

# Formation aux Réseaux de neurones convolutifs : les CNN



Stéphane Jamin-Normand & Thomas Wentz

# Plan

L'intelligence artificielle et  
l'apprentissage automatique

Les réseaux de neurones

Dissection d'un CNN

Application en vision par ordinateur

# Plan

## L'intelligence artificielle et l'apprentissage automatique

- Imiter l'intelligence / Modéliser le complexe
- Des succès importants
- Les grandes familles d'applications

## Les réseaux de neurones

## Dissection d'un CNN

## Application en vision par ordinateur

# Plan

L'intelligence artificielle et l'apprentissage automatique

Les réseaux de neurones

- Les réseaux de neurones simples
- Les limitations en traitement d'image
- Les réseaux de neurons convolutifs (CNN)

Dissection d'un CNN

Application en vision par ordinateur

# Plan

L'intelligence artificielle et  
l'apprentissage automatique

Les réseaux de neurones

Dissection d'un CNN

- Les briques élémentaires
- Architectures typiques
- Entraînement d'un modèle

Application en vision par ordinateur

# Plan

L'intelligence artificielle et  
l'apprentissage automatique

Les réseaux de neurones

Dissection d'un CNN

Application en vision par ordinateur

- Reconnaissance d'images
- Détection d'objets
- Segmentation sémantique
- Autres

# Plan

## L'intelligence artificielle et l'apprentissage automatique

- Imiter l'intelligence / Modéliser le complexe
- Des succès importants
- Les grandes familles d'applications

## Les réseaux de neurones

## Dissection d'un CNN

## Application en vision par ordinateur

# Définition : qu'est ce que l'intelligence

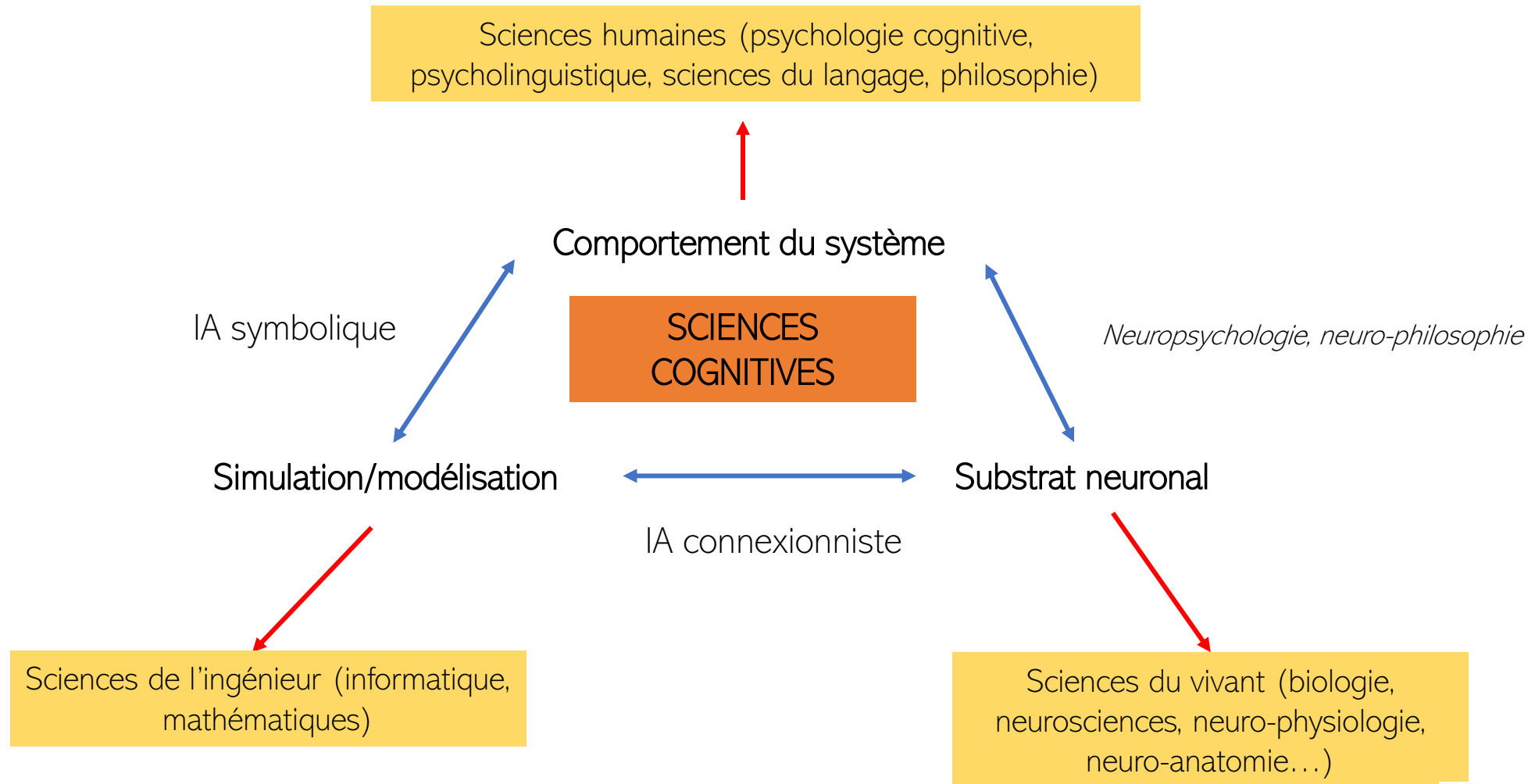
1 : Faculté de comprendre, de découvrir des relations (de causalité, d'identité, etc.) entre les faits et les choses.

2 : Aptitude à comprendre facilement, à agir avec discernement (Notion de « niveau d'intelligence », de comparaison par rapport à un système semblable)

3 : Capacité ou fait de comprendre une chose particulière. Exemple : avoir l'intelligence des affaires. (Notion de spécialisation.)



# Comment approcher le raisonnement



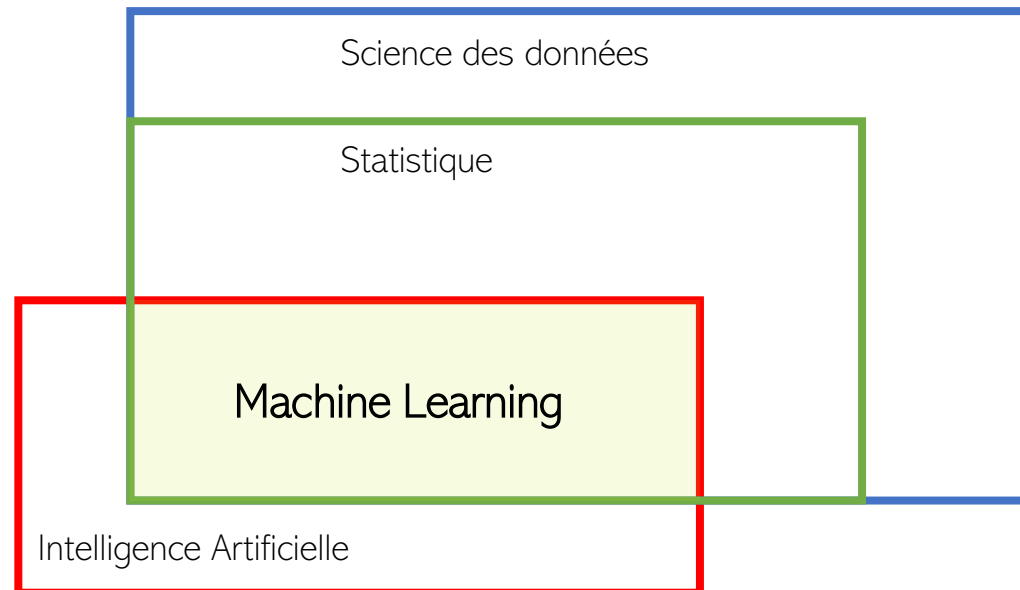
# Tentative de définition : qu'est ce que l'Intelligence Artificielle ?

- Une définition possible : l'Intelligence Artificielle est un domaine de l'informatique dont le but est de recréer un équivalent technologique à l'intelligence humaine. L'IA n'est pas une technologie à part entière mais un ensemble de technologies et d'outils.
- Discipline scientifique inventée en 1955 par deux mathématiciens, John MacCarthy et Marvin Lee Minsky

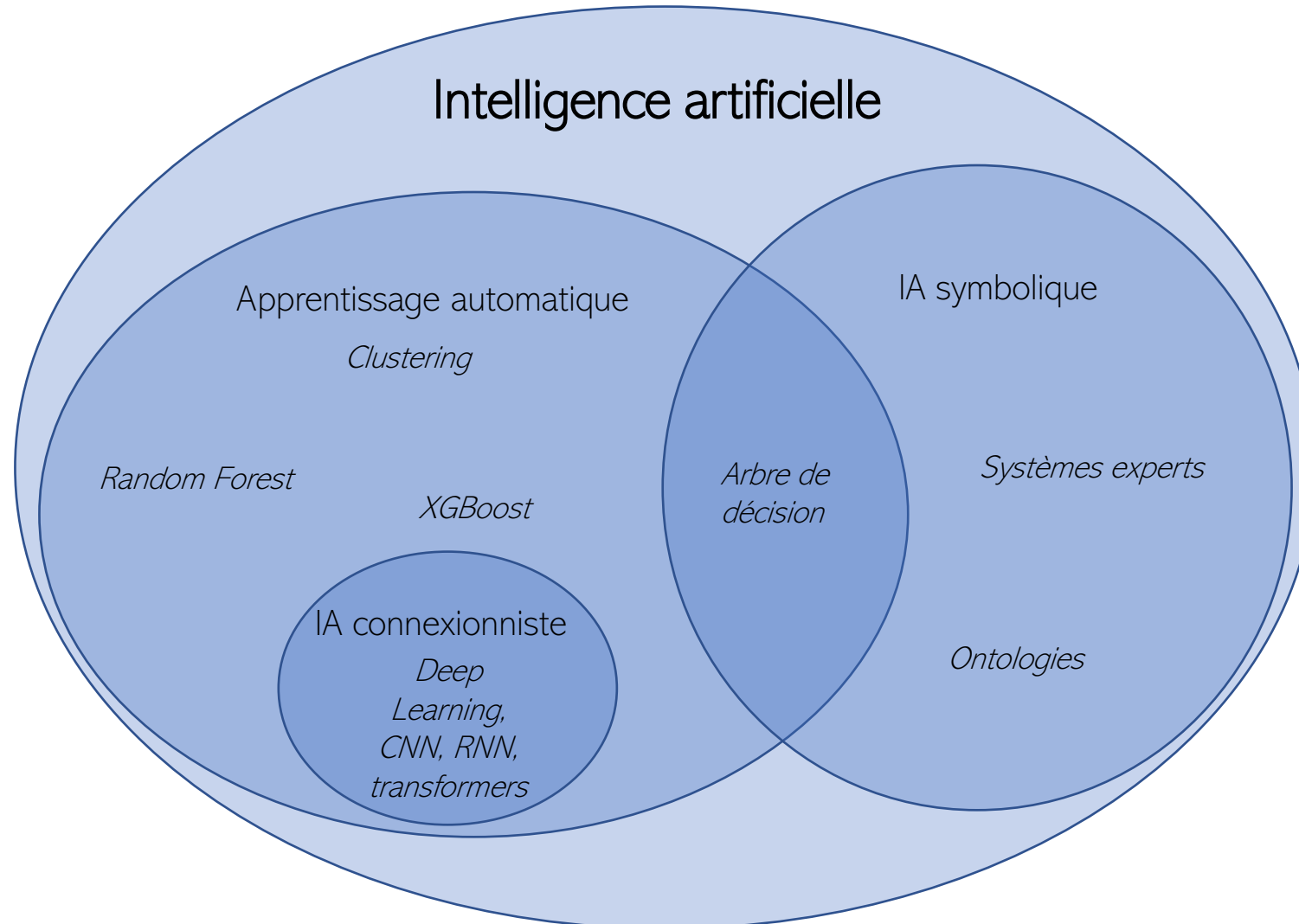
*« L'IA est la science de programmer les ordinateurs pour qu'ils réalisent des tâches qui nécessitent de l'intelligence lorsqu'elles sont réalisées par des êtres humains »* Marvin Lee Minsky

# Définition : qu'est ce que l'apprentissage automatique (Machine Learning)

- Le Machine Learning est une branche de l'IA qui concerne le développement d'algorithmes permettant d'accomplir des tâches complexes sans avoir été explicitement programmé dans ce but.
- Elle consiste à laisser des algorithmes découvrir des patterns, à savoir des motifs récurrents, dans les ensembles de données.



