 1 Iterations per Epoch = 2 Iterations per Epoch = 10



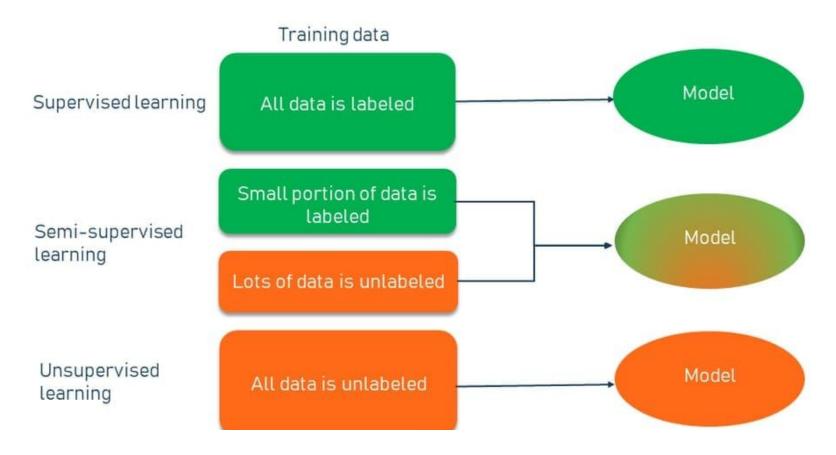
- L'epoch est un hyperparamètre qui définit le nombre de fois que l'algorithme d'apprentissage s'entrainera sur l'ensemble des données d'apprentissage.
- Une epoch signifie que chaque échantillon de l'ensemble de données d'apprentissage a eu l'occasion de mettre à jour les paramètres du modèle interne.
- Une epoch est composée d'un ou plusieurs lots ou batch size.
- On prend généralement une epoch de 25, 50, 100 ou 200. mais ce nombre peut varier grandement suivant la problématique ou la taille du dataset.

- Le batch size est un hyperparamètre qui définit le nombre d'échantillons à traiter avant de mettre à jour les paramètres du modèle interne.
- L'utilisation de batch permet de consommer moins de mémoire lors des phases d'entrainement.
- les tailles de lots populaires incluent 32, 64 et 128 échantillons

## Typologie d'apprentissage



# SUPERVISED LEARNING vs SEMI-SUPERVISED LEARNING vs UNSUPERVISED LEARNING



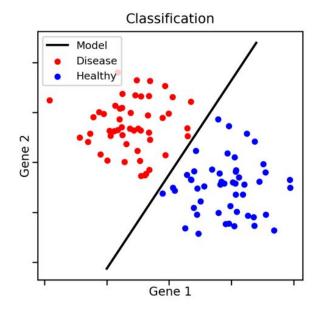


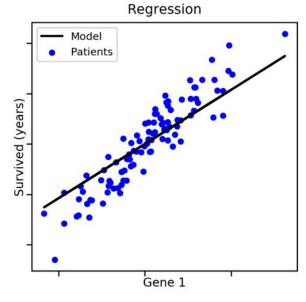
### Apprentissage supervisé vs non supervisé

L'apprentissage supervisé consiste en des variables d'entrée (x) et une variable de sortie (Y). Vous utilisez un algorithme pour apprendre la fonction de mapping de l'entrée à la sortie.

$$Y = f(X)$$

C'est ce qu'on appelle l'apprentissage supervisé, car le processus d'un algorithme tiré de l'ensemble de données de formation (training set) peut être considéré comme un enseignant supervisant le processus d'apprentissage.





### Apprentissage supervisé vs non supervisé

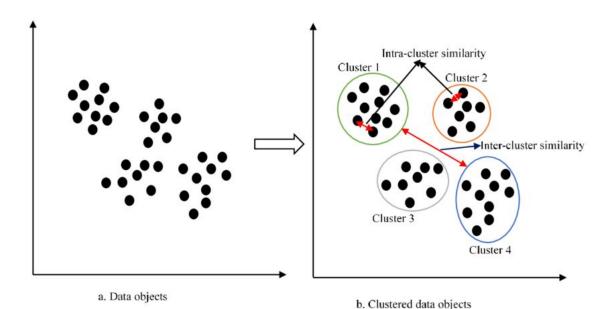
L'apprentissage non supervisé (Unsupervised Learning) consiste à ne disposer que de données d'entrée (X) et pas de variables de sortie correspondantes.

L'objectif de l'apprentissage non supervisé est de modéliser la structure ou la distribution sous-jacente dans les données afin d'en apprendre davantage sur les données.

