

Intelligence Artificielle pour le Traitement d'Images

Apprentissage profond et vision par ordinateur

Ecole Doctorale
SNS

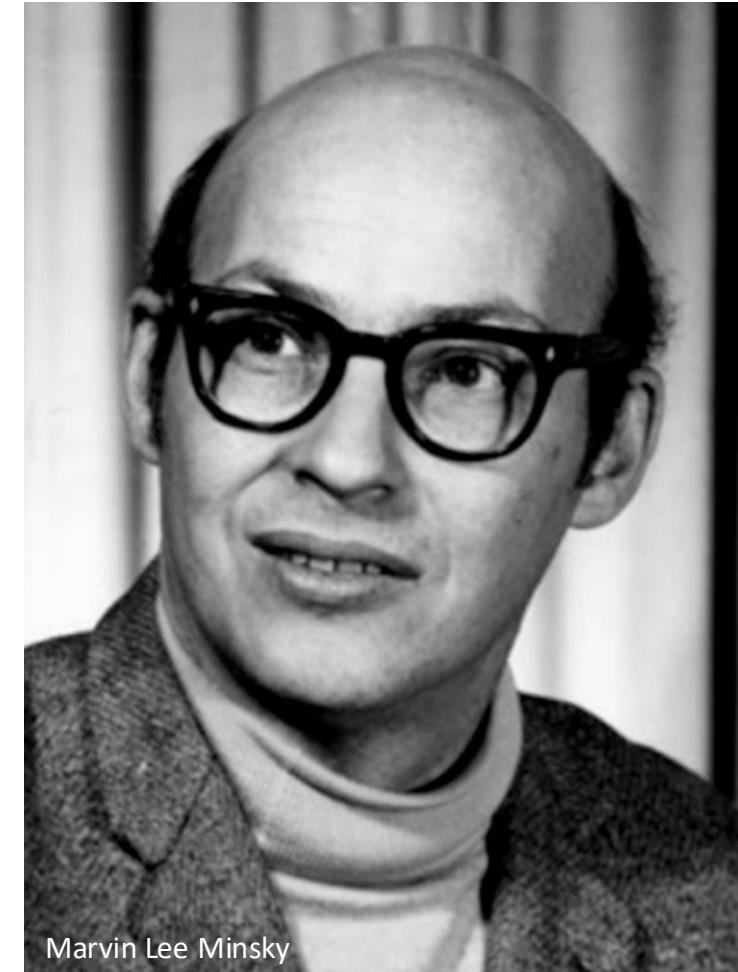


L'Intelligence Artificielle

construction de programmes informatiques capables de prendre en charge des tâches habituellement effectuées par des humains

car demandant un apprentissage, une organisation de la mémoire et un raisonnement. L'objectif est de parvenir à transmettre à une machine des fonctions propres au vivant : rationalité, raisonnement, mémoire et perception.

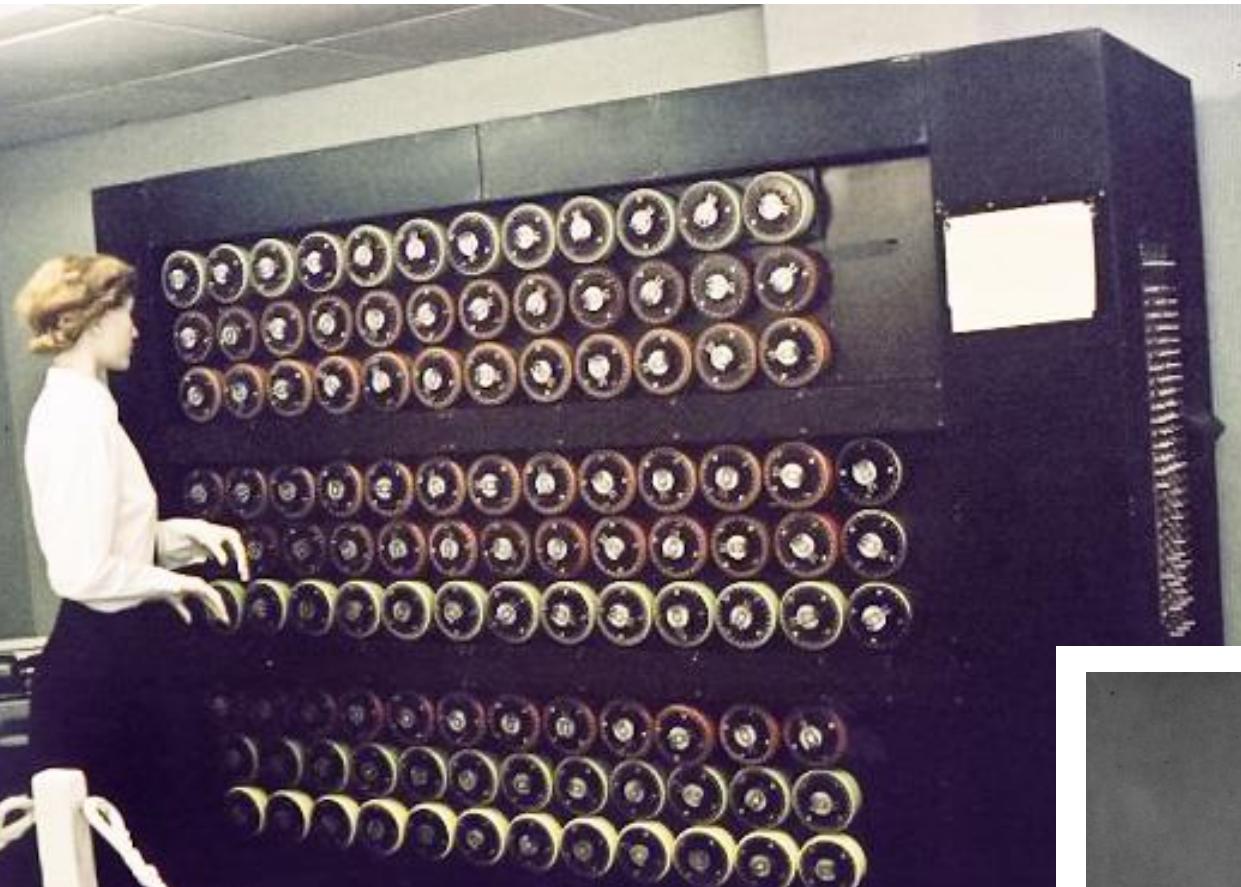
Raisonner / Comprendre / Interagir



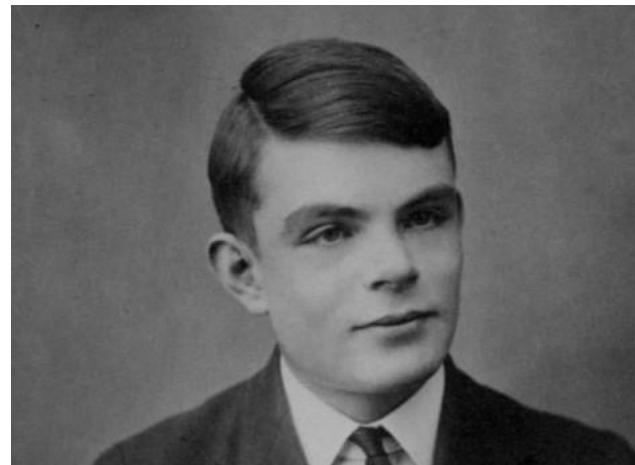
Marvin Lee Minsky

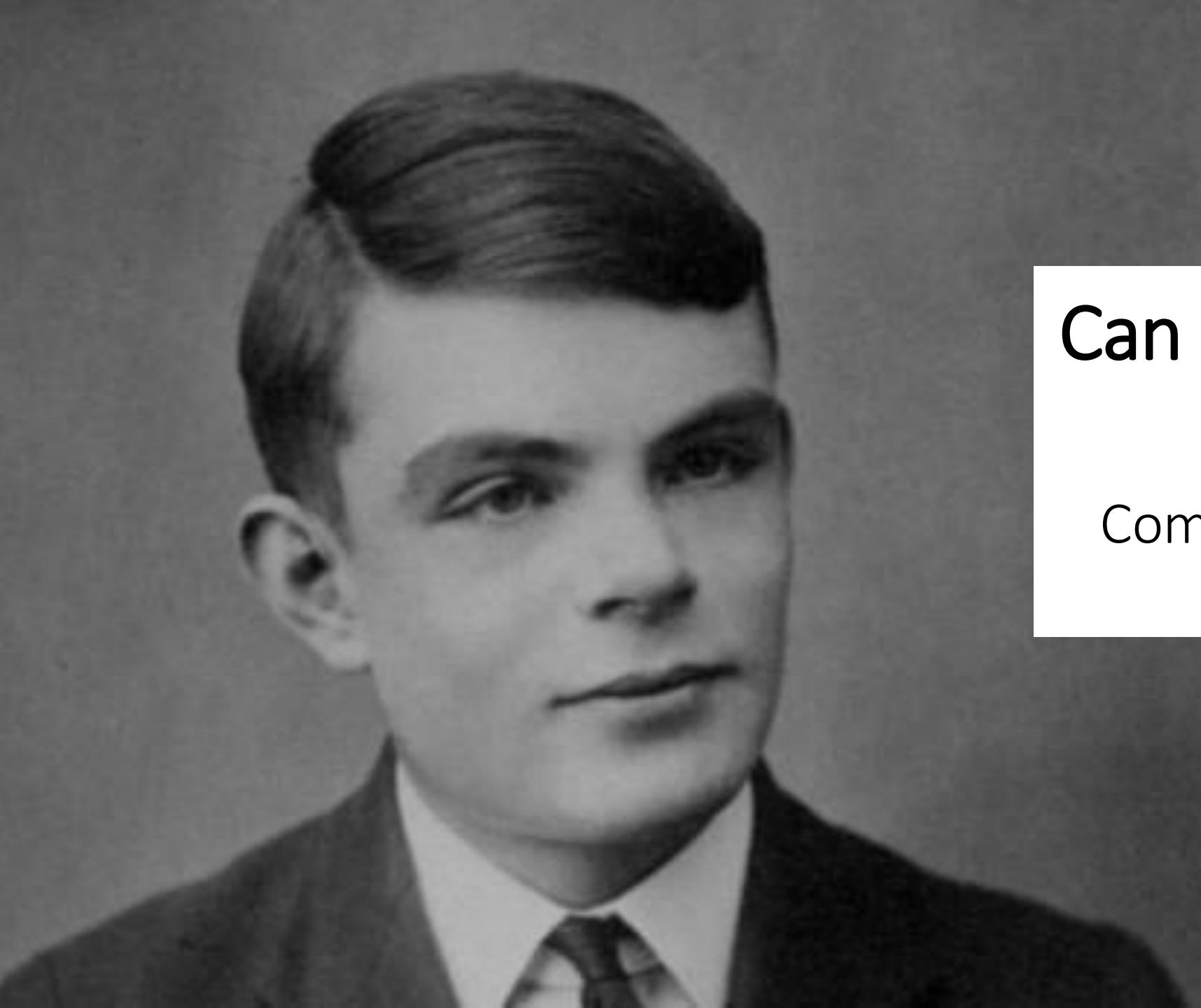
Un ordinateur peut-il
devenir intelligent ?

The Bomb



- Alan Turing
- Enigma
- The bomb





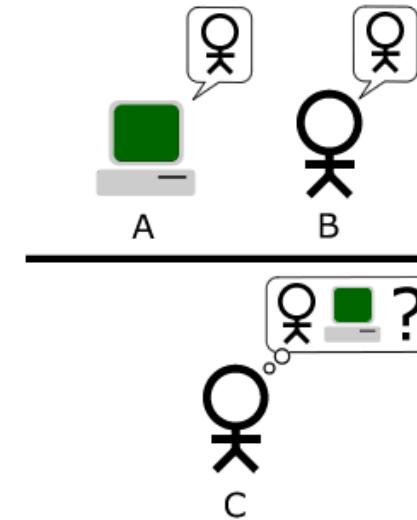
Can machines think ?

A. Turing

Computing machinery and
intelligence, 1950

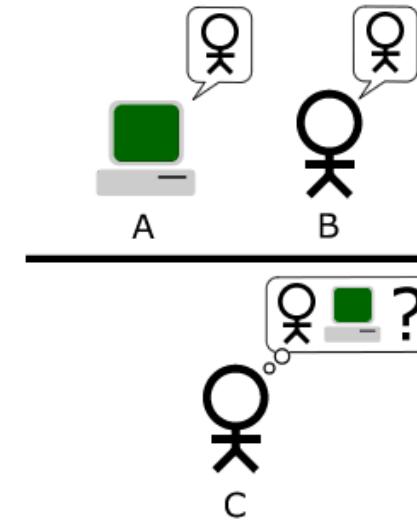
Mind (en), Oxford University Press,
vol. 59, no 236, octobre 1950 (DOI
[10.1093/mind/LIX.236.433](https://doi.org/10.1093/mind/LIX.236.433))

The imitation game



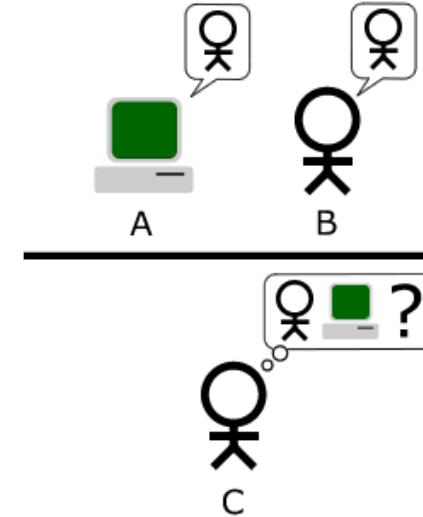
- Test de Turing
- « Un ordinateur peut-il penser »

The imitation game



- Test de Turing
- « Un ordinateur peut-il penser »
- Intelligence multiple
- Faire semblant de faire des erreurs

The imitation game

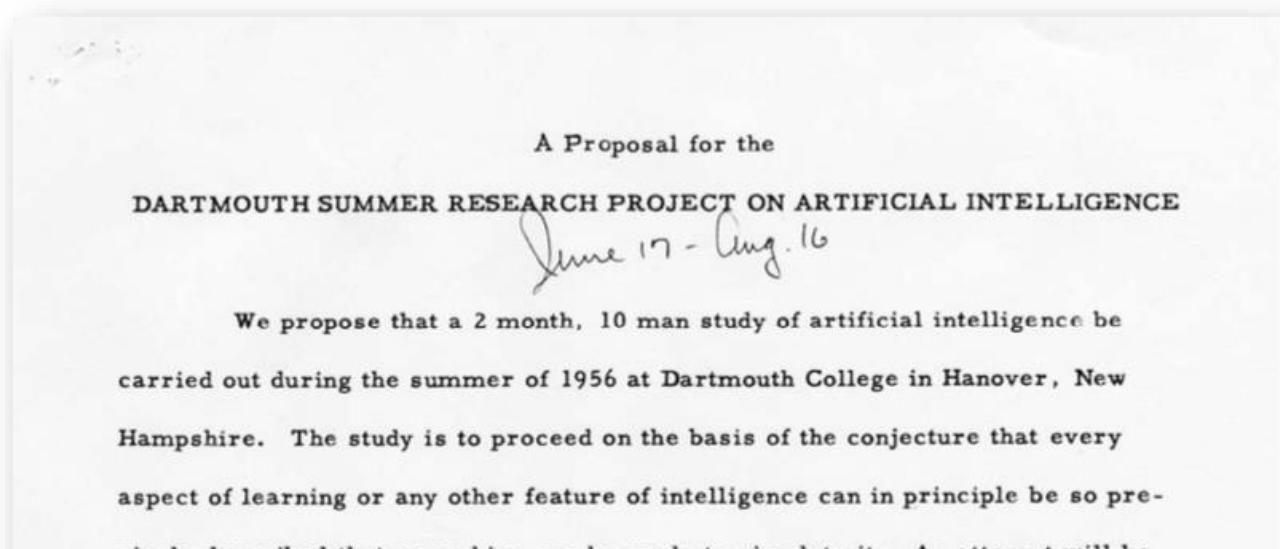


- Test de Turing
- « Un ordinateur peut-il penser »
- Intelligence multiple
- Faire semblant de faire des erreurs
- Machine expérimentale (pas de preuve ou d'algo)
- Ordinateur digital : disponibilité, **puissance**
- Ordinateur à variable discrète : mémoire ?

Historique de l'IA

- 1956 : L'intelligence artificielle devient un véritable domaine scientifique
 - Dartmouth Summer Research Project on Artificial Intelligence
 - Organisée par John McCarthy

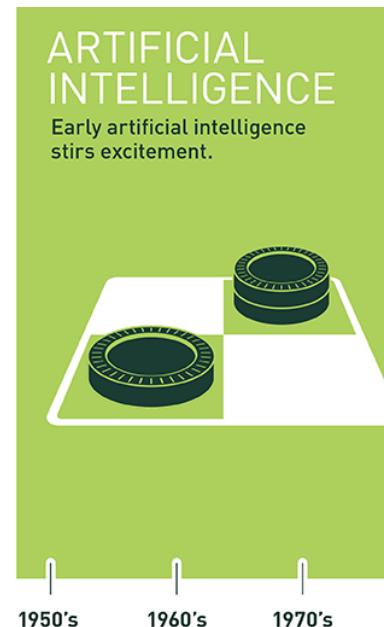
“ find how to make machines [...] solve kinds of problems now reserved for humans ”



août 1956

Historique de l'IA

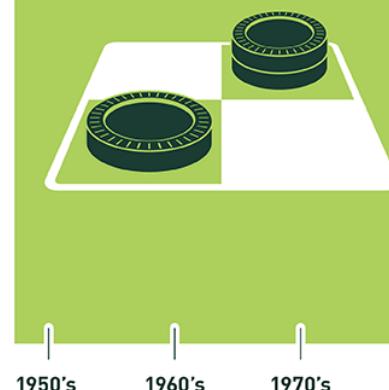
- 1956 : L'intelligence artificielle devient un véritable domaine scientifique
- 60' : financements, laboratoires USA & UK
 - « des machines seront capables, d'ici 20 ans, de faire le travail que toute personne peut faire »



Historique de l'IA

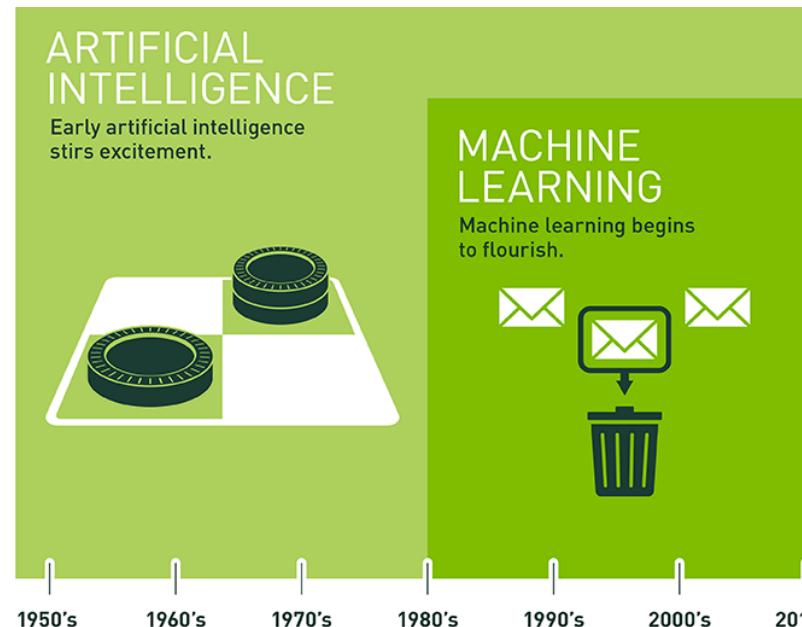
- 1956 : L'intelligence artificielle devient un véritable domaine scientifique
- 60' : financements, laboratoires USA & UK
 - « des machines seront capables, d'ici 20 ans, de faire le travail que toute personne peut faire »
- 1974 : AI Winter : les projets n'aboutissent pas, les ordinateurs ne sont pas assez puissants, les financements sont réduits

ARTIFICIAL
INTELLIGENCE
Early artificial intelligence
stirs excitement.



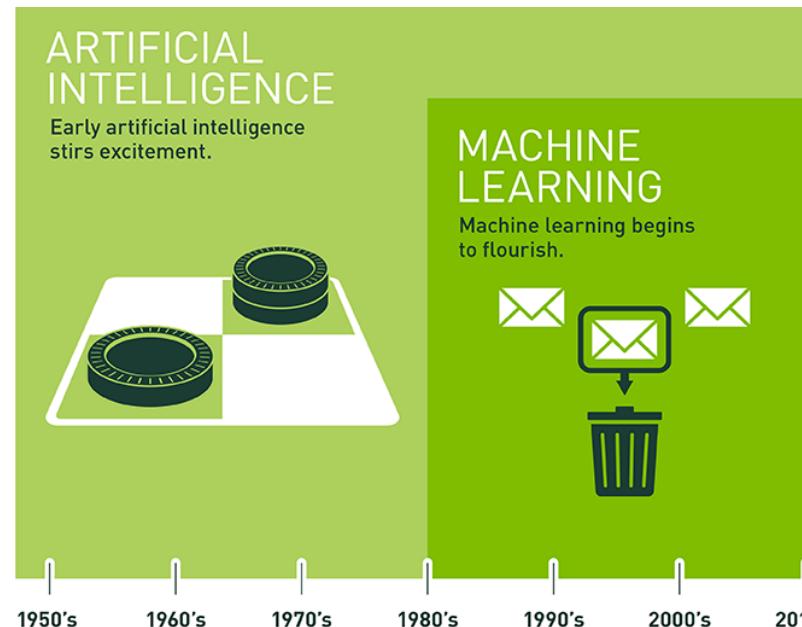
Historique de l'IA

- 1956 : L'intelligence artificielle devient un véritable domaine scientifique
- 60' : financements, laboratoires USA & UK
 - « des machines seront capables, d'ici 20 ans, de faire le travail que toute personne peut faire »
- 1974 : AI Winter : les projets n'aboutissent pas, les ordinateurs ne sont pas assez puissants, les financements sont réduits
- 80' : Systèmes experts (une seule tâche)



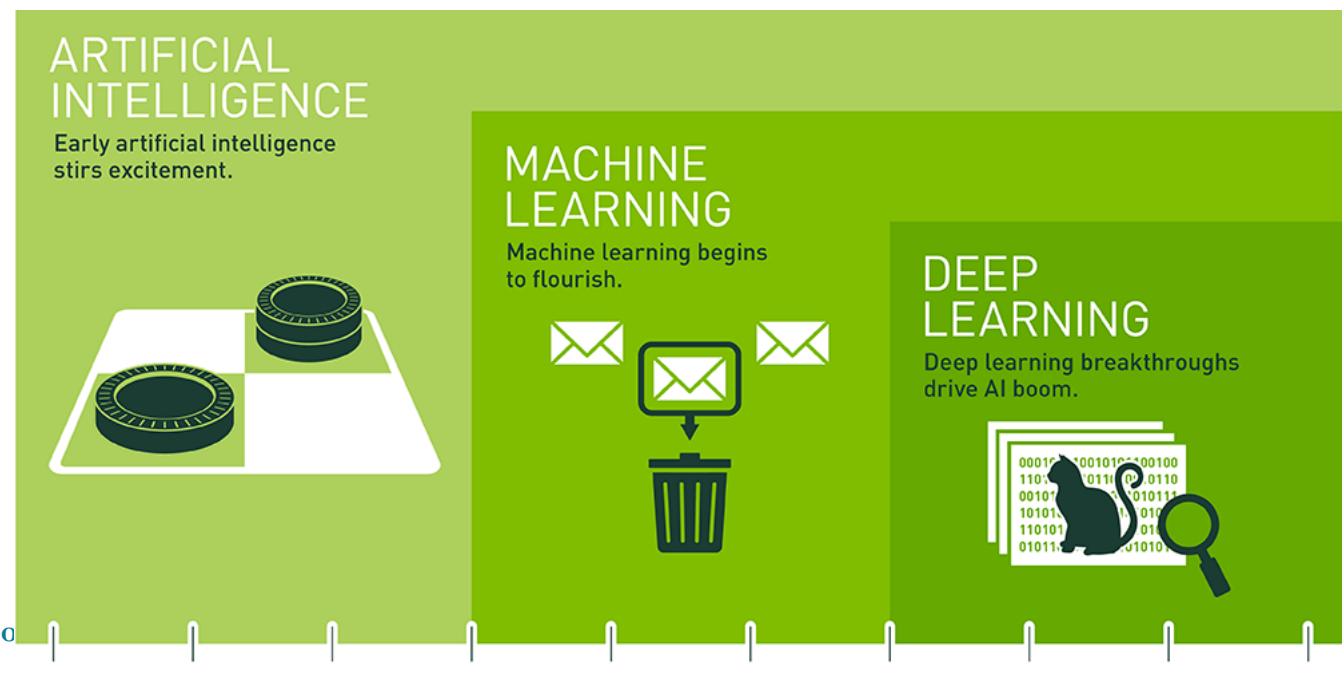
Historique de l'IA

- 1956 : L'intelligence artificielle devient un véritable domaine scientifique
- 60' : financements, laboratoires USA & UK
 - « des machines seront capables, d'ici 20 ans, de faire le travail que toute personne peut faire »
- 1974 : AI Winter : les projets n'aboutissent pas, les ordinateurs ne sont pas assez puissants, les financements sont réduits
- 80' : Systèmes experts (une seule tâche)
- 90' : Ordinateurs puissants, nouveaux domaines

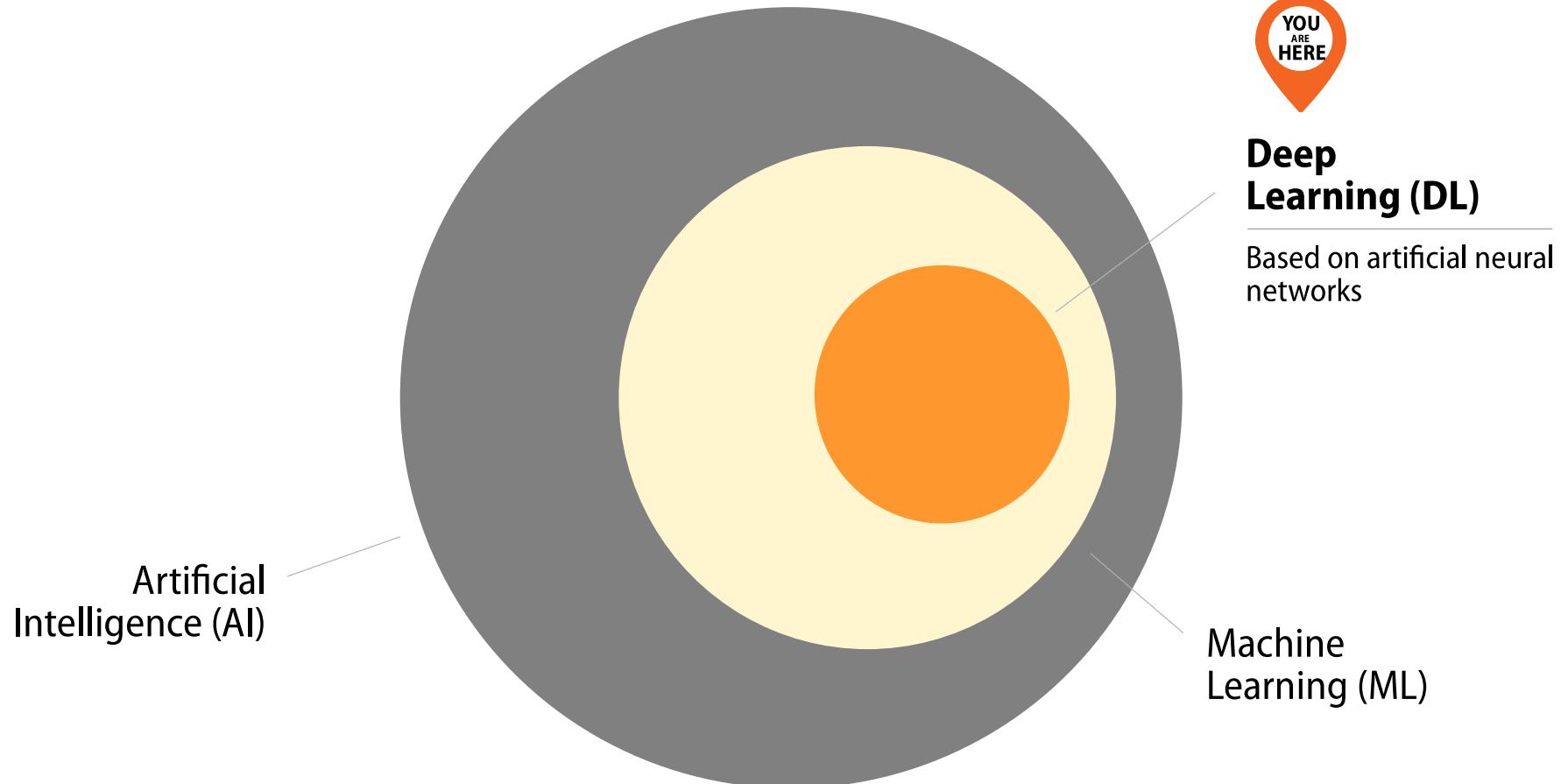


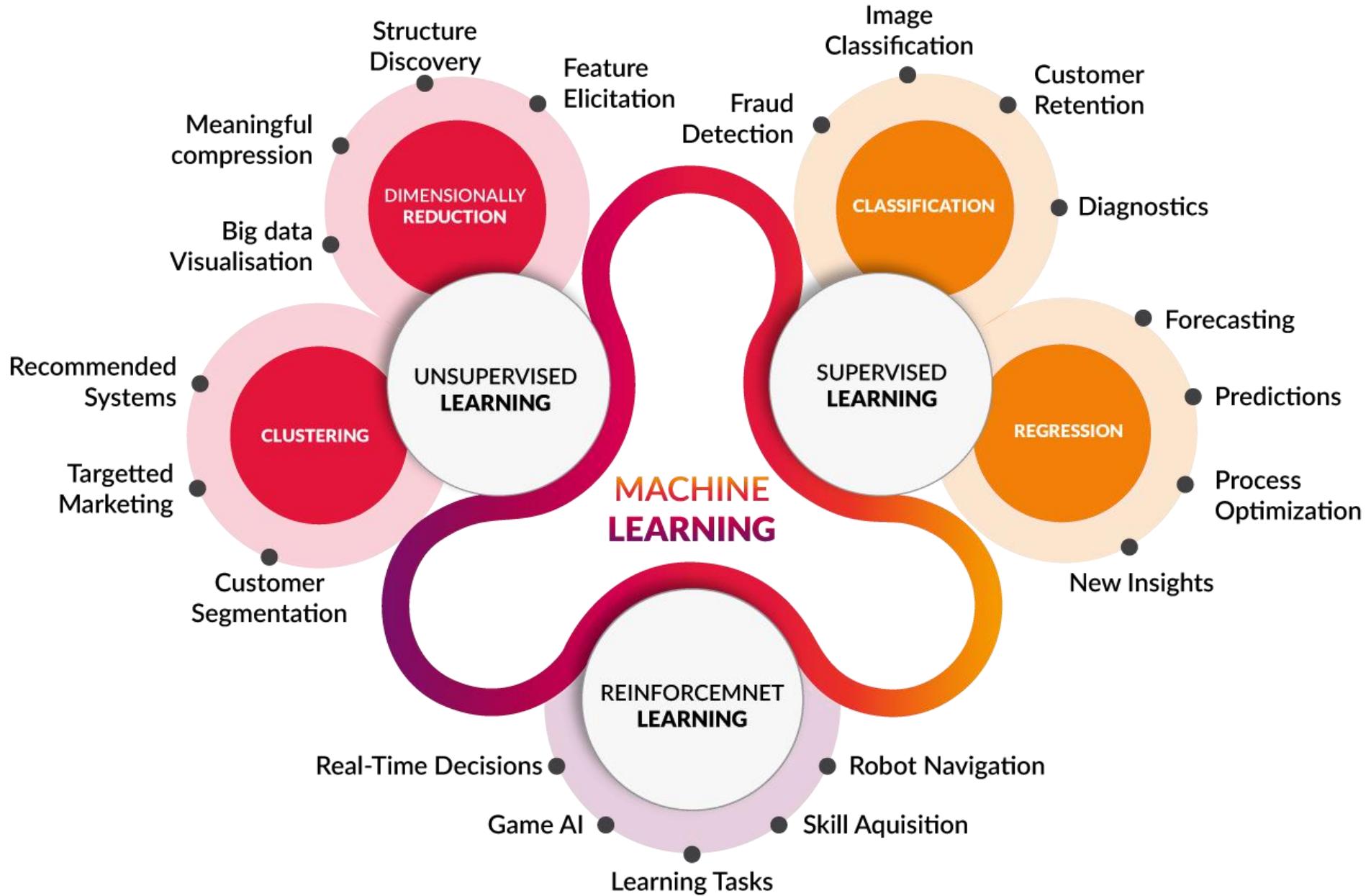
Historique de l'IA

- 2000' : l'avènement des données
 - Les nouvelles algorithmes
 - Les puissances de calcul phénoménales
 - Apprentissage profond : Deep Learning

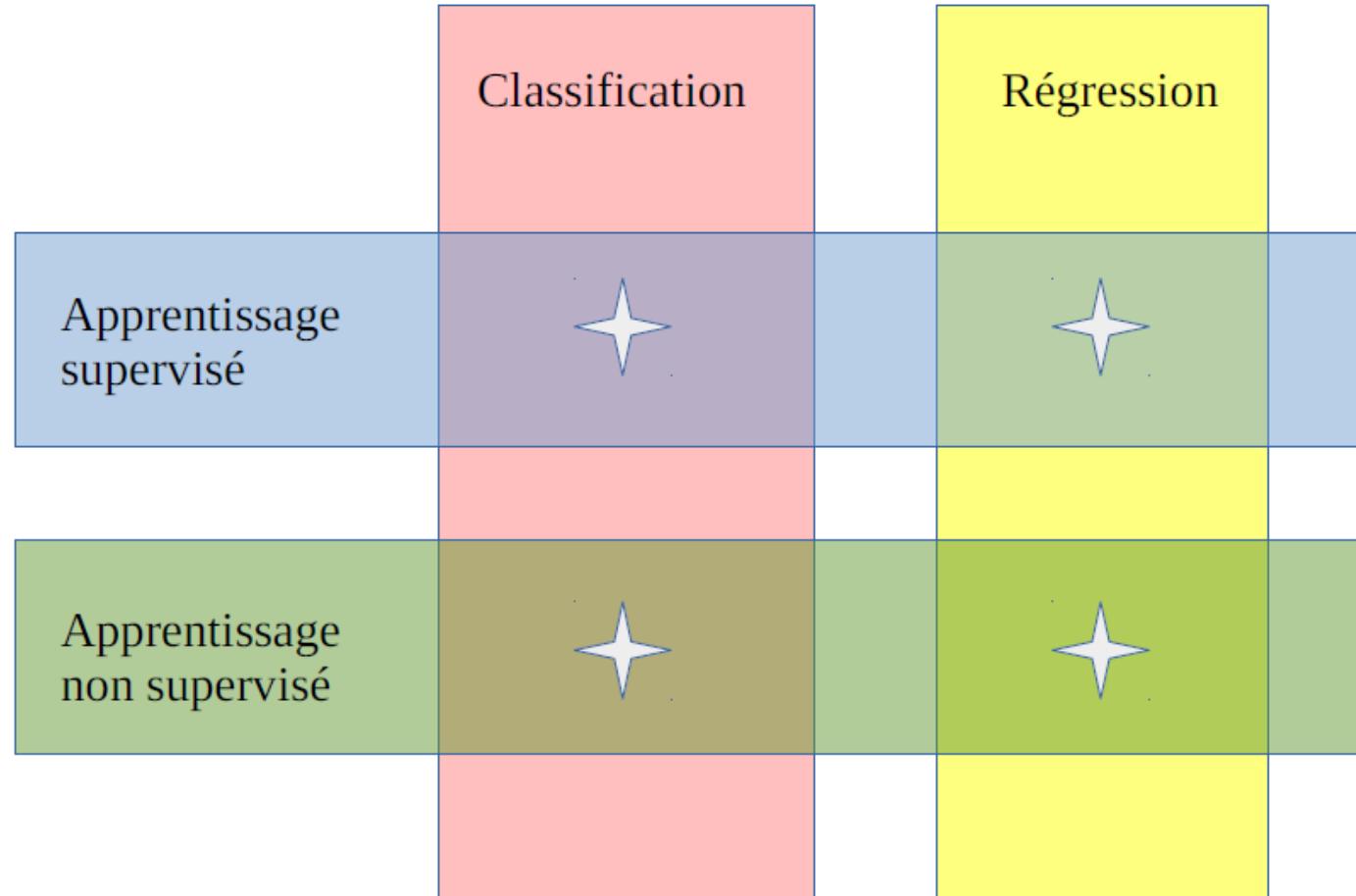


C'EST QUOI L'INTELLIGENCE ARTIFICIELLE ?