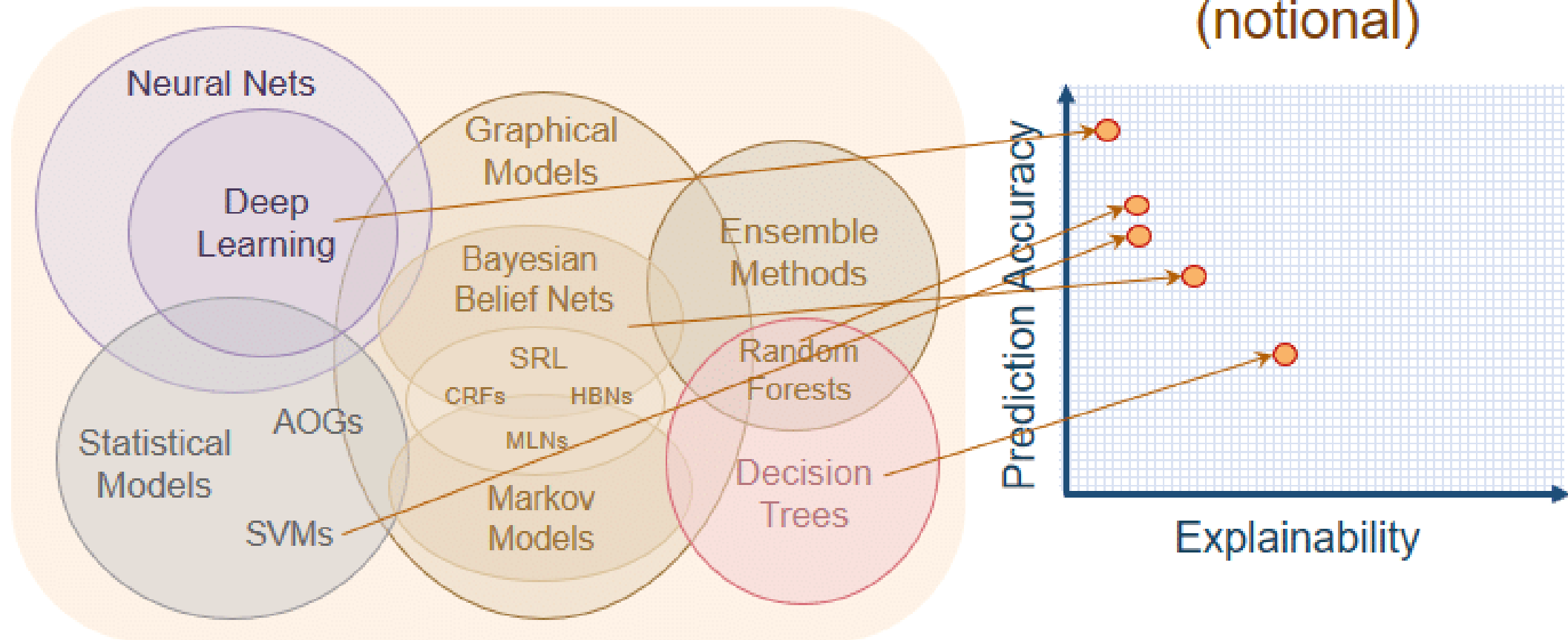
# Cartographie de l'Intelligence Artificielle



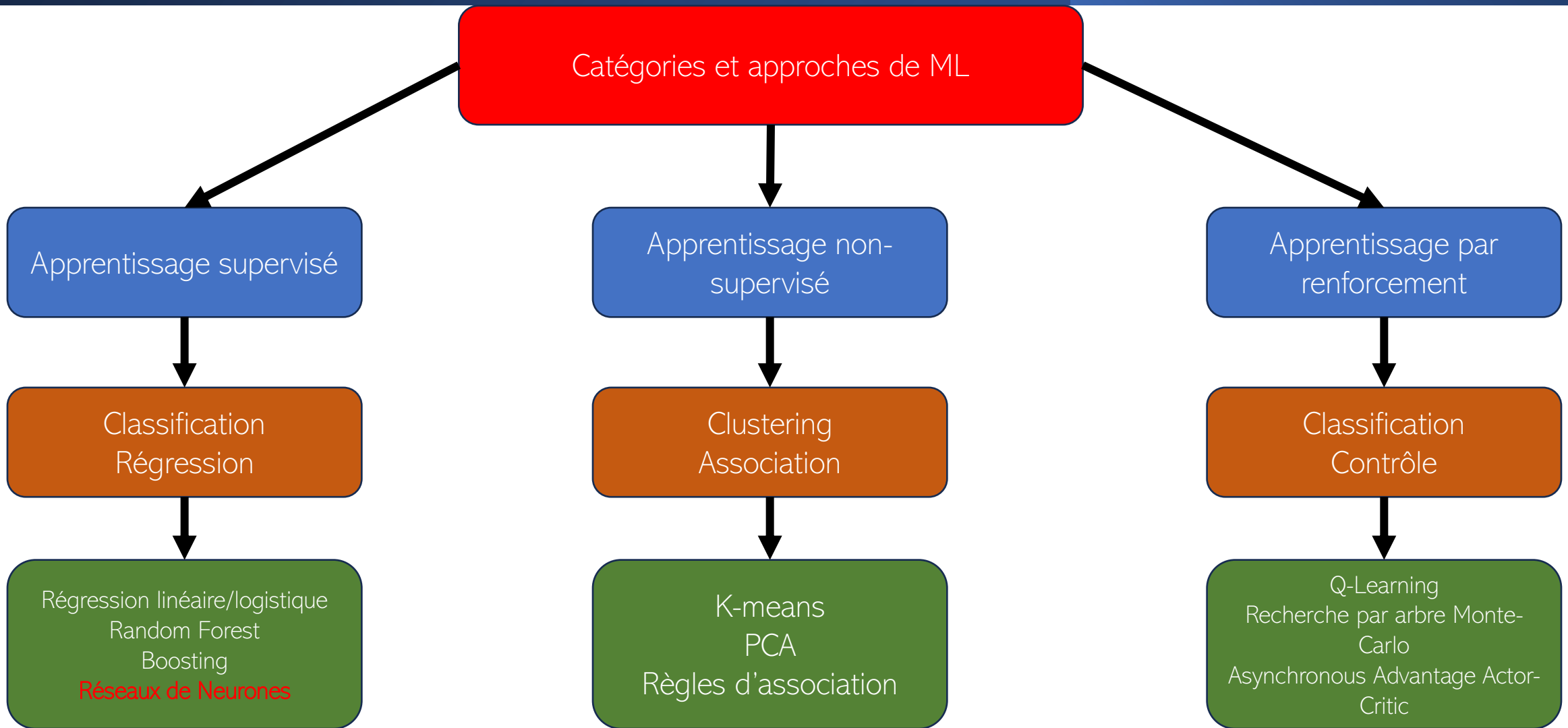
# Cartographie de l'Intelligence Artificielle

## Learning Techniques (today)

## Explainability (notional)



# Les catégories et approches de Machine Learning



# Plan

## L'intelligence artificielle et l'apprentissage automatique

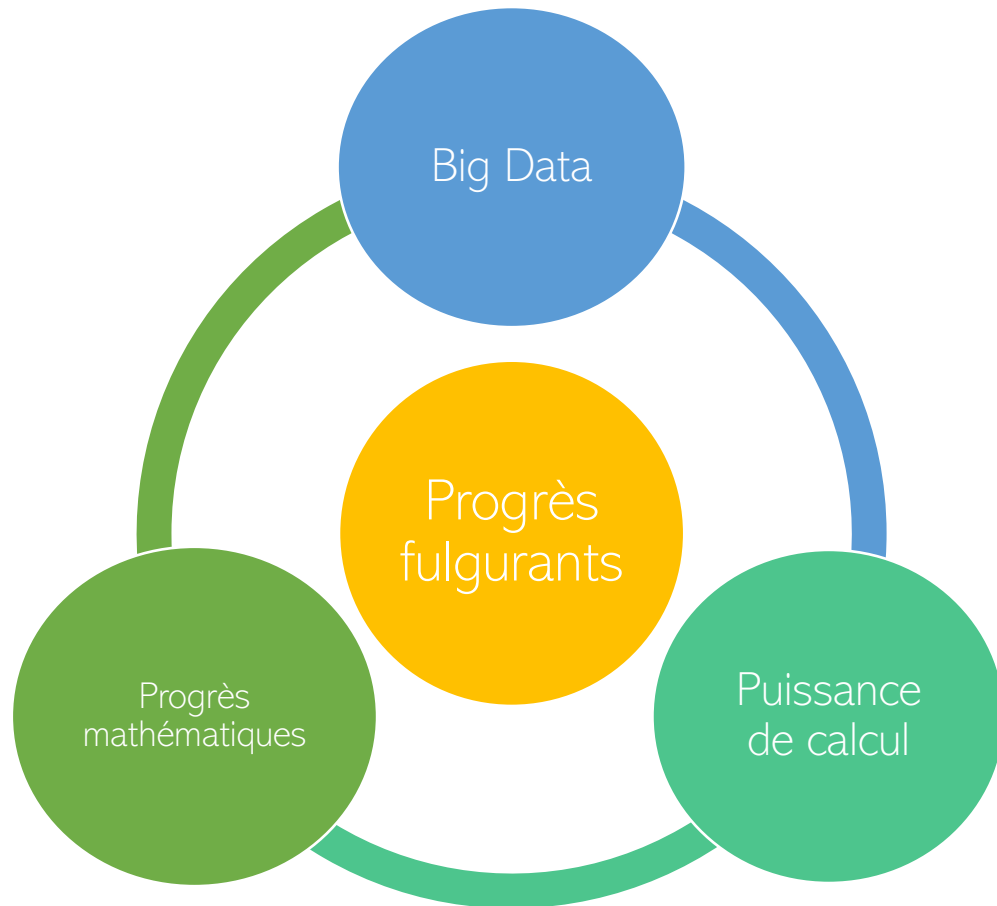
- Imiter l'intelligence / Modéliser le complexe
- Des succès importants
- Les grandes familles d'applications

## Les réseaux de neurones

## Dissection d'un CNN

## Application en vision par ordinateur

# Pourquoi maintenant ?



Rien que dans le domaine de la santé :

1. Assister les praticiens dans leur art
2. Augmenter les fonctions des praticiens
3. Accélérer le développement des traitements médicaux
4. Améliorer la prise en charge des maladies mentales
5. Anticiper des épidémies à un stade très précoce



# Pourquoi maintenant ?

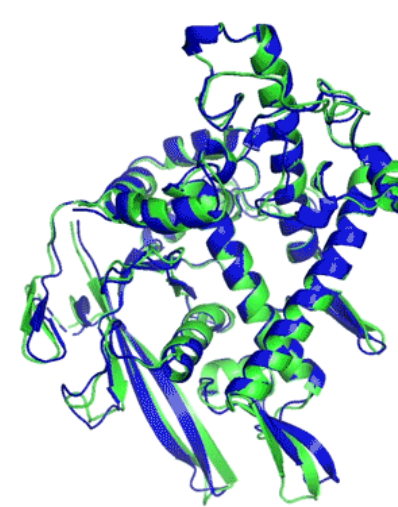
## Highly accurate protein structure prediction with AlphaFold

[John Jumper](#) , [Richard Evans](#), [Alexander Pritzel](#), [Tim Green](#), [Michael Figurnov](#), [Olaf Ronneberger](#), [Kathryn Tunyasuvunakool](#), [Russ Bates](#), [Augustin Žídek](#), [Anna Potapenko](#), [Alex Bridgland](#), [Clemens Meyer](#), [Simon A. A. Kohl](#), [Andrew J. Ballard](#), [Andrew Cowie](#), [Bernardino Romera-Paredes](#), [Stanislav Nikolov](#), [Rishub Jain](#), [Jonas Adler](#), [Trevor Back](#), [Stig Petersen](#), [David Reiman](#), [Ellen Clancy](#), [Michal Zielinski](#), ... [Demis Hassabis](#) 

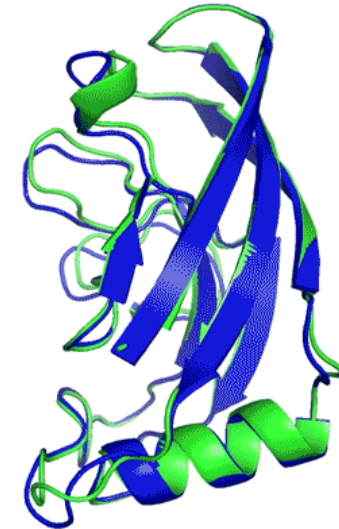
[+ Show authors](#)

[Nature](#) **596**, 583–589 (2021) | [Cite this article](#)

- ✓ Gain de temps phénoménal
- ✓ Perspective pour le développement de médicaments
- ✓ Nouveaux développements autour des protéines?