On l'appelle apprentissage non supervisé car, contrairement à l'apprentissage supervisé ci-dessus, il n'y a pas de réponse correcte ni d'enseignant.

Les algorithmes sont laissés à leurs propres mécanismes pour découvrir et présenter la structure intéressante des données.





### Typologie de modèles



- Il existe deux types de modèles : les modèles discriminants et génératifs.
- Les modèles génératifs et discriminants sont deux approches différentes qui sont largement étudiées dans la tâche de classification. Ils suivent un itinéraire différent les uns des autres pour atteindre le résultat final.
- Les modèles discriminants sont très populaires et sont utilisés de manière plus comparative pour effectuer la tâche car ils donnent de meilleurs résultats lorsqu'ils sont fournis avec une bonne quantité de données.

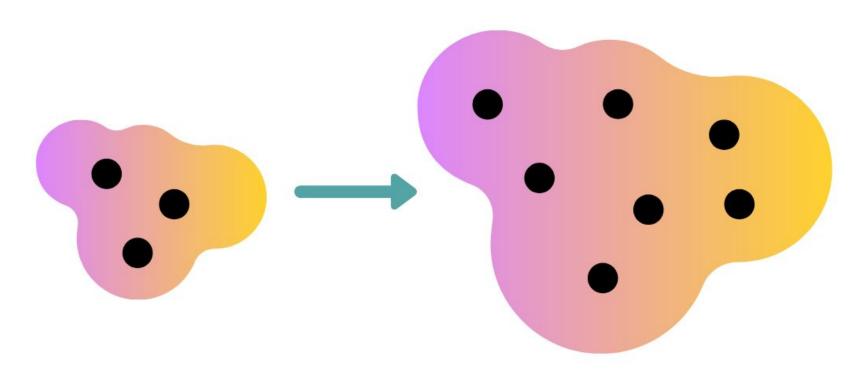
#### Modèle discriminant



- La tâche d'un modèle discriminant est simple, si on lui montre des données de différentes classes, il devrait être capable de les discriminer.
- Par exemple, si je montre au modèle un ensemble d'images de chien et de chat, il devrait pouvoir dire ce qu'est un chien et ce qu'est un chat en utilisant des caractéristiques discriminantes telles que la forme des yeux, la taille des oreilles, etc.

### **Generative Models**







**ChatGPT** 

**Generating new data using past data** 

### **Generative Models**





#### **Generative Models**



- Les modèles génératifs représentent une catégorie à part. Contrairement aux modèles discriminatifs qui associent les valeurs d'entrée à des étiquettes de sortie, ils ont pour objectif de générer de nouvelles données suivant des règles et des conditions spécifiques.
- Une explication simplifiée serait que les modèles génératifs sont capables d'apprendre les caractéristiques communes d'une collection de données et ensuite, de générer des données similaires





Deep learning framework















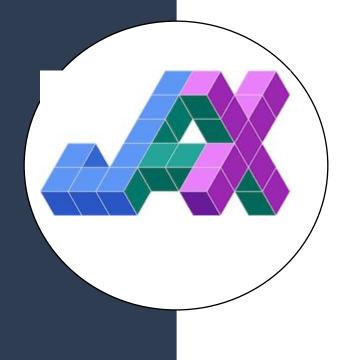
- PyTorch est une bibliothèque logicielle Python open source d'apprentissage machine qui s'appuie sur Torch développée par Meta
- PyTorch permet d'effectuer des calculs tensoriels nécessaires notamment pour du deep learning. Ces calculs sont optimisés et effectués soit par CPU ou GPU





- TensorFlow est un outil open source d'apprentissage automatique développé par Google.
- TensorFlow peut être lancé sur plusieurs CPU et GPU (avec des extensions optionnelles telles que CUDA ou SYCL





- Jax est un outil open source d'apprentissage automatique développé par Google.
- Il est conçu pour suivre au plus près la structure et le flux de travail de NumPy et fonctionne avec divers frameworks existants tels que TensorFlow et PyTorch.

Ne confondre pas avec ce Jax

### Que choisir?



- TensorFlow et PyTorch offrent tous deux des abstractions utiles qui facilitent le développement de modèles en réduisant le code passe-partout
- PyTorch a une approche plus "pythonique" et est orienté objet, tandis que TensorFlow offre une variété d'options.
- TensorFlow offre une meilleure visualisation, ce qui permet aux développeurs de mieux déboguer
- Dans le domaine du parallélisme des données, PyTorch obtient des performances optimales en s'appuyant sur un support natif pour l'exécution asynchrone via Python.



## PyTorch vs TF