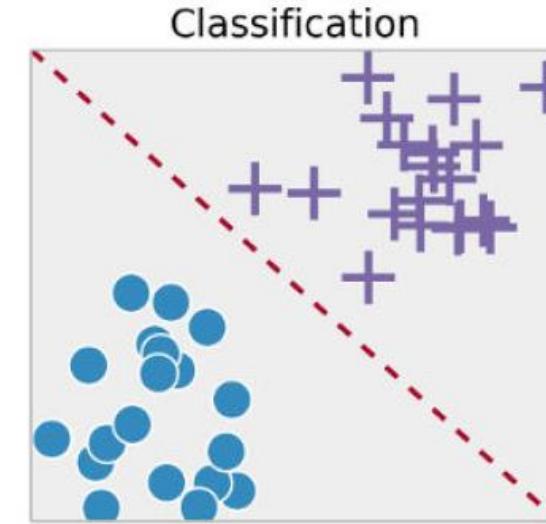
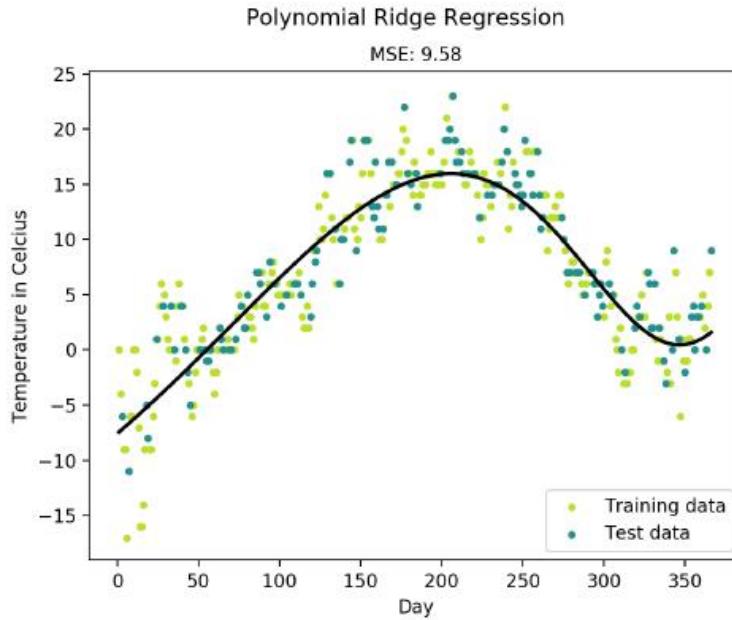




LES GRANDES FAMILLES



CLASSIFICATION VS REGRESSION

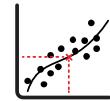


Régression

Prédire une variable quantitative



Tell me,
what's the
price ?



Classification

Prédire une classe (qualitative, discrète)



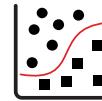
This is a cat



This is a rabbit

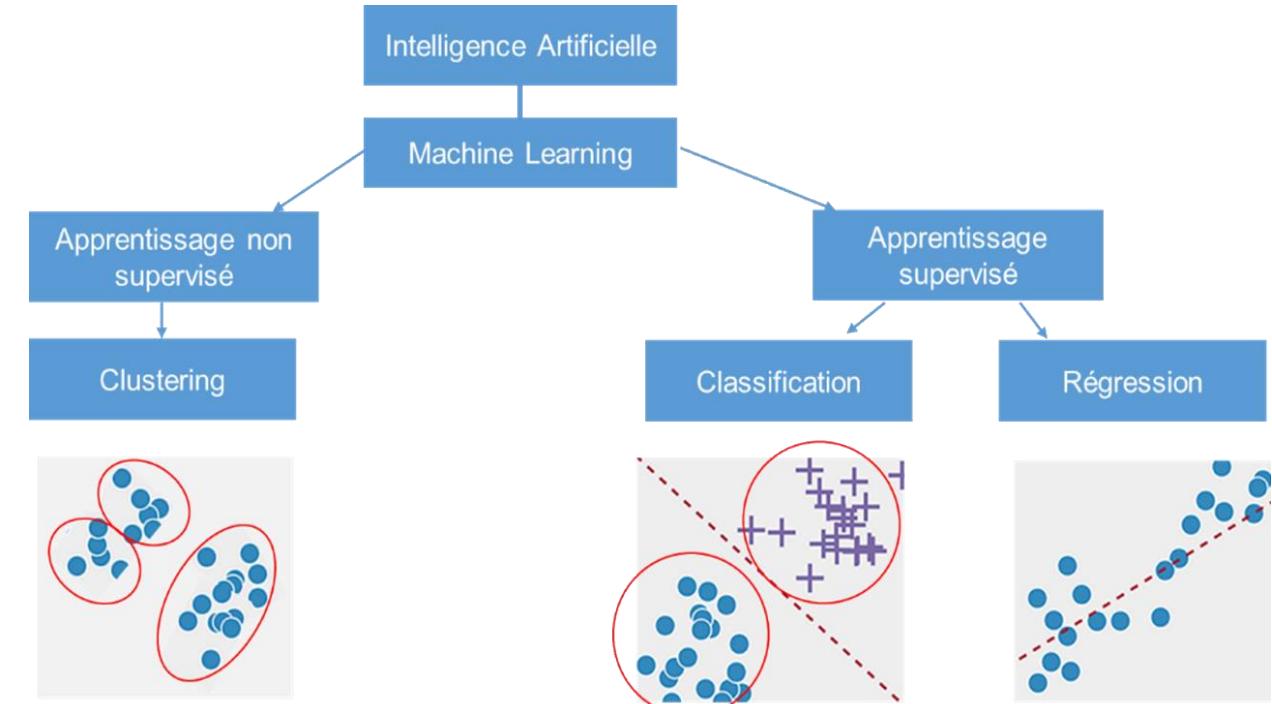


Tell me,
what is it ?



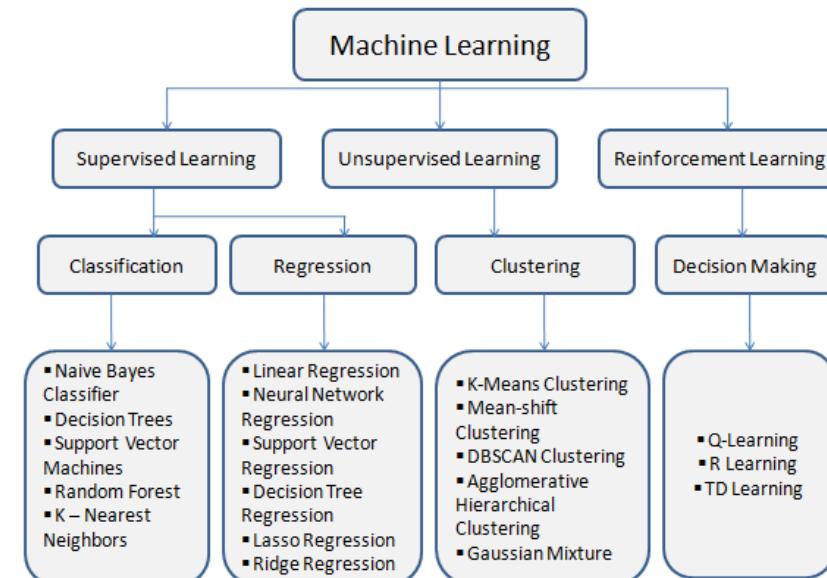
APPRENTISSAGE SUPERVISÉ VS NON-SUPERVISÉ

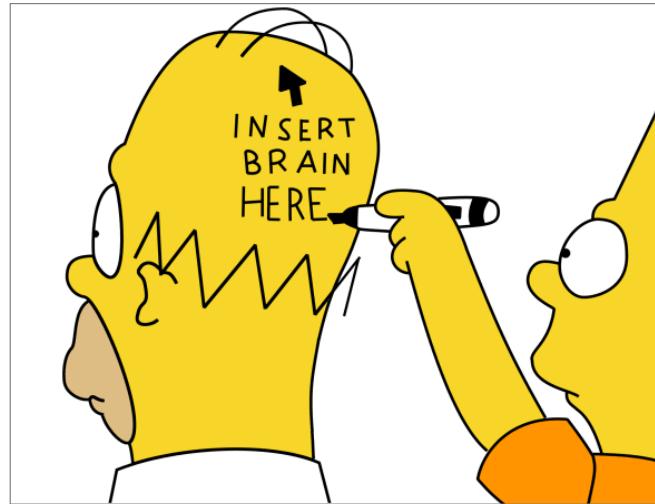
- Apprentissage supervisé :
 - Nécessite un jeu d'entraînement **X**, **y**
 - **X : prédicteurs**
 - **y : variable à prédire**
- Apprentissage non supervisé :
 - Nécessite un jeu d'entraînement X
 - Application principale : le clustering
 - Exemple : classer des situations météo en groupes homogènes



LES BASES : LE MACHINE LEARNING

- Un modèle mathématique nécessite un certain nombre de paramètres pour fonctionner
 - Ex : le périmètre P d'un cercle de rayon r s'écrit : $P = 2 \times \pi \times r$
 - Que se passerait-il si on n'avait pas π mais juste plusieurs P et r ? Peut-on déduire une valeur de π ?
- Le ML couvre plusieurs techniques mathématiques/statistiques permettant d'apprendre des paramètres à partir des données
- Le ML "classique" comprend souvent des techniques qui s'appliquent facilement à des données numériques
 - Régression linéaire
 - Régression logistique
 - Arbres de décision
 - ...
- On regarde rapidement deux de ces méthodes





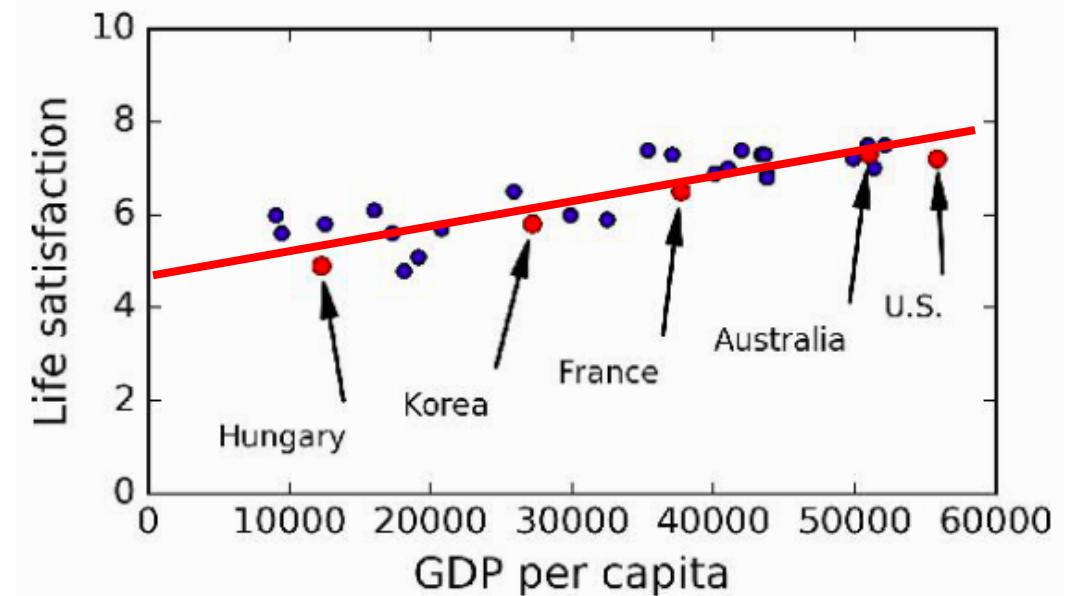
La Régression Linéaire

Une première méthode de IA

LA RÉGRESSION LINÉAIRE

- À partir d'observations d'un phénomène, extraire un comportement "linéaire"
 - Exemple : prévoir le taux de satisfaction d'un pays en fonction de son revenu per capita
- Méthode d'apprentissage supervisé : un jeu d'entraînement X, y
 - X : Le revenu des pays
 - y : le taux de satisfaction
- Objectif : Trouver la droite qui se rapproche le plus du nuage de points

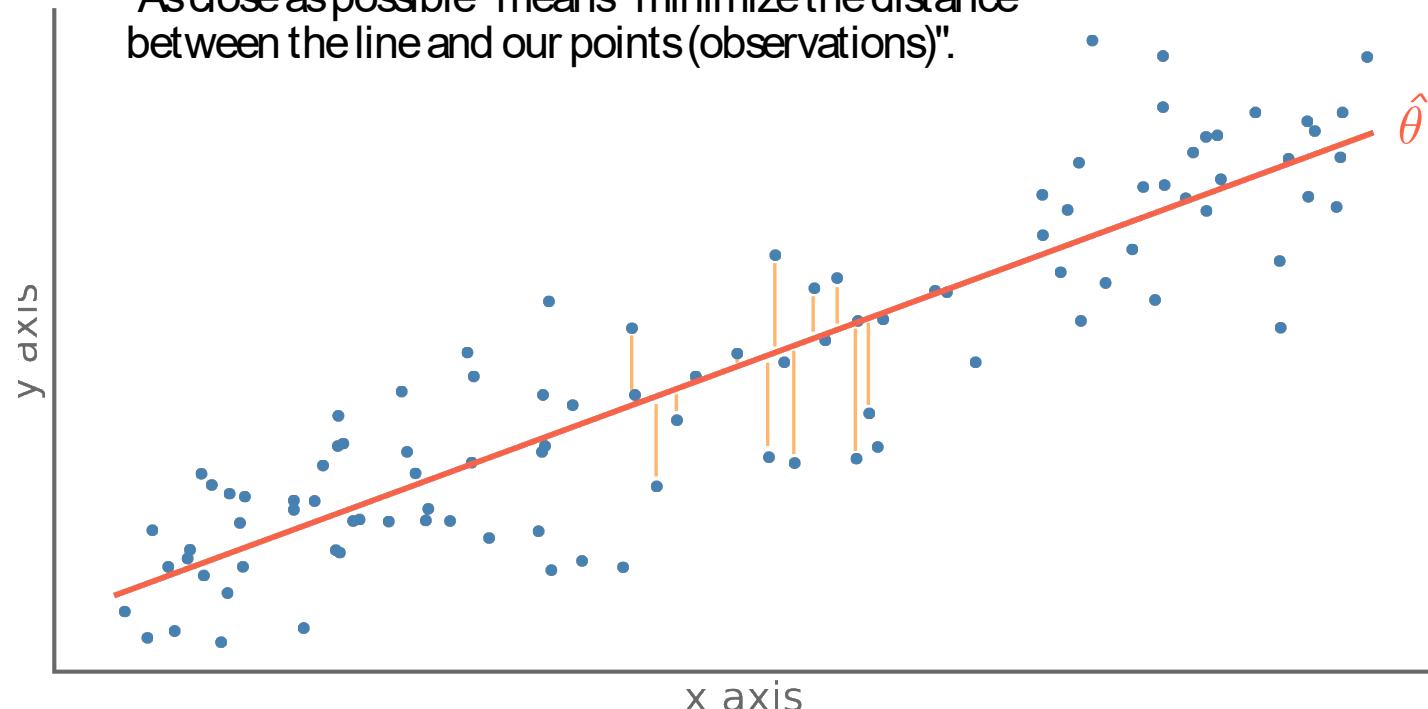
Country	GDP per capita (USD)	Life satisfaction
Hungary	12,240	4.9
Korea	27,195	5.8
France	37,675	6.5
Australia	50,962	7.3
United States	55,805	7.2



LA RÉGRESSION LINÉAIRE

- Le modèle suppose une fonction de prédiction de forme
 - $f(x) = a_1x_1 + \dots + a_px_p + b = a \cdot x + b$
- L'apprentissage consiste à calculer les coefficients a et b qui minimisent l'erreur de prédiction (coût)
- Mais comment définir le coût ?

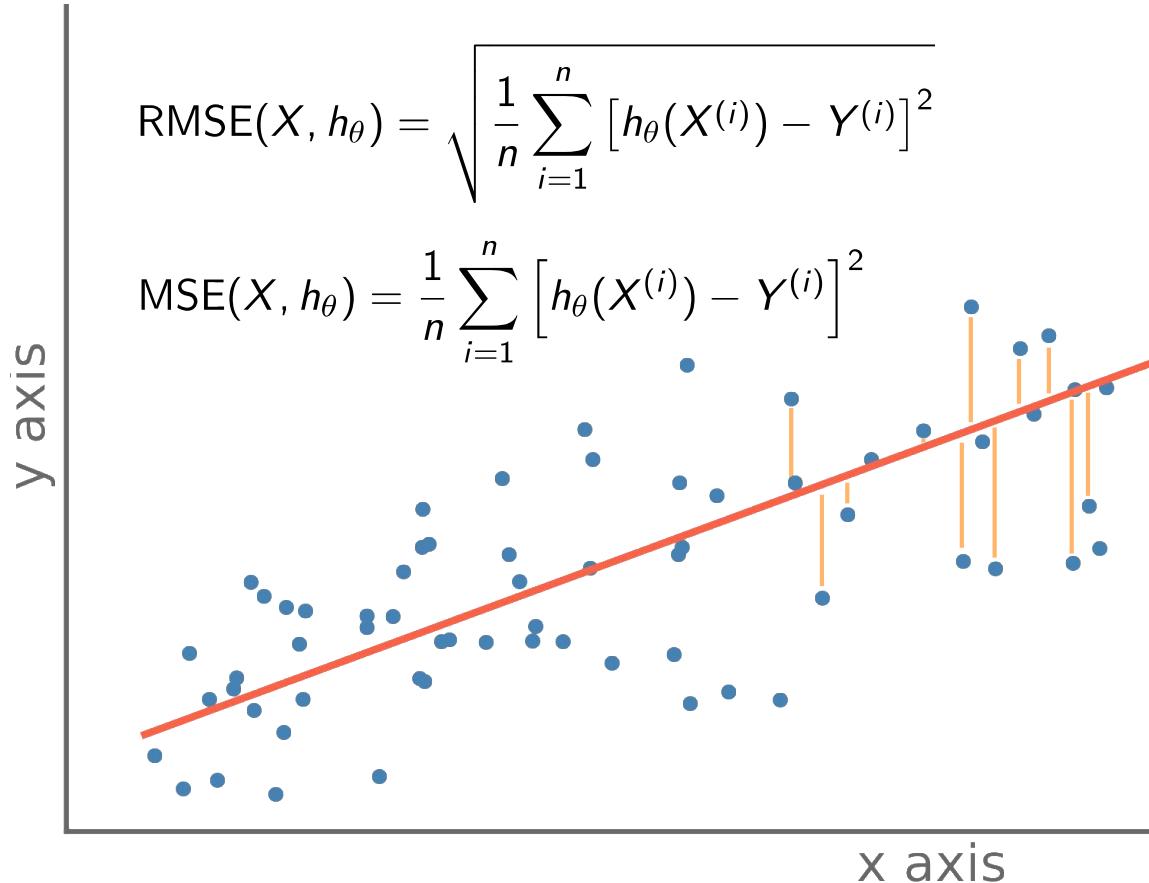
"As close as possible" means "minimize the distance between the line and our points (observations)".



LA FONCTION DE COÛT (LOSS)

- C'est l'écart moyen entre les prédictions et la vérité terrain

$$MAE(X, h_{\theta}) = \frac{1}{n} \sum_{i=1}^n |h_{\theta}(X^{(i)}) - Y^{(i)}|$$



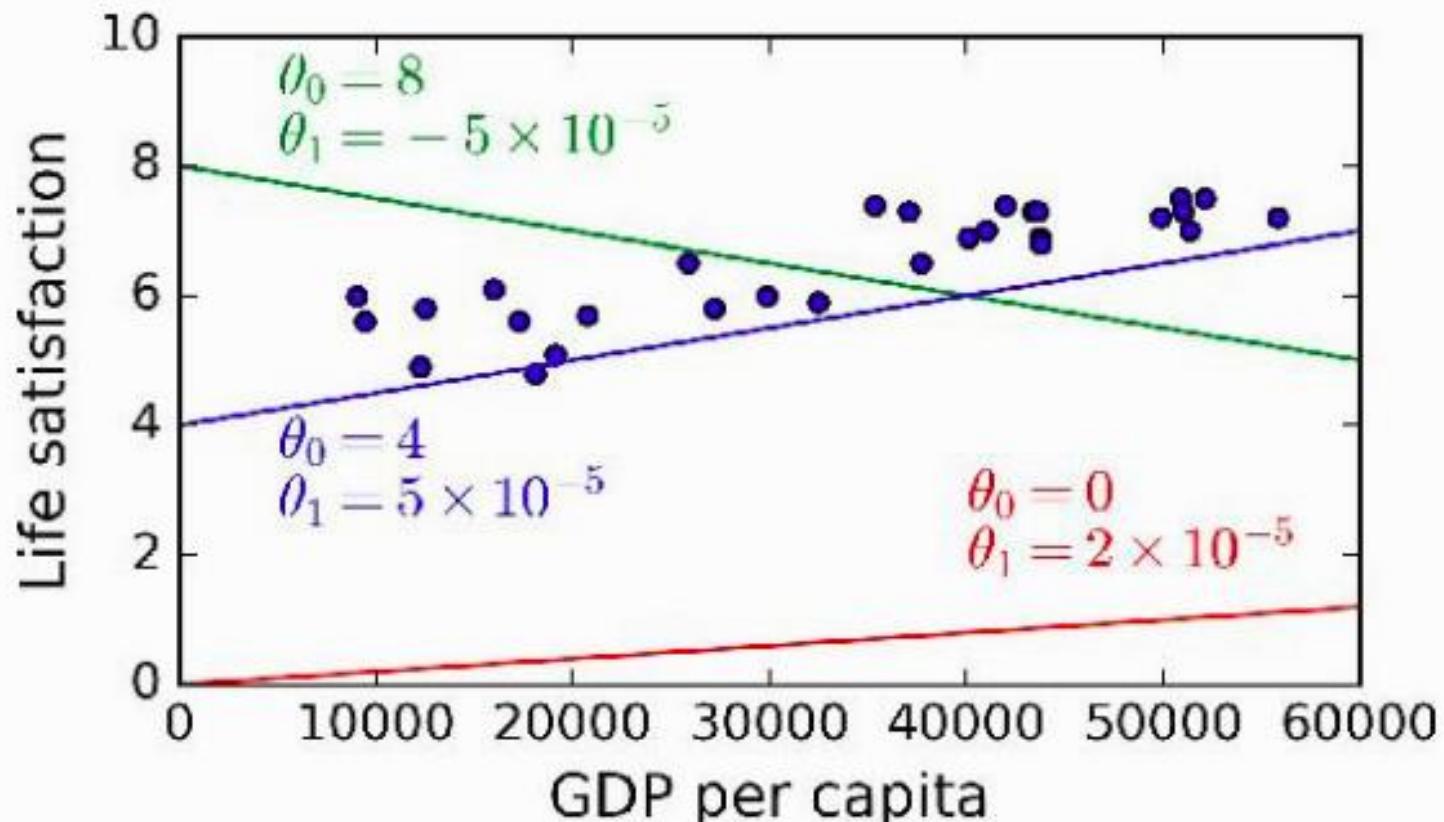
RMSE: Root Mean Square Error
Erreur quadratique moyenne

MSE: Mean Squared Error
Moyenne du carré des erreurs

MAE : Mean Absolute Error
Erreur absolue moyenne

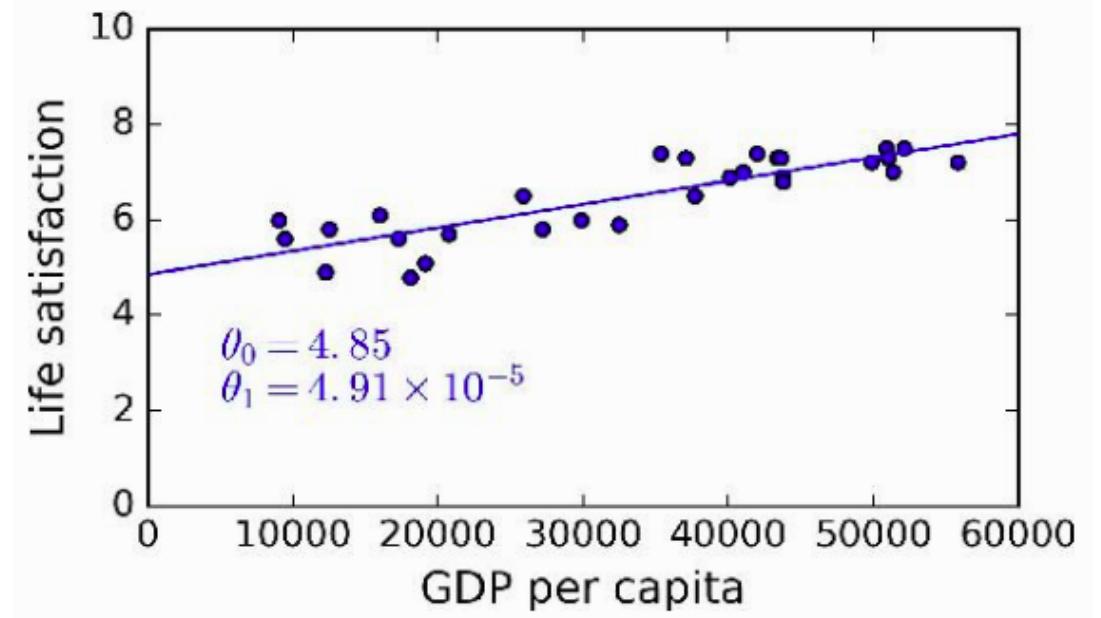
EXEMPLE : EST-CE QUE L'ARGENT REND HEUREUX ?

- Ce modèle a deux paramètres, θ_0 et θ_1 .
- On peut créer une formule simple et essayer plusieurs combinaisons
 - $satisfaction = \theta_0 + \theta_1 \times GDP_per_capita$

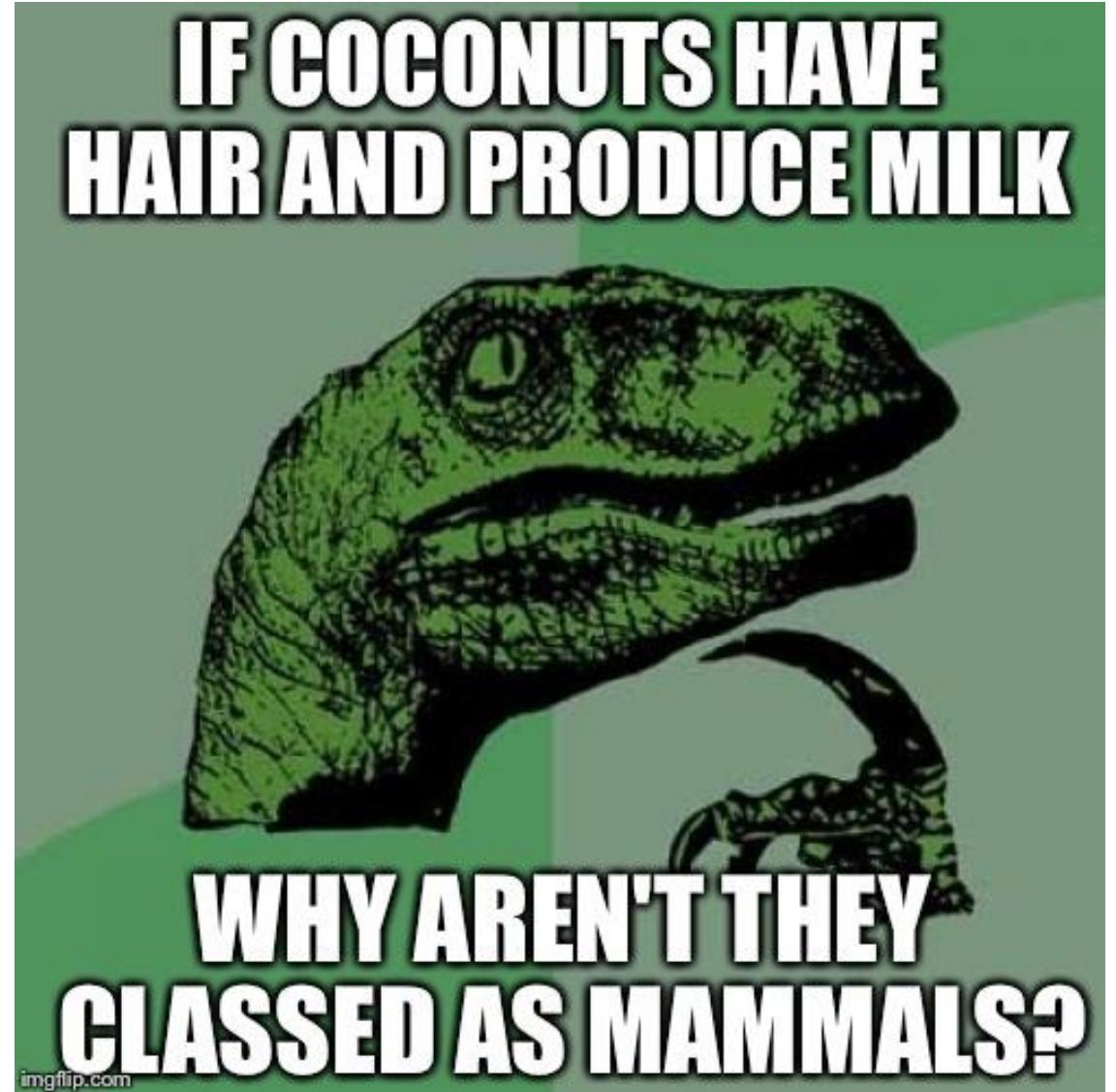


EXEMPLE : EST-CE QUE L'ARGENT REND HEUREUX ?

- Le modèle qui mieux correspond aux données est
 - $\theta_0 = 4.85$
 - $\theta_1 = 4.91 \times 10^{-5}$
- Grâce à ce modèle, on peut essayer d'estimer la satisfaction de la population de Chypre
 - `GDP_per_capita = 22875 USD`
 - `Estimation = 4.85 + 22875 * 4.91 * 10-5`
 - `Estimation = 5.96`

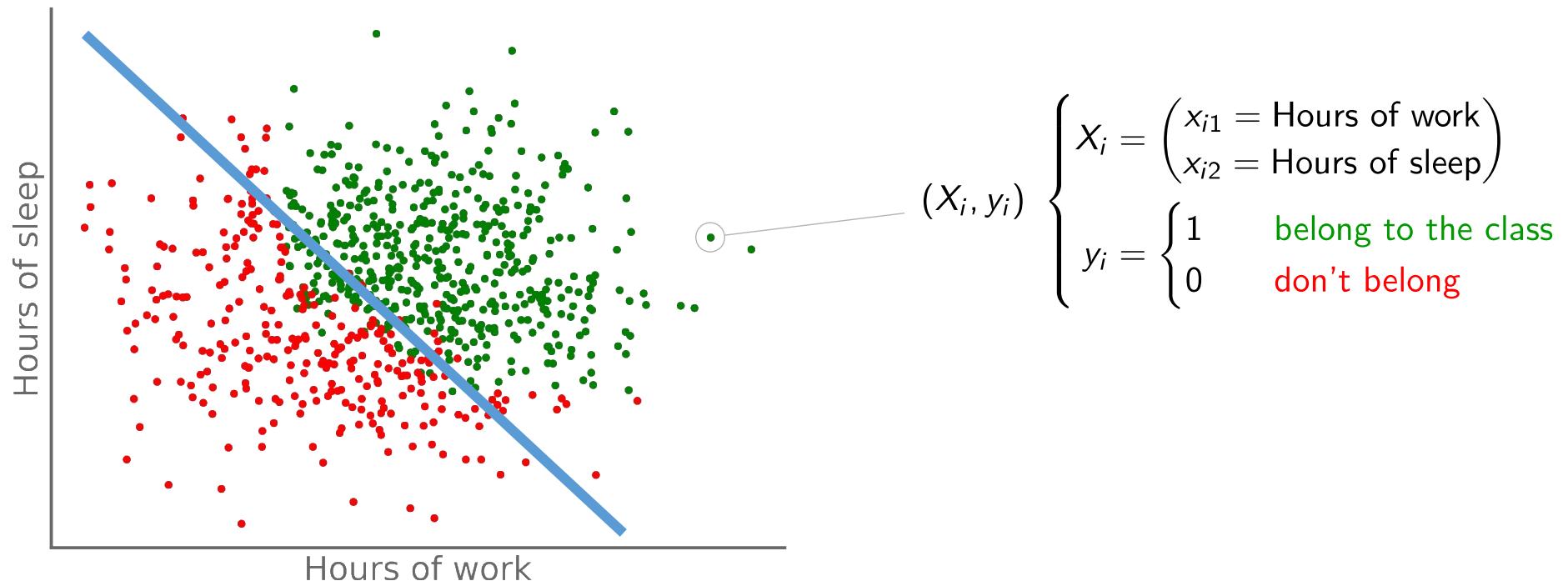


De la Régression à la Classification



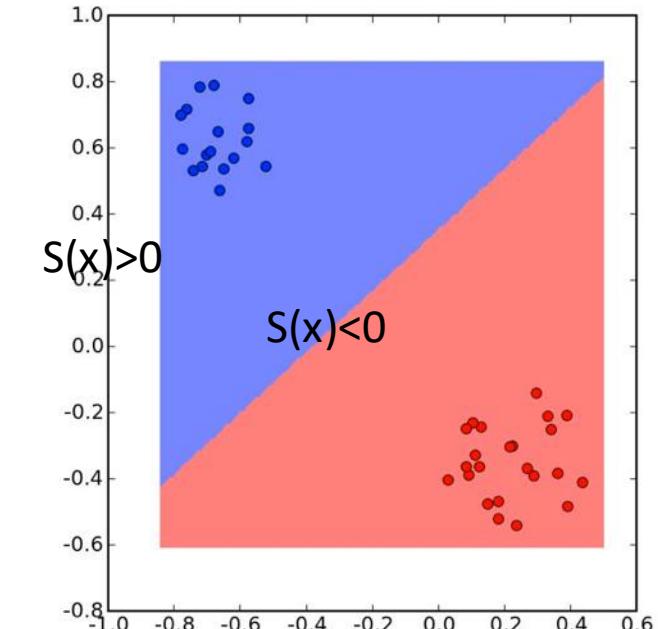
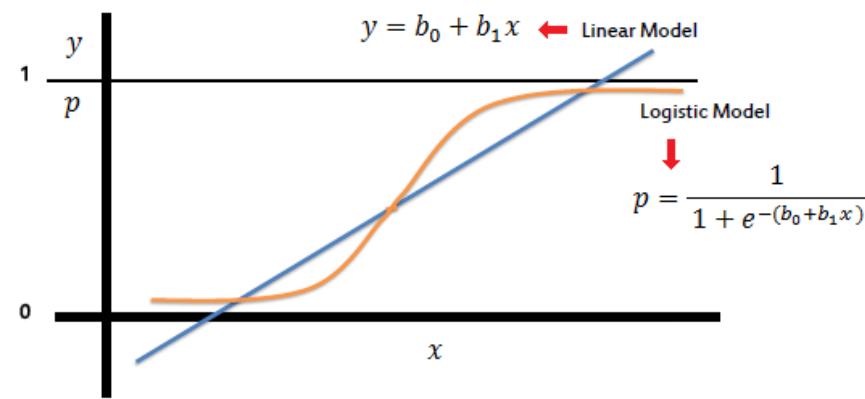
LE CAS DE LA RÉGRESSION LOGISTIQUE

- Au lieu de trouver la ligne qui est la moins éloignée des points, **on cherche la ligne qui mieux les sépare**
 - Une Régression Logistique sert à donner une probabilité d'appartenance à une classe



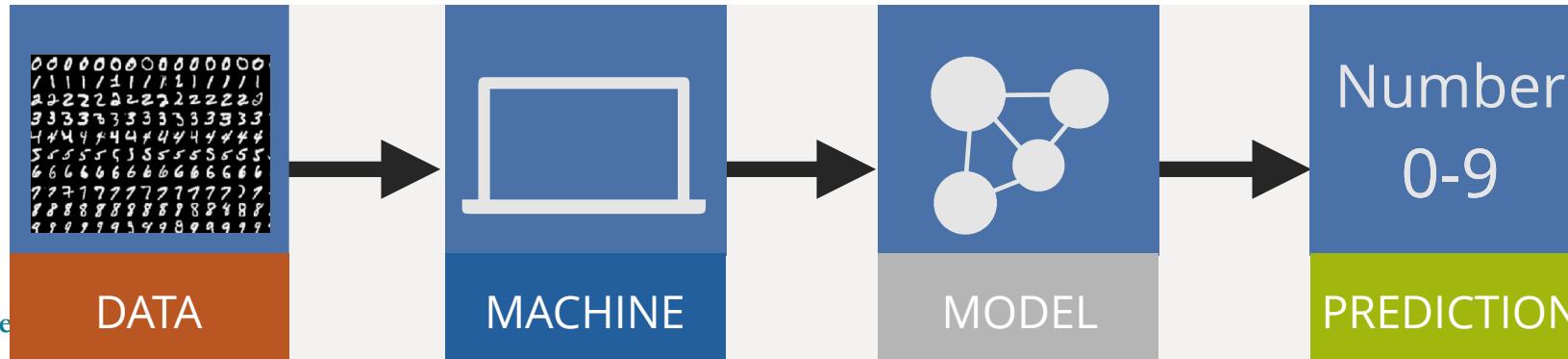
RÉGRESSION LOGISTIQUE

- Comme pour la régression linéaire, on cherche une fonction $S(x) = a_1x_1 + \dots + a_px_p$ qui doit délimiter deux groupes (classes) de données
 - Trouver des coefficients a tel que
 - $S(x)$ est positive lorsque les chances d'appartenir au groupe 1 sont grandes
 - $S(x)$ est négative si la probabilité est grande pour le groupe 0
- Il faut trouver un moyen de "récompenser" les bons scores
 - Une fonction d'interpolation $\text{logit}(S) = 1/[1 + \exp(-S)]$ est souvent utilisée pour exprimer cette probabilité



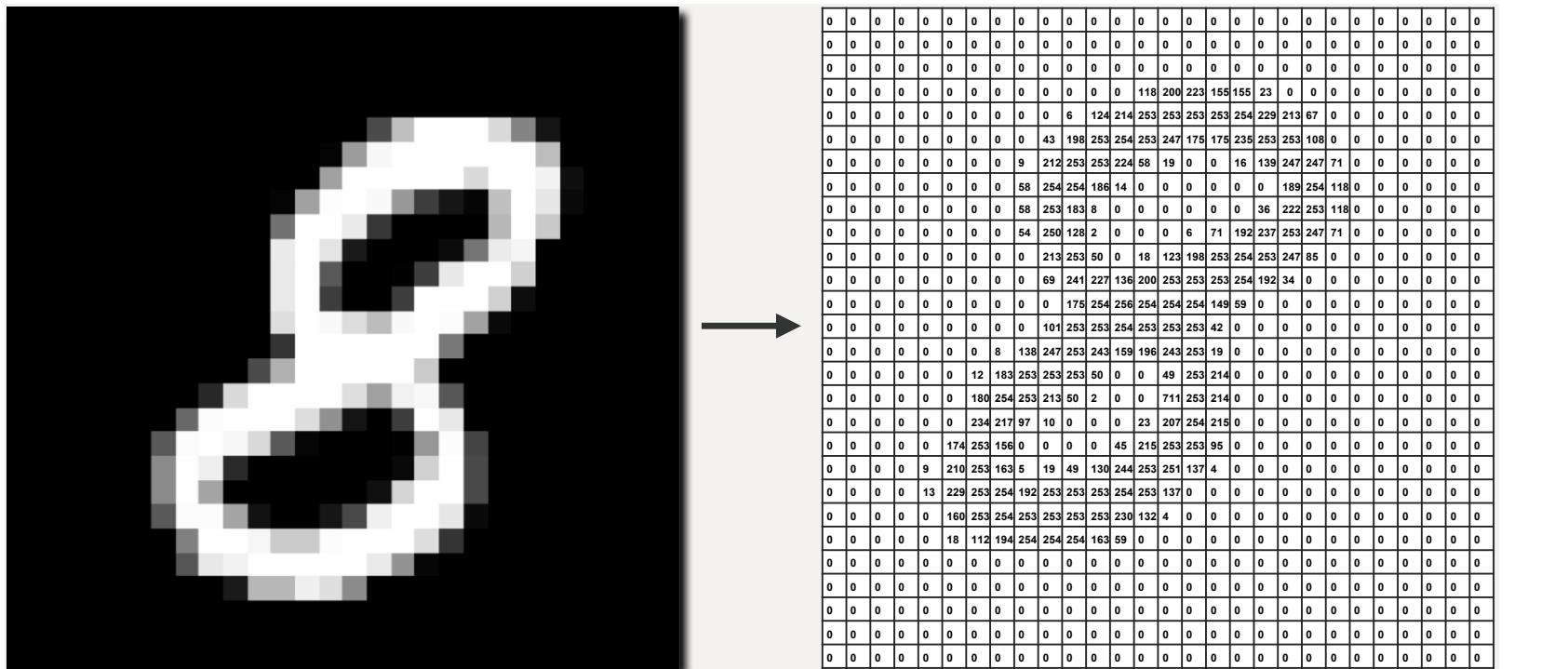
COMMENT ÇA MARCHE DANS LA PRATIQUE ?