**T1037 / 6vr4**  
90.7 GDT  
(RNA polymerase domain)



**T1049 / 6y4f**  
93.3 GDT  
(adhesin tip)

● Experimental result  
● Computational prediction

# Plan

## L'intelligence artificielle et l'apprentissage automatique

- Imiter l'intelligence / Modéliser le complexe
- Des succès importants
- Les grandes familles d'applications

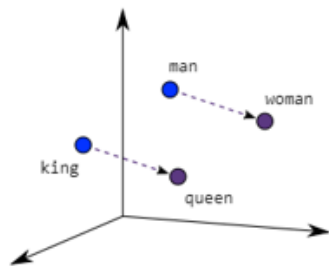
## Les réseaux de neurones

## Dissection d'un CNN

## Application en vision par ordinateur

# Le traitement du langage naturel (NLP)

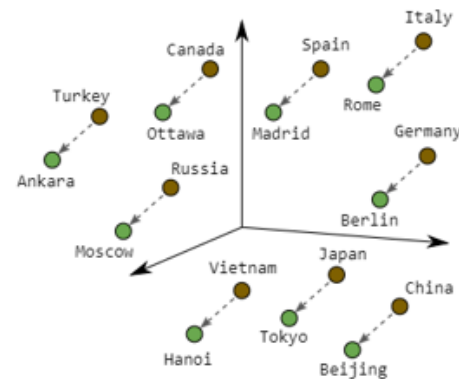
- **Domaine historique de l'IA**, mais très peu performant jusqu'en 2017, le langage humain étant trop « variable », sujet aux ambiguïtés
- **Explosion** depuis :
  - 2013 et le premier vrai plongement sémantique : Google, Word2Vec
  - 2017 et l'architecture Transformers



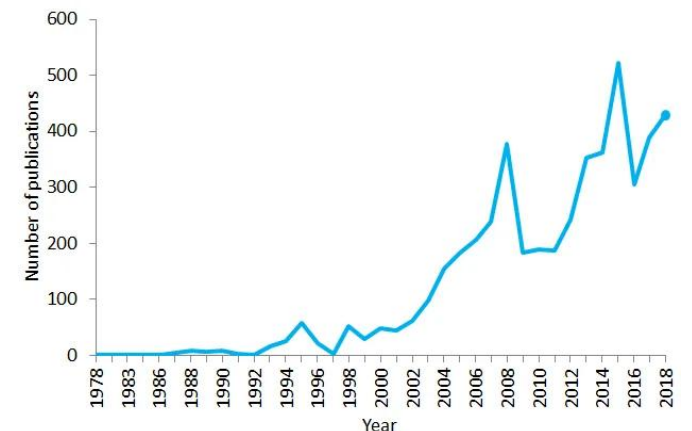
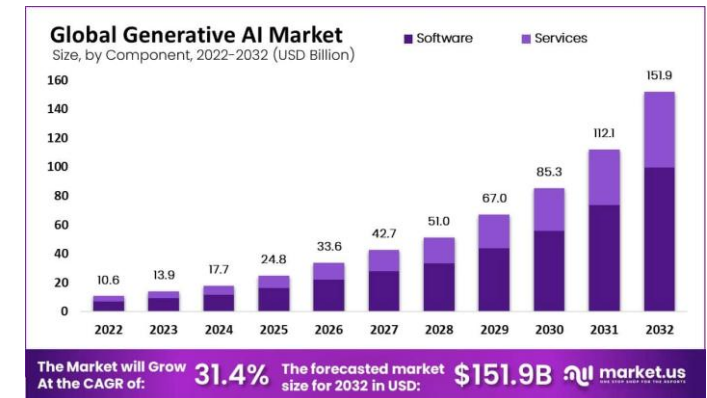
Male-Female



Verb Tense

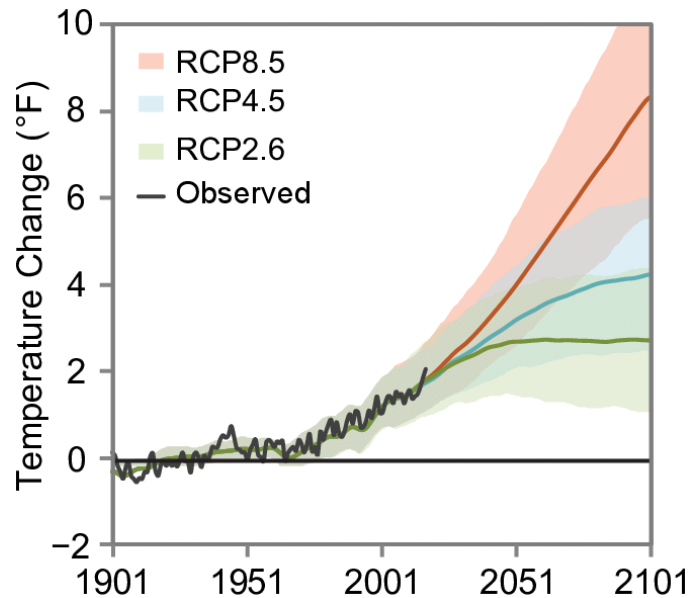


Country-Capital



# Les séries temporelles

- Se réfère au traitement de **données historicisées**.
- Les approches sont particulières car elles doivent être capables de capturer les relations et influences entre les données, ce qui induit l'introduction d'une mémoire



# Les données tabulaires

- Se réfère au traitement **de données sous formes de tableaux**, avec comme données des valeurs, numériques ou catégorielles, qui sont des caractéristiques (« features »)
- C'est le domaine dans lequel les approches de « Deep Learning » ne sont pas supérieures aux techniques classiques de Machine Learning



Information Fusion  
Volume 81, May 2022, Pages 84-90



Full length article

## Tabular data: Deep learning is not all you need

Ravid Shwartz-Ziv , Amitai Armon

[Show more](#)

[+ Add to Mendeley](#) [Share](#) [Cite](#)

<https://doi.org/10.1016/j.inffus.2021.11.011>

[Get rights and content](#)

### Highlights

- Deep neural networks are not good for all type of tabular data.
- XGboost outperforms deep models on tabular data.
- It harder to optimize deep neural networks compared to XGBoost.

# Le traitement d'images

- Se réfère au traitement de données structurées, en 2 ou 3 dimensions, avec potentiellement plusieurs canaux
- C'est ce domaine qui a le plus profité de l'arrivée des approches profondes, le traitement d'images étant hautement adapté au concept de la **convolution**





# Plan

L'intelligence artificielle et  
l'apprentissage automatique

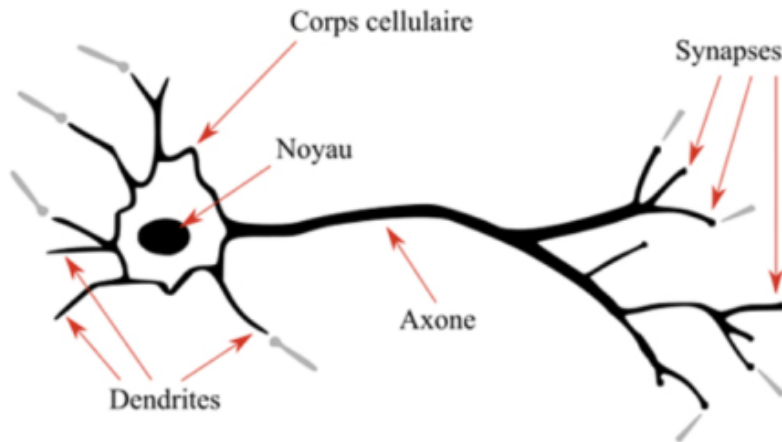
Les réseaux de neurones

- Les réseaux de neurones simples
- Les limitations en traitement d'image
- Les réseaux de neurons convolutifs (CNN)

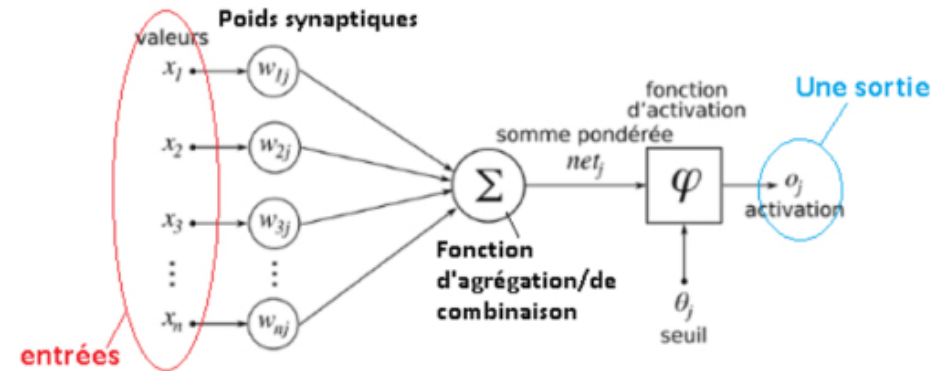
Dissection d'un CNN

Application en vision par ordinateur

# Les réseaux de neurones



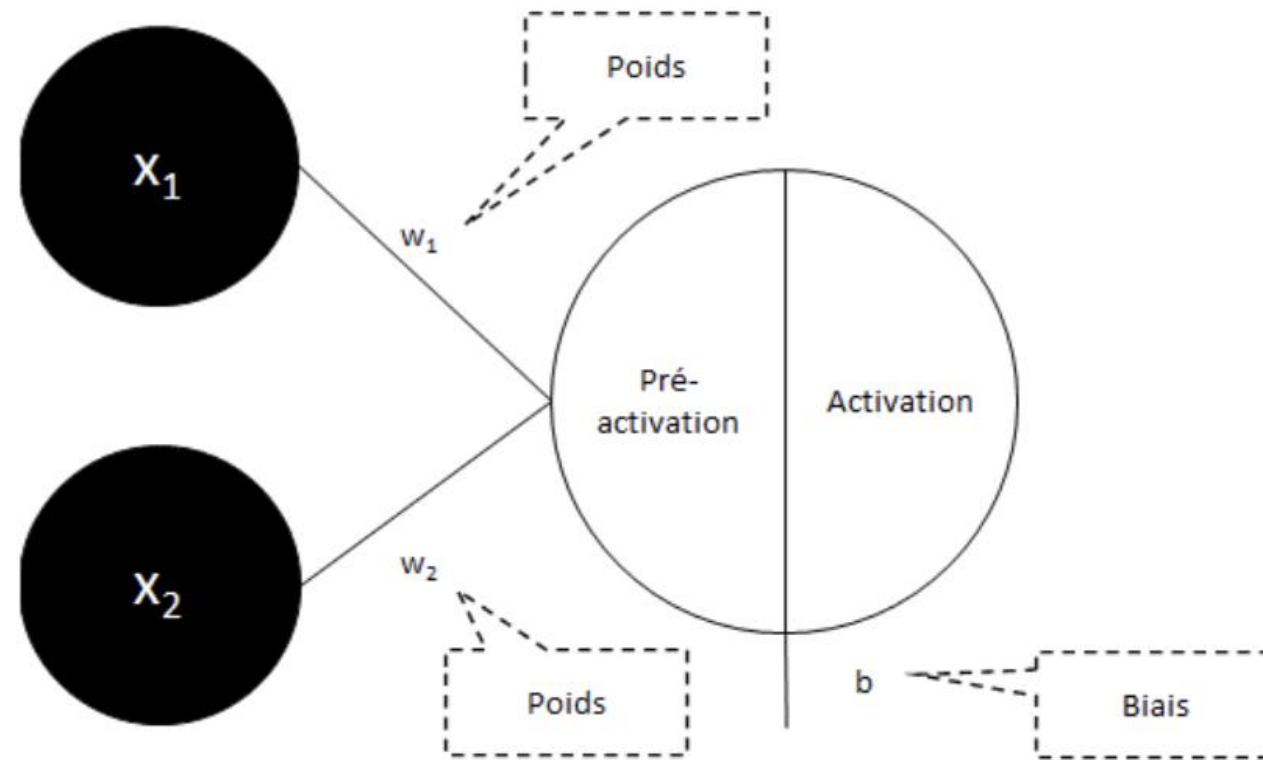
NEURONE BIOLOGIQUE



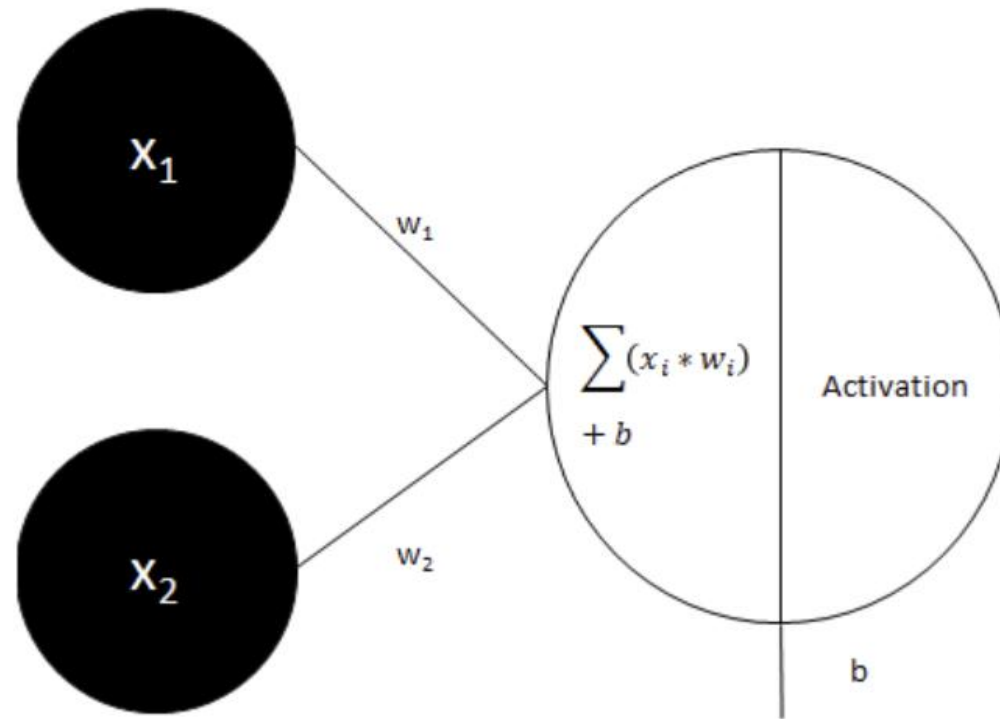
NEURONE ARTIFICIEL



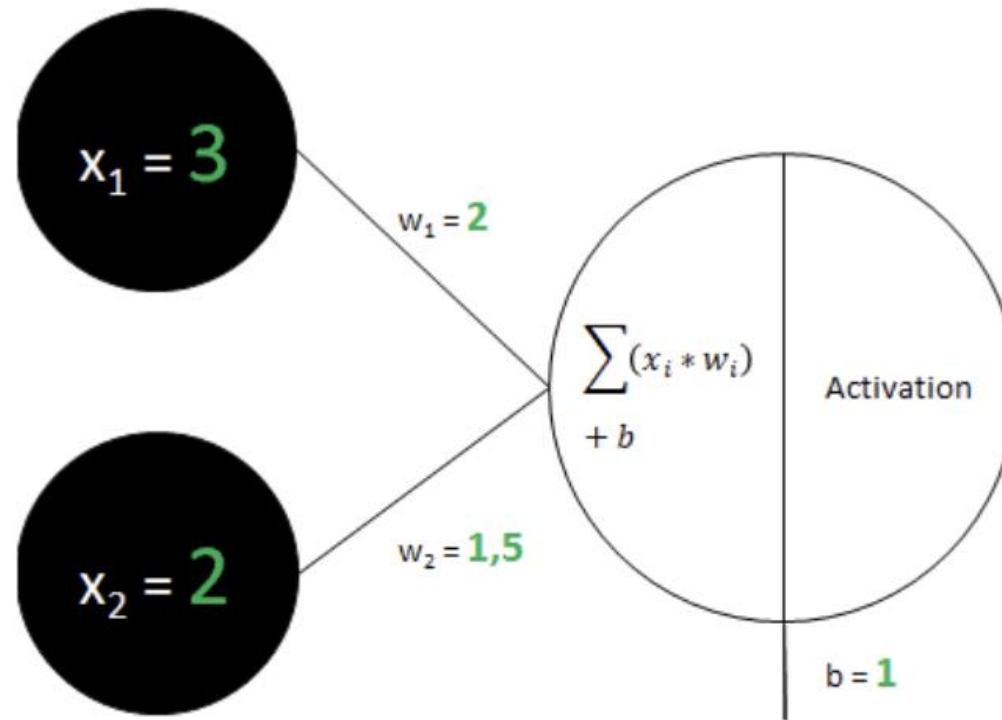
# Les réseaux de neurones



# Les réseaux de neurones



# Les réseaux de neurones



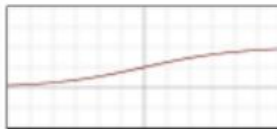
# Les réseaux de neurones

Fonction "Marche/Heaviside"



$$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$$

Fonction Sigmoide



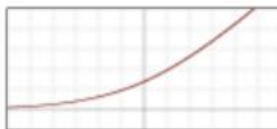
$$f(x) = \frac{1}{1 + e^{-x}}$$

Fonction "Unité de rectification linéaire" (ReLU)

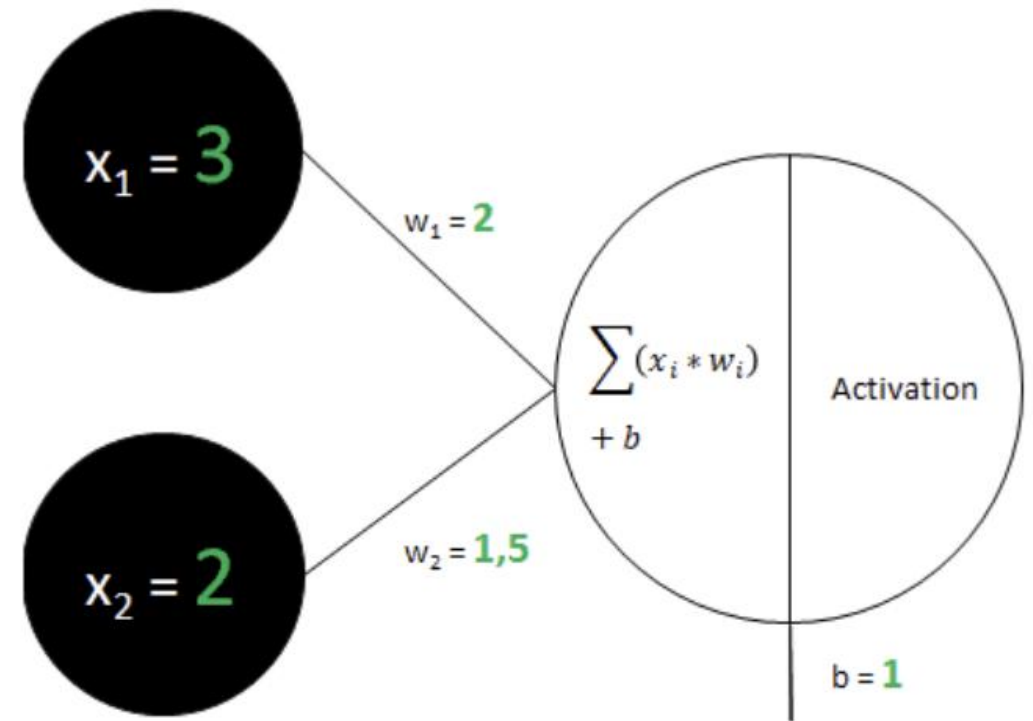


$$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$$

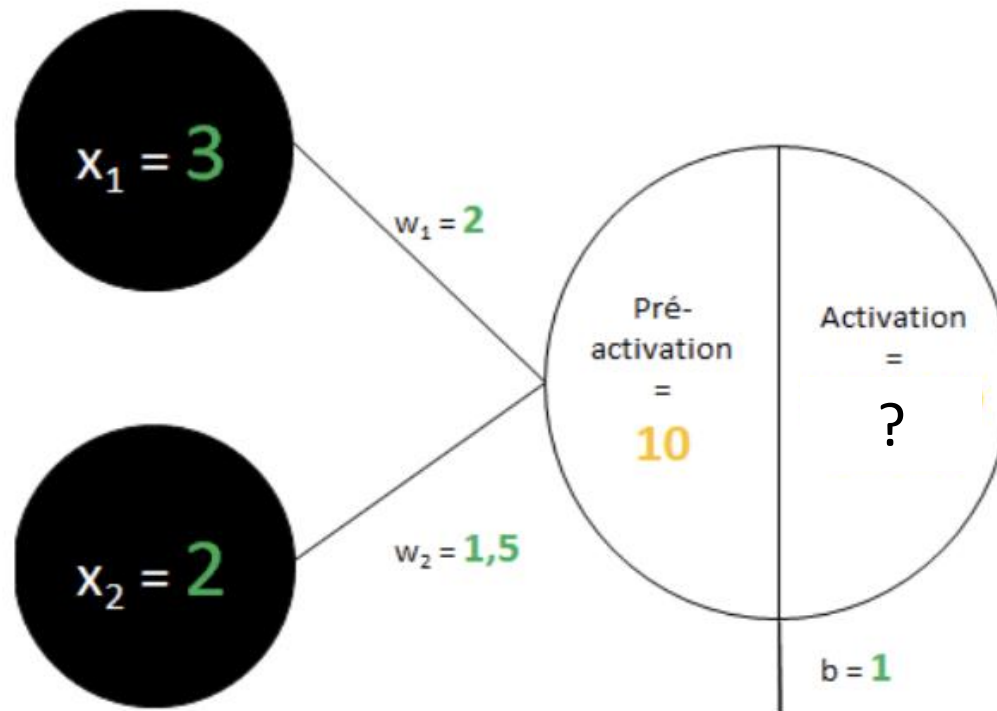
Fonction "Unité de rectification linéaire douce" (SoftPlus)



$$f(x) = \ln(1 + e^x)$$



# Les réseaux de neurones

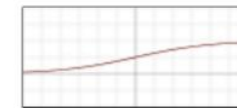


Fonction "Marche/Heaviside"



$$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$$

Fonction Sigmoide



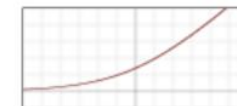
$$f(x) = \frac{1}{1 + e^{-x}}$$

Fonction "Unité de rectification linéaire" (ReLU)



$$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$$

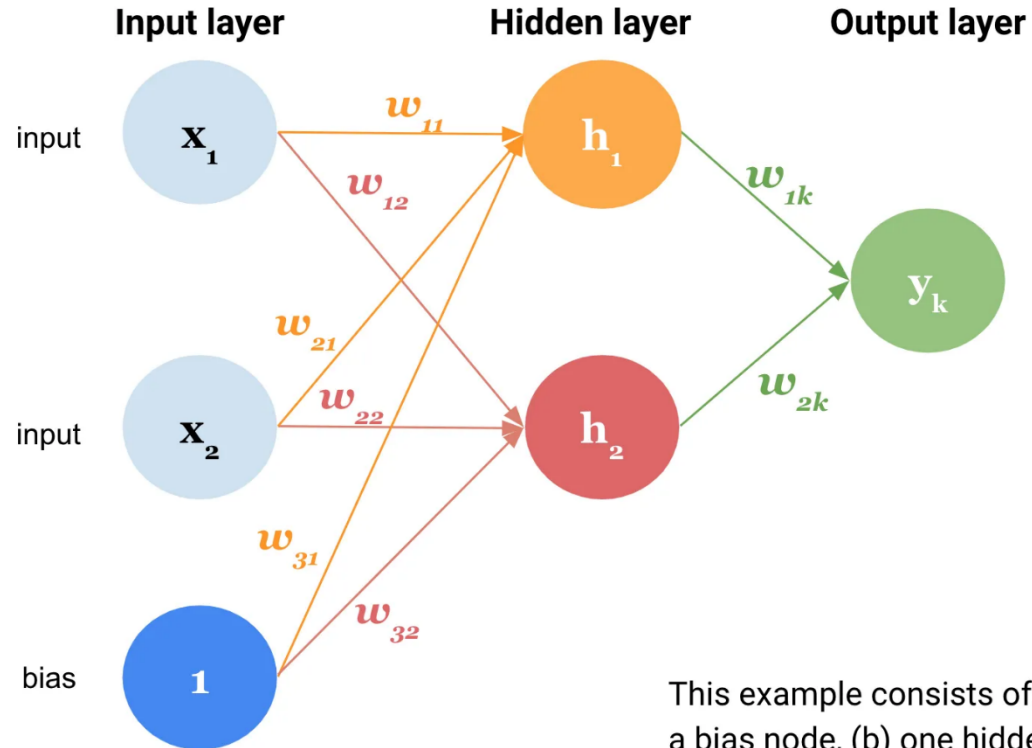
Fonction "Unité de rectification linéaire douce" (SoftPlus)