- Il est facile de comprendre un système de classification basé sur une ou deux variables
 - Comment travailler avec multiples variables, comme dans le cas d'une image ?
- Exemple : le dataset MNIST
 - Ensemble de chiffres écrits à la main
- Objectif : identifier le chiffre
 - 0 à 9



MISE EN ROUTE

- Les chiffres sont des images, chaque pixel a une valeur numérique

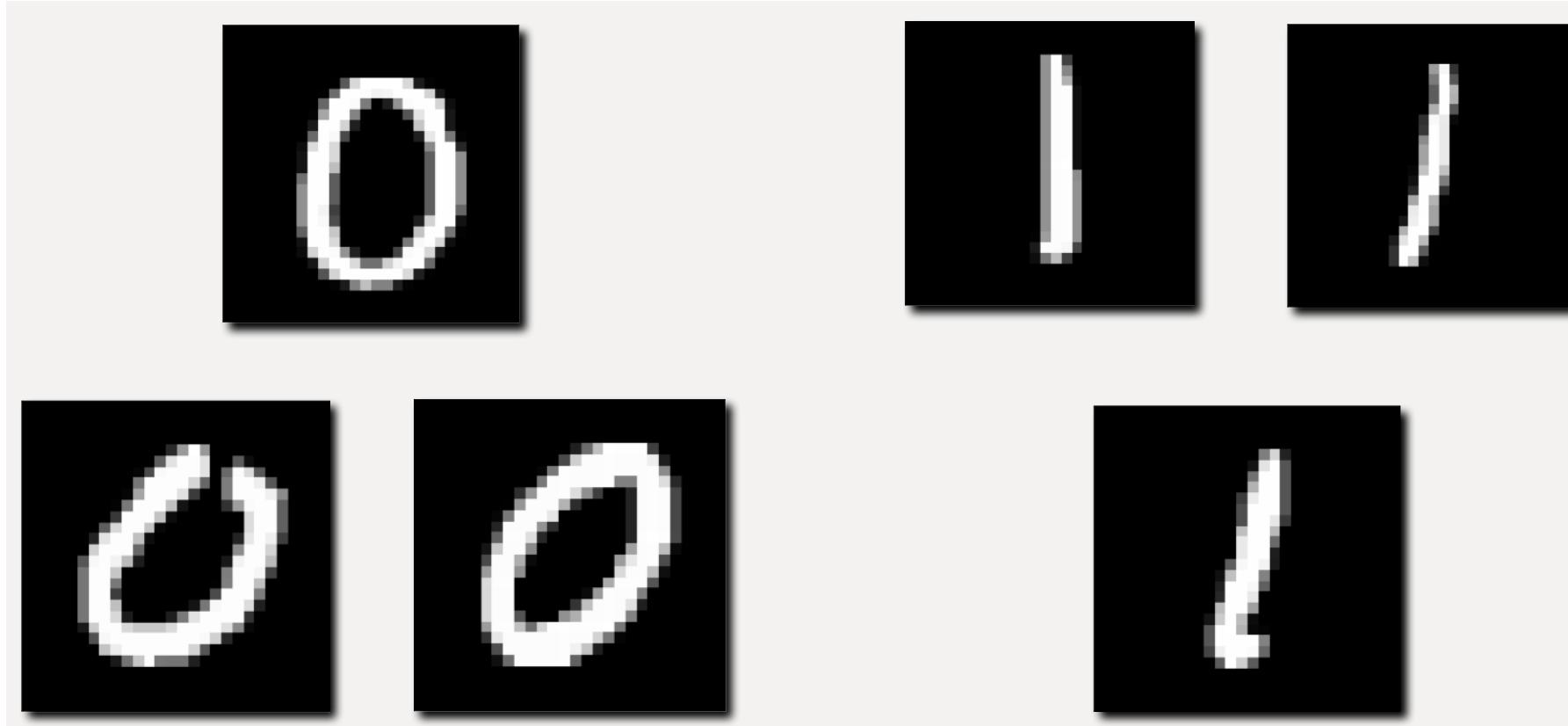


= X

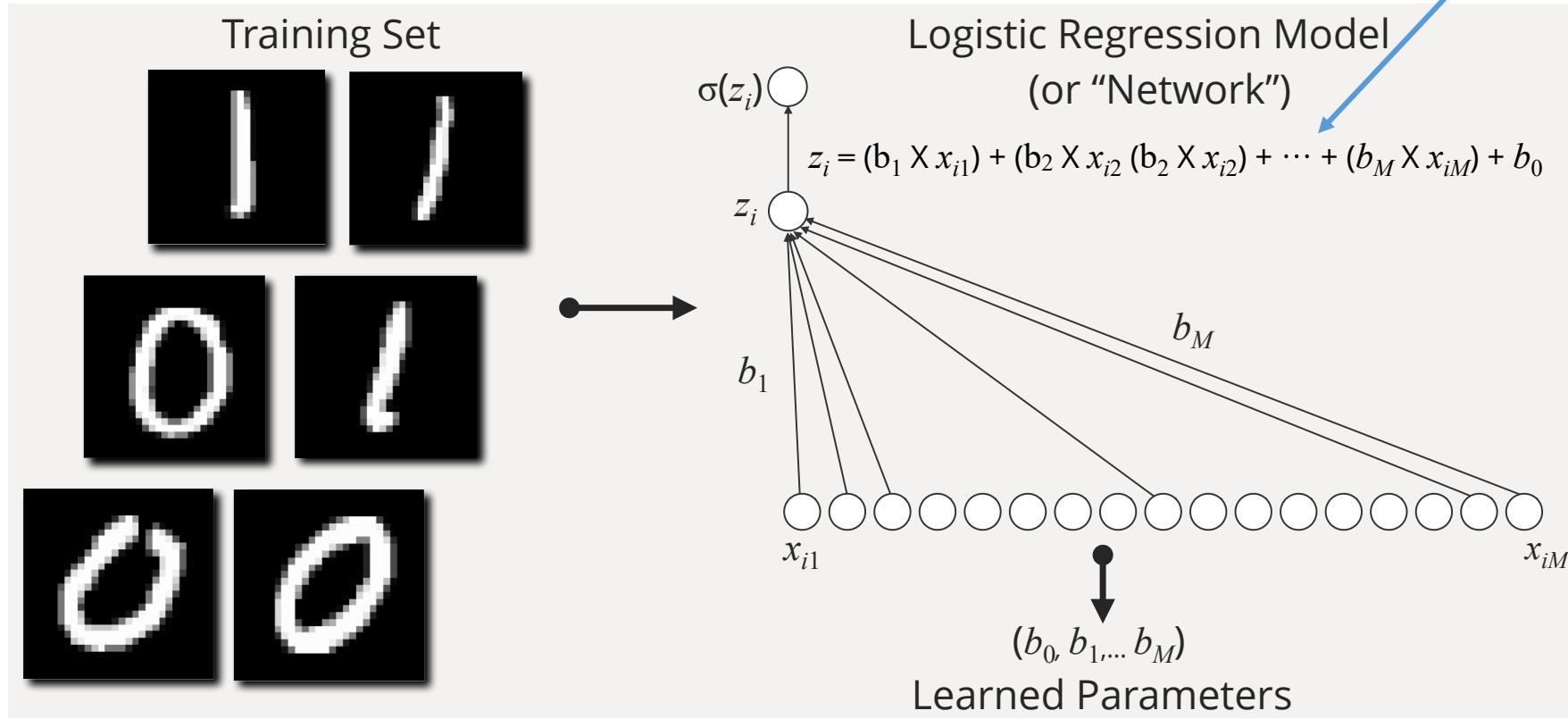
MNIST Dataset of Handwritten Digits (Images)

Yann LeCun (Courant Institute, NYU) and Corinna Cortes (Google Labs, New York) CC-by-SA 3.0
<http://yann.lecun.com/exdb/mnist/>

LE CAS SIMPLE : SORTIE BINAIRE



APPRENTISSAGE DU MNIST

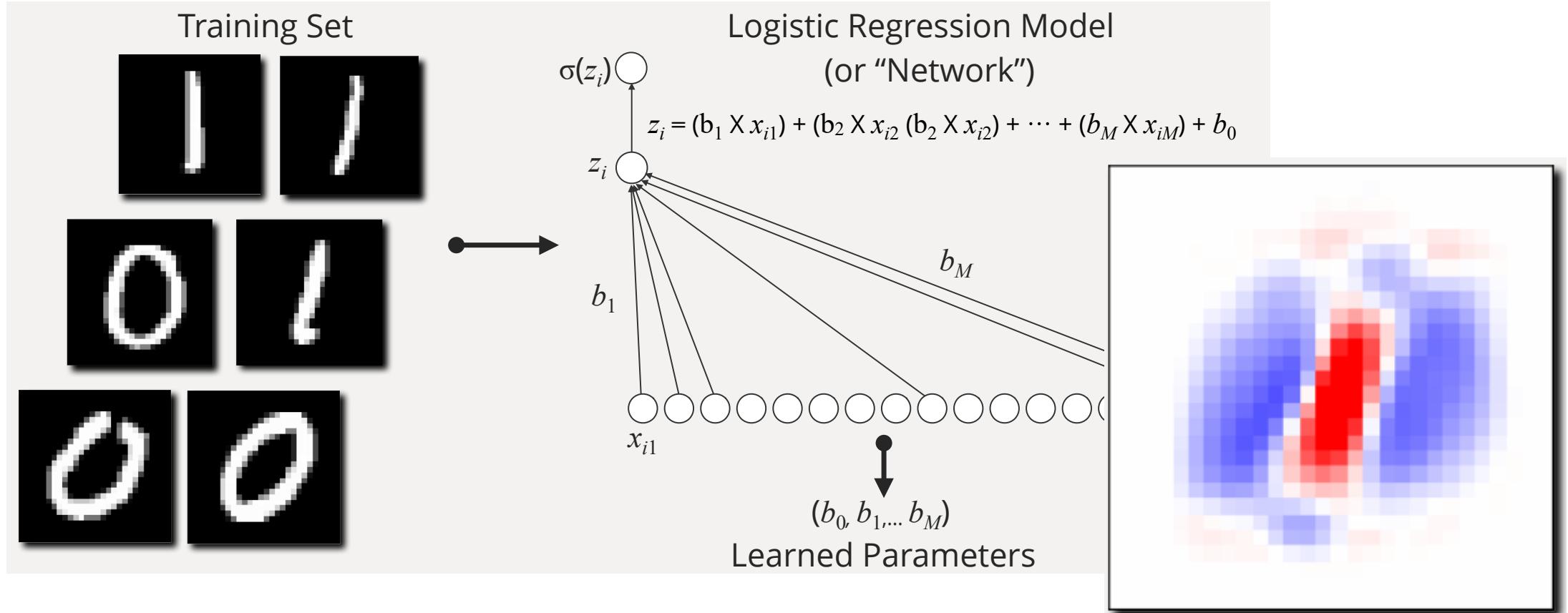


C'est la même formule de la régression linéaire !!!



Chaque position de la matrice est étiquetée avec une valeur négative (0), positive (1) ou rien. Plus on a des 0 ou 1, plus la "case" est fortement marquée (poids de b)

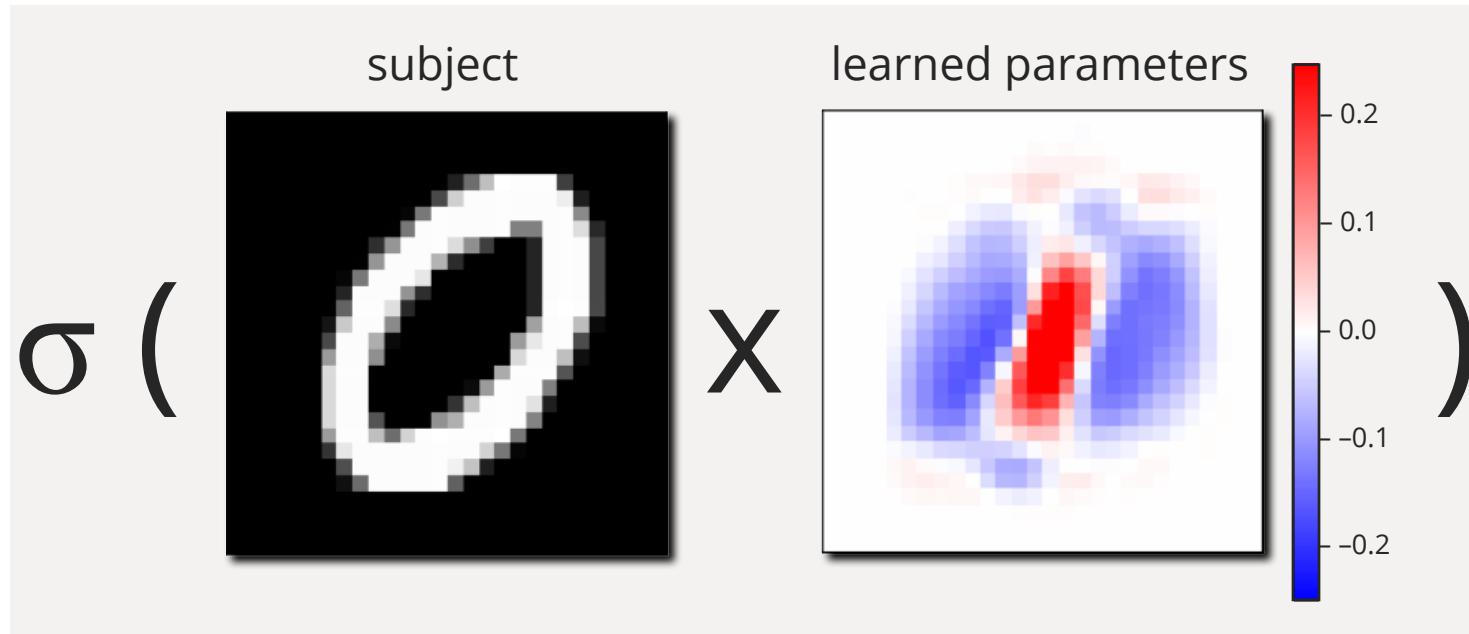
APPRENTISSAGE DU MNIST



Chaque position de la matrice est étiquetée avec une valeur négative (0), positive (1) ou rien. Plus on a des 0 ou 1, plus la "case" est fortement marquée (poids de b)

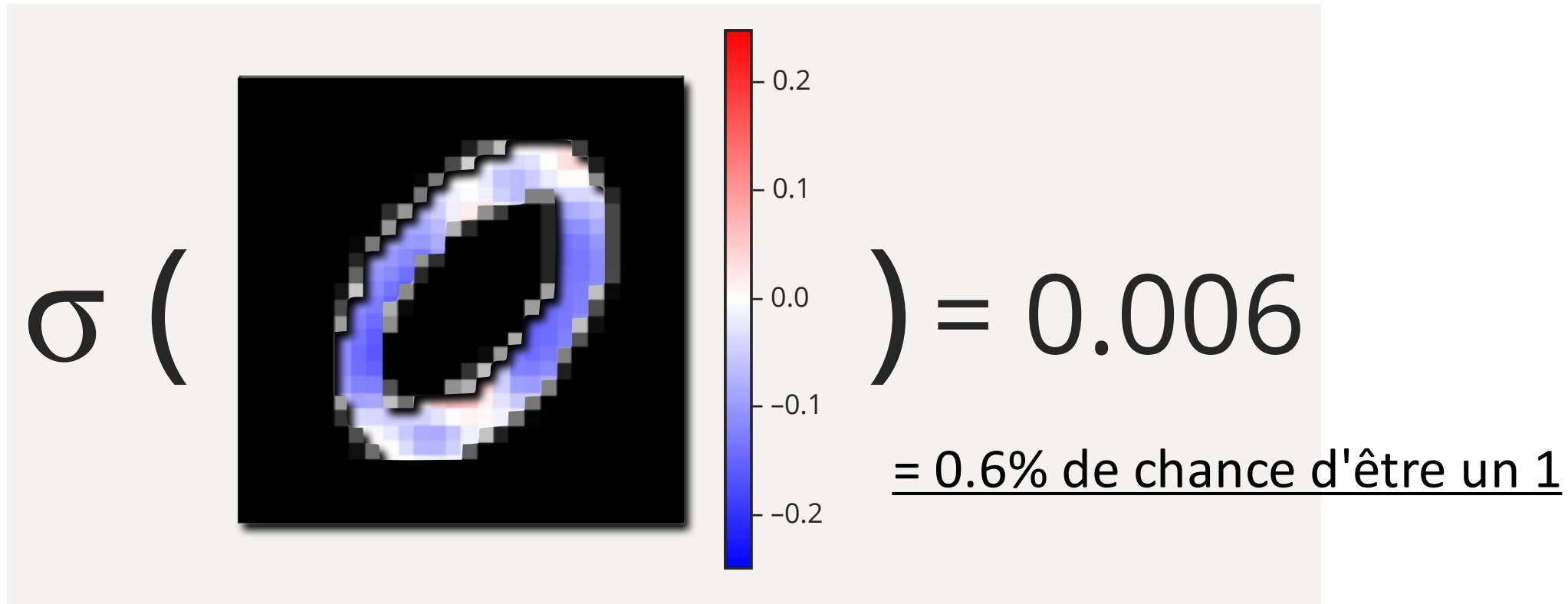
QUELLE EST LA PROBABILITÉ D'ÊTRE UN ZÉRO ?

- Pour un chiffre donné, on compare sa matrice avec la matrice entraînée :



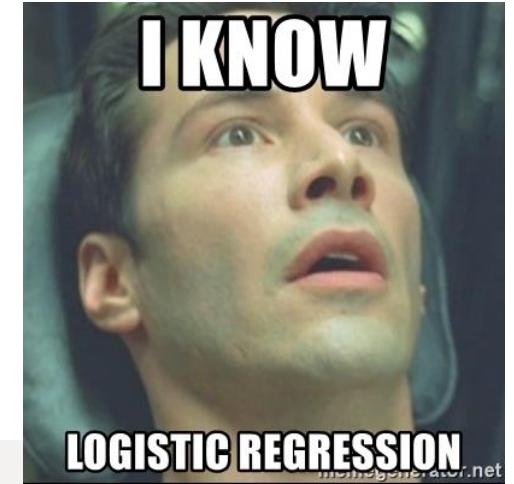
PROBABILITÉ D'UN ZÉRO

- La "superposition" donne une note pour les parties communes

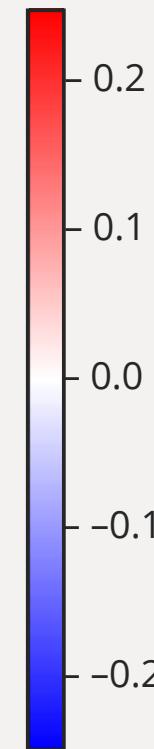
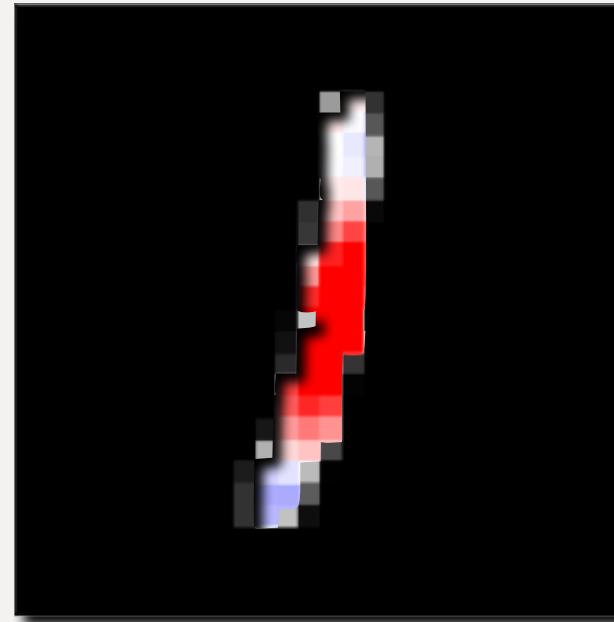


PROBABILITÉ D'UN UN

- La "superposition" donne une note pour les parties communes



$\sigma ($



) = 0.991

= 99,1% de chance d'être un 1

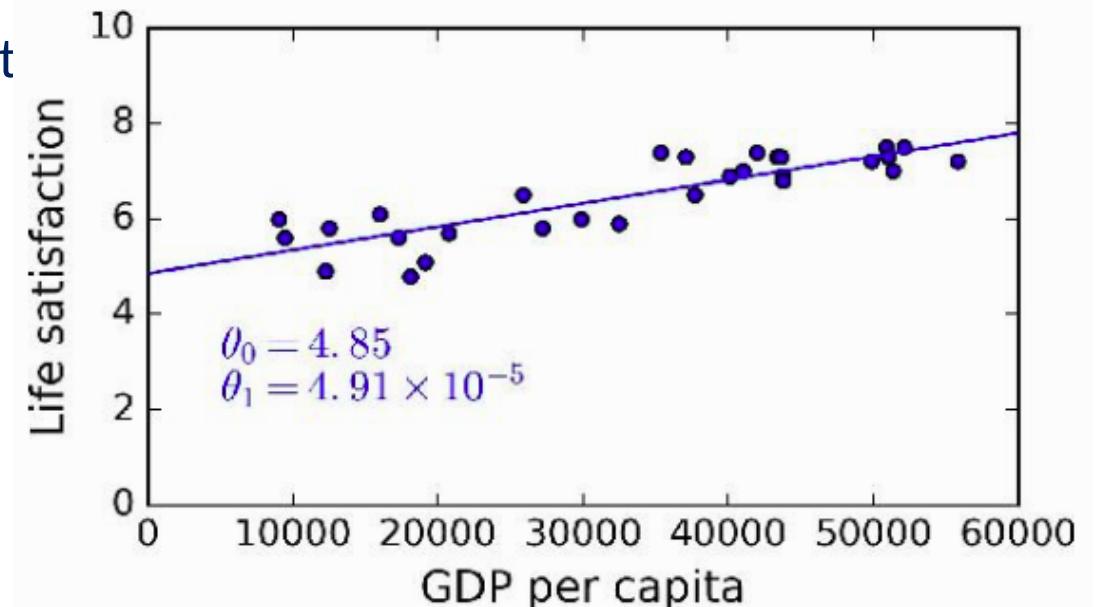
Comment entraîner un modèle ?

Un peu de jargon technique



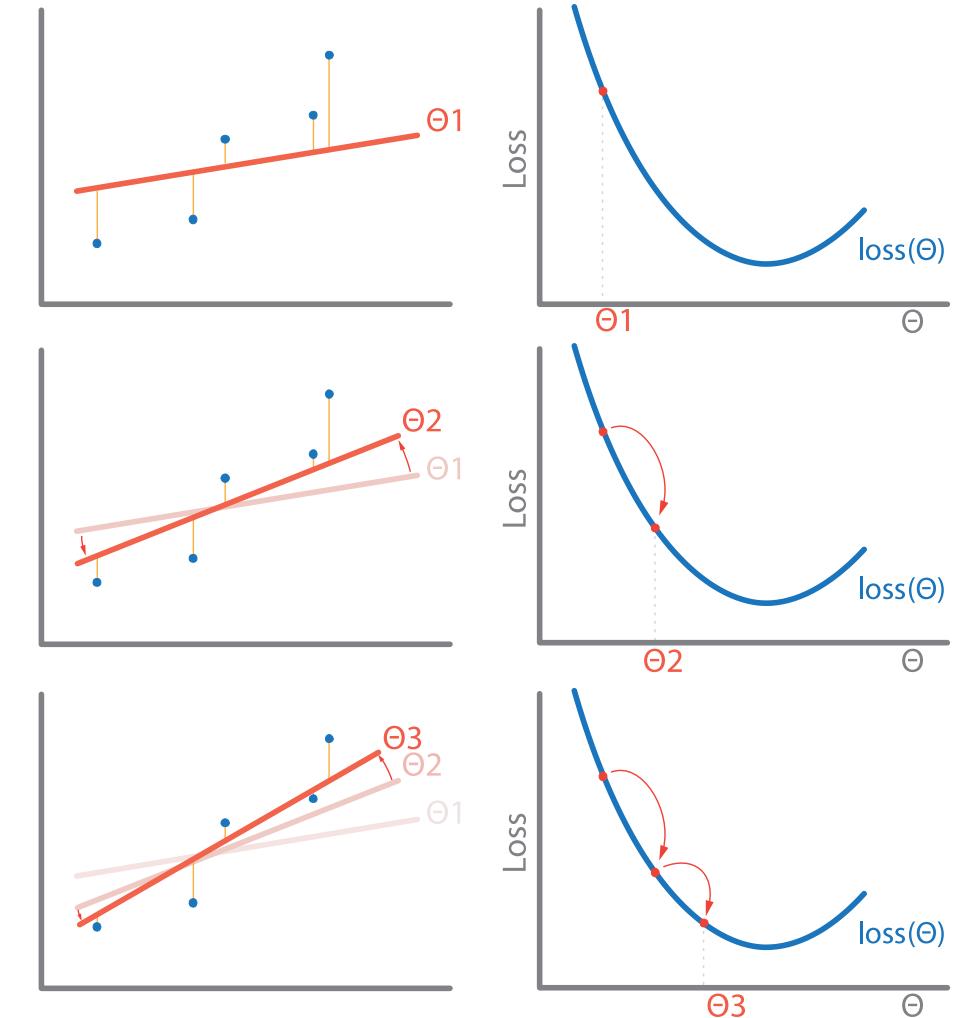
RAPPEL : EST-CE QUE L'ARGENT REND HEUREUX ?

- Le modèle qui mieux correspond aux données avait
 - $\theta_0 = 4.85$
 - $\theta_1 = 4.91 \times 10^{-5}$
- Mais comment trouver les bons paramètres ?
 - Tester **plein de combinaisons de θ_0 et θ_1**
- Problème : complexité $O(n^3)$



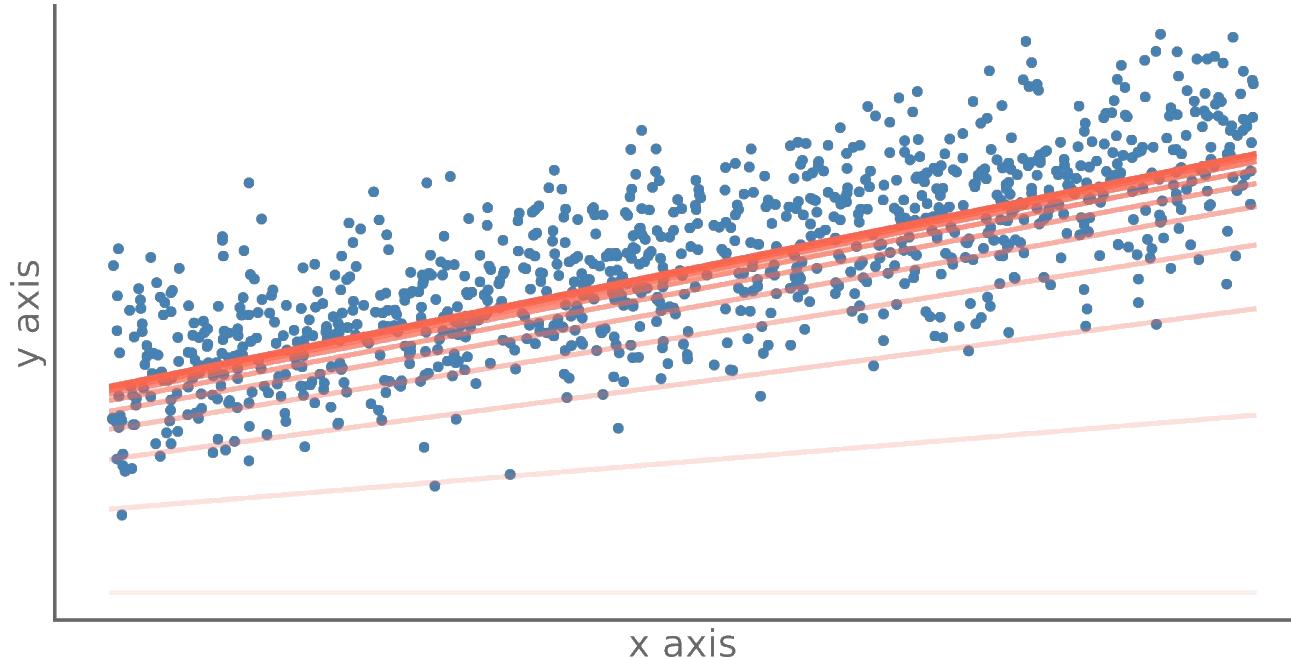
AVANCER EN FAISANT FACE À LA DESCENTE

- La descente de gradient permet de prendre l'élan tant que la pente nous est favorable
 - Avancer par des petits sauts, répéter plusieurs fois



EXEMPLE DE DESCENTE DE GRADIENT

- La solution itérative passe par $\Theta \leftarrow \Theta - \eta \cdot \nabla_{\Theta} MSE(\Theta)$ où η est le taux d'apprentissage (**learning rate**)



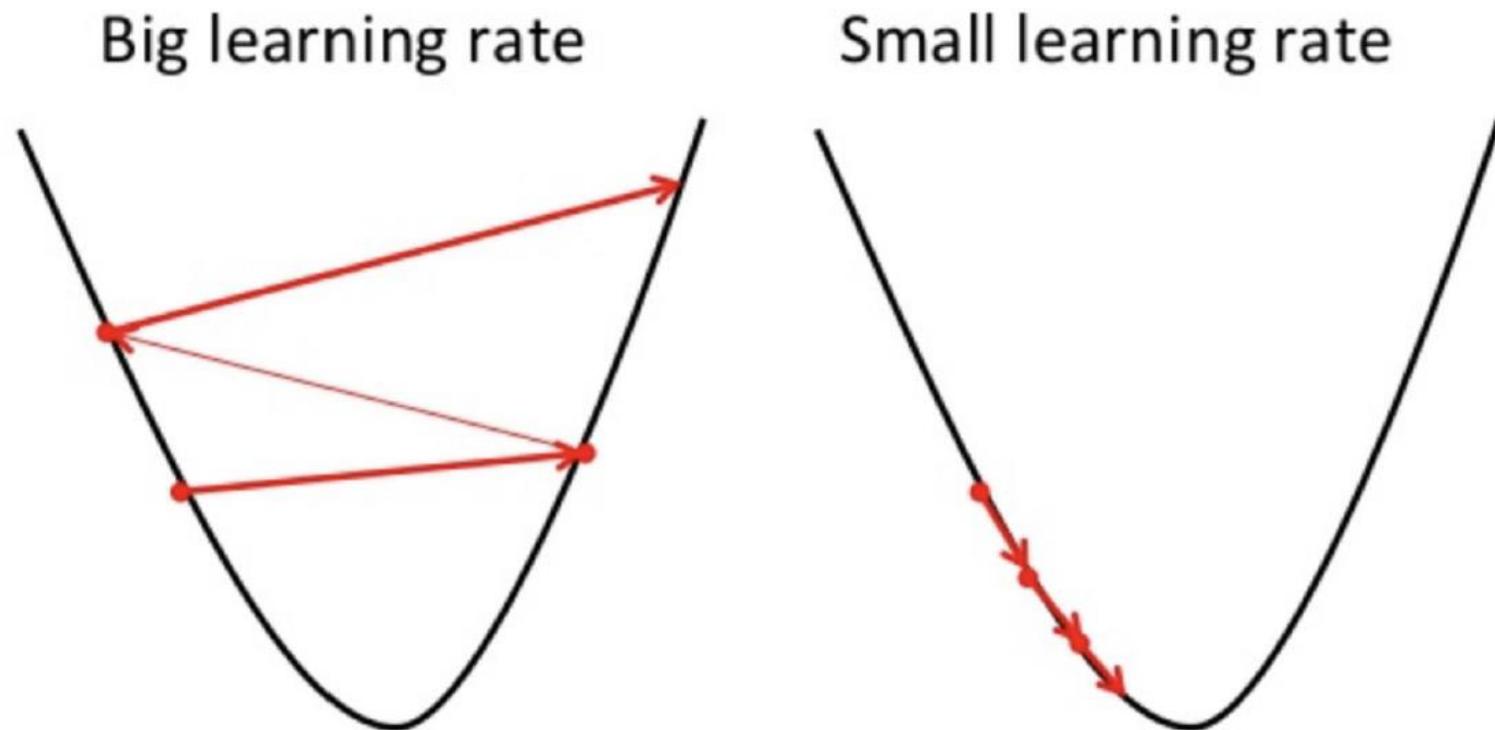
#i	Loss	Gradient	Theta
0	+12.481	-6.777	-3.388
20	+4.653	-4.066	-2.033
40	+1.835	-2.440	-1.220
60	+0.821	-1.464	-0.732
80	+0.455	-0.878	-0.439
100	+0.324	-0.527	-0.263
120	+0.277	-0.316	-0.158
140	+0.260	-0.190	-0.095
160	+0.253	-0.114	-0.057
180	+0.251	-0.068	-0.034
200	+0.250	-0.041	-0.020

$$MSE(X, h_{\theta}) = \frac{1}{n} \sum_{i=1}^n \left[h_{\theta}(X^{(i)}) - Y^{(i)} \right]^2$$

$$\nabla_{\theta} MSE(\Theta) = \begin{bmatrix} \frac{\partial}{\partial \theta_0} MSE(\Theta) \\ \frac{\partial}{\partial \theta_1} MSE(\Theta) \\ \vdots \\ \frac{\partial}{\partial \theta_n} MSE(\Theta) \end{bmatrix} = \frac{2}{m} X^T \cdot (X \cdot \Theta - Y)$$