$$f(x) = \ln(1 + e^x)$$

# Comment fonctionne un réseau de neurones classique ?

## Illustrative example of Multilayer perceptron, a Feedforward neural network

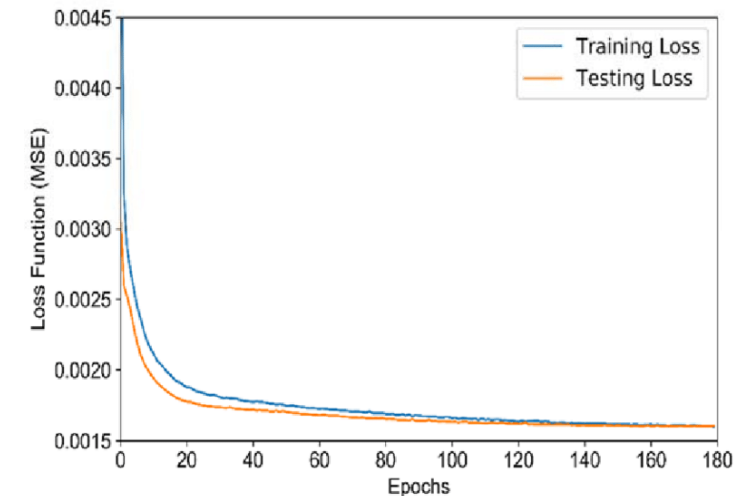
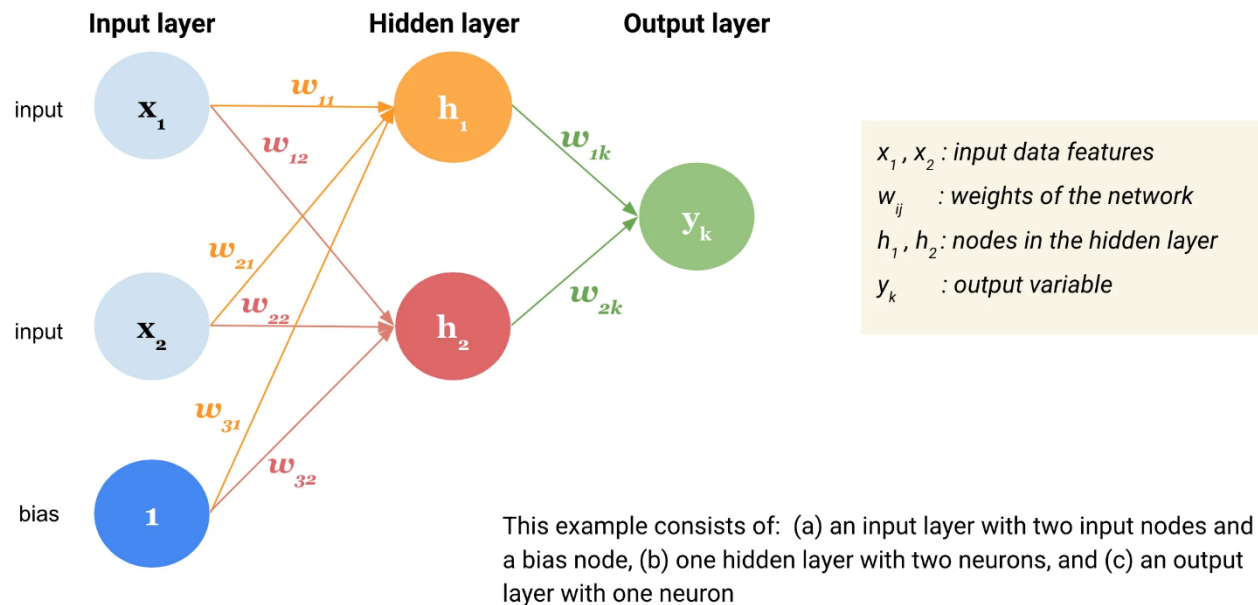


$x_1, x_2$  : input data features  
 $w_{ij}$  : weights of the network  
 $h_1, h_2$  : nodes in the hidden layer  
 $y_k$  : output variable

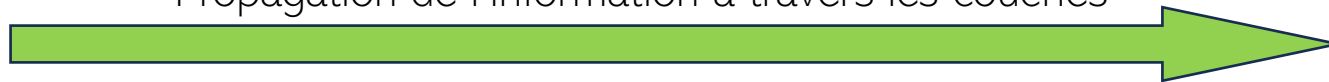
This example consists of: (a) an input layer with two input nodes and a bias node, (b) one hidden layer with two neurons, and (c) an output layer with one neuron

# Comment un MLP apprend-il ?

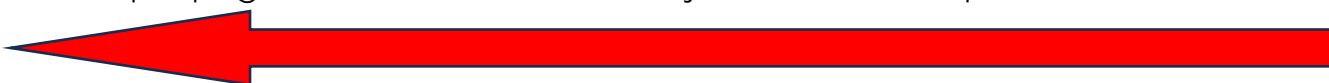
Illustrative example of Multilayer perceptron, a Feedforward neural network



Propagation de l'information à travers les couches



Rétropropagation de l'erreur avec ajustement des poids du modèle



**Objectif : minimisation d'une fonction de coût**



# Plan

L'intelligence artificielle et l'apprentissage automatique

Les réseaux de neurones

- Les réseaux de neurones simples
- Les limitations en traitement d'image
- Les réseaux de neurons convolutifs (CNN)

Dissection d'un CNN

Application en vision par ordinateur



# Pourquoi les MLP échouent avec les images ?

- Problème 1 : La perte de la structure spatiale : on transforme une image 2d en 3d en une liste plate : une image  $32 \times 32 = 1024$  pixels  $\rightarrow$  toutes les informations locales disparaissent
- Problème 2 : L'explosion des paramètres : pour une image  $32 \times 32 = 1024$  entrées  $1024$  entrées reliées à  $1000$  neurones  $= > 1\text{M}+$  paramètres !
- Résultat : les MLP sont inefficaces, lents et incapables de capter les motifs locaux (bords, textures).

# Plan

L'intelligence artificielle et l'apprentissage automatique

Les réseaux de neurones

- Les réseaux de neurones simples
- Les limitations en traitement d'image
- Les réseaux de neurons convolutifs (CNN)

Dissection d'un CNN

Application en vision par ordinateur

# Pourquoi les CNN sont la solution ?

- Problème : comment réussir à exploiter la localité et la hiérarchie des motifs ?
  - Un bord dépend des pixels voisins, pas de l'image entière
  - Une texture se dessine sur une sous-partie locale de l'image
- Solution : les couches convolutives :
  - On remplace des couches de neurones entièrement connectés par des filtres locaux : des fonctions de convolution
  - Les couches convolutives vont extraire les textures et motifs caractéristiques d'une image
  - Les couches connectées restantes vont combiner ces textures et motifs pour effectuer la tâche de classification
  - On résout ainsi les 2 grands problèmes des MLP :
    - beaucoup moins de paramètres
    - conservation de la structure spatiale et donc possibilité de capter cette information par le modèle

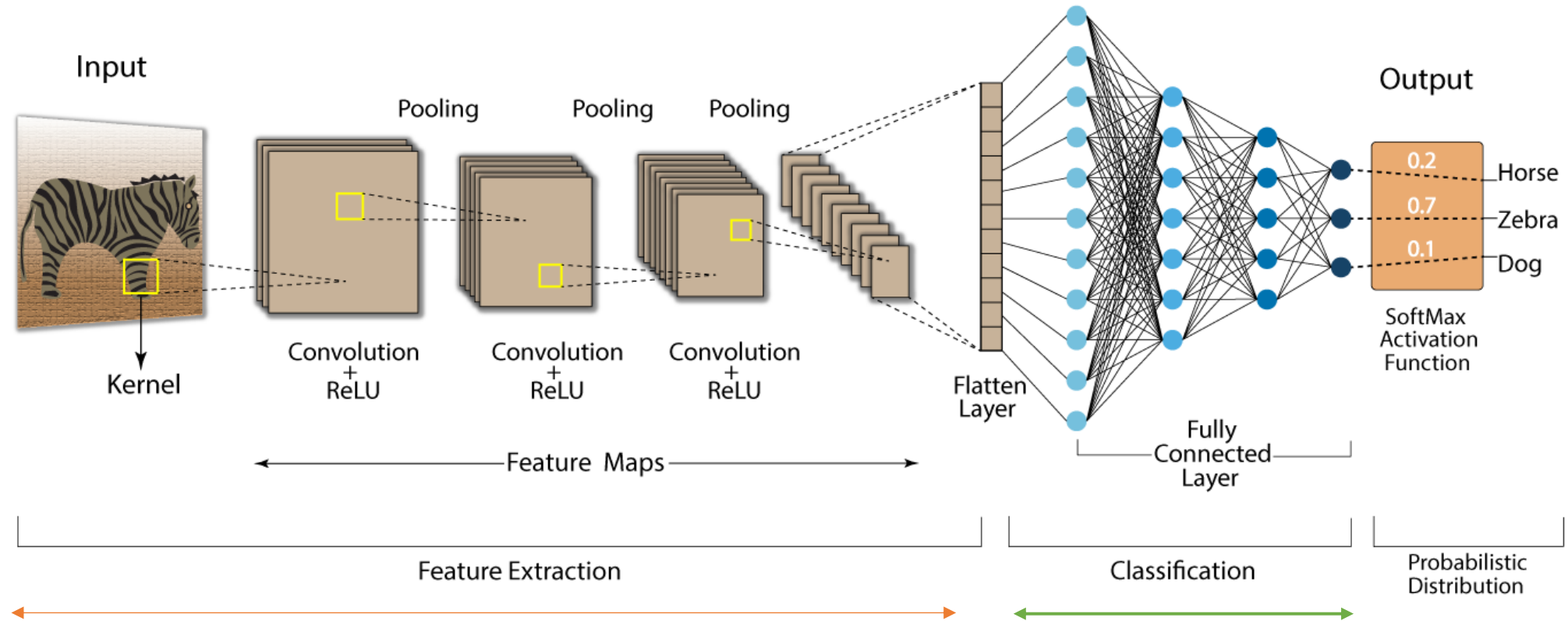
# Les CNNs

Les CNNs sont des MLP à qui on a rajouté une couche convolutive en amont.

Architecture classique d'un CNN :

- **Une partie convolution** qui a comme objectif **d'extraire les caractéristiques propres à l'image** en les compressant afin de réduire leur taille initiale
  - => L'image fournie en entrée passe à travers une succession de filtres, créant ainsi de nouvelles images, des « cartes de convolutions ».
  - => Les cartes de convolutions sont ensuite concaténées en un vecteur de caractéristique appelé « Code CNN »
- **Une partie classification** : le code CNN est fourni à un MLP, qui a comme objectif de **combiner les caractéristiques du code CNN** afin de classer l'image.

## Convolution Neural Network (CNN)

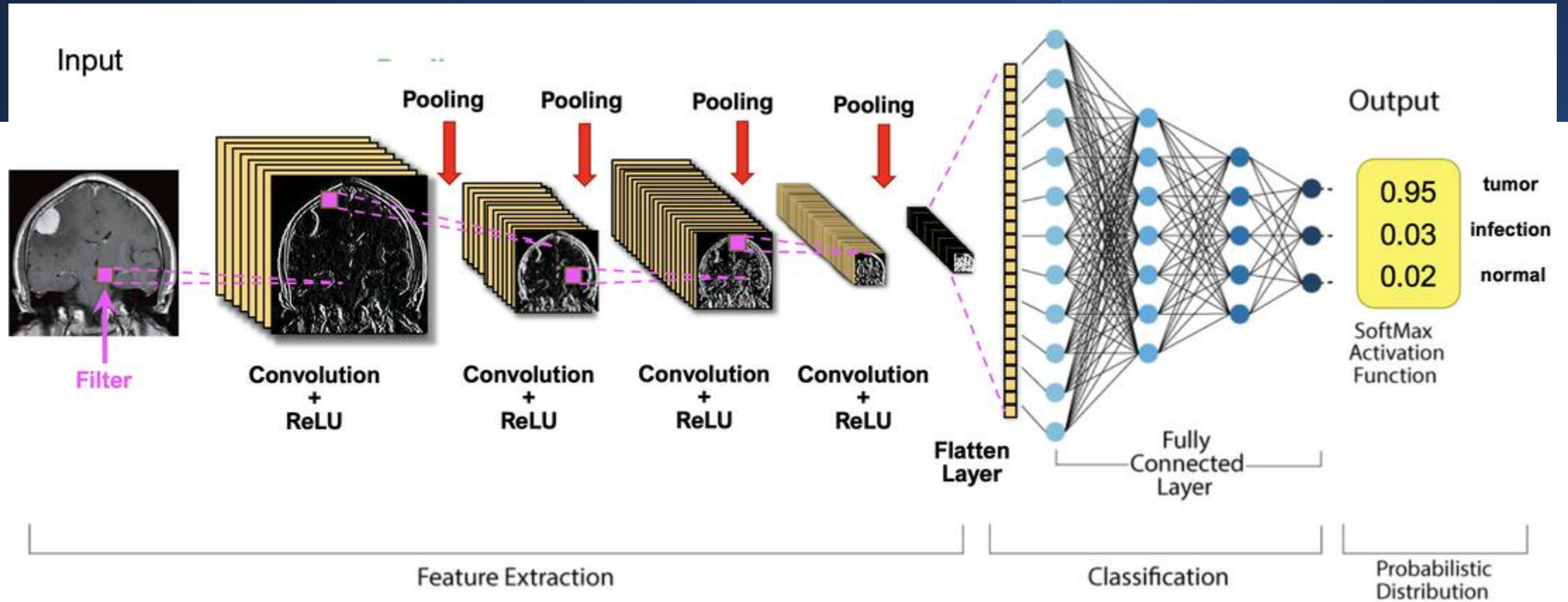


### Partie Convolution

*Les couches convolutives extraient les motifs et textures*

### Partie Classification

*Les neurones connectés  
combinent les motifs et textures  
pour décrire les classes de sortie*



Partie Convolution

*Les couches convolutives extraient les motifs et textures*

Partie Classification

*Les neurones connectés  
combinent les motifs et textures  
pour décrire les classes de sortie*

# En résumé

- Les MLP : sont des approches puissantes mais inadaptées aux images (structure ignorée, trop de paramètres)
- Les CNN : pensés et conçus pour les données visuelles grâce aux convolutions
- Prochaine étape : comprendre les couches des CNN !



# Plan

L'intelligence artificielle et  
l'apprentissage automatique

Les réseaux de neurones

Dissection d'un CNN

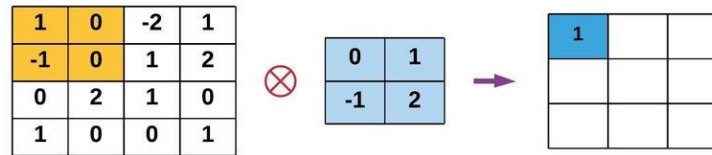
- Les briques élémentaires
- Architectures typiques
- Entraînement d'un modèle

Application en vision par ordinateur

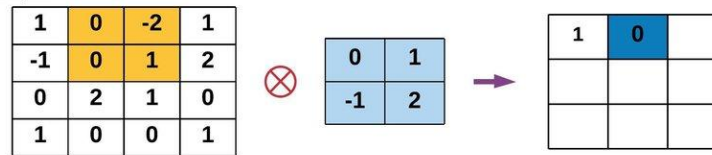


# La convolution

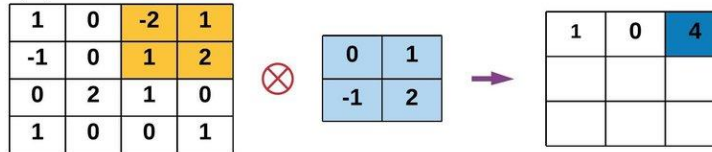
Step-1



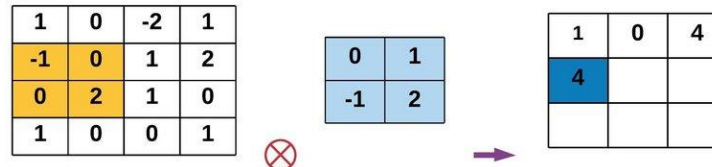
Step-2



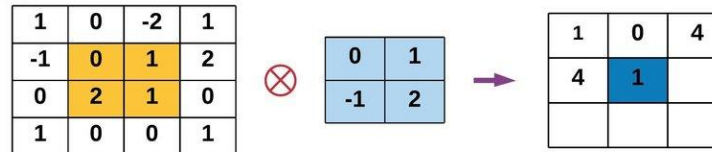
Step-3



Step-4



Step-5



# La convolution : extraction de features locaux

Input



Convolution (Kernel)

Edge Detection

$$\times \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} =$$

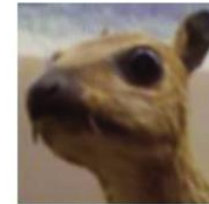
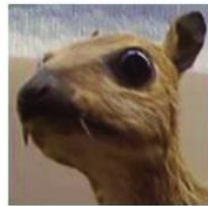
Feature



Edge

Box Blur

$$\times \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$



Blurred

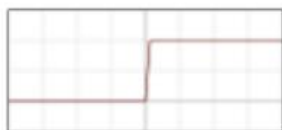
Sharpen

$$\times \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} =$$



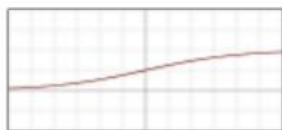
# Les fonctions d'activation

Fonction "Marche/Heaviside"



$$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$$

Fonction Sigmoide



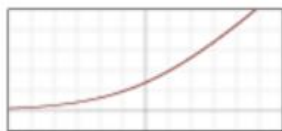
$$f(x) = \frac{1}{1 + e^{-x}}$$

Fonction "Unité de rectification linéaire" (ReLU)



$$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$$

Fonction "Unité de rectification linéaire douce" (SoftPlus)



$$f(x) = \ln(1 + e^x)$$

✓ **Fonction Marche (Step Function)** : Active ou désactive un neurone de façon binaire, inutilisable pour l'apprentissage profond à cause de l'absence de gradient.

✓ **Fonction Sigmoide** : Convertit une valeur en une probabilité entre 0 et 1, idéale pour la classification binaire mais sujette au problème du gradient qui disparaît.

✓ **Fonction ReLU** : Remplace les valeurs négatives par zéro, introduisant de la non-linéarité et accélérant l'apprentissage en réduisant les problèmes de gradient.

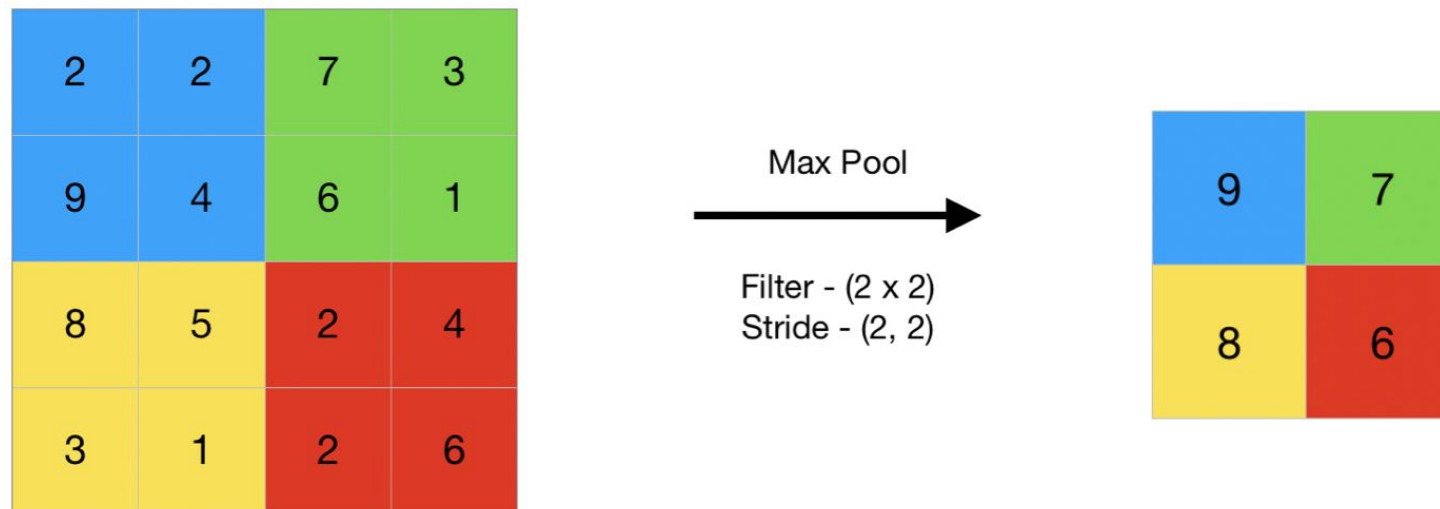
✓ **Fonction Softplus** : Lisse la version de ReLU en assurant une différentiabilité partout, utile pour éviter les discontinuités mais moins efficace que ReLU en pratique.

# Le pooling

L'opération de *pooling* consiste à **réduire la taille des images, tout en préservant leurs caractéristiques importantes.**

Le pooling permet :

- 1 : une réduction de la dimensionalité du modèle
- 2 : D'aider le modèle à être invariant aux petites translations
- 3 : de limiter le risque d'overfitting
- 4 : une hiérarchie des features extraits, les couches les plus basses capturant les détails fins, les couches les plus hautes capturant les features globaux ou plus abstraits

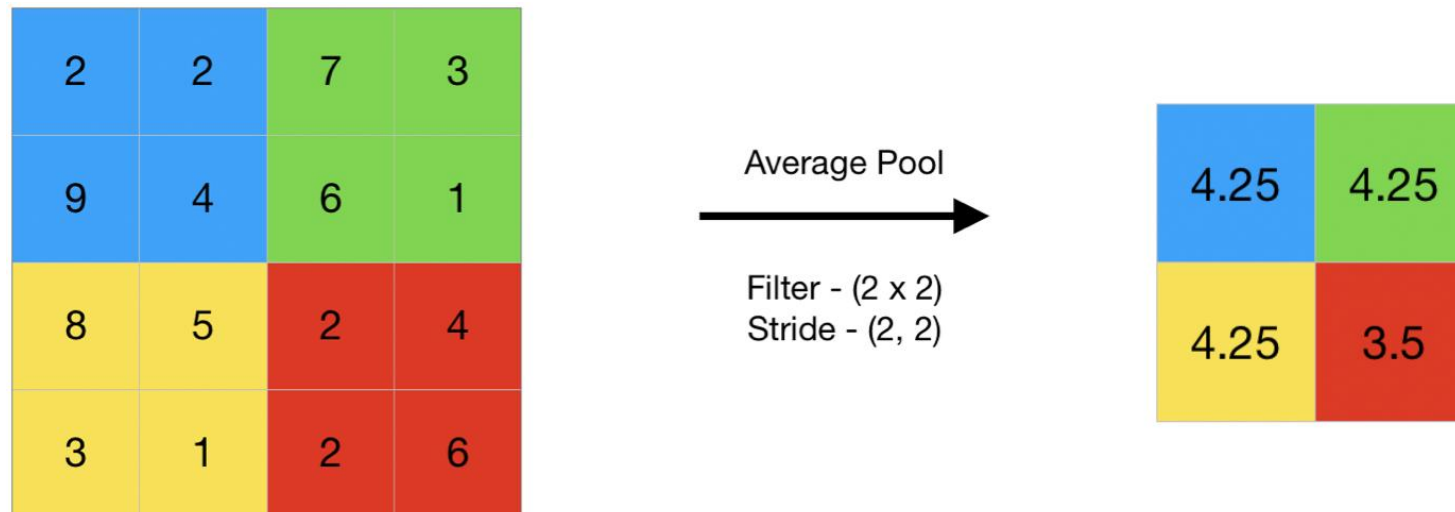


# Le pooling

L'opération de *pooling* consiste à **réduire la taille des images, tout en préservant leurs caractéristiques importantes.**