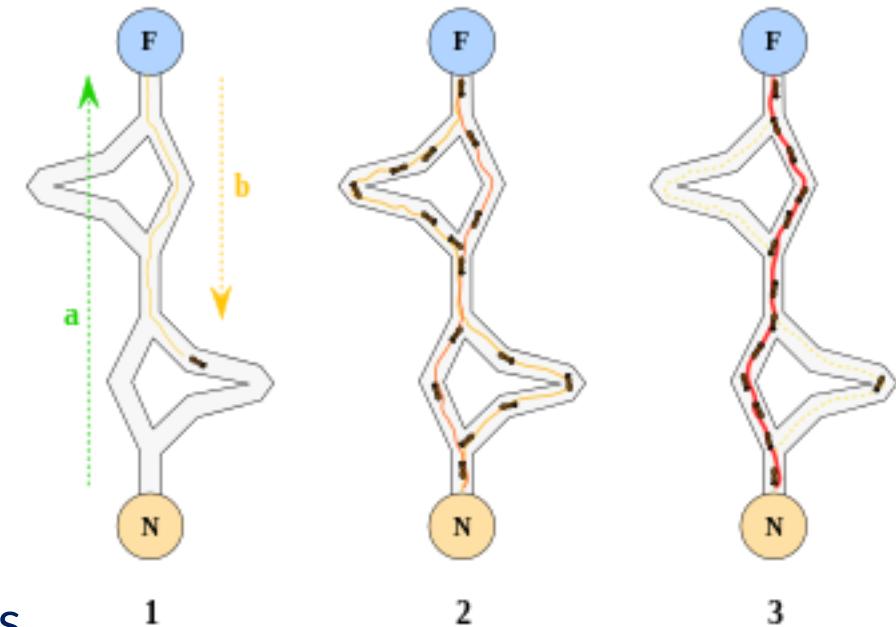
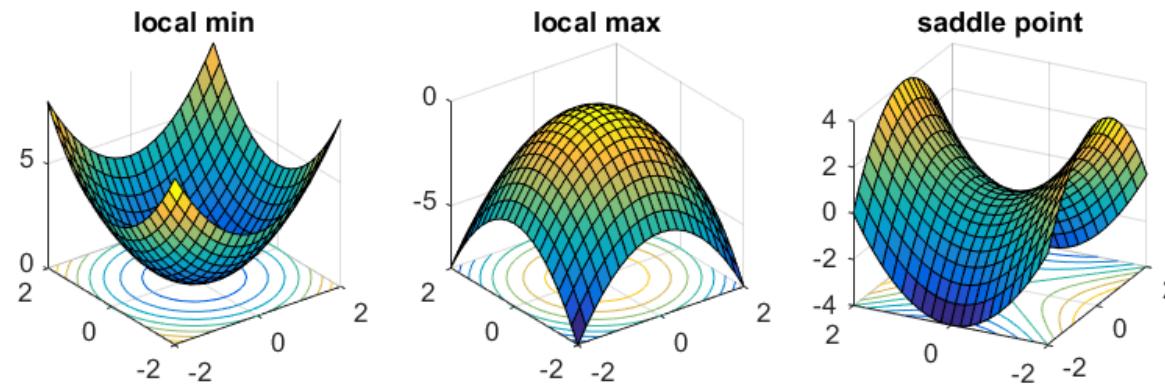
INFLUENCE DU PAS CHOISI (LEARNING RATE)

- Le choix du pas peut influencer le temps de convergence mais aussi le résultat
- Pas trop grand : risque de "sortir" du minimum
- Pas trop petit : risque de rester coincé dans un minimum local



LA NOTION D' EPOCH

- Dans la descente de gradient, on estime le « gradient »
 - échantillon par échantillon ou
 - par mini-batches de quelques échantillons
- Une passe complète sur le jeu de données est un **EPOCH**

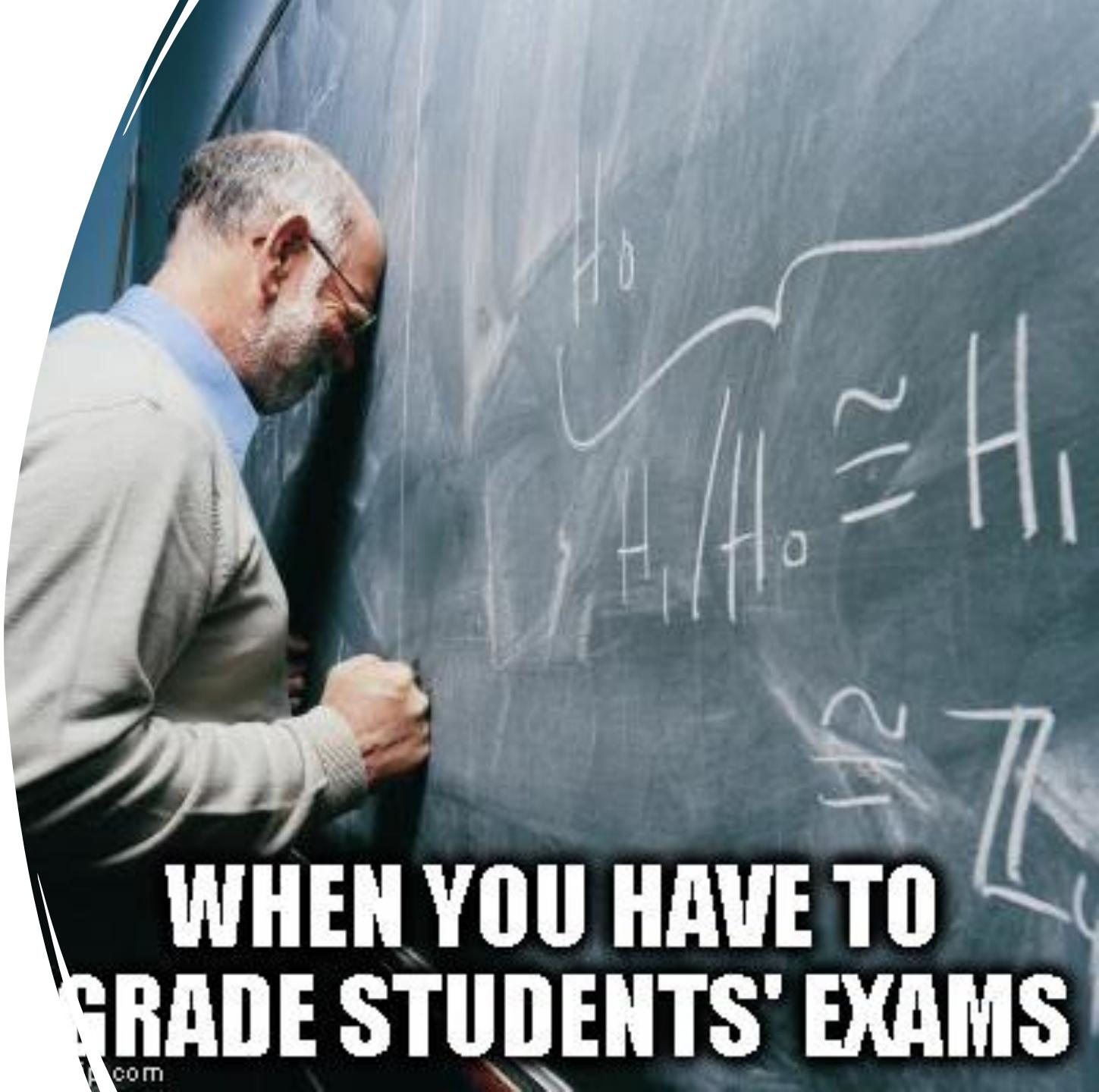


- Souvent un entraînement fait plusieurs passes sur les données
 - Sert à affiner les poids des modèles, un peu comme les traces de phéromones des fourmis

AUTRES HYPERPARAMÈTRES

- Les hyperparamètres sont les "paramètres" fournis par l'utilisateur qui influencent le déroulement de l'entraînement
 - Certains paramètres spécifiques du modèle
 - Pour un réseau de neurones : nombre de couches, nombre de neurones par couche...
 - Le learning rate
 - La taille des mini-batches
 - Le nombre d'epochs
- Comment choisir ces hyperparamètres ? C'est tout un problème d'optimisation
 - Souvent, on procède par des petites adaptations ou par des "balayages"
 - Méthodes GridSearch, etc.

Comment
évaluer un
modèle ?



COMMENT ÉVALUER UN MODÈLE ?

- Dans une classification, on veut savoir si notre modèle fait bien son boulot
- Dans le cas d'une **classification binaire**, on peut faire usage d'une **matrice de confusion**
- Ex : est-ce que la valeur X appartient à la classe "1" ?

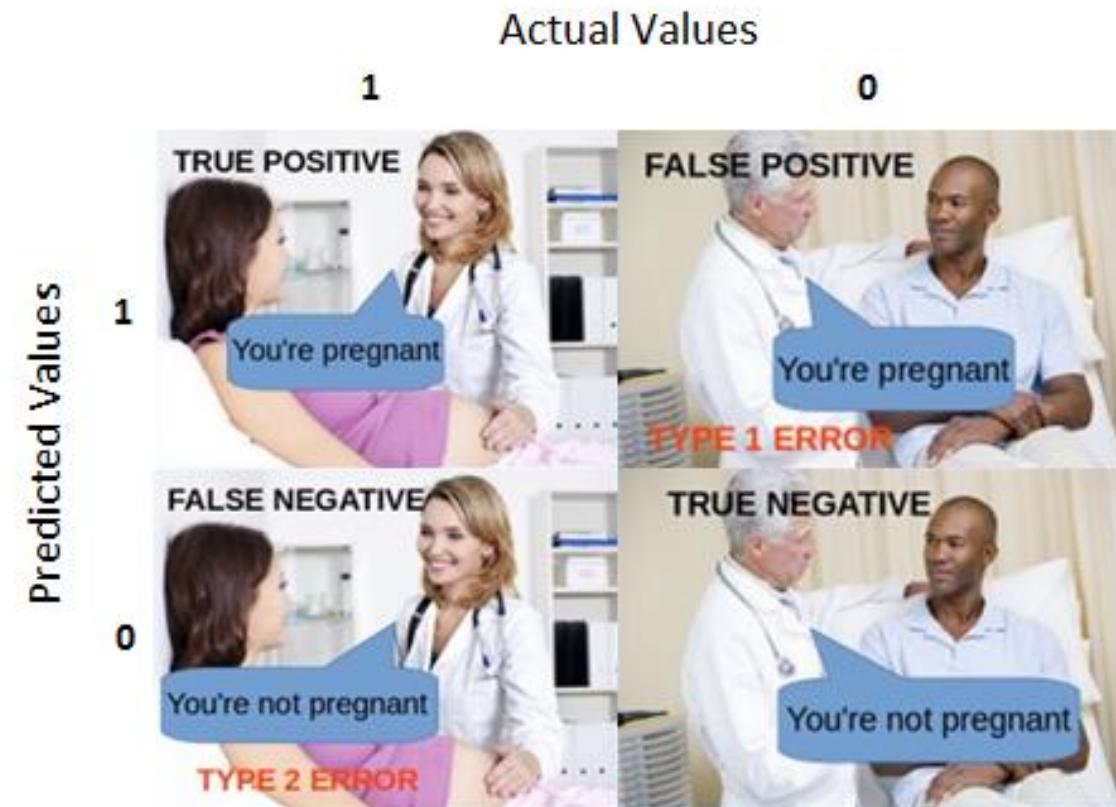
		Classe réelle	
		-	+
Classe prédictive	-	True Negatives (vrais négatifs)	False Negatives (faux négatifs)
	+	False Positives (faux positifs)	True Positives (vrais positifs)

Nom du cas	Abréviation	Description
Vrai positif	VP	La donnée appartient à la classe "1" et a été prédite comme tel
Vrai négatif	VN	La donnée appartient à la classe "0" et a été prédite comme tel
Faux positif	FP	La donnée appartient à la classe "0" mais a été prédite comme classe "1" -> Erreur type 1
Faux négatif	FN	La donnée appartient à la classe "1" mais a été prédite comme classe "0" -> Erreur type 2

IMPACT D'UNE ERREUR

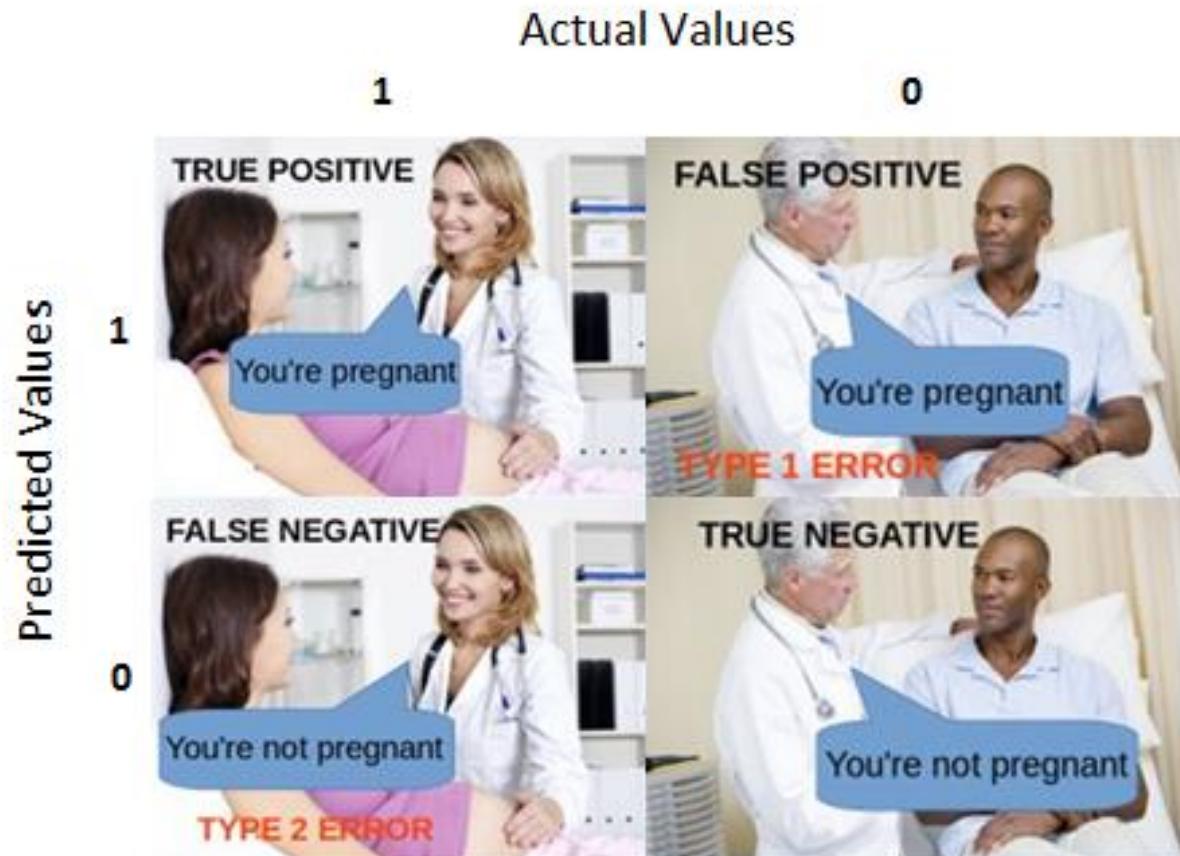
- On voit qu'il y a deux types d'erreur dans une classification binaire
- Erreur type 1 – faux positif
- Erreur type 2 – faux négatif

- En général, une erreur type 1 est moins grave qu'un type 2
- Ex : recherche d'une maladie
 - En cas de faux-positif, les tests cliniques / médecin permettront de vérifier la présence de la maladie
 - En cas de faux-négatif, le patient est renvoyé sans traitement
- Ex : recherche de défauts dans une chaîne de montage
 - FP -> la pièce est écartée puis sera recyclée/jetée
 - FN -> le produit aura une pièce défectueuse

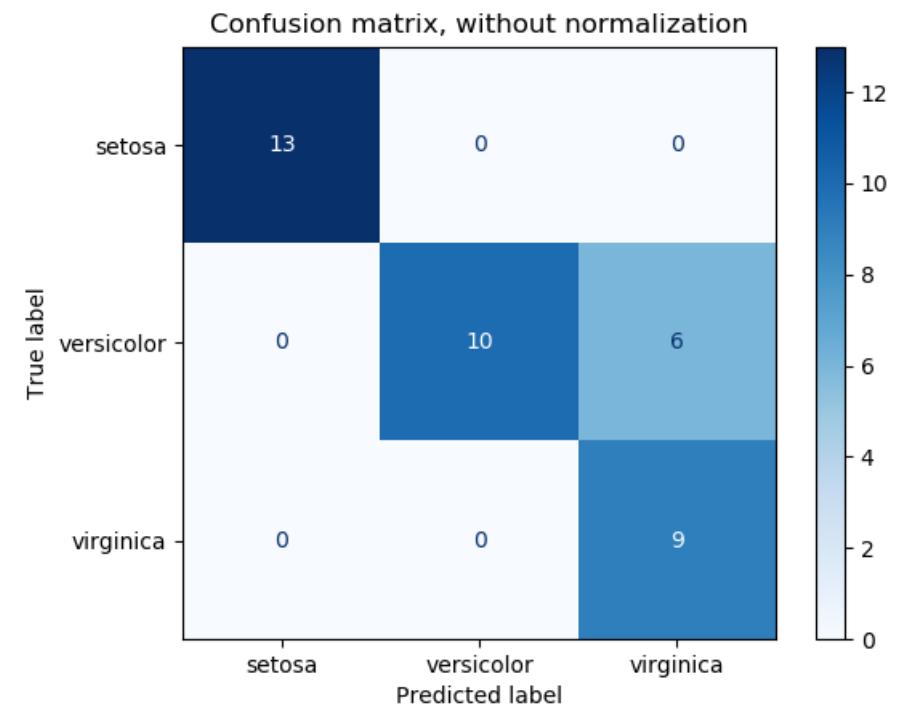


MATRICE DE CONFUSION

Cas binaire (deux classes seulement)



Cas multi-classe



EVALUATION D'UN MODÈLE DE CLASSIFICATION

- Dans le cas d'une classification (régression logistique), comment évaluer un modèle ?
- Plusieurs possibilités
 - L'accuracy (taux de réussite, justesse)
 - Le recall (rappel)
 - Le F1 score
 - Une matrice de confusion
- L'accuracy est l'un des critères permettant d'évaluer les modèles de classification
 - Désigne la proportion des prédictions correctes effectuées par le modèle

$$Accuracy = \frac{\text{nombre de prédictions correctes}}{\text{Nombre total de prédictions}} = \frac{TN + TP}{TN + TP + FN + FP}$$

NE PAS SE LIMITER À L'ACCURACY

- La métrique Accuracy a trois défauts majeurs:

1. Ne donne aucune information sur les erreurs commises
 - L'impact n'est pas le même si c'est type 1 ou type 2
2. Peu informative si les classes sont déséquilibrées
 1. Ex : détection de défauts où seulement 1% des pièces sont défectueuses
 2. Accuracy 99% -> soit il détecte une partie des défauts (type 2), soit il considère que toutes sont bonnes (type 1)
3. Accuracy 100% -> quasiment impossible, à se méfier !
 1. Surapprentissage ?
 2. Données val/test trop proches des données train ?
 3. Des informations qui "fuitent" la classe ?



RAPPEL ET PRÉCISION

- Le rappel (Recall) est l'un des critères permettant d'évaluer la sensibilité du modèle
 - Proportion de la classe X détectée vraiment

$$Rappel = \frac{\text{nombre de vrai positifs}}{\text{Nombre de vrai positifs} + \text{faux négatifs}} = \frac{TP}{TP + FN}$$

- Un grand recall indique que presque tous les éléments de la classe cible ont été détectés
- Plus le recall est grand, moins d'erreurs de type 2 (faux négatifs), qui sont les plus graves
- La précision (Precision) indique la proportion de vrais positifs dans l'ensemble de positifs détectés

$$Precision = \frac{\text{nombre de vrai positifs}}{\text{Nombre de vrai positifs} + \text{faux positifs}} = \frac{TP}{TP + FP}$$

- Permet d'estimer le pourcentage d'erreurs de type 1

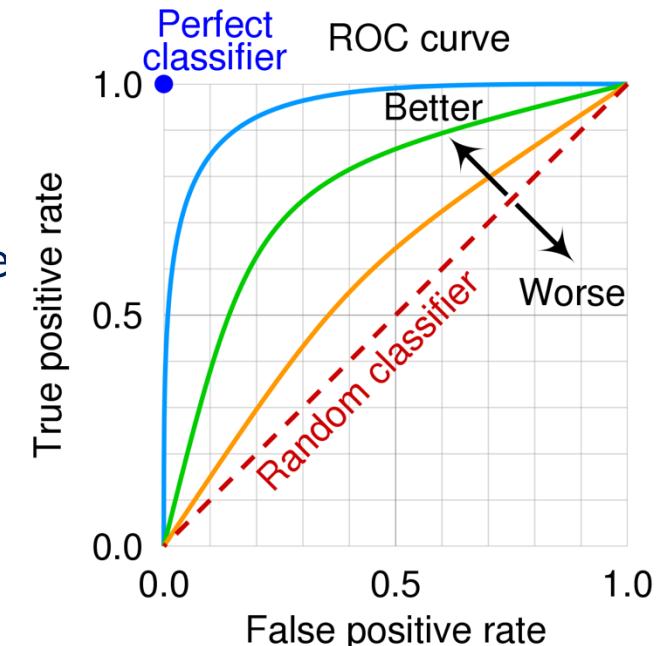
LE SCORE F1

- Le F1-score fait la moyenne harmonique entre precision et recall
 - Permet d'éviter les pièges de l'accuracy
 - Favorise les cas où nous avons une haute precision et haute recall

$$F1 = 2 * \frac{precision * rappel}{precision + rappel}$$

LA COURBE ROC ET L'AUC

- Souvent, les modèles estiment une probabilité d'appartenance à une classe :
 - Ex Titanic : passager 33 -> Survie 76%, Décès 24%
- En générale, la matrice de confusion utilise un seuil 0,5
 - Valeurs supérieures à 0,5 sont classés positivement, sinon négativement
- Lors du déploiement du modèle, on peut choisir des seuils différents en fonction des erreurs à minimiser
- **La courbe ROC** permet de visualiser l'impact des différents seuils
- Pour tracer la courbe ROC on a besoin de :
 - Le taux de vrais positifs (TPR) -> le rappel
 - Le taux de faux positifs (FPR) -> 1-specificité
- Idéalement, la courbe devrait monter rapidement pour avoir un fort TPR et un faible FPR
- L'AUC (Area Under Curve) est la surface sous la courbe
 - Idéalement la valeur serait 1, permet de comparer différents curves ROC indépendamment du seuil



QUELLE MÉTRIQUE FAVORISER ?

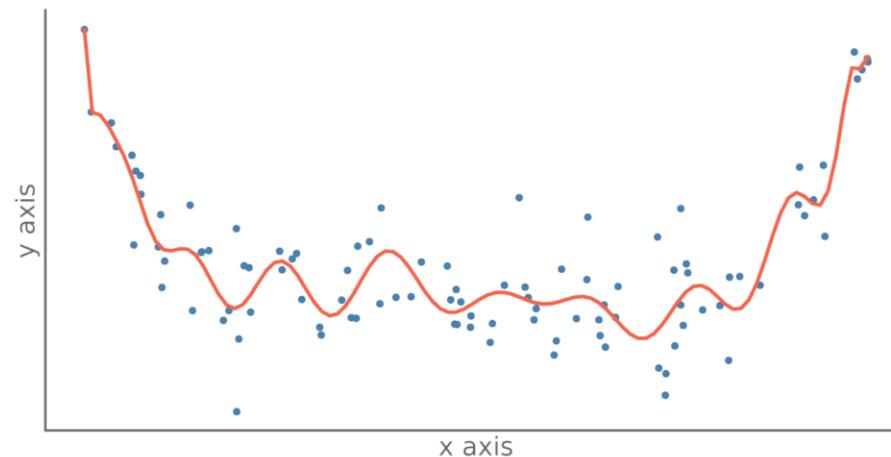
- Tout dépend de l'objectif
- Une messagerie pourra laisser passer quelques spams mais devra éviter au maximum de classer en tant que spam un mail qui n'en n'est pas un (faux negatif)
 - L'utilisateur se chargera d'effacer les spam qui passent le filtre
- Un algorithme de reconnaissance de cancer préférera se tromper et faire un faux positif
 - Le patient aura donc des examens supplémentaires et un avis éclairé du médecin
- Il y a aussi d'autres métriques spécifiques pour certains types de problème
 - IOU (segmentation sémantique), mAP (détection d'objets)...

PRÉPARATION DES DONNÉES

- Première idée : utiliser toutes les données

Jeu de données d'entraînement

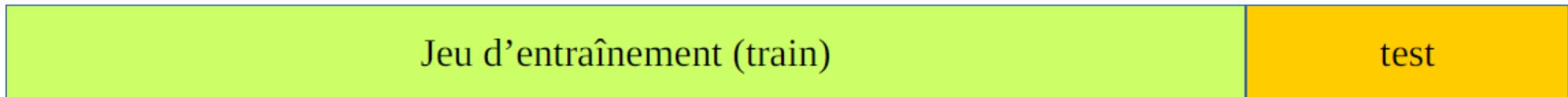
- Pas bon : on risque de surentraîner (apprendre par cœur) et le modèle ne pourra pas généraliser



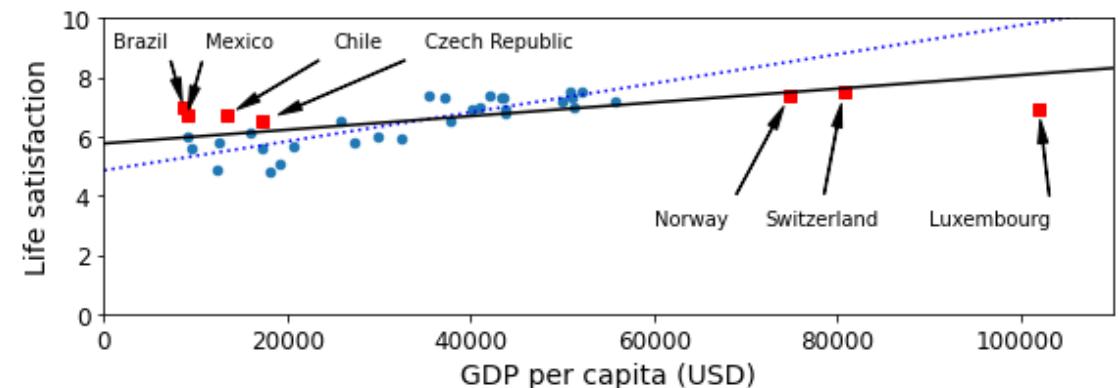
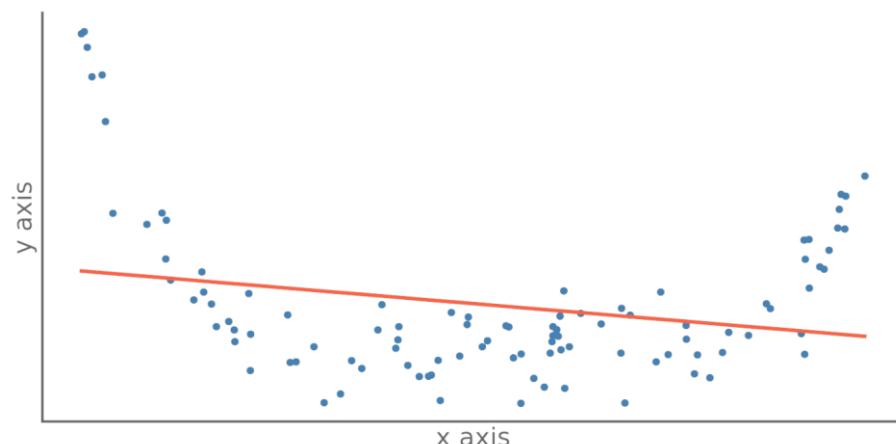
Overfitting

PRÉPARATION DES DONNÉES

- Deuxième idée : séparer les données en deux parties (train/test) et vérifier régulièrement avec le jeu de test

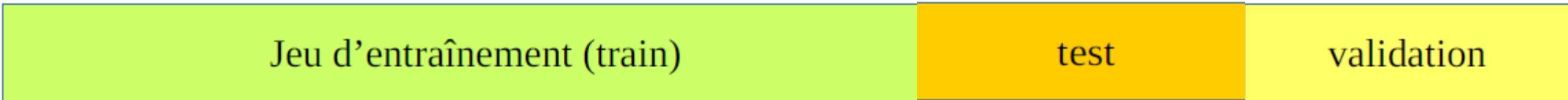


- **Pratique courante**, mais qui a quelques risques
- Problème : Aucune garantie que l'algorithme fonctionnera bien sur de nouvelles données
 - Underfitting : modèle trop simple pour expliquer la variance



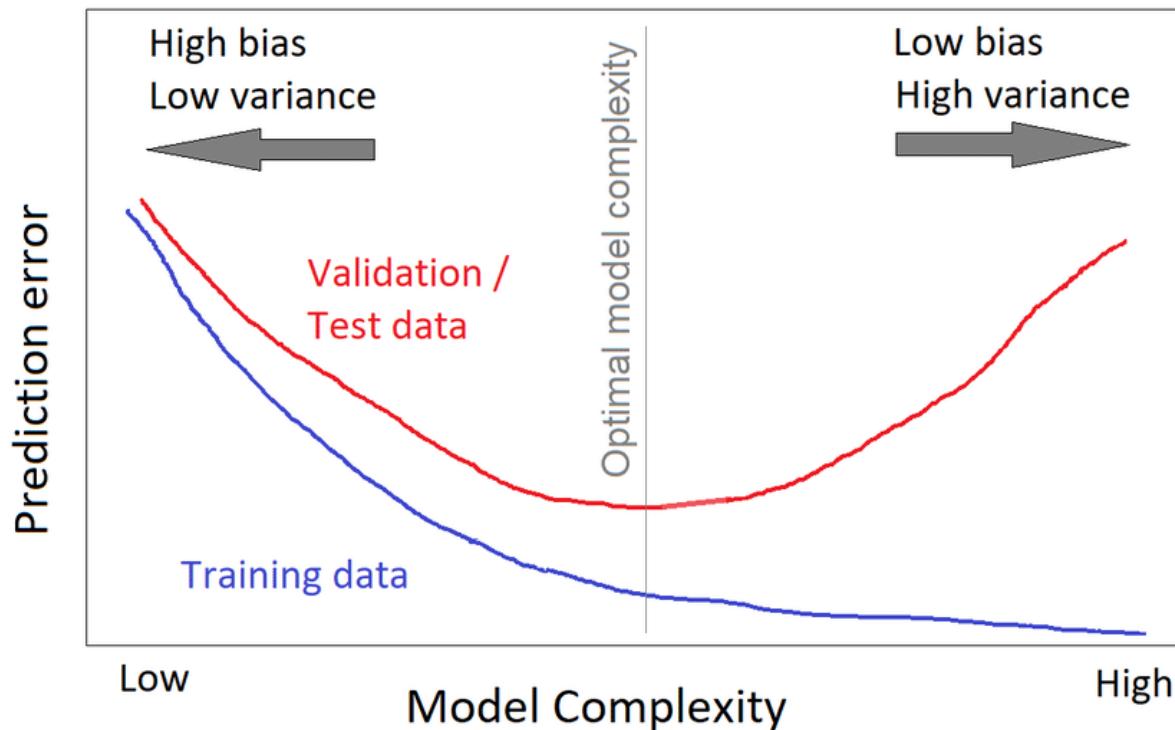
PRÉPARATION DES DONNÉES

- Troisième idée :
 1. entraîner sur le jeu d'entraînement,
 2. choisir les hyper-paramètres qui fonctionnent le mieux sur un jeu de test,
 3. puis une fois le modèle réglé, l'évaluer sur un jeu de validation

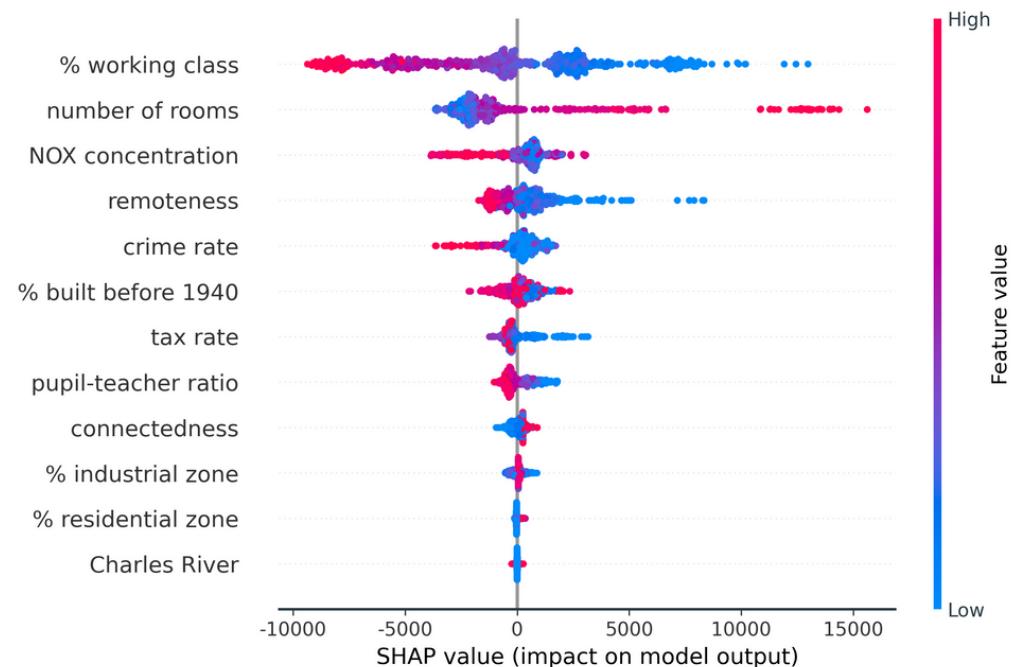


Attention : explicabilité vs performance

- Précision x erreur



Explicabilité (ex : shap)

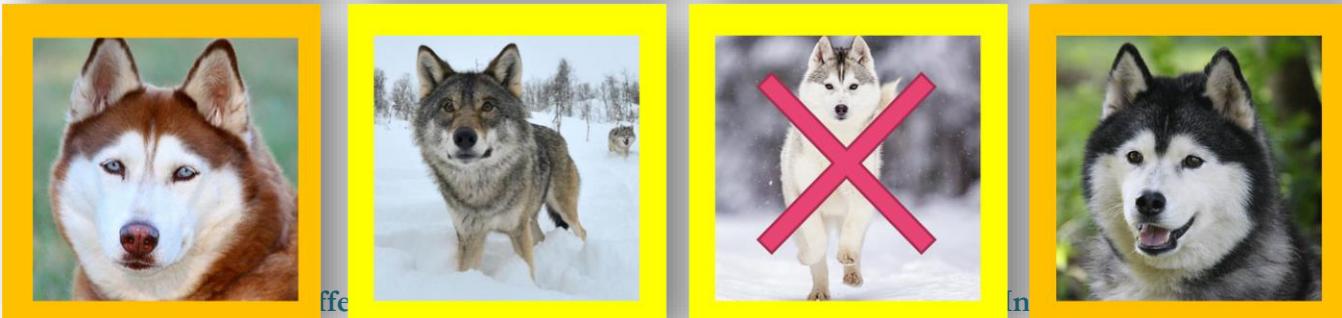


LES POINTS D'ATTENTION : LES BIAIS

l'IA amplifie
les biais

Raciste et détestable, l'intelligence artificielle tente de progresser

Depuis ses premiers pas, l'outil GPT-3 – une IA capable d'écrire des textes originaux – a fait l'admiration de tout un chacun. OpenAI, avec cette troisième évolution du projet parvenait à faire rédiger poésie, articles de presse ou même code de programmation. Cependant, les dérives sont rapidement arrivées, avec une Intelligence qui virait à la grossièreté, voire aux propos toxiques. Que faire ? Une bonne correction, tout simplement.