Le pooling permet :

- 1 : une réduction de la dimensionalité du modèle
- 2 : D'aider le modèle à être invariant aux petites translations
- 3 : de limiter le risque d'overfitting
- 4 : une hiérarchie des features extraits, les couches les plus basses capturant les détails fins, les couches les plus hautes capturant les features globaux ou plus abstraits



# Plan

L'intelligence artificielle et  
l'apprentissage automatique

Les réseaux de neurones

Dissection d'un CNN

- Les briques élémentaires
- Architectures typiques
- Entraînement d'un modèle

Application en vision par ordinateur

# Des modèles puissants

## Image Classification on ImageNet

Leaderboard

Community Models

Dataset

View

Top 1 Accuracy



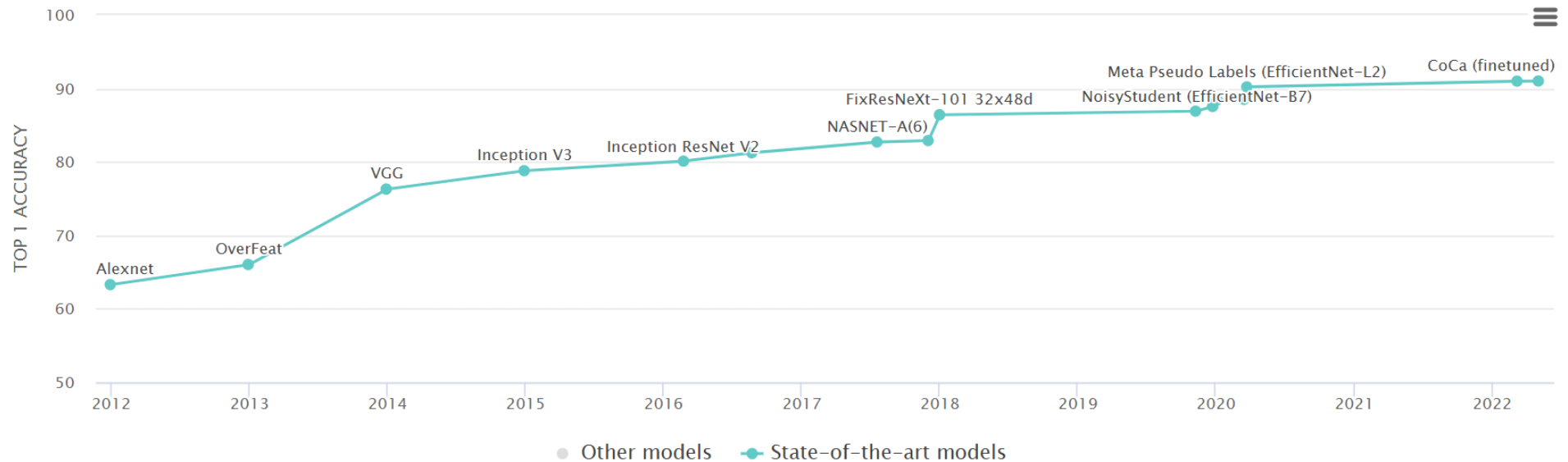
by

Date



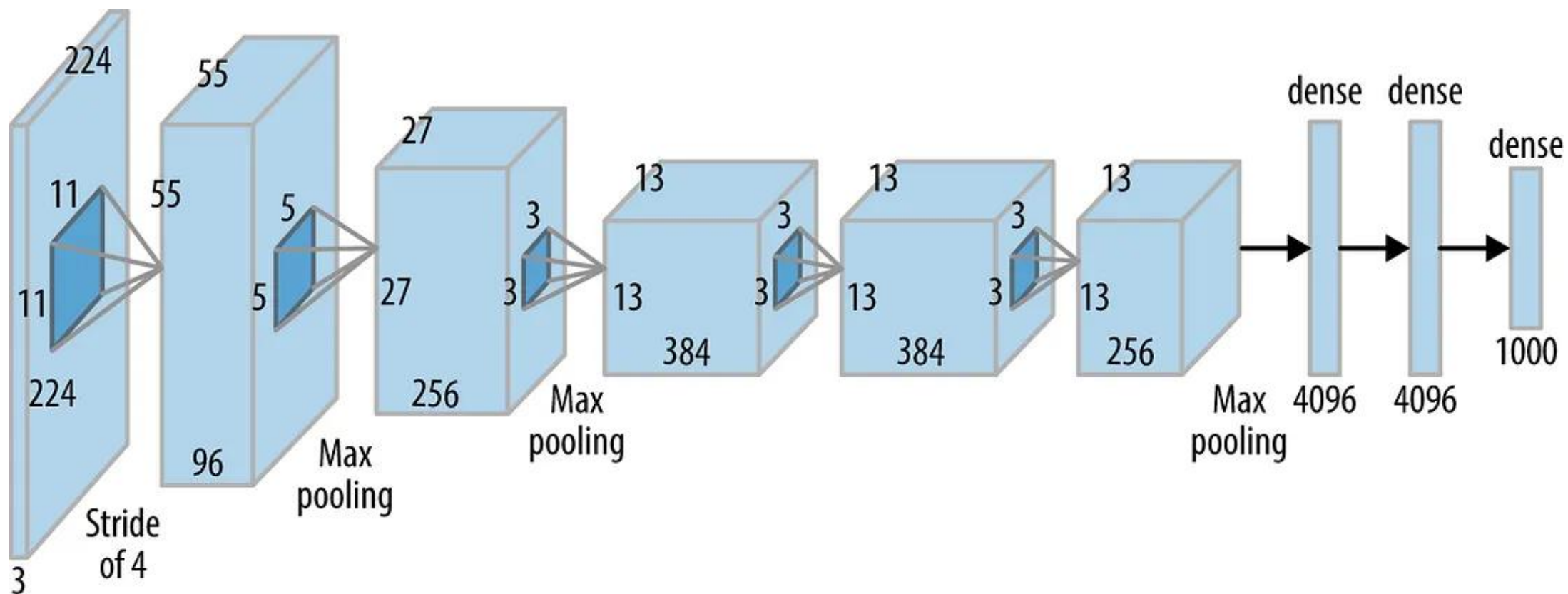
for

All models



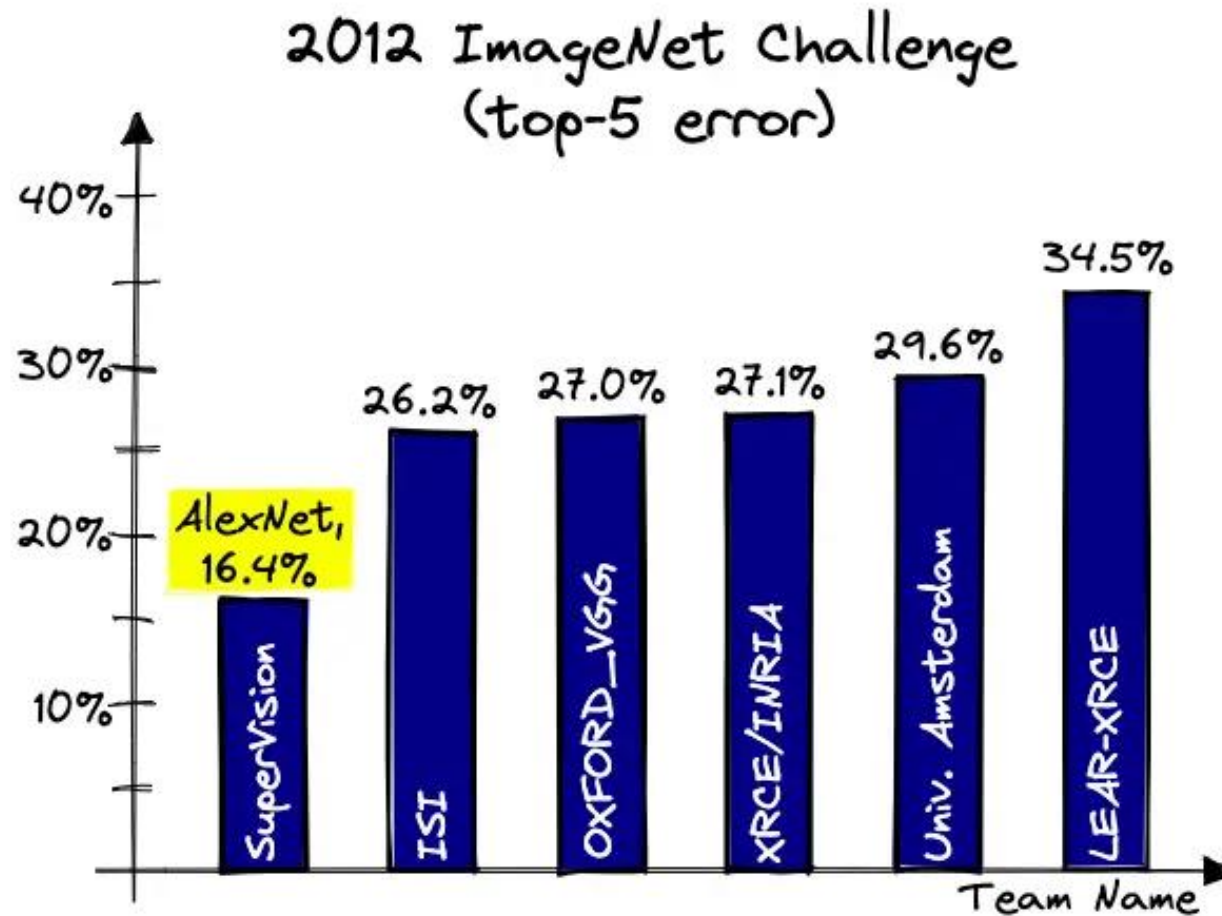


# AlexNet, 2012





# AlexNet, 2012



# AlexNet, 2012

Vainqueur du challenge 2012, passant le record de 26.1% d'erreurs à 15,3%

AlexNet Network - Structural Details													
Input			Output			Layer	Stride	Pad	Kernel size		in	out	# of Param
227	227	3	55	55	96	conv1	4	0	11	11	3	96	34944
55	55	96	27	27	96	maxpool1	2	0	3	3	96	96	0
27	27	96	27	27	256	conv2	1	2	5	5	96	256	614656
27	27	256	13	13	256	maxpool2	2	0	3	3	256	256	0
13	13	256	13	13	384	conv3	1	1	3	3	256	384	885120
13	13	384	13	13	384	conv4	1	1	3	3	384	384	1327488
13	13	384	13	13	256	conv5	1	1	3	3	384	256	884992
13	13	256	6	6	256	maxpool5	2	0	3	3	256	256	0
						fc6			1	1	9216	4096	37752832
						fc7			1	1	4096	4096	16781312
						fc8			1	1	4096	1000	4097000
<b>Total</b>						<b>62,378,344</b>							

# Plan

L'intelligence artificielle et  
l'apprentissage automatique

Les réseaux de neurones

Dissection d'un CNN

- Les briques élémentaires
- Architectures typiques
- Entraînement d'un modèle

Application en vision par ordinateur

# La fonction de perte ou loss function

La **fonction de perte** mesure l'écart entre la prédiction du modèle et la vérité terrain (label réel). Elle guide l'optimisation du modèle en minimisant cette erreur.