écolier



écolière



ATTENTION : DONNÉES PERSONNELLES

Contexte : RGPD

- **Transparence** : « Que fait-on de vos données »
- **Évaluation des risques** : évaluer et atténuer les risques de confidentialité à l'avance
- **Audits** : mieux éclairer la position de l'entreprise sur le plan de l'IA et la confidentialité. Difficulté de l'audit des algorithmes et de l'
- **Explicabilité**

ETHIQUE



Nous prenons au quotidien des décisions morales
Une voiture autonome va heurter une grand-mère et un enfant.
En déviant un peu, un des deux peut être sauvé. Qui choisir ?
Ou un fossé : Qui choisir entre les piétons et les passagers

ETHIQUE

2016, « the Next Rembrandt »

- un tableau de Rembrandt
- conçu par un ordinateur
- réalisé par une imprimante 3D,
351 ans après la mort du peintre.

2019, « la Symphonie inachevée ... achevé »

- Les 2 derniers mouvements de la Symphonie n°8
- la Symphonie inachevée que Franz Schubert
- commencée en 1822
quelques 197 années auparavant.



ETHIQUE



L'IA pourrait-elle évaluer des affaires judiciaires et appliquer la justice mieux qu'un juge ?

Les défis éthiques :

- Le manque de transparence des outils d'IA
- L'IA n'est pas neutre : les décisions inexactes, discriminatoires, ou de contenant des biais.
- Collecte de données & la protection de la vie privée.
- Equité et risques pour les droits humains et d'autres valeurs fondamentales.

Premiers pas

TOUR DE TABLE

- Pour démarrer notre exploration du machine learning, nous avons besoin de quelques outils
 - Notebooks Jupyter
 - Les bibliothèques Pandas, Scikit-Learn et Tensorflow



NOTEBOOK JUPYTER

- C'est quoi un Notebook Jupyter ?
 - Document « actif » contenant des **blocs de texte** (en format « **markdown** ») et des **blocs de code Python**

The screenshot shows a Jupyter Notebook interface with the following components:

- Bloc de texte Texte libre et formaté**: A green box pointing to the first text cell in the notebook.
- Bloc de code Code Python et résultat d'exécution**: A green box pointing to the second code cell, which includes the Python code and its output.
- Series**: The title of the first text cell.
- Entrée [4]:** The input cell containing the Python code:

```
from pandas import Series
maserie = Series ([8, 70, 320, 1200], index=["Suisse", "France", "USA", "Chine"])
print(maserie)
```
- Résultat d'exécution**: The output of the code:

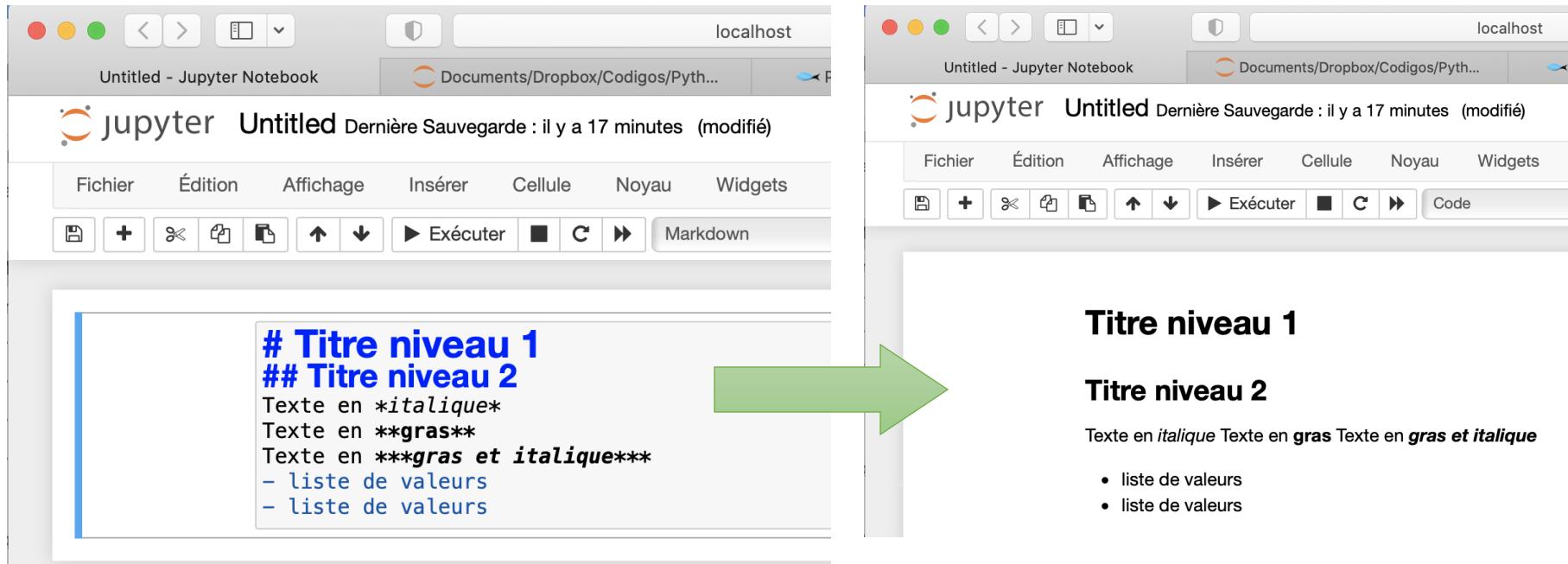
```
Suisse    8
France   70
USA     320
Chine   1200
dtype: int64
```
- Entrée [5]:** The input cell containing the Python code:

```
import numpy as np
monarray = np.random.randn(5)
s2 = Series (monarray, index=["A","B","C","D","E"])
print (monarray, '\n', s2)
```
- Résultat d'exécution**: The output of the code:

```
[-0.96850466 -1.99478449 -1.6655539 -1.22318014 -0.35159315]
A   -0.968505
B   -1.994784
```
- Bloc de texte**: A green box pointing to the third text cell in the notebook.
- Code Python**: A green box pointing to the fourth code cell in the notebook.

NOTEBOOK

- Texte en « markdown »
 - Langage de marquage simple pour formater le texte



NOTEBOOK

■ Bloc de code Python

- Des petits **blocs de code** qu'on peut exécuter
- Equivalent au mode « **itératif** »

Attention : ça dépend de l'ordre dans laquelle **on exécute les blocs, pas nécessairement de l'ordre des blocs**

The screenshot shows a Jupyter Notebook interface with two cells. The top bar includes standard file operations (Fichier, Édition, Affichage, Insérer, Cellule, Noyau, Widgets) and a toolbar with various icons. A green circle highlights the 'Exécuter' button in the toolbar.

Cell 1:

- Titre niveau 1**
- Titre niveau 2**
- Texte en *italique* Texte en **gras** Texte en **gras et italique**
- liste de valeurs
- liste de valeurs

Entrée [2]:

```
a = 2
b = 'To'
print (a*b)
```

A large green arrow points from the bottom of Cell 1 to the beginning of Cell 2.

Cell 2:

Entrée [2]:

```
a = 2
b = 'To'
print (a*b)
```

ToTo

Google Collab

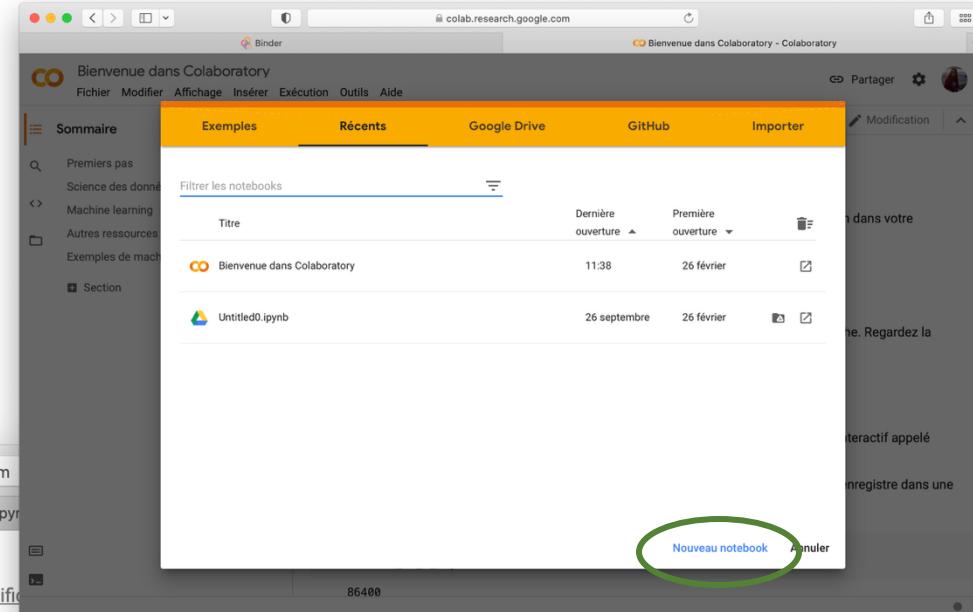


Accessible avec un simple compte Google

<https://colab.research.google.com>

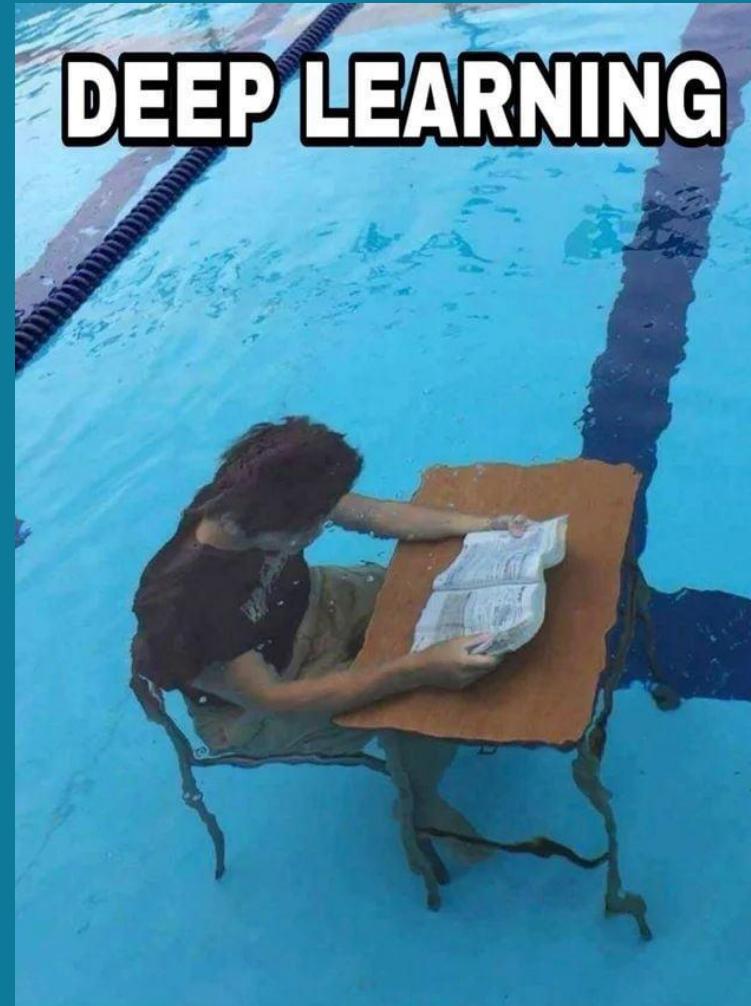
```
print('hello')
print(2+2)

hello
4
```



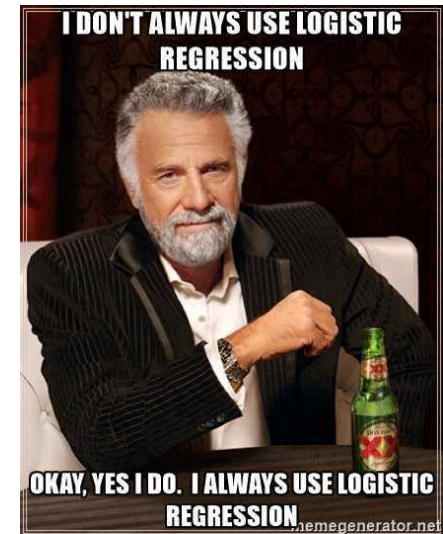
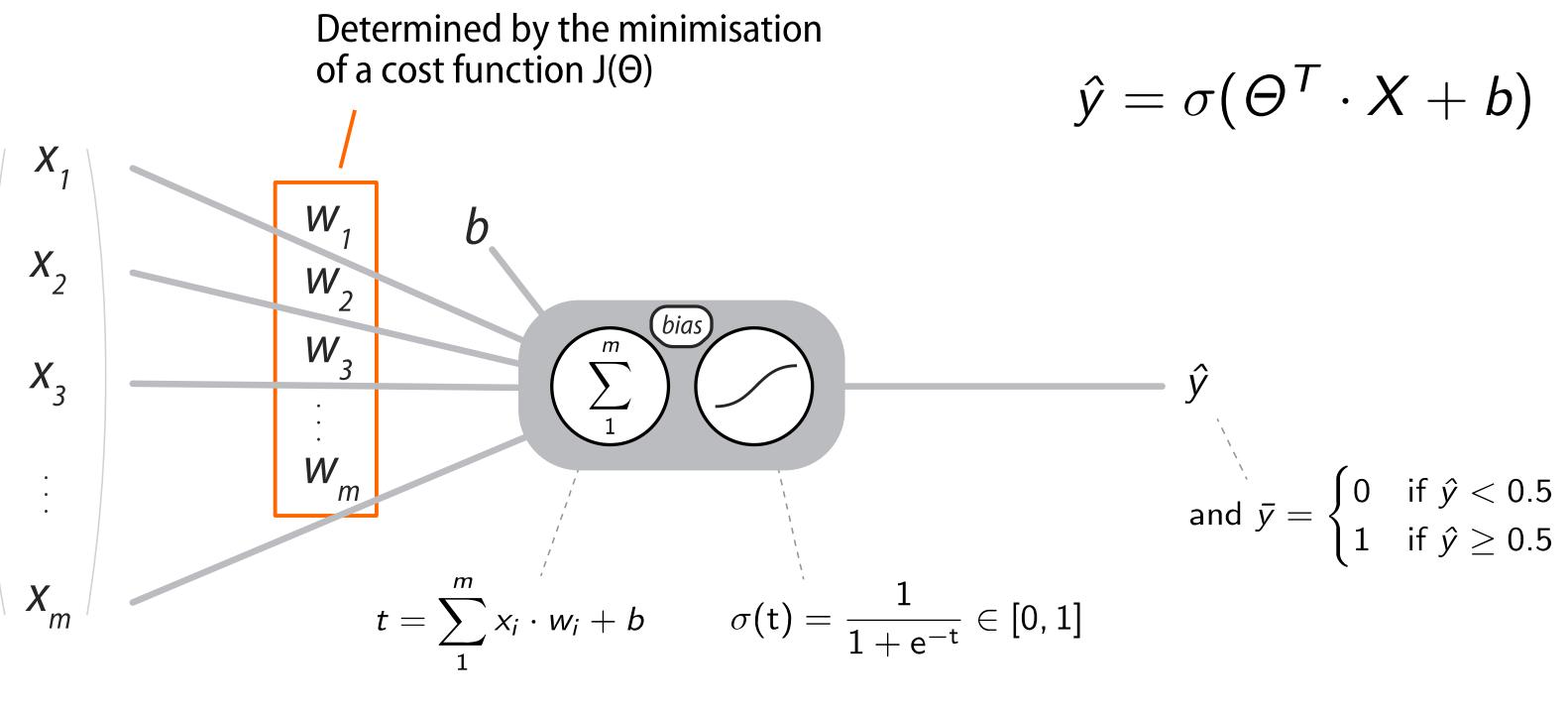
- On peut créer ses Notebooks et les enregistrer sur son Google Drive
- Partage d'un Notebook avec d'autres personnes est aussi possible
- Surtout, possibilité d'utiliser des machines avec GPU

Partie 2 :



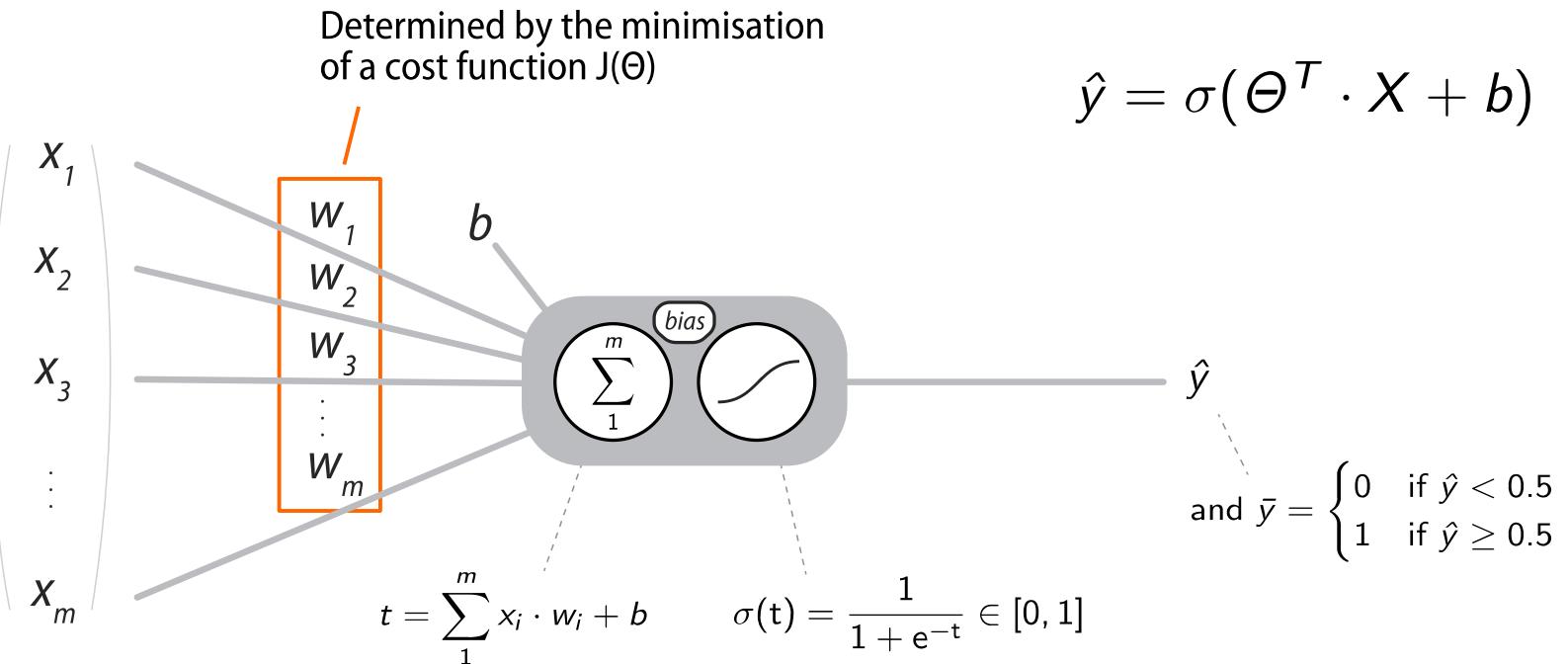
Ecole Doctorale SNS

SCHÉMA D'UNE RÉGRESSION LOGISTIQUE



Input	Bias / Weight	Activation function	Output
X	Θ	$\sigma(t)$	\hat{y}

SCHÉMA D'UNE RÉGRESSION LOGISTIQUE

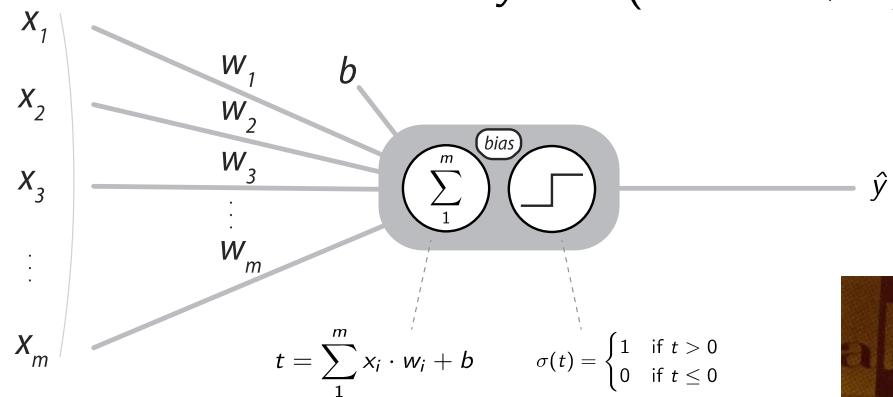


Input	Bias / Weight	Activation function	Output
X	Θ	$\sigma(t)$	\hat{y}

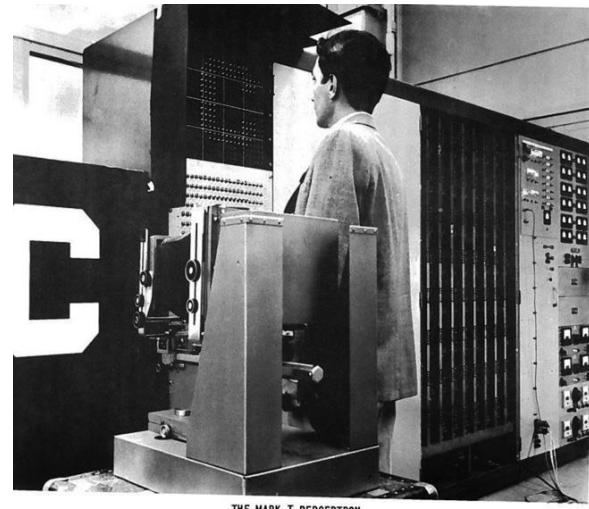
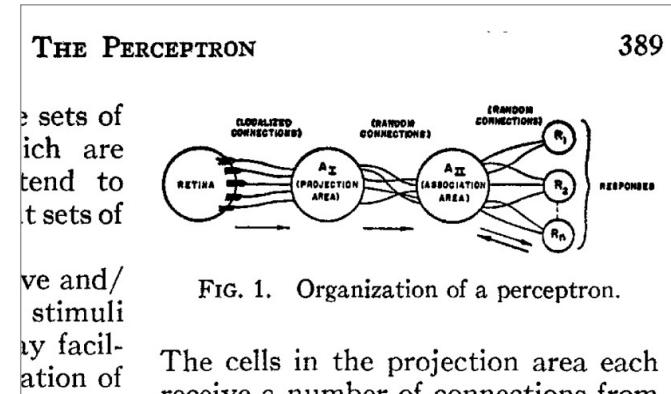
Surprise !!! Ceci est le schéma d'un neurone artificiel (perceptron). On a un réseau d'un neurone !!

L'HISTOIRE DES RÉSEAUX DE NEURONES

- Création du "Perceptron": Frank Rosenblatt (1958)



$$\hat{y} = \sigma(\Theta^T \cdot X + b)$$



NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI)—The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's \$2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of \$100,000.

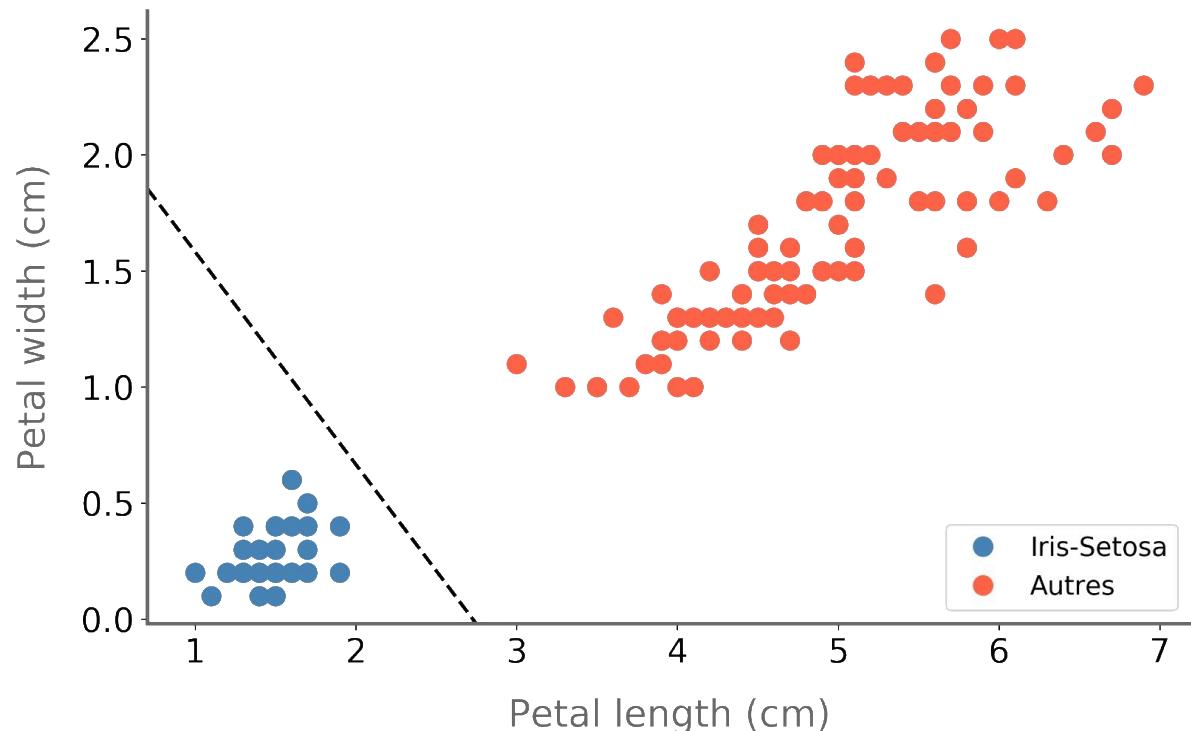
Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

LE PERCEPTRON MARCHE (LINÉAIREMENT) !

Iris plants dataset

Dataset from : Fisher, RA. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936)



Length	Width	Iris Setosa (0/1)
x_1	x_2	y
1.4	1.4	1
1.6	1.6	1
1.4	1.4	1
1.5	1.5	1
1.4	1.4	1
4.7	4.7	0
4.5	4.5	0
4.9	4.9	0
4.0	4.0	0
4.6	4.6	0
(...)		

EXERCICE 1

- Pour ce premier exercice, vous allez utiliser un neurone simple (perceptron) pour faire la classification sur un dataset historique : IRIS
 - IRIS est un dataset créé originellement en 1936
 - Il regroupe des mesures (largeur, longueur) des pétales et sépales de trois espèces de la fleur Iris
 - Iris versicolor
 - Iris virginica
 - Iris setosa
 - Le perceptron sera utilisé pour séparer les plantes Iris setosa des autres
 - Nous allons utiliser la bibliothèque Scikit Learn
 - Accéder ce site : <https://t.ly/jek-x>



You are not a data scientist..

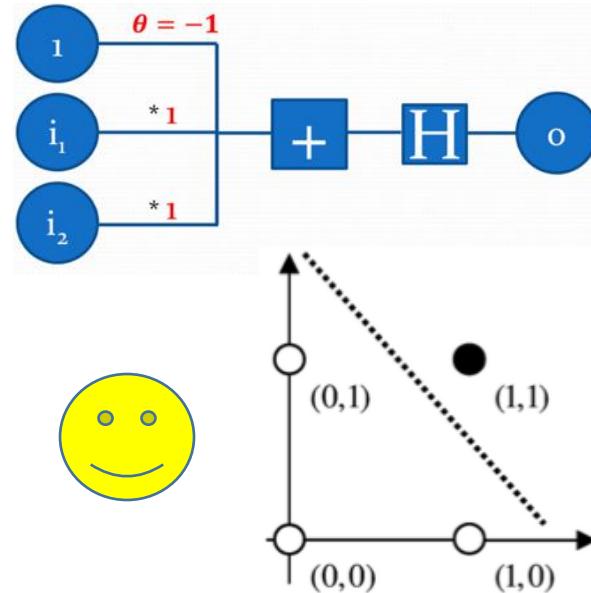


if you don't know this flower

LES RÉSEAUX DE NEURONES PROFONDES

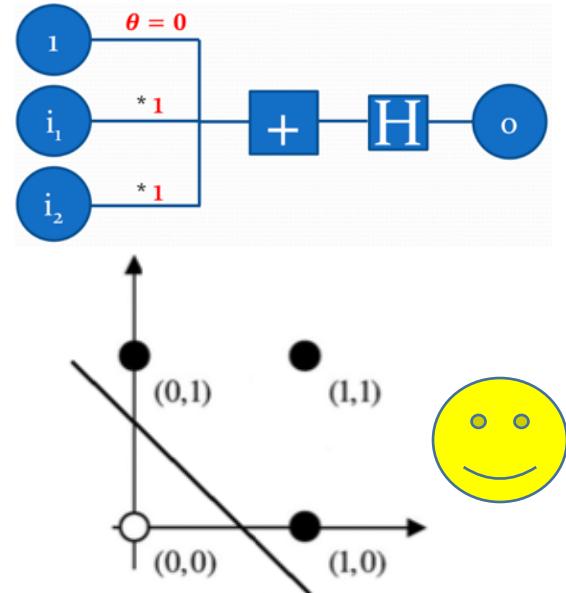
- Un perceptron peut apprendre tant que les résultats sont "linéairement séparables" :
- Ex ET logique

i1	i2	o
0	0	0
0	1	0
1	0	0
1	1	1



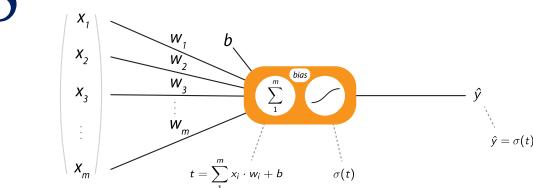
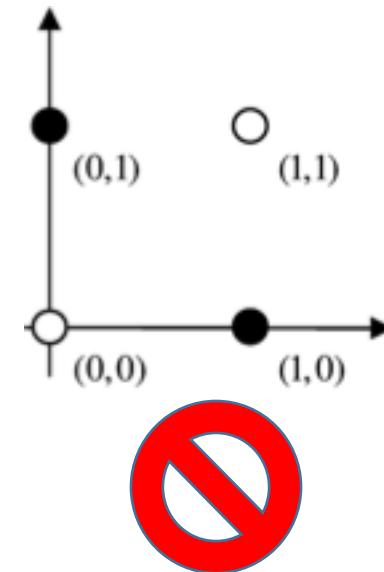
- Ex OU logique

i1	i2	o
0	0	0
0	1	1
1	0	1
1	1	1

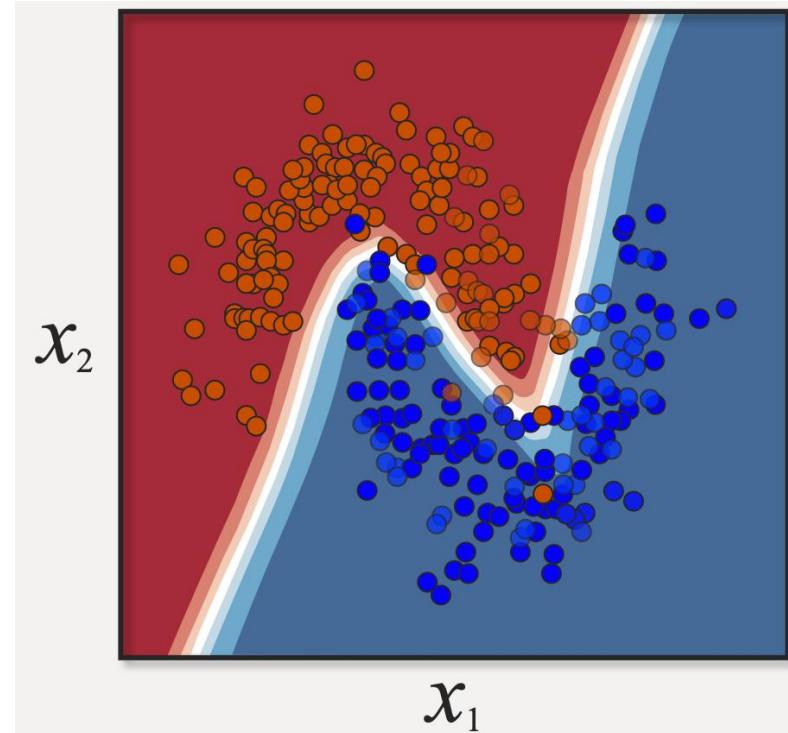
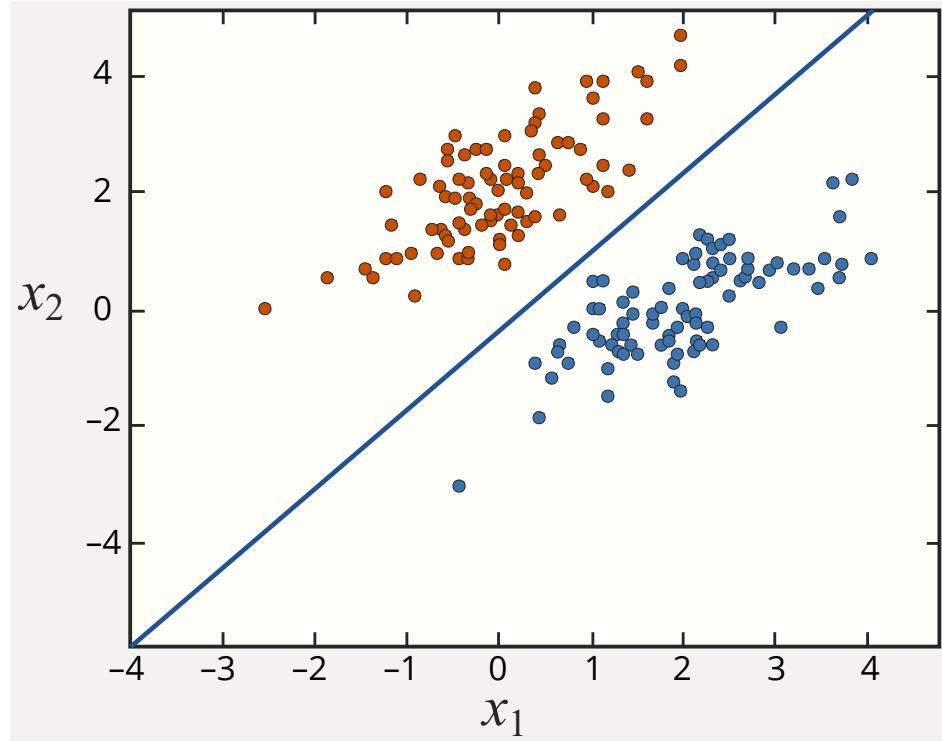


- Ex XOU logique

	E1 = 0	E1 = 1
E2 = 0	0	1
E2 = 1	1	0

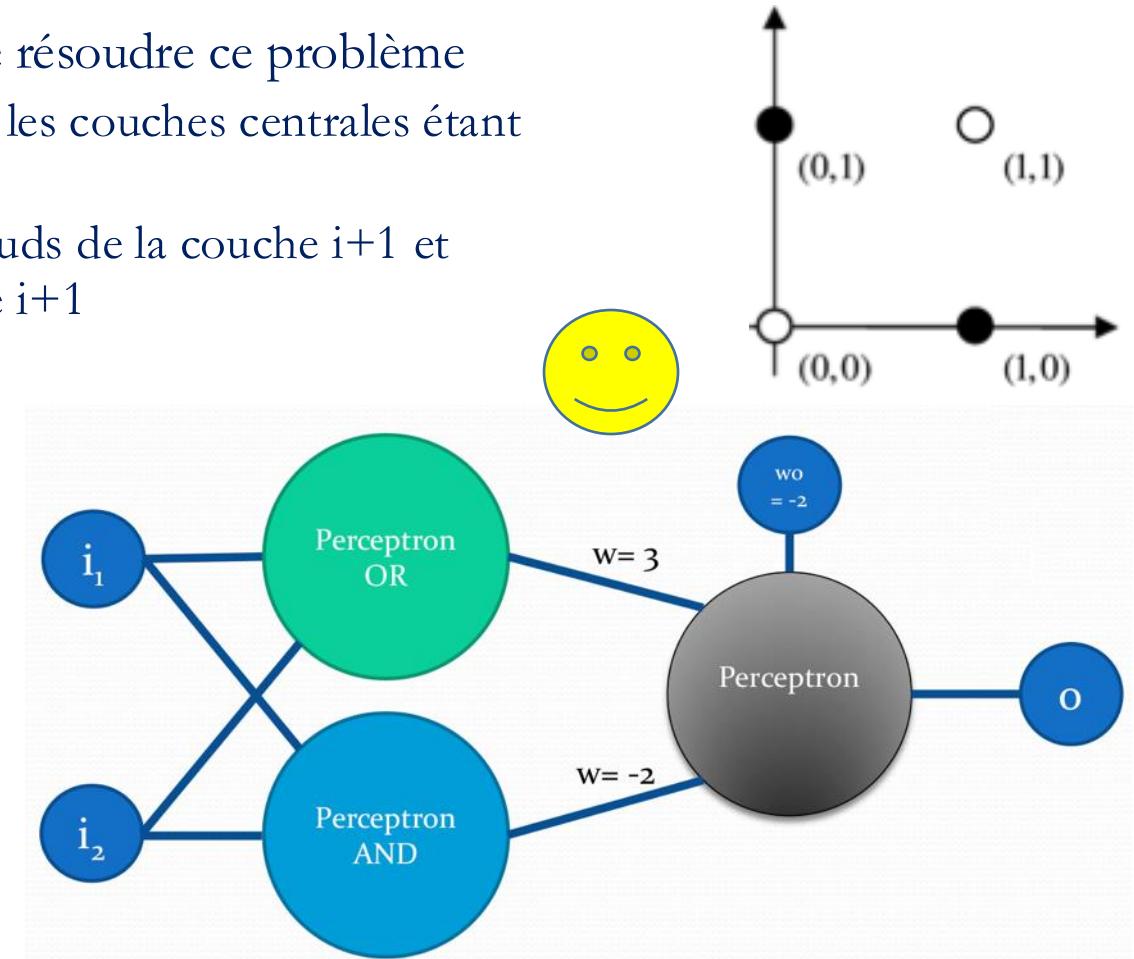
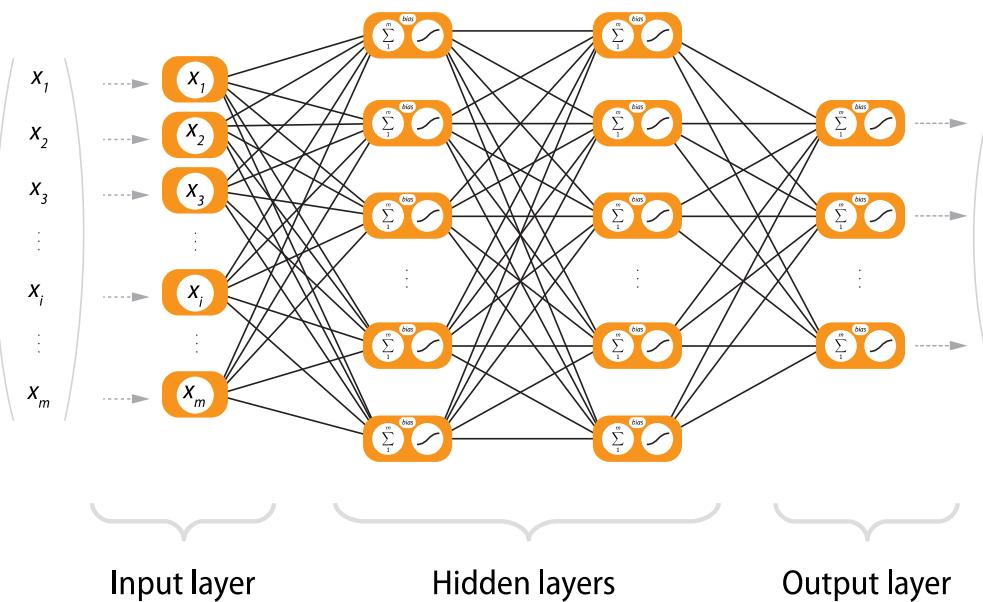


UN SEUL NEURONE NE FAIT PAS TOUT

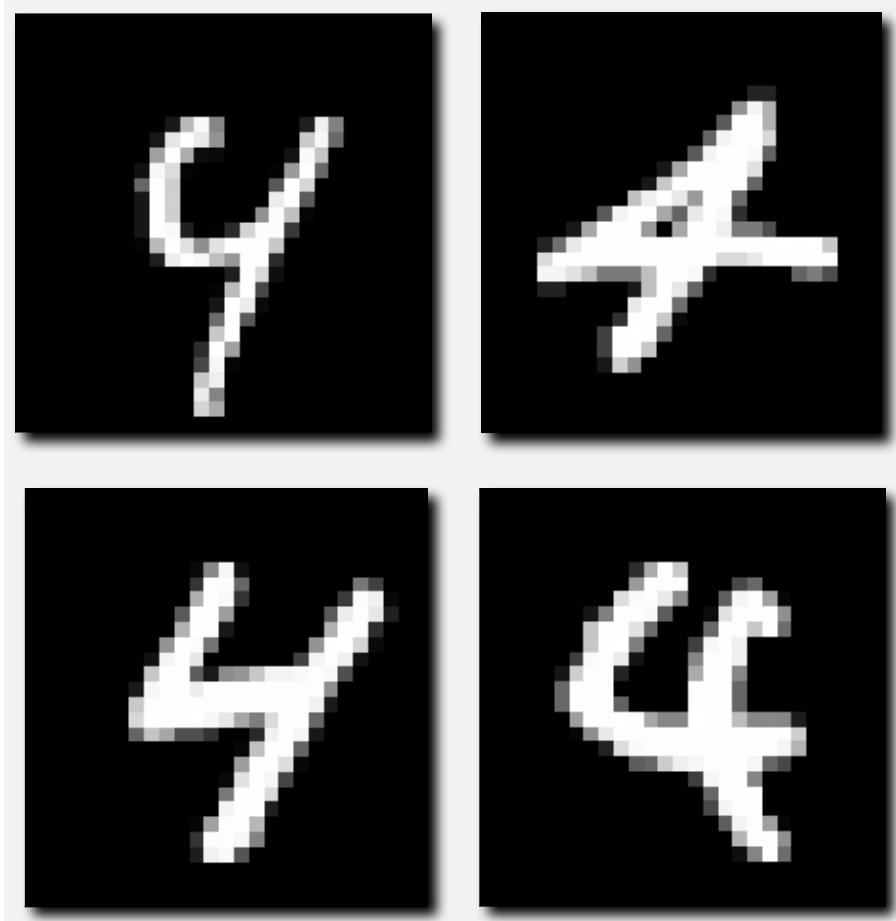


RÉSEAUX DE NEURONES

- Le concept de perceptron multicouche (MLP) permet de résoudre ce problème
 - On utilise plusieurs perceptrons en couche successives, les couches centrales étant dites cachées
 - Chaque nœud de la couche i est connecté à tous les nœuds de la couche $i+1$ et l'information circule toujours de la couche i à la couche $i+1$

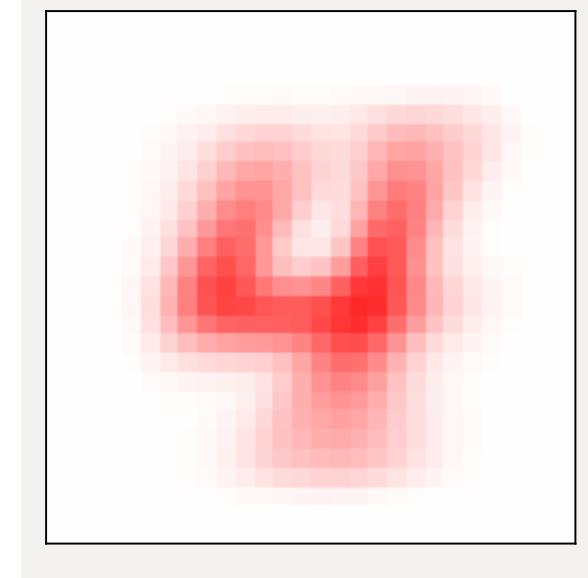


PLUS DE NEURONES = PLUSIEURS FAÇONS D'ÉCRIRE

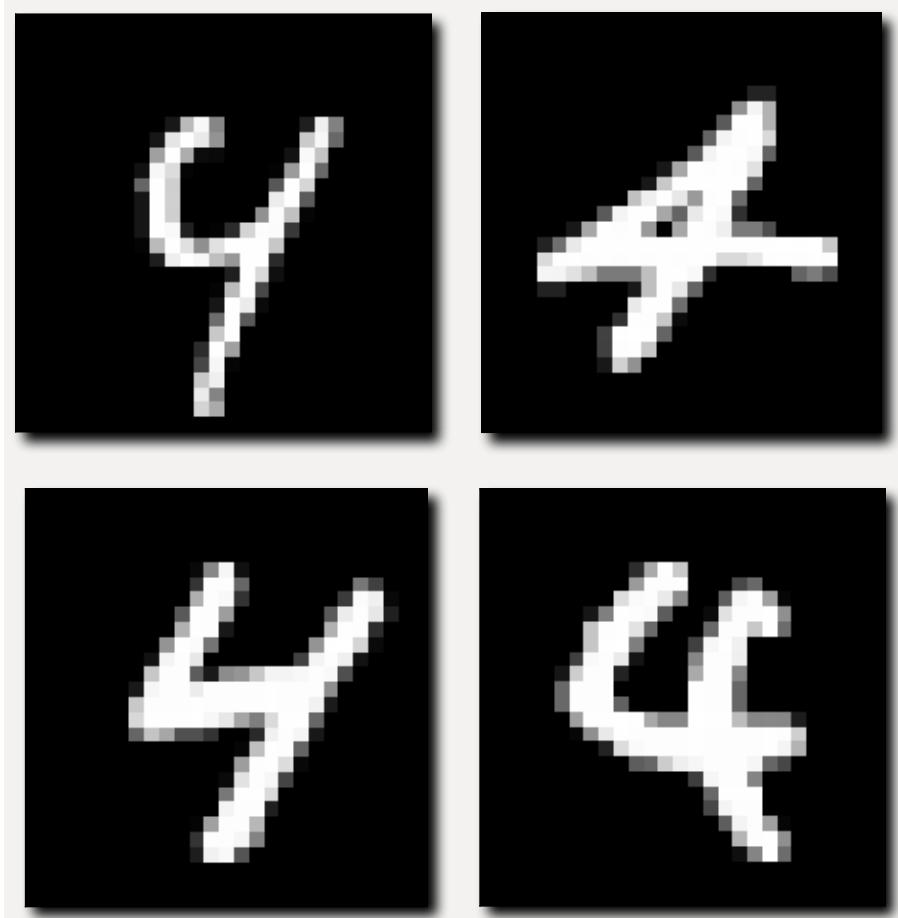


Filtre unique

- pas très précis, adapté juste au cas moyen

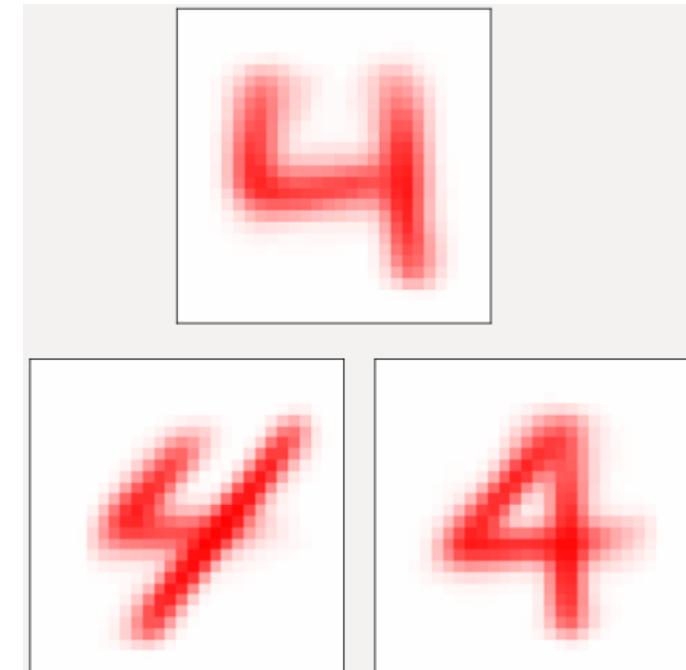


PLUSIEURS FAÇONS D'ÉCRIRE 4

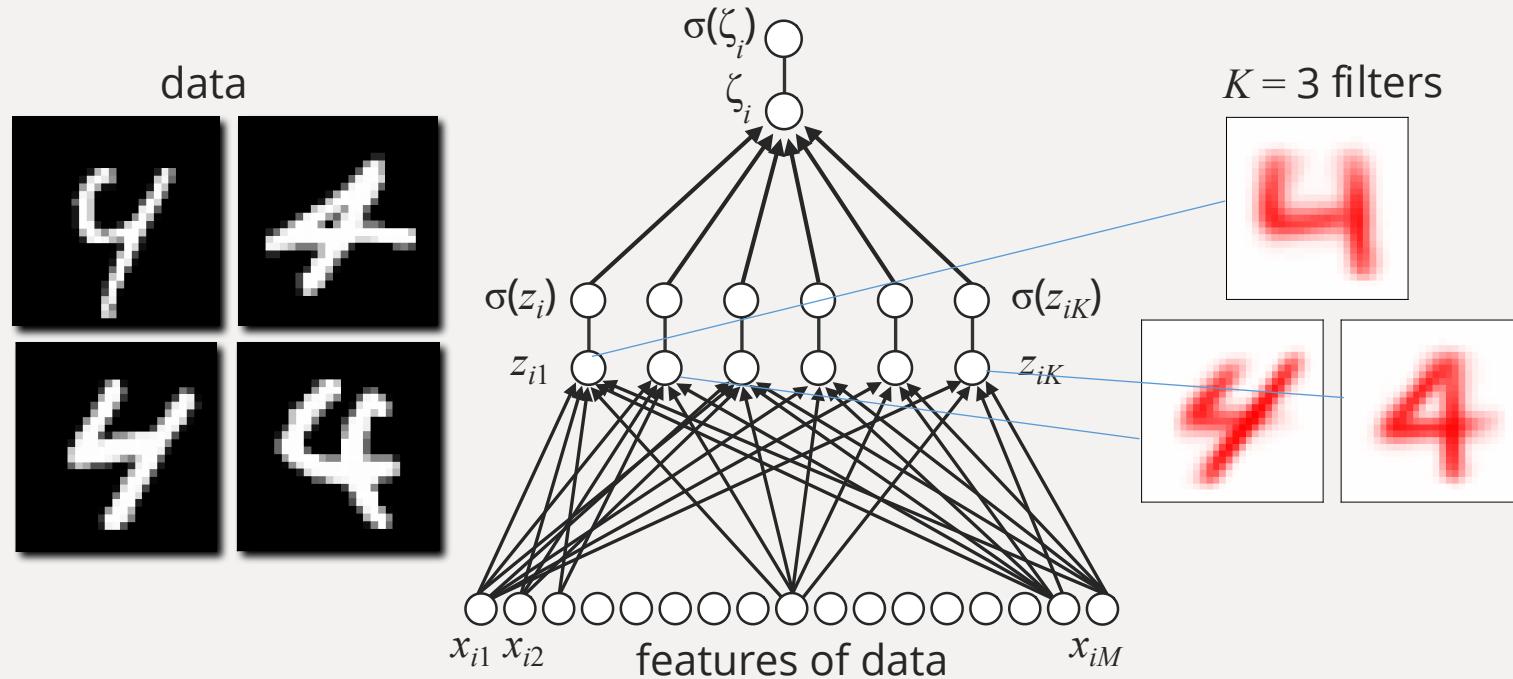


Et si on avait plusieurs filtres ?

- Chacun est spécialisé sur une forme d'écrire 4
- Une correspondance "forte" avec un filtre est un bon signe



RÉSEAU DENSE DE NEURONES



- Dans la pratique, les neurones sont initialisés avec des poids aléatoires
- L'entraînement permet à chacun de regarder les données avec des points de vue différents
- C'est la somme de ces points de vue qui donne une sortie

L'ACTION DES NEURONES SE CUMULE

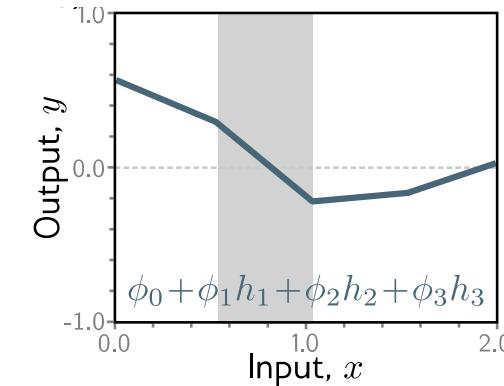
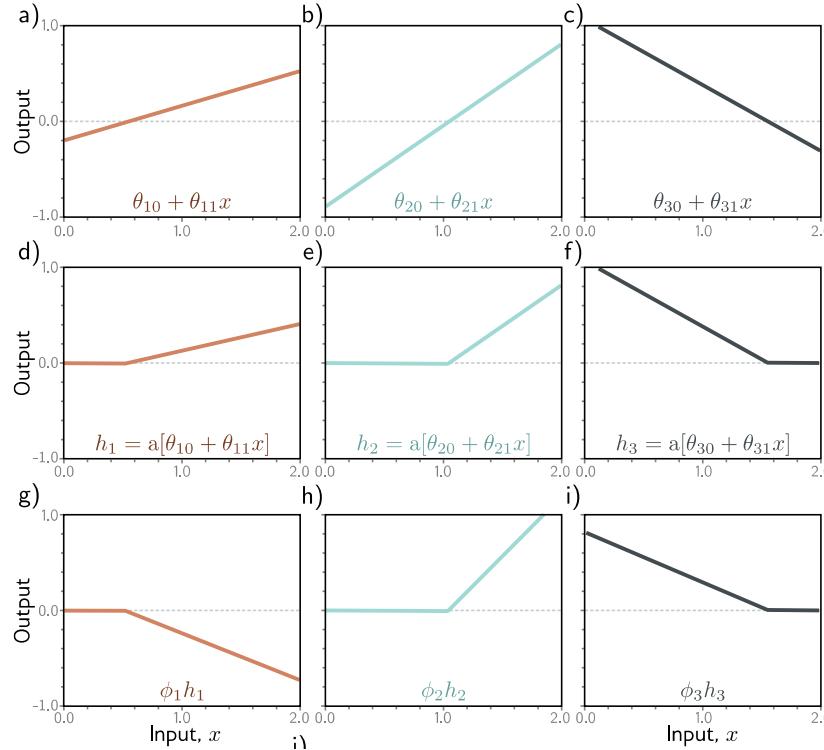
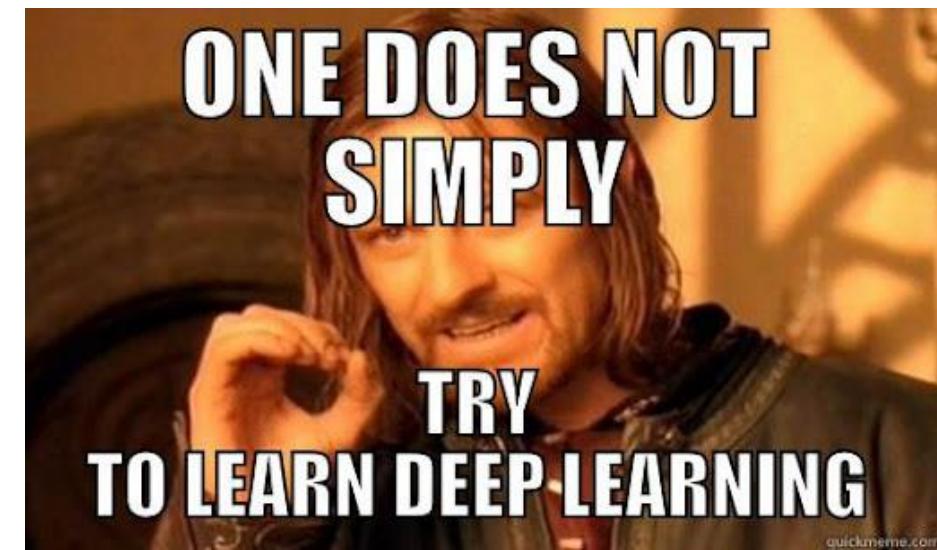
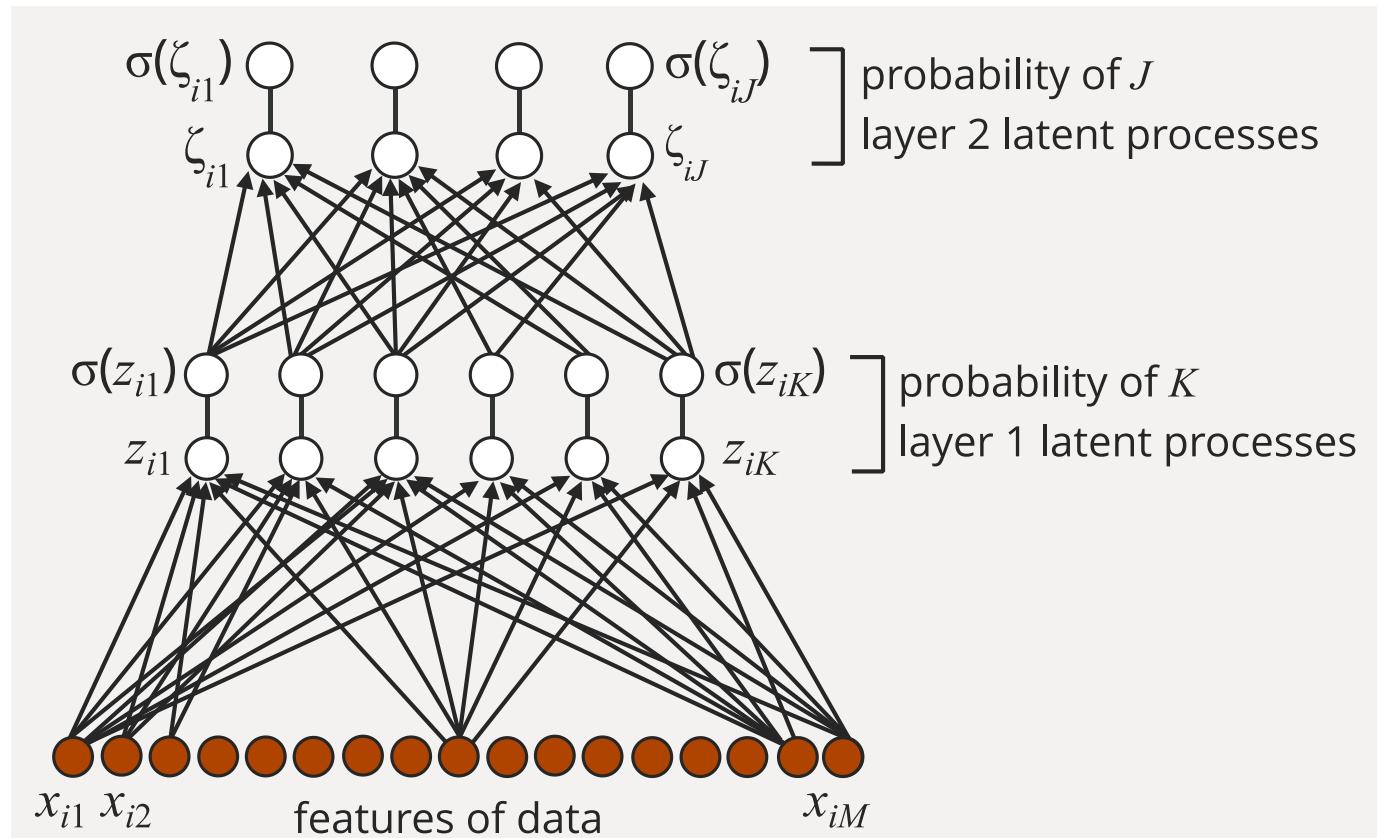


Figure 3.3 Computation for function in figure 3.2a. a–c) The input x is passed through three linear functions, each with a different y-intercept $\theta_{\bullet 0}$ and slope $\theta_{\bullet 1}$. d–f) Each line is passed through the ReLU activation function, which clips negative values to zero. g–i) The three clipped lines are then weighted (scaled) by ϕ_1, ϕ_2 , and ϕ_3 , respectively. j) Finally, the clipped and weighted functions are summed, and an offset ϕ_0 that controls the height is added. Each of the four linear regions corresponds to a different activation pattern in the hidden units. In the shaded region, h_2 is inactive (clipped), but h_1 and h_3 are both active.

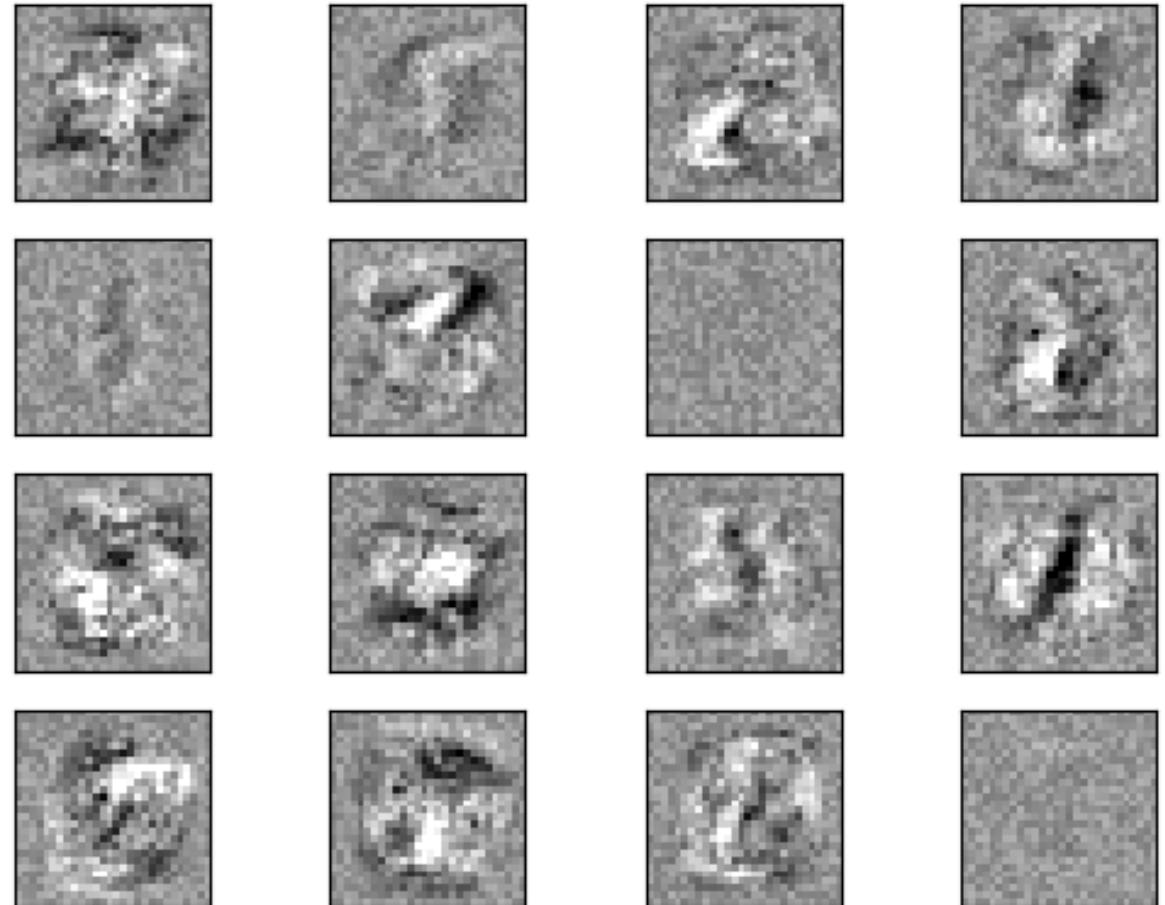
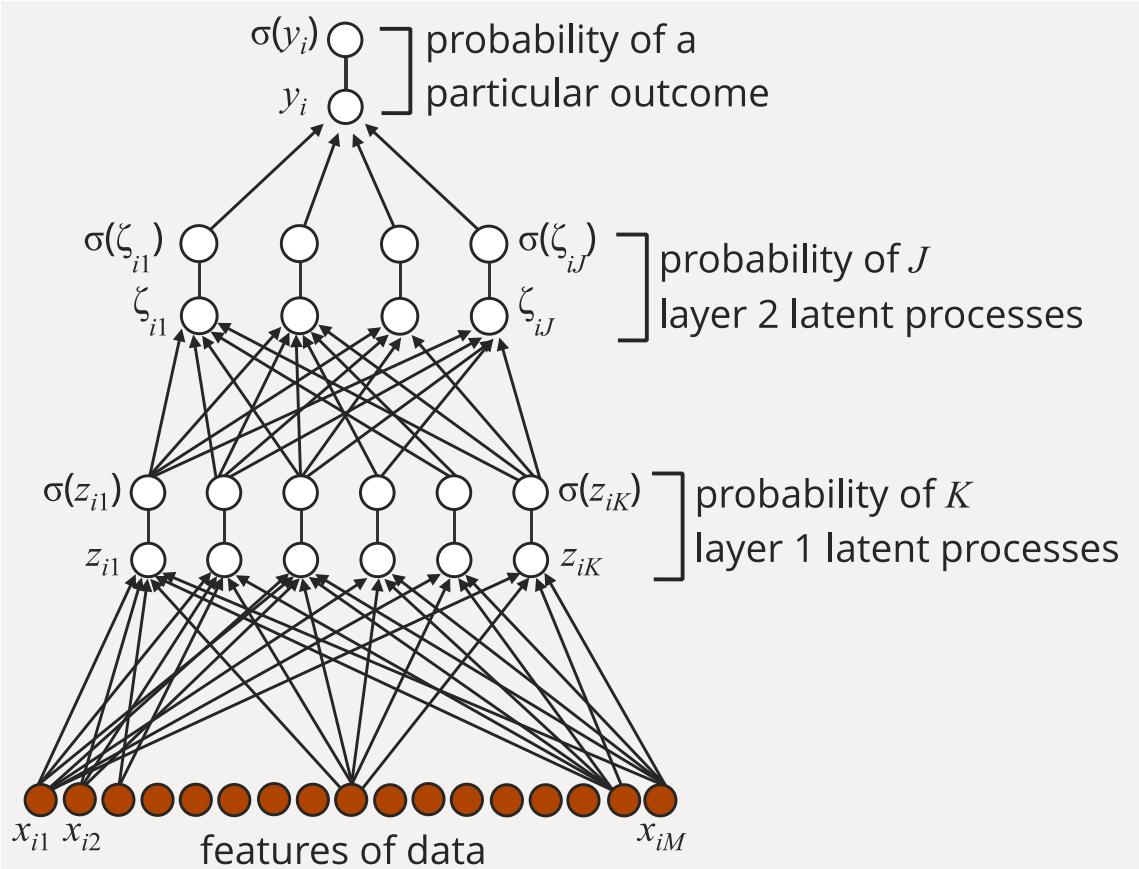
Source : Understanding Deep learning, Simon J.D. Price (<http://udlbook.com>)

POURQUOI S'ARRÊTER LÀ ?

- Différentes couches peuvent mettre en valeur des features distinctes



PERCEPTRON MULTI-COUCHES



PREMIER PROBLÈME DES DNN - SCALABILITÉ

For a fully connected layer of (only)
1000 neurons, we would need to



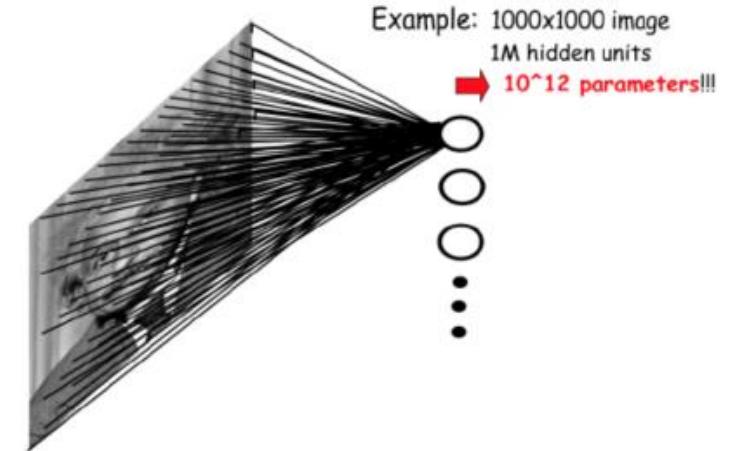
0.0008 M pixels
28x28, 8 bits

→ 785.000 params



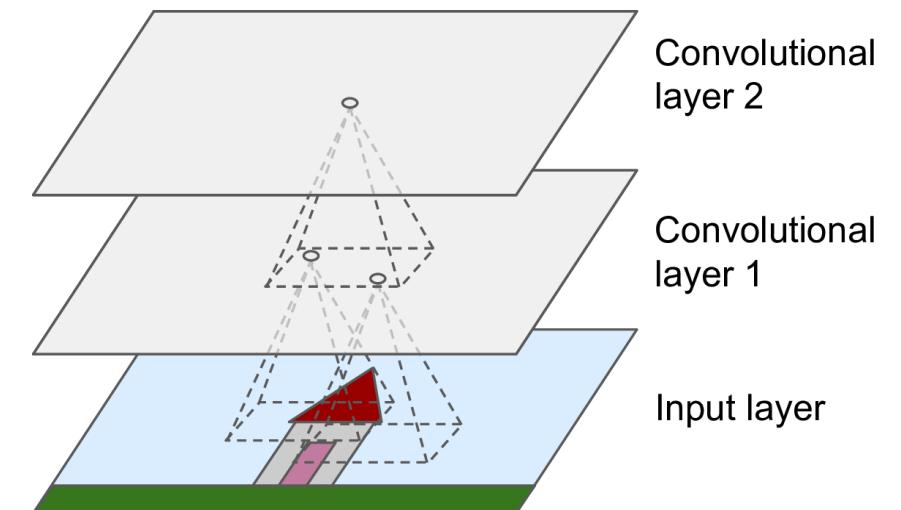
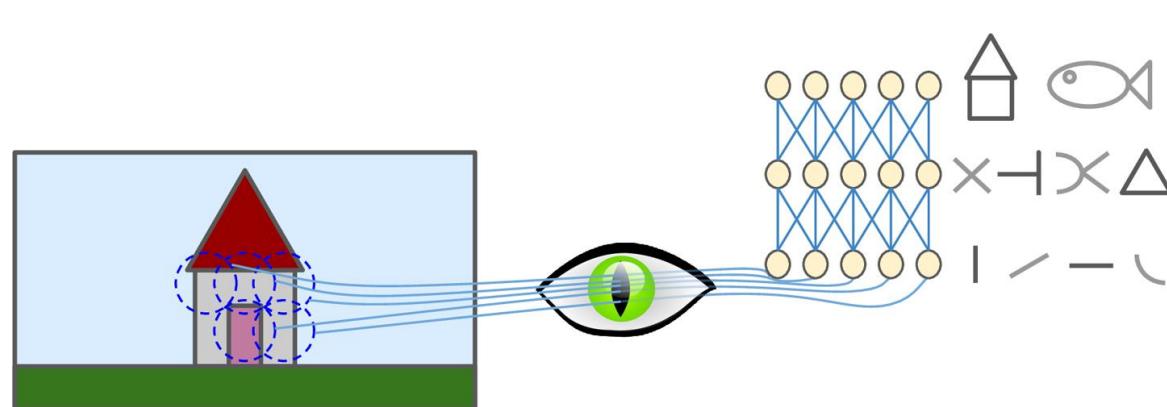
24 M pixels
(r,v,b) 3x8 bits

→ 72. 10^{E9} params...