Exemple : l'entropie croisée :

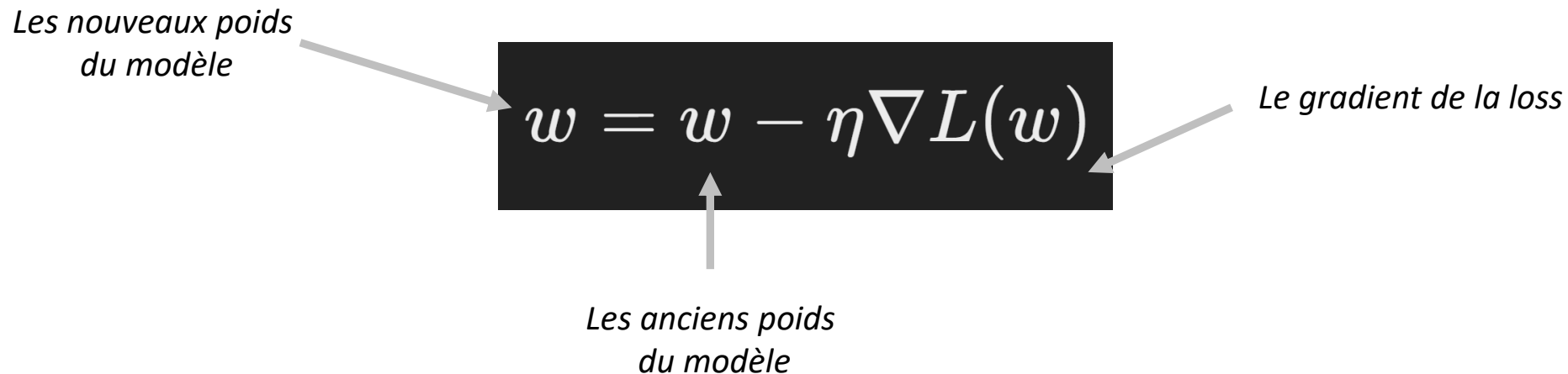
$$L = - \sum_i y_i \log(\hat{y}_i)$$

## Autres Fonctions de Perte

- La divergence Kullback-Leibler
- La focal loss
- **MSE (Mean Squared Error)** : Pour les régressions
- **MAE (Mean Absolute Error)** : Alternative à MSE

# Optimisation : la descente de gradient stochastique

La **SGD** est une méthode d'optimisation qui met à jour les poids du modèle en suivant la direction négative du gradient de la fonction de perte.



The diagram shows the SGD update equation  $w = w - \eta \nabla L(w)$  centered in a black box. Three arrows point to the equation with labels: an arrow from the top-left points to the first  $w$  with the label "Les nouveaux poids du modèle"; an arrow from the bottom points to the second  $w$  with the label "Les anciens poids du modèle"; and an arrow from the right points to the gradient term  $\nabla L(w)$  with the label "Le gradient de la loss".

*Les nouveaux poids du modèle*

$$w = w - \eta \nabla L(w)$$

*Le gradient de la loss*

*Les anciens poids du modèle*

**Adam** : un optimiseur basé sur la SGD mais qui module le learning rate pour chaque poids du modèle, permettant une meilleure gestion de la convergence

# Plan

L'intelligence artificielle et  
l'apprentissage automatique

Les réseaux de neurones

Dissection d'un CNN

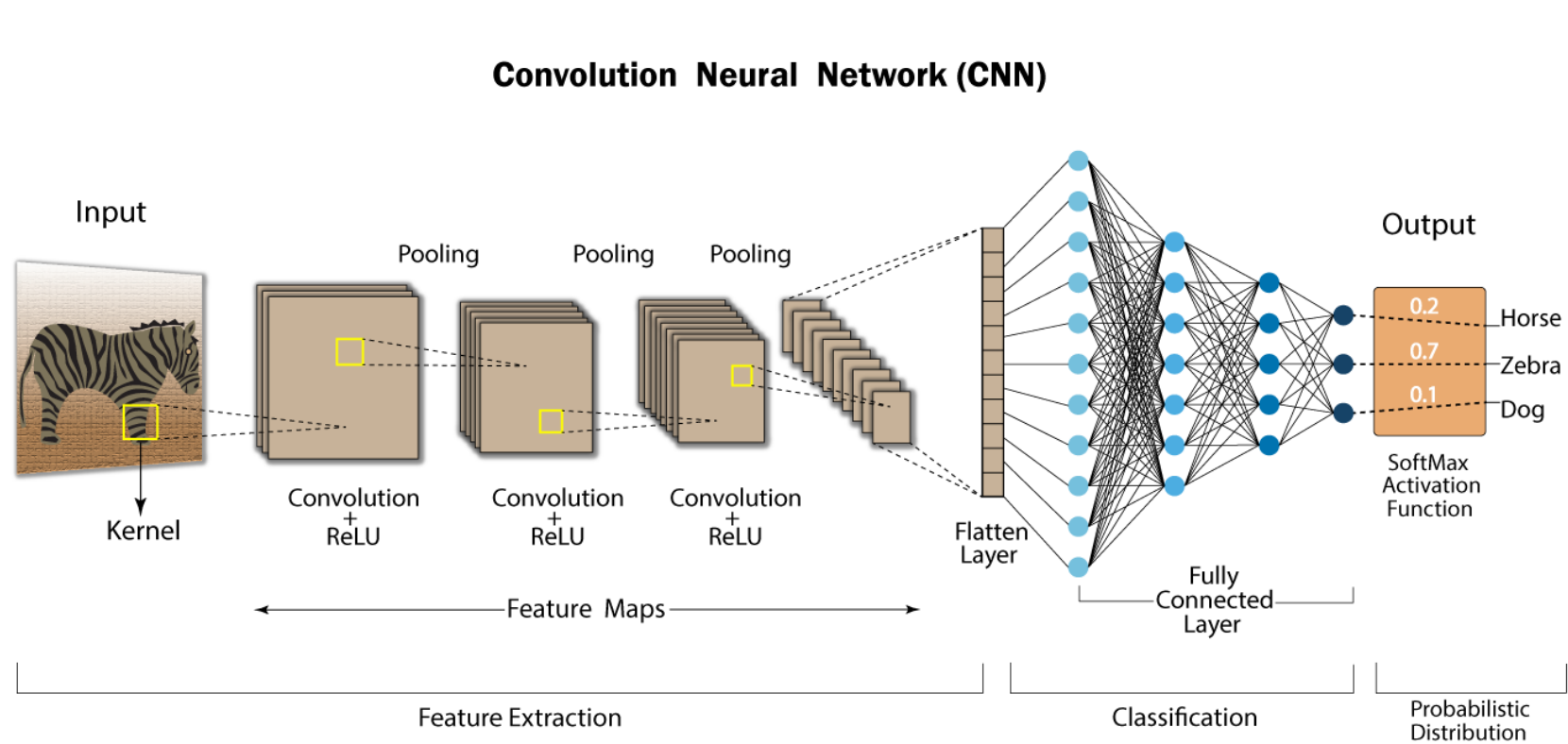
Application en vision par ordinateur

- Reconnaissance d'images
- Détection d'objets
- Segmentation sémantique
- Autres

# La reconnaissance d'images

La reconnaissance d'images permet d'identifier et de classer des objets, des personnes ou des scènes à partir d'images.

Les CNN sont particulièrement adaptés à cette tâche



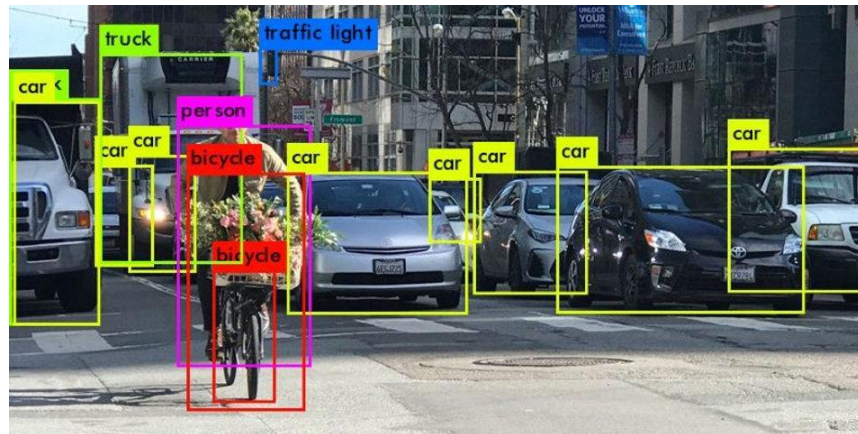


# La détection d'objets

La **détection d'objets** identifie et localise plusieurs objets dans une image en traçant des **bounding boxes** autour d'eux et en attribuant un **score de confiance**.

Modèles célèbres :

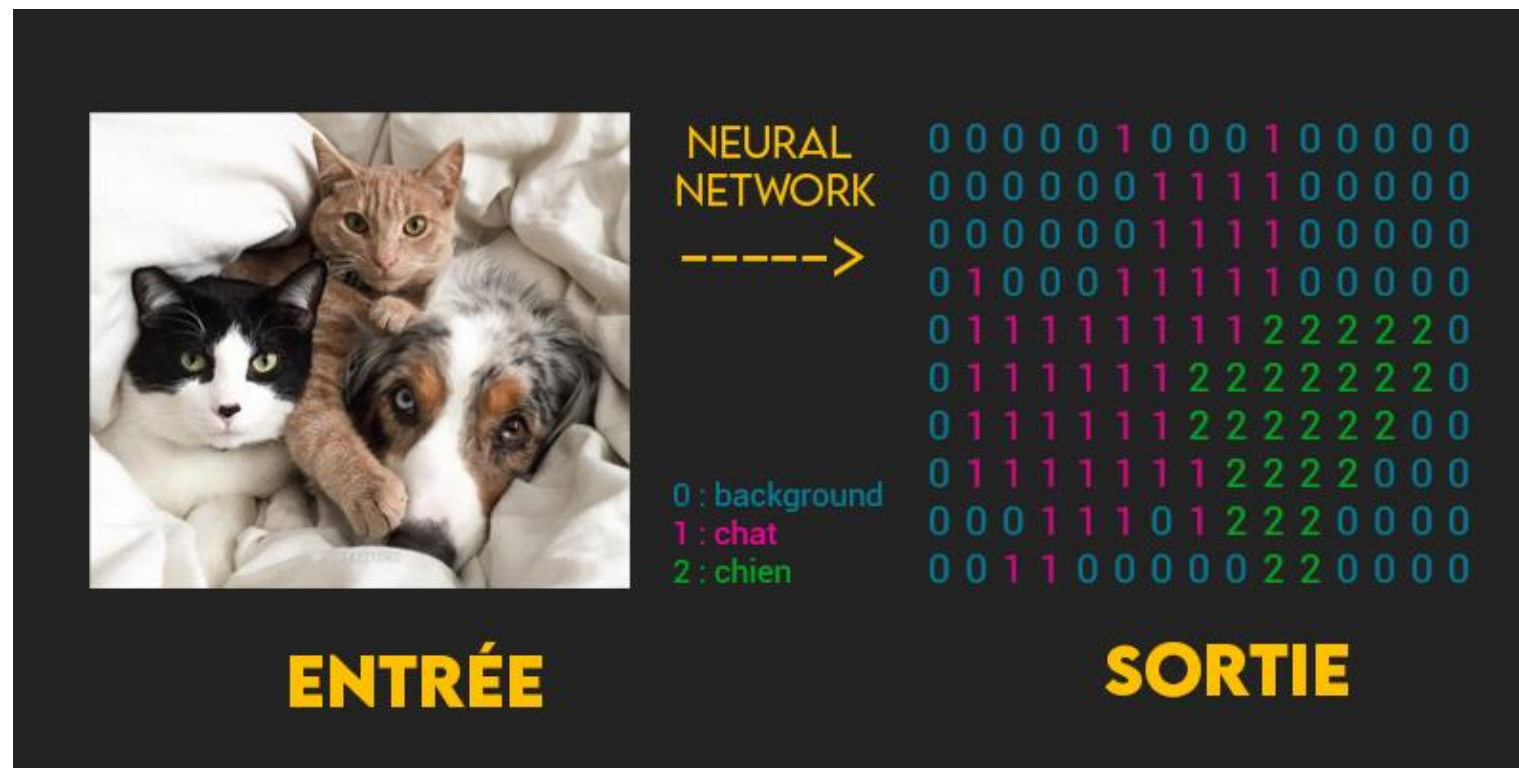
- **Yolo** : Analyse toute l'image en un seul passage, générant rapidement les bounding boxes (rapide et efficace)
- **R-CNN** : Divise l'image en régions candidates et applique un CNN sur chaque région (lent mais précis)





# La segmentation sémantique

La **segmentation sémantique** assigne une **classe** à **chaque pixel** d'une image, permettant une compréhension fine des objets et de leur structure.



# La segmentation sémantique

Le modèle produit en sortie une carte du forma de l'image d'entrée, il a donc une architecture d type encodeur/décodeur

Exemple U-Net :