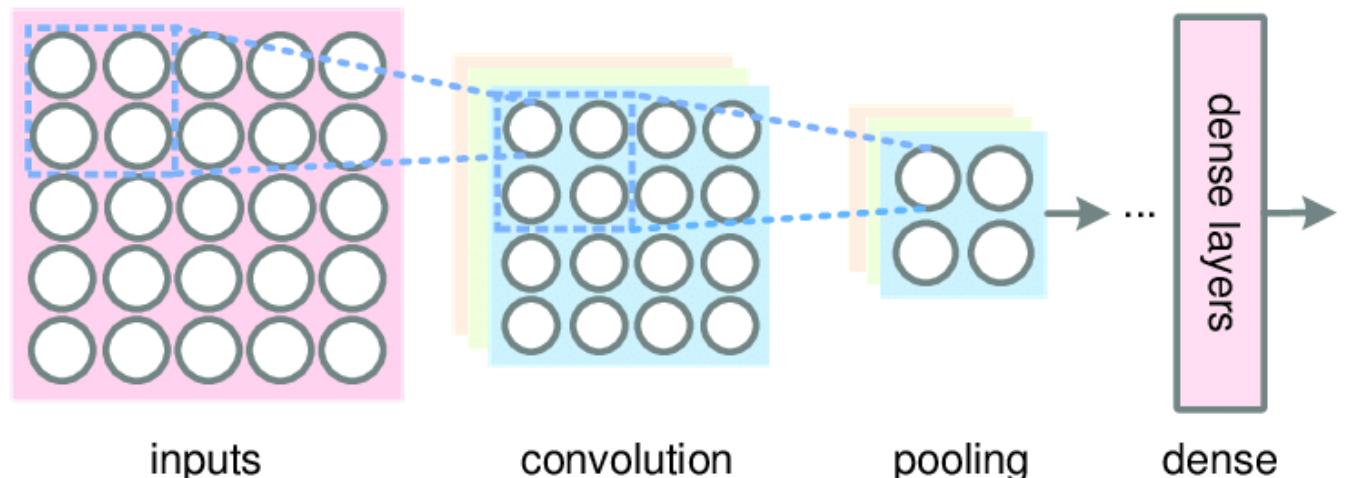
RETOUR À LA SOURCE → LE CORTEX VISUEL

- Des travaux faits par des biologistes/neurologistes dans les années 50 ont montré que plusieurs neurones ont un champ d'activation limité
 - Ne réagissent qu'à des certaines zones du champ visuel
 - La réceptivité de ces neurones peut se superposer
 - Certains neurones ne répondent qu'à certains motifs : ex : lignes horizontales
 - D'autres étaient plus "avancés", combinant de signaux des neurones plus basiques



LES RÉSEAUX CONVOLUTIFS (CNN)

- Introduits par Yan Lecun en 1998 dans l'architecture LeNet-5
 - Utilisé encore pour la reconnaissance de l'écriture manuelle
- Utilisation d'éléments déjà connus :
 - Réseaux totalement connectés
 - Fonctions d'activation (sigmoid et ReLU, notamment)
- Introduction de deux nouveaux éléments :
 - Les couches convolutionnelles
 - Les couches de pooling



C'EST QUOI UNE CONVOLUTION ?

filter (W)

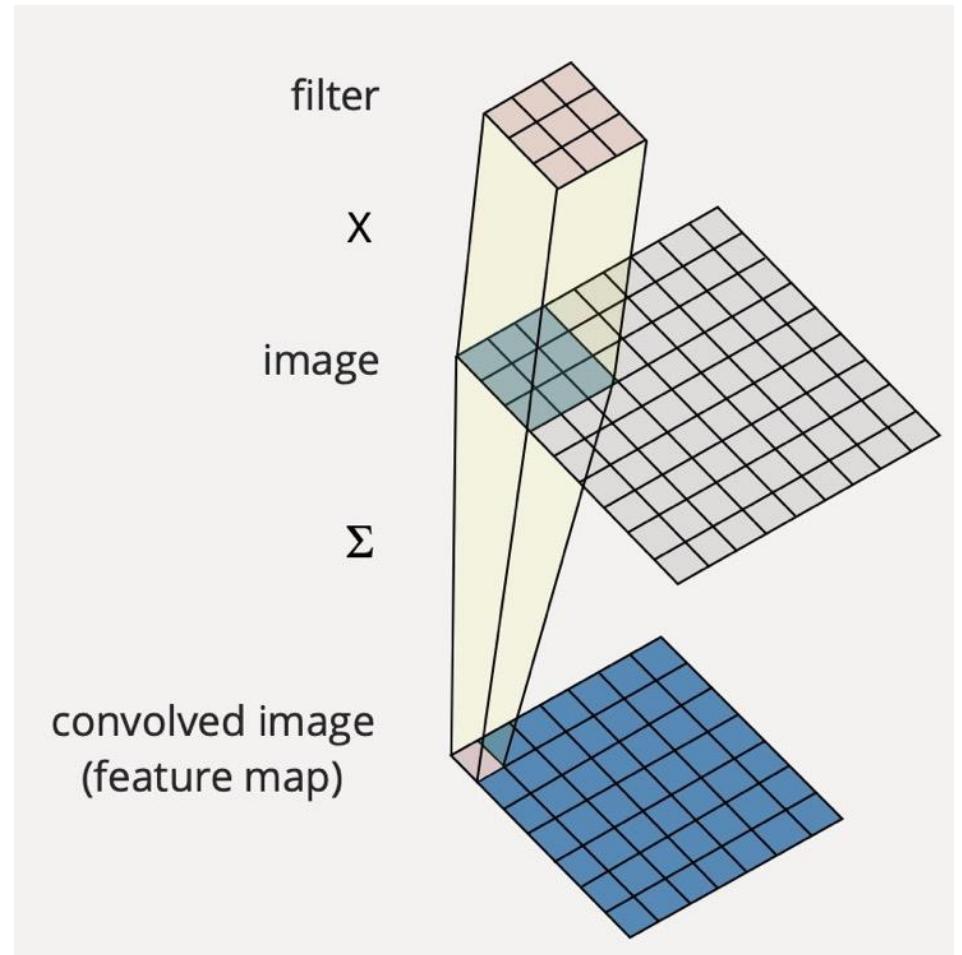
0	1	0
1	-4	1
0	1	0

input (i)

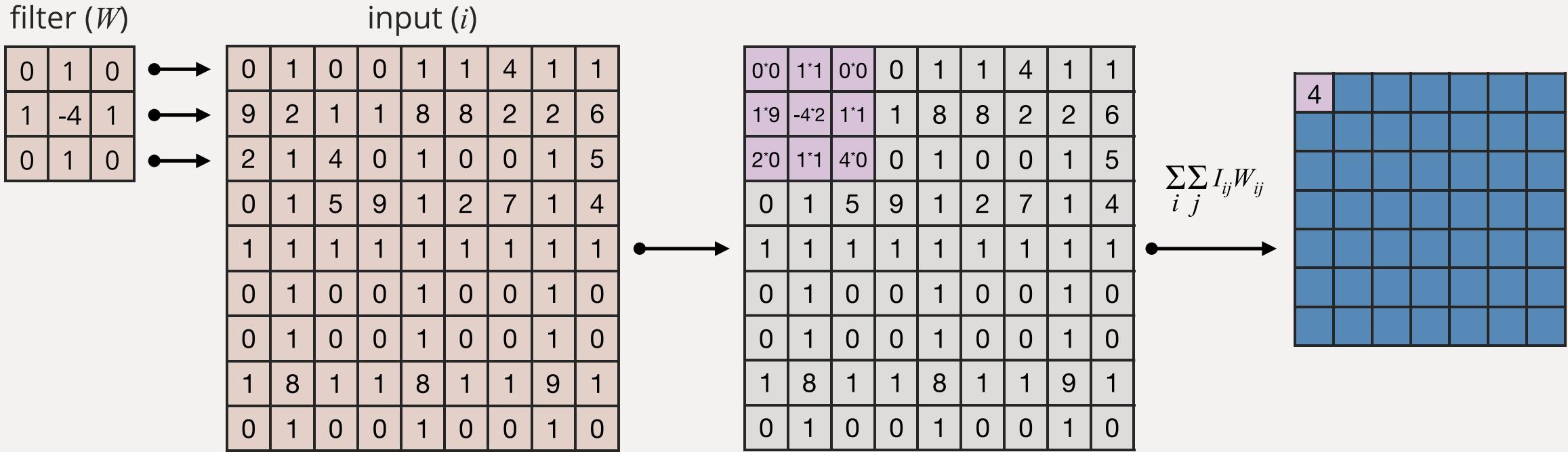


LE CAS DE LA CONVOLUTION

- Une convolution (2D dans cet exemple) est une opération de type multiplication de matrices
- Un filtre "balaye" la matrice de données, résumant l'information selon un motif défini (ou aléatoire)
- Un filtre peut avoir différentes tailles
 - Ex : 5×5 , 3×3 , 4×6 ...

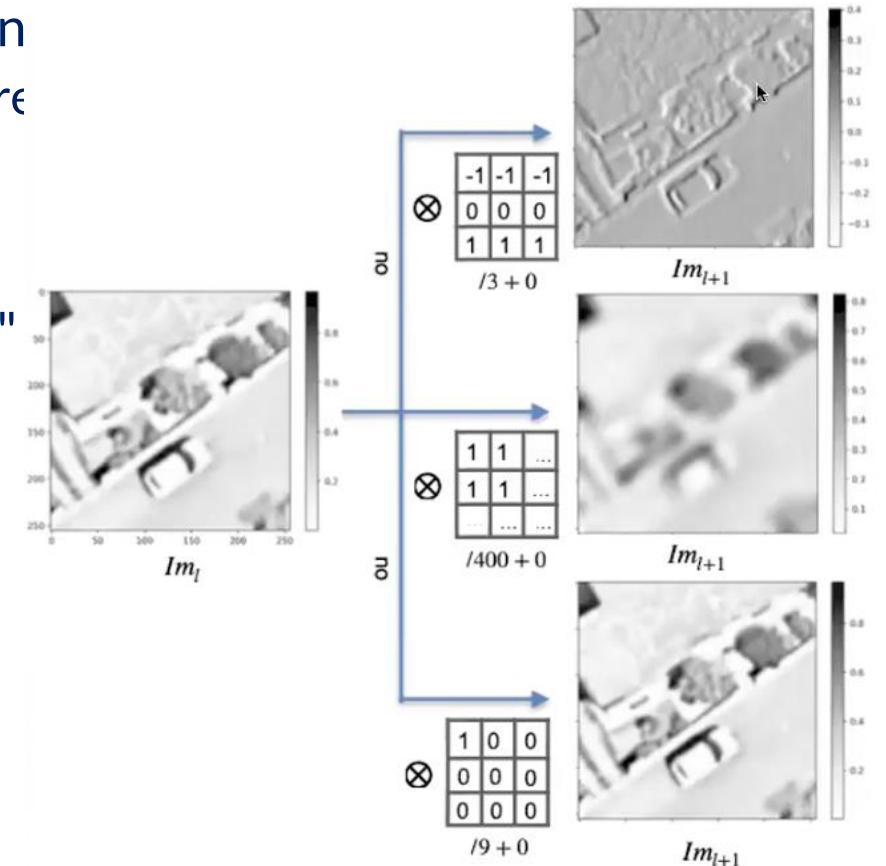


LE CAS DE LA CONVOLUTION 2D



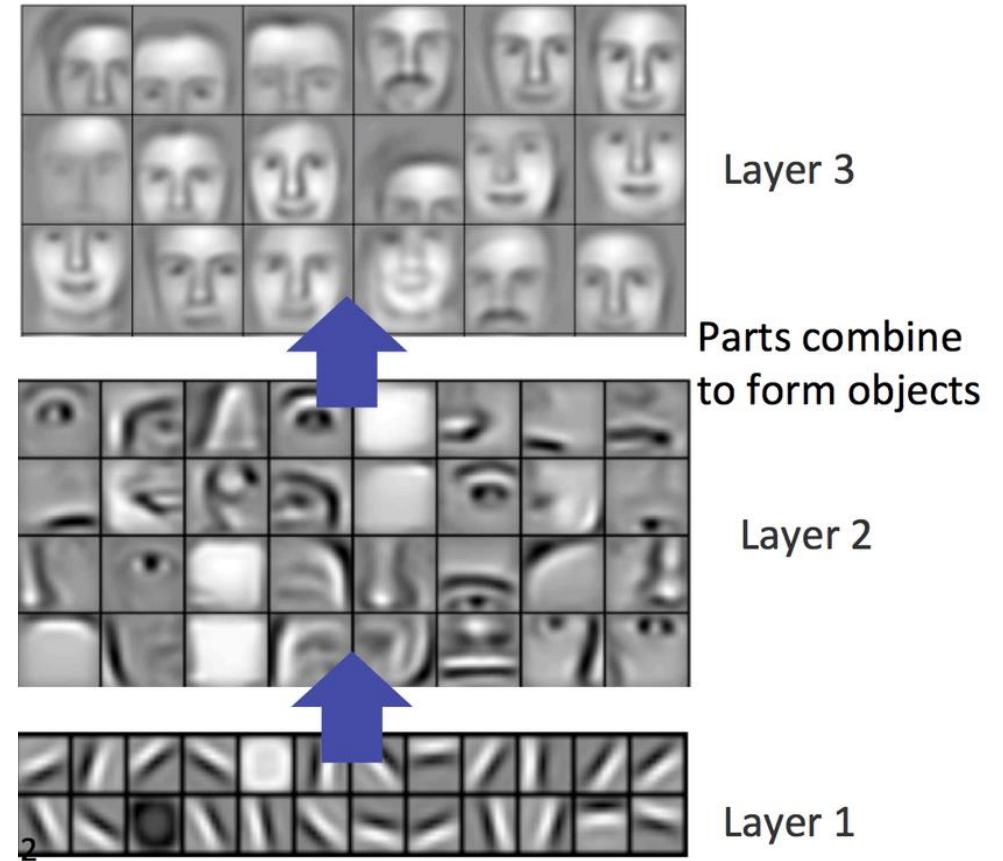
L'INTÉRÊT DES CONVOLUTIONS

- Les filtres permettent de chercher des caractéristiques différentes
 - On peut dire que chaque neurone est responsable par un filtre
- Souvent, les filtres sont initialisés de manière aléatoire
 - Certains se dédoublent, d'autres ne servent à rien
- En spécialisant les neurones des couches initiales, on "enrichit" l'observation sans trop augmenter le nombre de paramètres
 - 1 – chaque convolution produit une sortie "réduite"
 - 2 – les neurones intermédiaires ne seront pas "figés" sur une seule caractéristique
 - 3 – comparé à un DNN, on aura moins de connexions



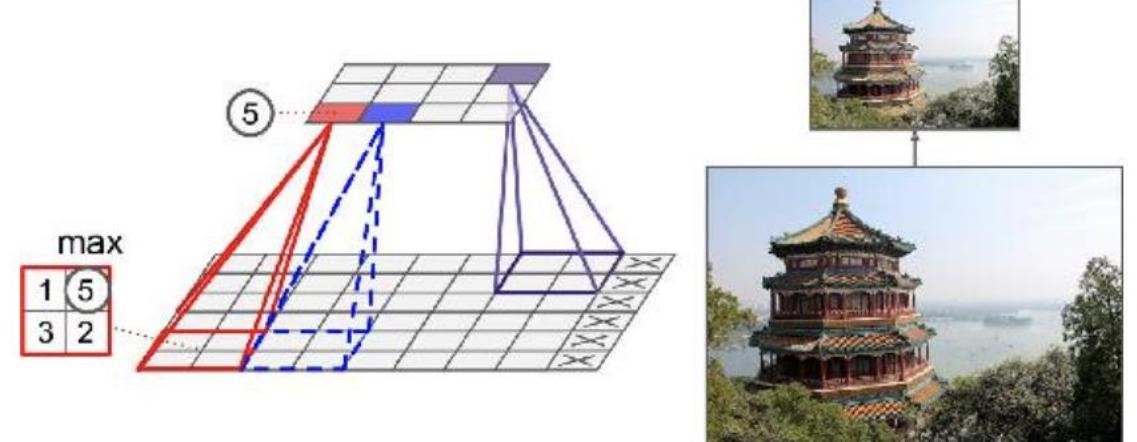
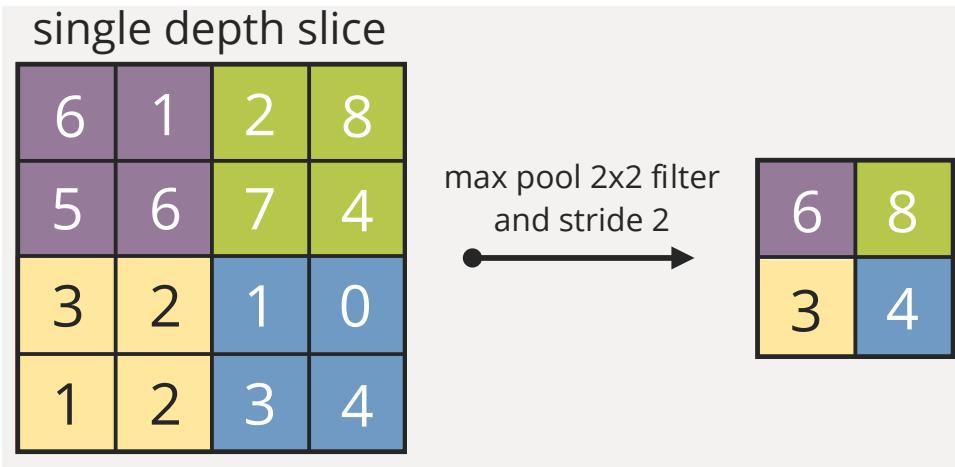
EST-CE SI RÉVOLUTIONNAIRE ?

- Comme on a vu avant, il faut un nombre important de filtres pour représenter différentes features
- Les neurones pour des feature maps différents utilisent des poids différents
- Un DNN ne peut reconnaître le motif là où il l'a appris
- Un CNN apprend à identifier un motif, il est capable d'identifier ce motif partout dans l'image

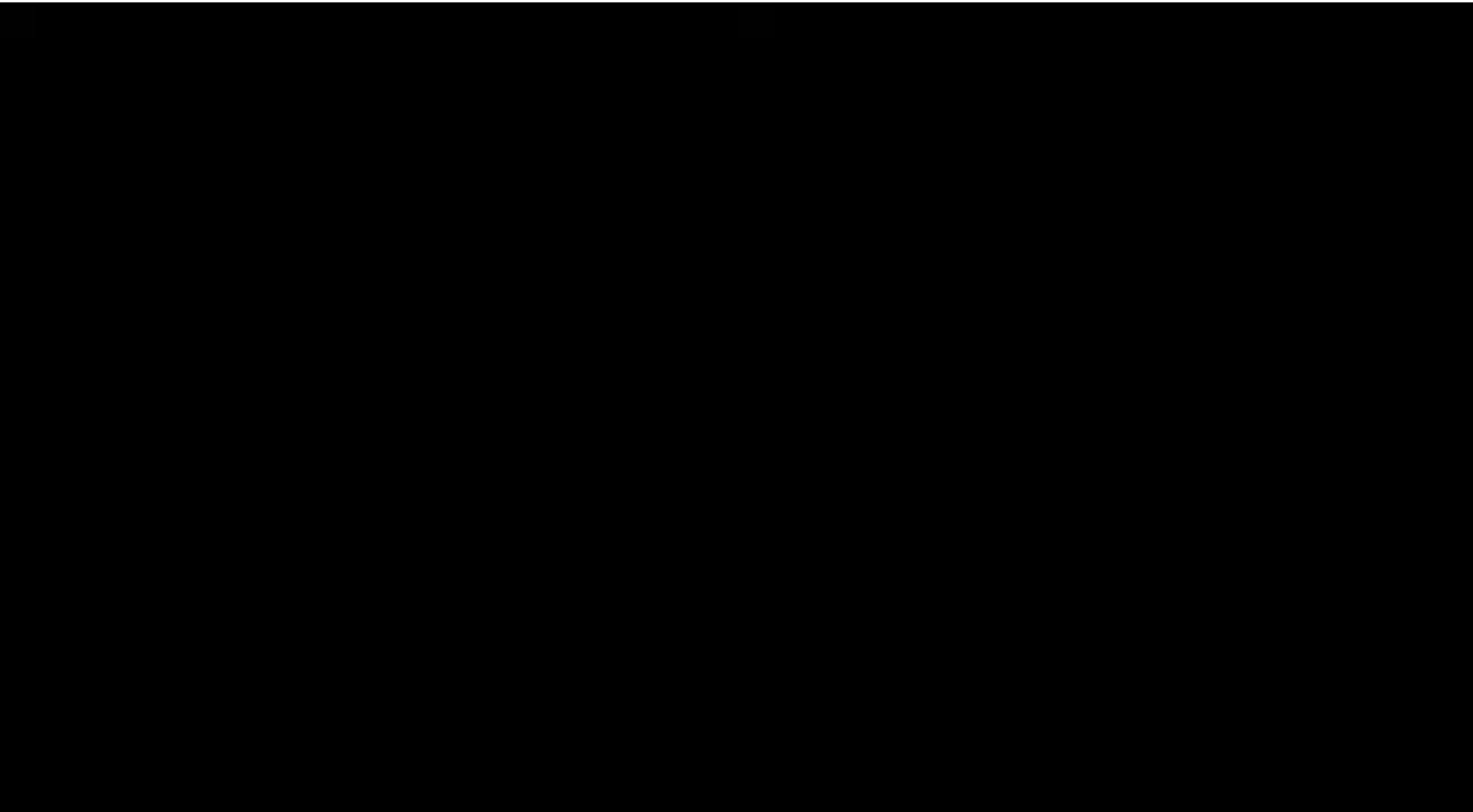


LA COUCHE POOLING

- Similaire à une convolution, sert à réduire la taille de la matrice (et donc la complexité du modèle)
- On utilise typiquement un filtre "max" (*maxpooling*)
 - Permet de réduire la taille des matrices tout en gardant les valeurs plus importantes
- On peut également utiliser la "moyenne"

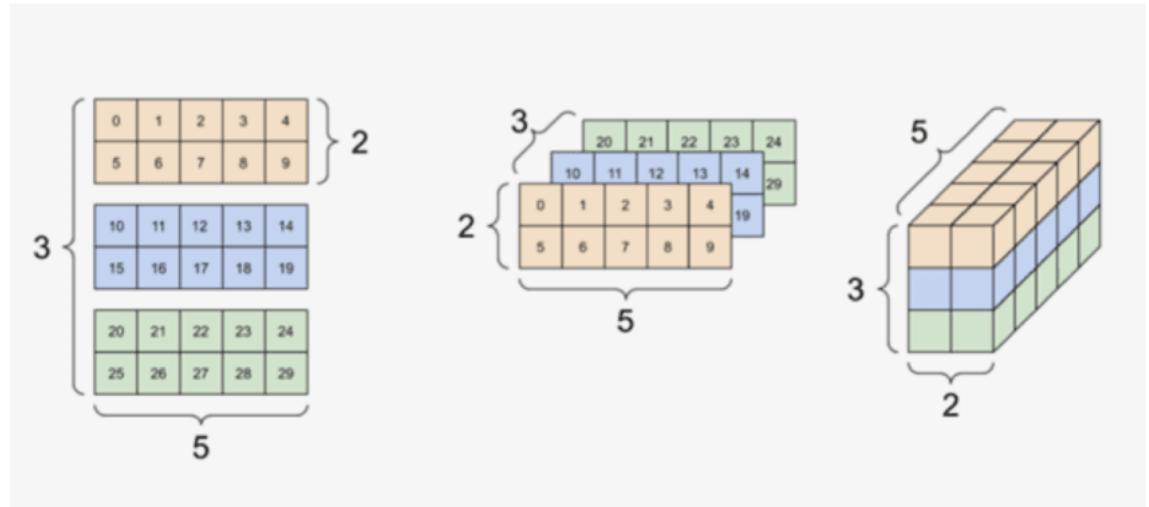


Y A-T-IL UNE PERTE D'INFORMATION ?



TENSOR : LA REPRÉSENTATION DES DONNÉES

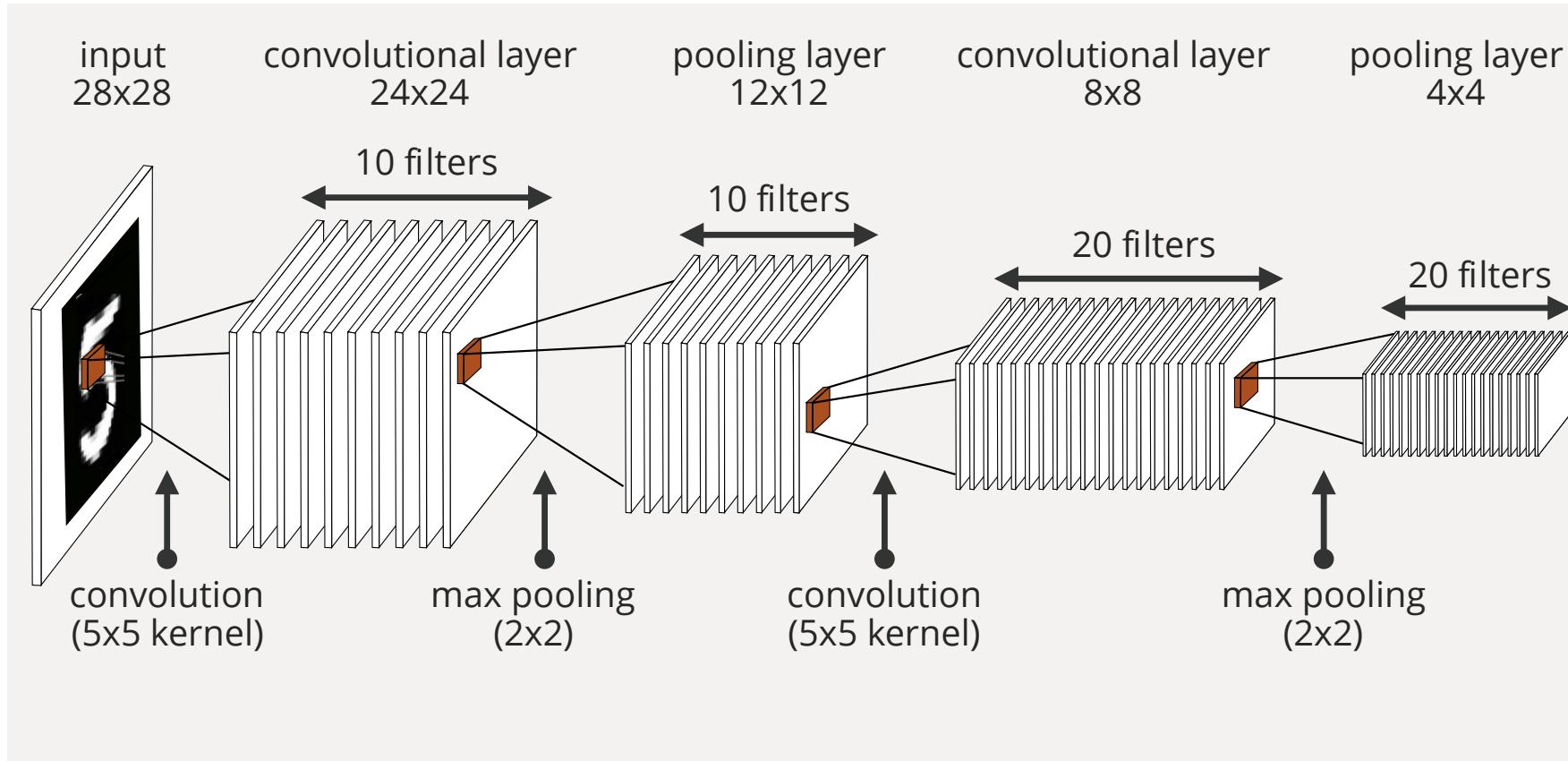
- Toutes les données (entrées, poids, biais, sortie) sont des matrices numériques
 - Une simple image RGB peut être représentée par 3 matrices bidimensionnelles : une matrice 3D
- En mathématique, un **tensor** est un objet dans un espace vectoriel :
 - Ordre 0 = scalaire
 - Ordre 1 = vecteur
 - Ordre 2 = matrice
 - Ordre 3... = tensor
- Cette représentation est donc la base de tout "calcul" dans les réseaux de neurones
 - D'où le nom de certains frameworks comme TensorFlow
- Les frameworks ont souvent des objets "tensor" propres
 - Objets pour le stockage des informations et leur manipulation optimisée (opérations, passage CPU-GPU, etc.)



MAIS PAS QUE ÇA

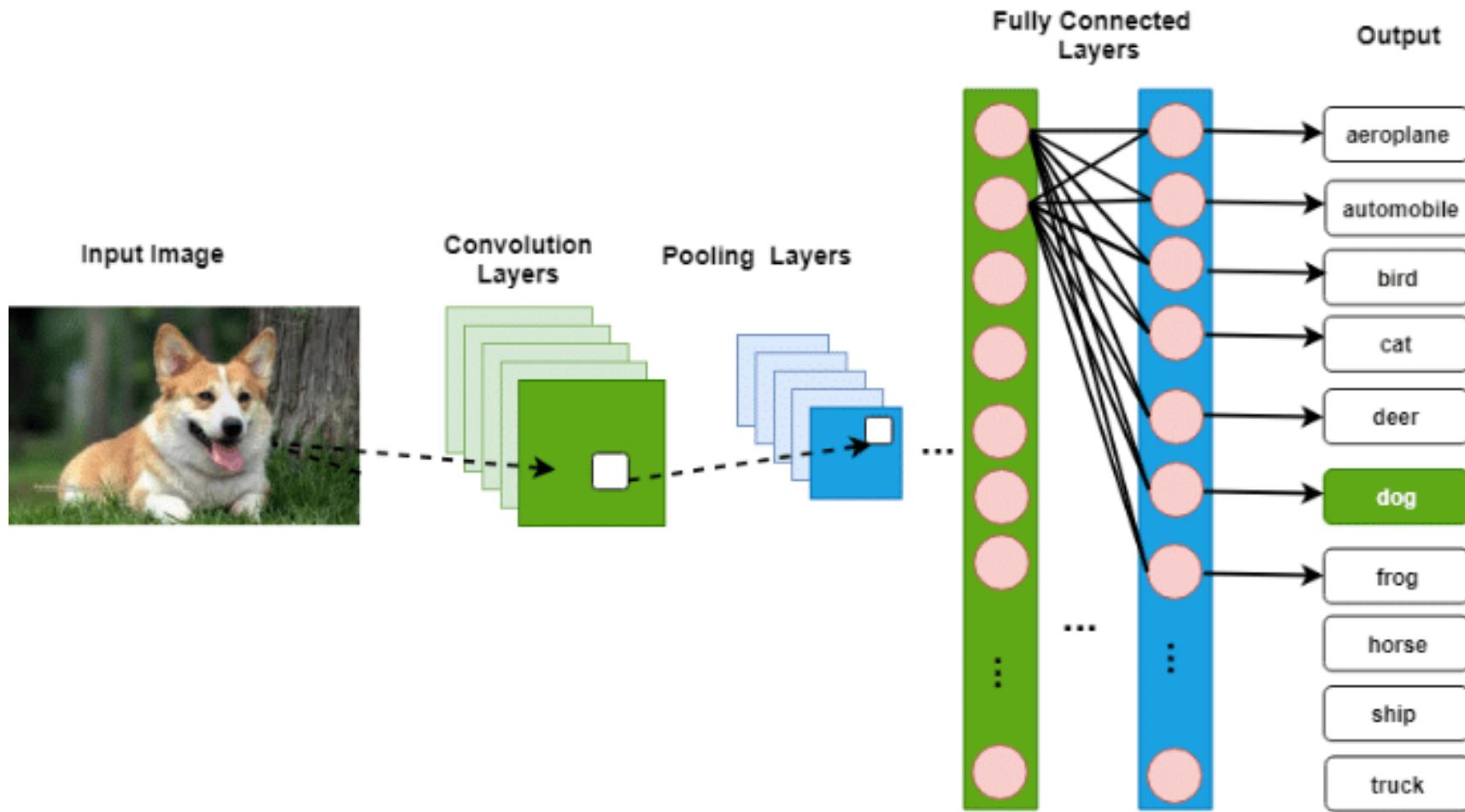
- Tout un tas d'autres éléments sont nécessaires :
 - Fonctions d'activation
 - Fonctions de perte
 - Optimiseurs
 - Fonctions de normalisation et régularisation...
- Les frameworks de programmation se chargent d'offrir tous ces éléments, plus le support à un entraînement et déploiement efficace
 - Gestion CPU-GPU / multiGPUs / multimachines
 - Enregistrement/export dans des formats interchangeables

DEEP LEARNING = ENCHAÎNER PLUSIEURS COUCHES



On arrive à un point où il devient impossible de faire d'autres convolutions (2×2)
Comment transformer ça pour la sortie ?

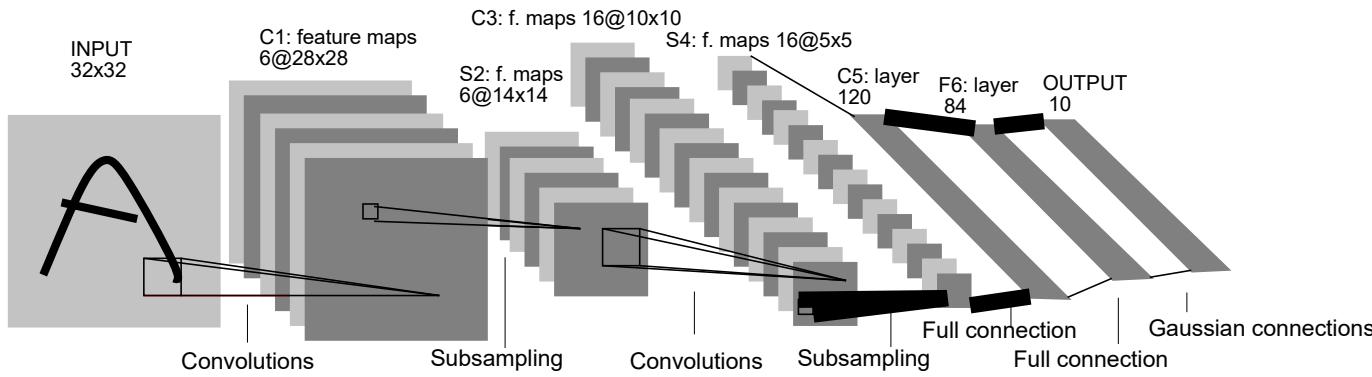
ON APLATIT TOUT ET HOP, UN COUP DE RÉSEAU DENSE POUR FAIRE LA CLASSIFICATION



Exemples de réseaux et utilisations

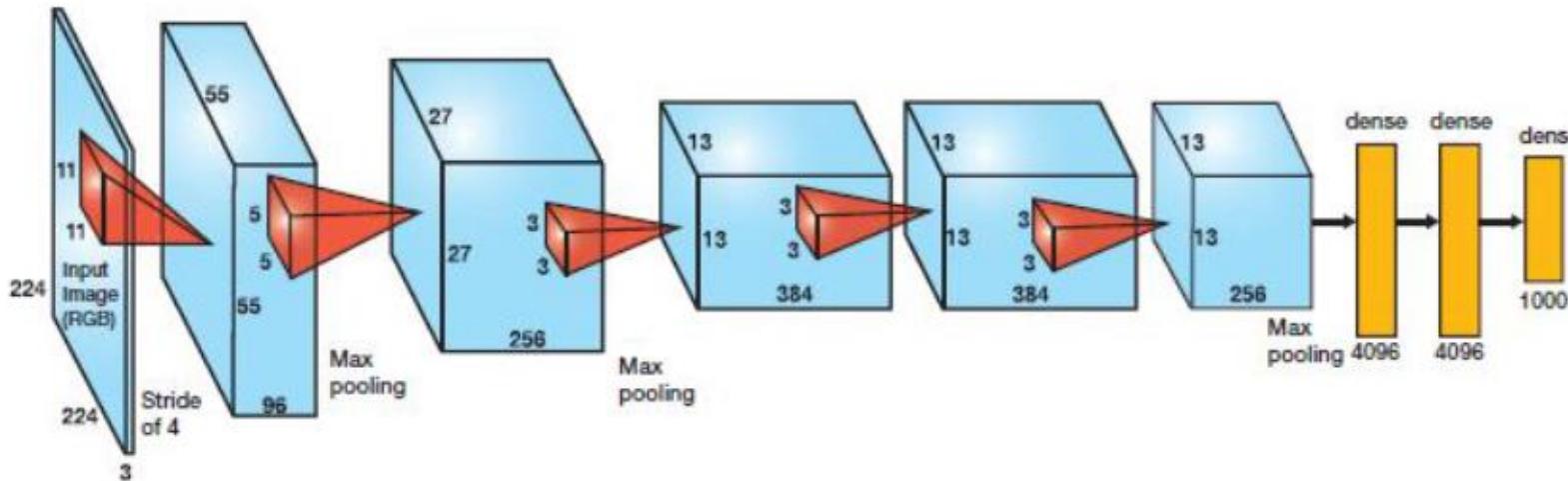
LENET, LE PIONNIER

- Modèle proposé par Yan Lecun en 1998 dans le but de reconnaître le l'écriture (MNIST)
- Peu de convolutions, peu de couches

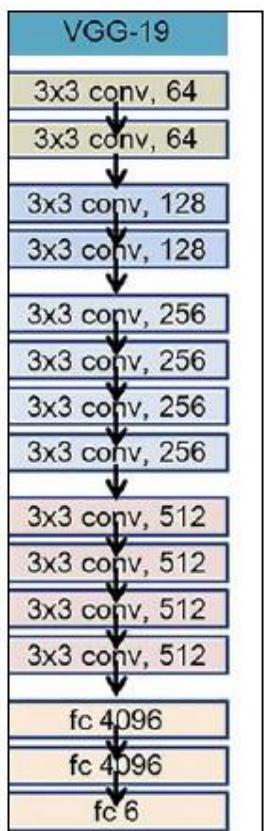


Layer	Type	Maps	Size	Kernel size	Stride	Activation
Out	Fully connected	—	10	—	—	RBF
F6	Fully connected	—	84	—	—	tanh
C5	Convolution	120	1×1	5×5	1	tanh
S4	Avg pooling	16	5×5	2×2	2	tanh
C3	Convolution	16	10×10	5×5	1	tanh
S2	Avg pooling	6	14×14	2×2	2	tanh
C1	Convolution	6	28×28	5×5	1	tanh
In	Input	1	32×32	—	—	—

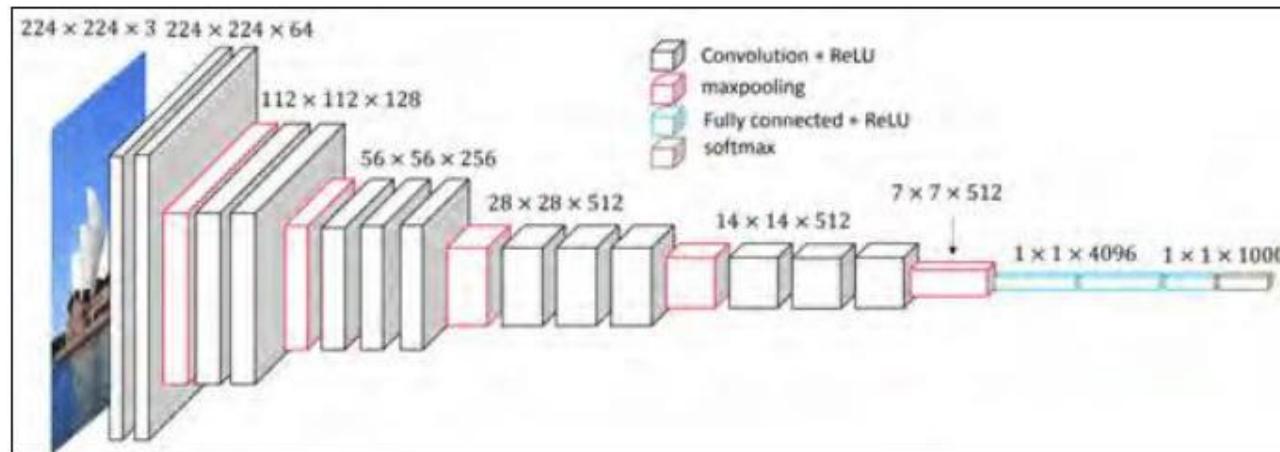
AlexNet



VGG



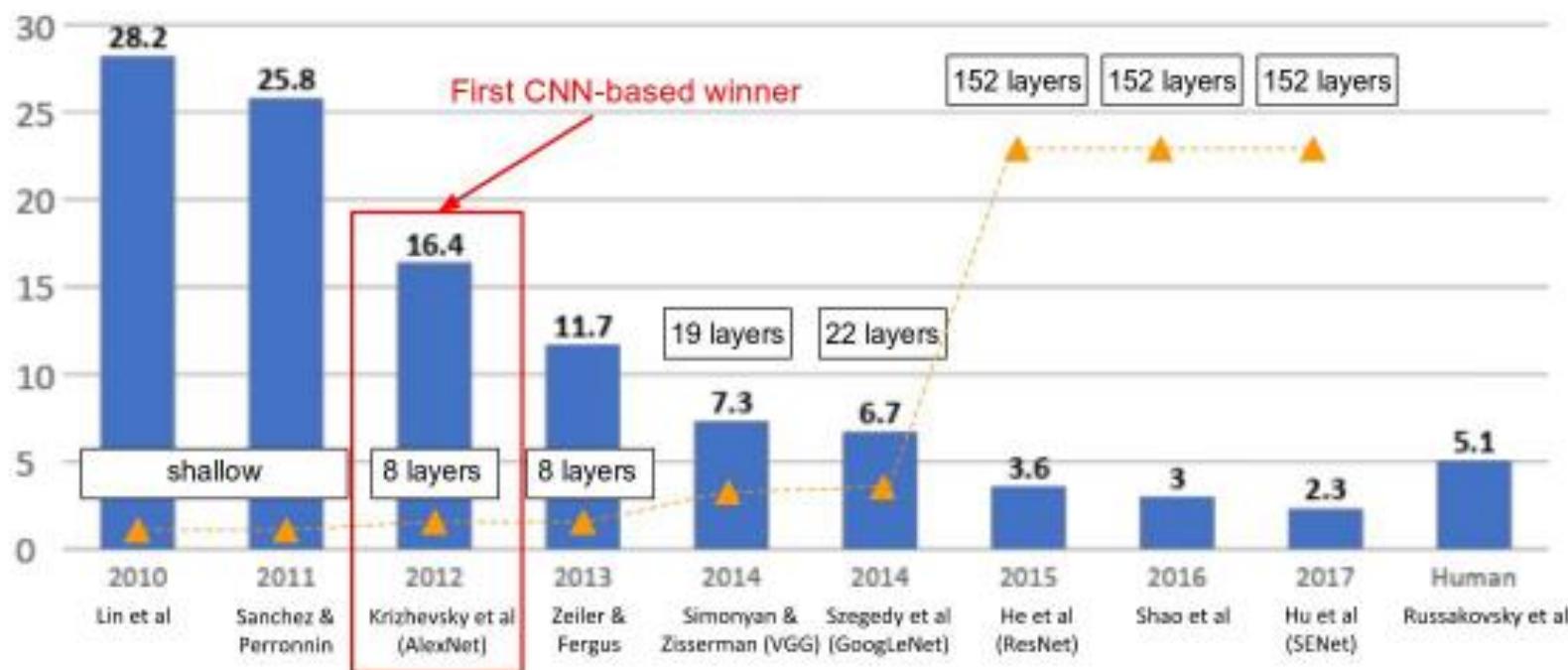
- Gagnant ImageNet 2014
- encore largement utilisé pour le transfert learning



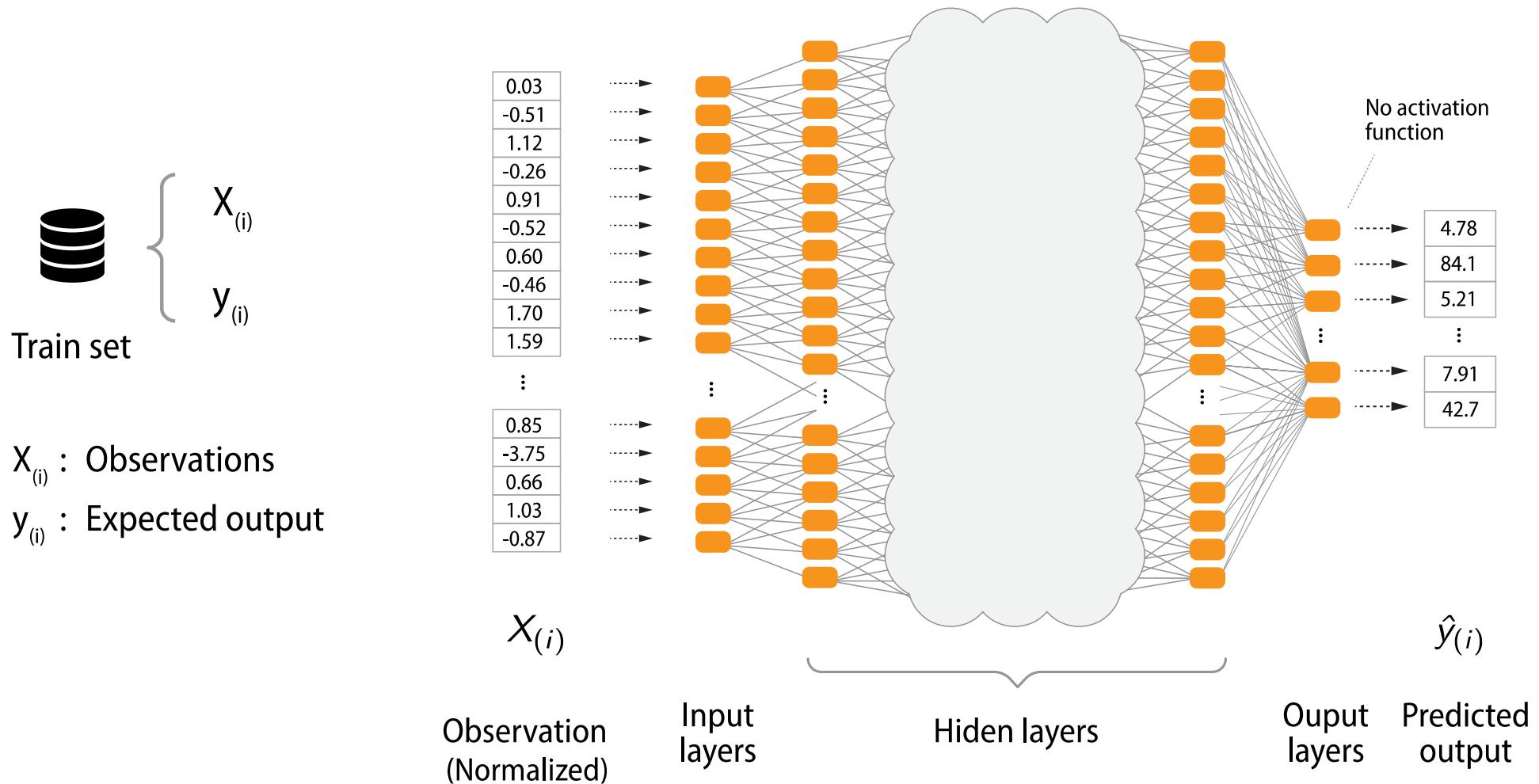
LA RÉVOLUTION DU DEEP LEARNING

- Depuis 2012, des modèles DL se sont imposés dans les challenges de classification d'images

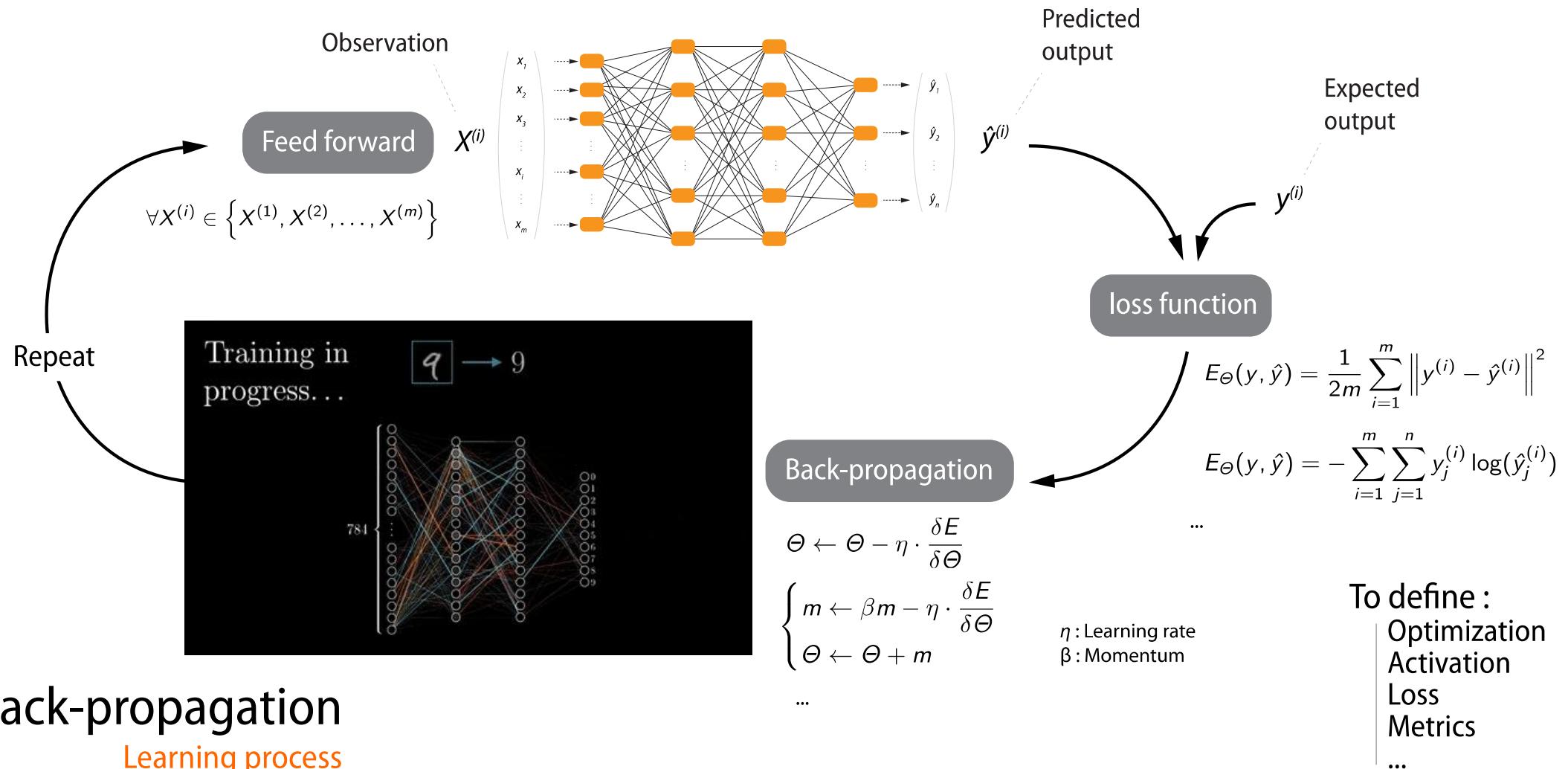
ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



DES PERCEPTRONS AU DEEP LEARNING



ENTRAÎNEMENT PAR RETROPROPAGATION (1986)

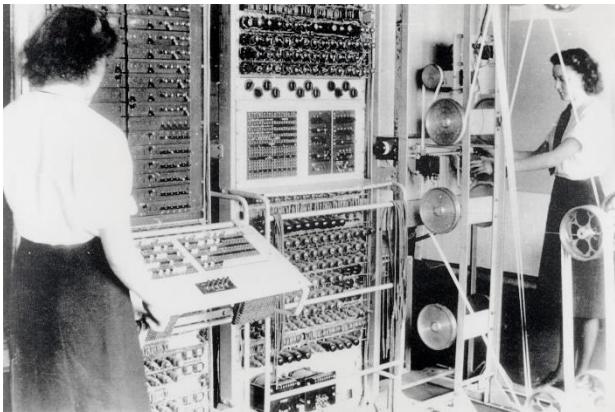


Back-propagation
Learning process

WE'RE GONNA NEED

A GPU

ENJEU : PUISSANCE DE CALCUL



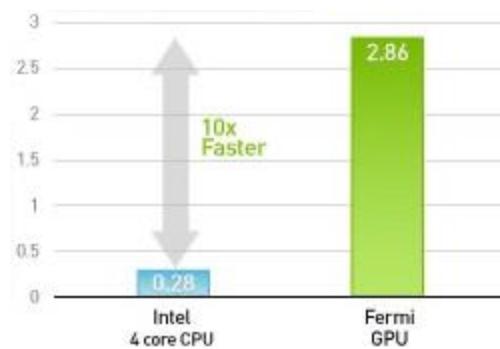
1. Depuis 1943, des supercalculateurs



2. ROMEO, 5 générations de supercalculateurs depuis 2001
Luiz Angelo Steffenel

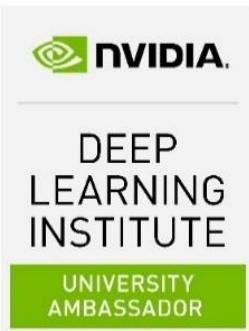


3. ROMEO, 3 et 4, les supercalculateurs GPU

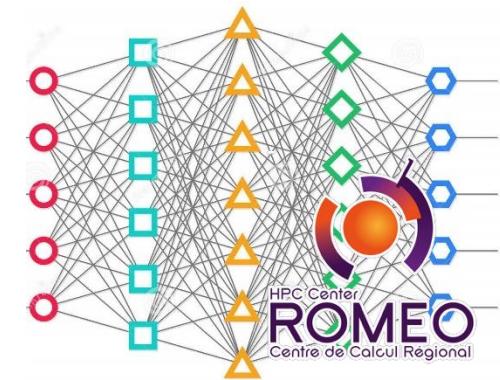


4. GPUs are very good at DeepLearning

ED SNI – Introduction au Deep Learning



5. DLI Certified Educator & University Ambassador



6. ROMEO, Le supercalculateur IA

LE SUPERCALCULATEUR ROMEON [2018]



6,5 M€



Mai 2016 – Octobre 2018



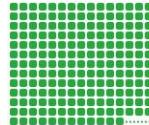
117 serveurs
96 / 192 Go DDR



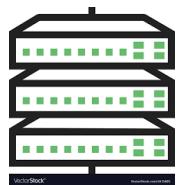
Dernière génération de processeurs Intel Skylake gold 6132



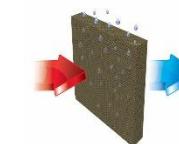
4 espaces de Stockages



4x NVIDIA Tesla P100/16GB SXM2 Nvlink - 280 GPU (+ 2 serveurs V100)



Réseau BXI nouvelle génération.
100 / 200 Gbits



Direct Liquid Cooling
Free Cooling adiabatique

Juin 2018



ED SNI – Introduction au Deep Learning

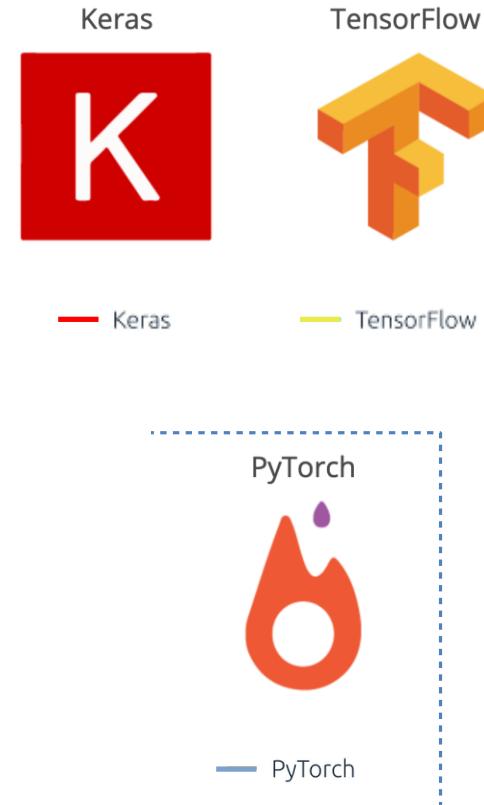
- Interconnexion innovante
- Processeurs GPU optimisés HPC et IA
- Eco-efficacité



Comment programmer un
Deep Learning ?

DEUX GRANDS FRAMEWORKS

- Scikit Learn est une référence pour des algorithmes machine learning classiques
 - Régressions, arbres de décision, SVM
- Il n'a pas été développé pour explorer les réseaux de neurones
- En ce moment, deux frameworks dominent le "marché" du Deep Learning
- Tensorflow
 - Initialement développé par Google
 - Bien testé et disposant d'une base importante d'outils
 - Dispose d'une API "haut niveau" : Keras
- PyTorch
 - Initialement développé par Facebook/Meta
 - Plus récent, support croissant
 - Quelques APIs permettent un usage haut niveau
 - Fast.ai, PyTorch Lightning



Keras

- Keras a été créé comme une interface "haut niveau" pour faire des réseaux de neurones
 - Jusqu'à la version 2.3 il supportait plusieurs types de bibliothèques
 - TensorFlow
 - Microsoft Cognitive Toolkit (CNTK)
 - Theano
 - PlaidML
 - Depuis la version 2.4 (Juin 2020), support prioritaire Tensorflow
 - Ça rend Tensorflow un peu plus simple...
- La version 3 de Keras supporte également PyTorch

```
hidden1      = 100
hidden2      = 100

model = keras.Sequential([
    keras.layers.Input((28, 28)),
    keras.layers.Flatten(),
    keras.layers.Dense(hidden1, activation='relu'),
    keras.layers.Dense(hidden2, activation='relu'),
    keras.layers.Dense(10,      activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

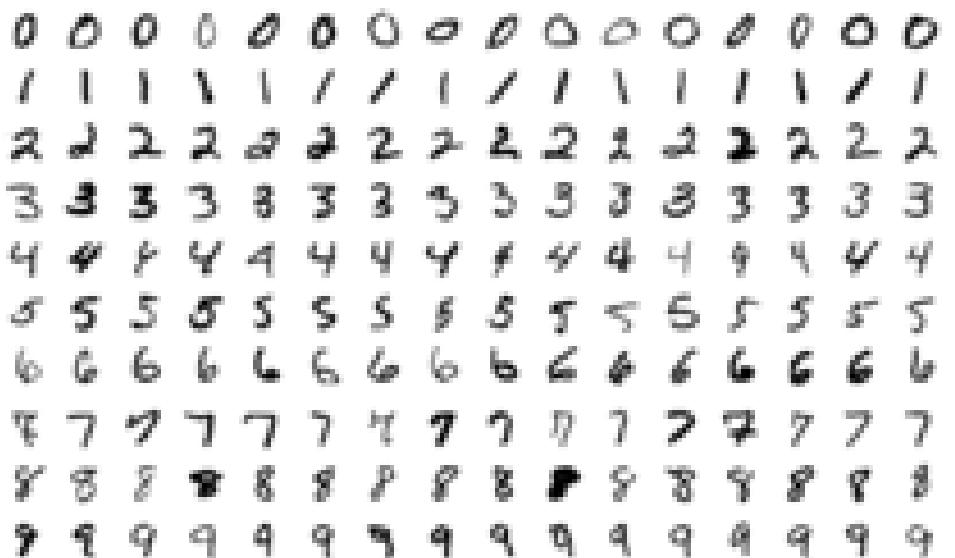
batch_size   = 512
epochs       = 16

history = model.fit(x_train, y_train,
                     batch_size      = batch_size,
                     epochs         = epochs,
                     verbose        = 1,
                     validation_data = (x_test, y_test))

score = model.evaluate(x_test, y_test, verbose=0)
```

EXERCICE 2

- Ici, nous allons apprendre à programmer avec Keras en reproduisant l'expérience d'Yan Lecun en 1998
 - Le dataset MNIST
- Utilisation de réseaux denses (DNN)
- Accéder le site
- <https://t.ly/5WWz>



EXERCICE 3

- Et si on faisait de la classification de vêtements ?
 - Le dataset Fashion-MNIST (Zalando)
 - Utilisation de réseaux convolutionnels (CNN)
-
- Accéder le site
 - <https://t.ly/a0IY>

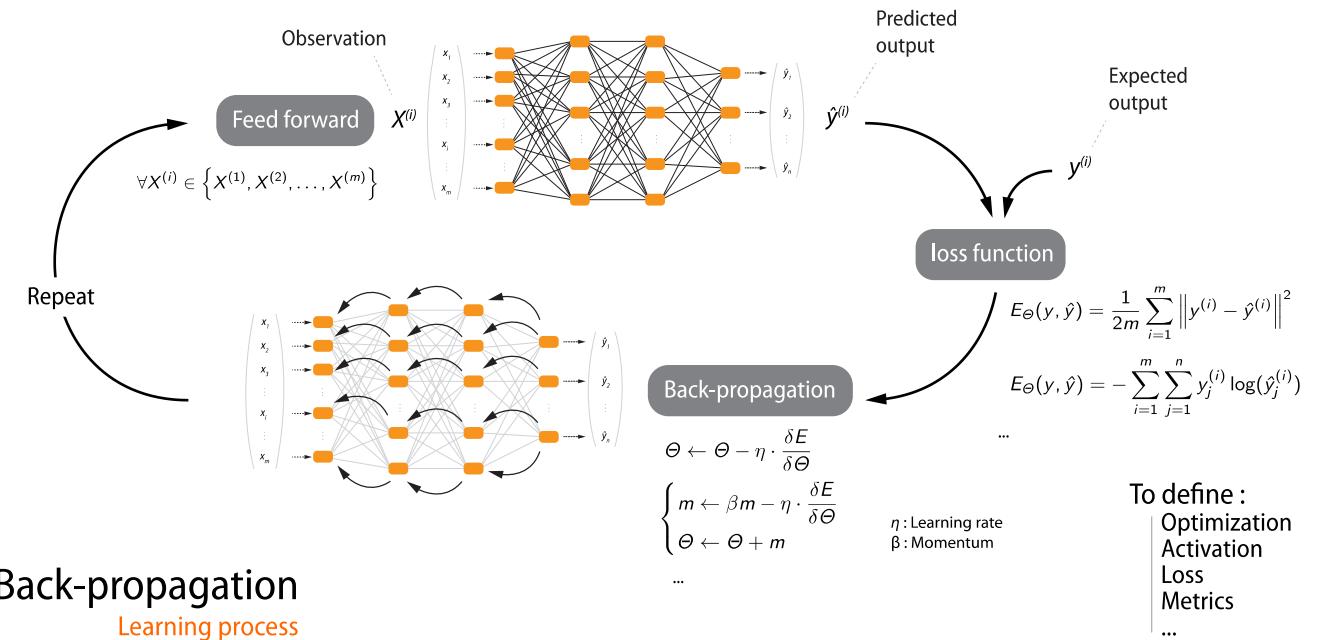


Reposer sur les
épaules
des géants



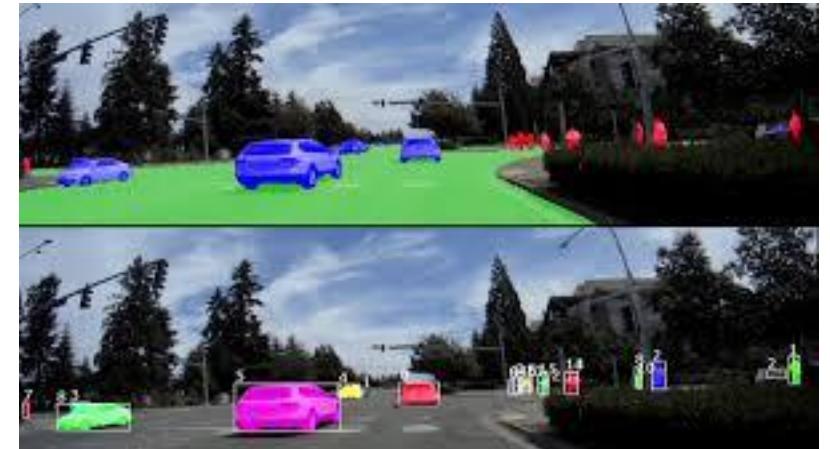
COÛT D'UN ENTRAÎNEMENT COMPLET

- La rétropropagation s'effectue sur l'ensemble des couches
- Opération lourde, surtout si le réseau est profond et le dataset grand et complexe
 - Très lent sans une GPU
- La question cependant est :
 - Faut-il entraîner un modèle du zéro à chaque fois ??
 - Après tout, les couches inférieures détectent des "patrons" bien connus



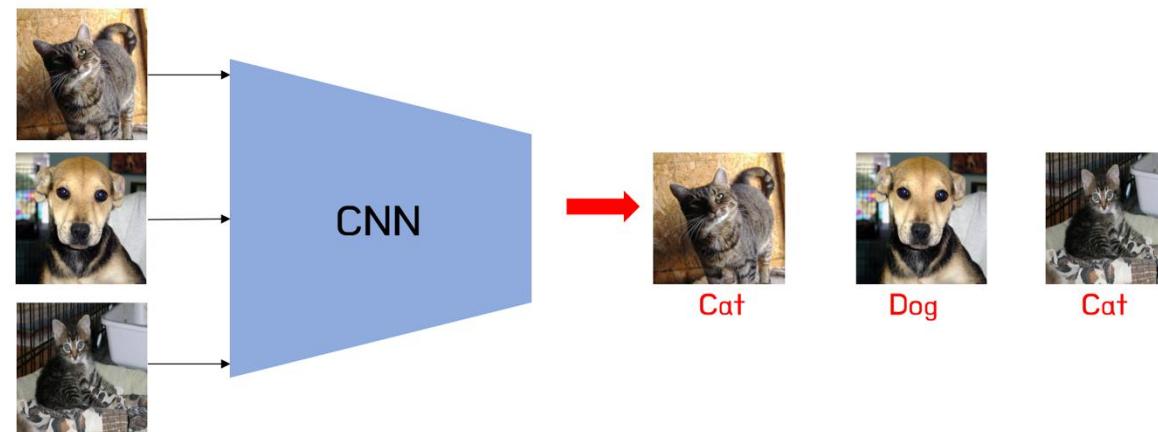
OPTION 1 : UTILISER UN MODÈLE PROCHE

- Plusieurs auteurs proposent des modèles extensivement entraînés sur des datasets publics
 - Ex : ImageNet -> 1000 classes, des millions d'images
- On peut facilement "emprunter" un modèle prêt et juste utiliser les classes qui nous intéressent
- Plusieurs sources disponibles
 - Tensorflow Hub
 - Nvidia NGC
 - Github
 - Une recherche sur Google...



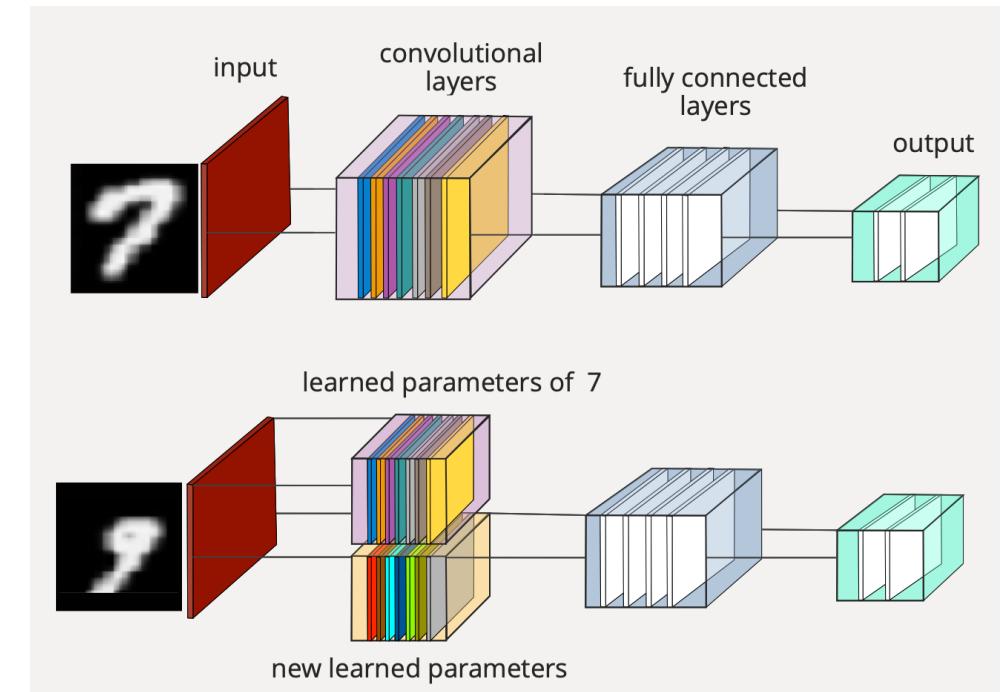
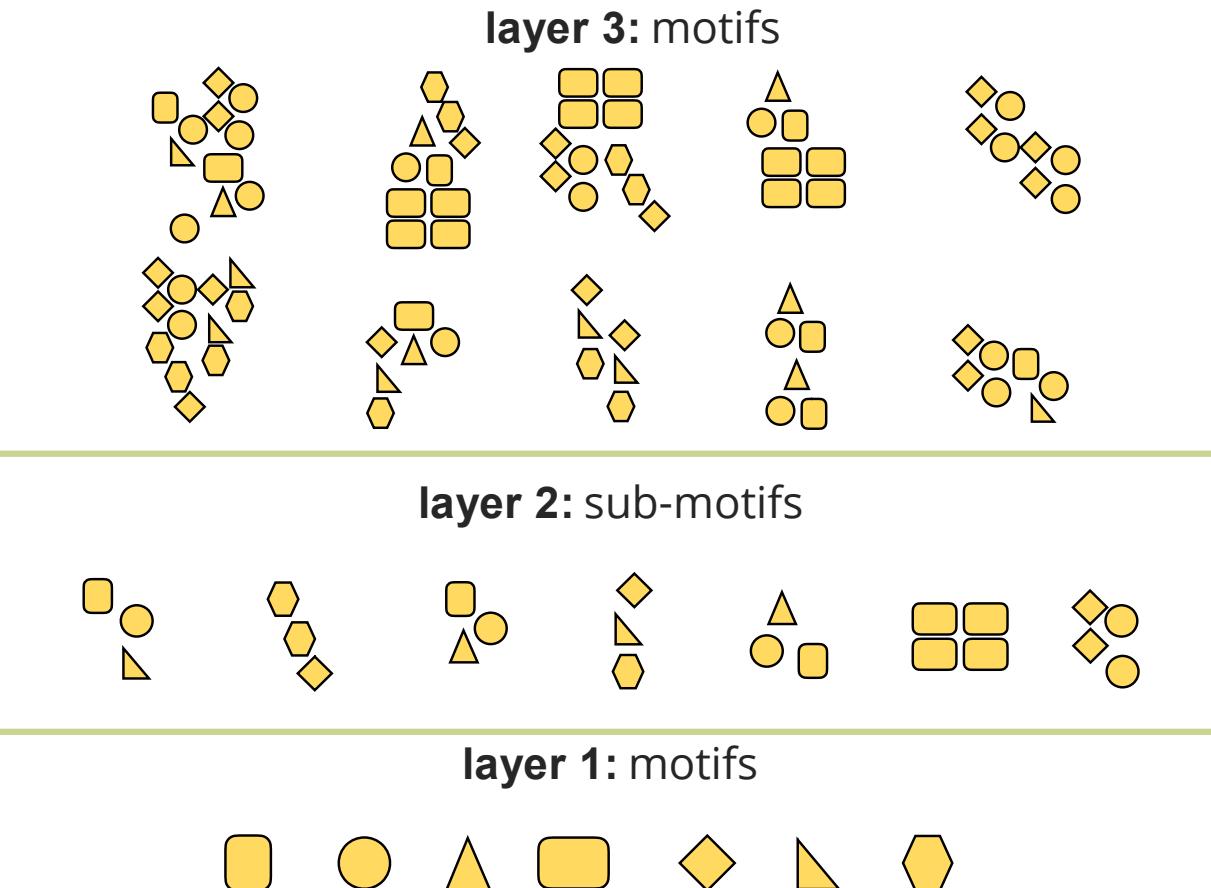
EXERCICE 4

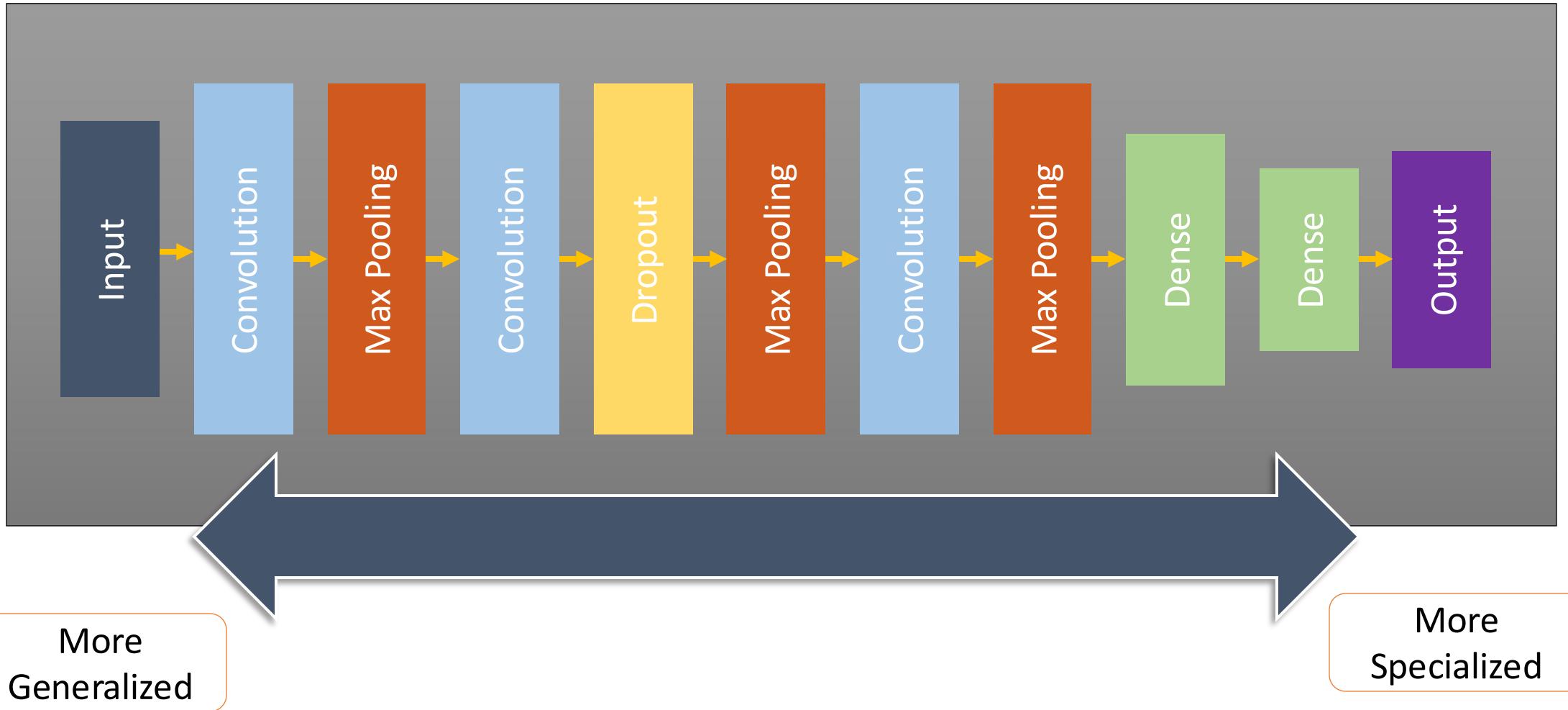
- Utiliser un modèle existant pour identifier Chiens x Chats
- Utilisation du modèle VGG 16
- Accéder le site <https://t.ly/ygyG>



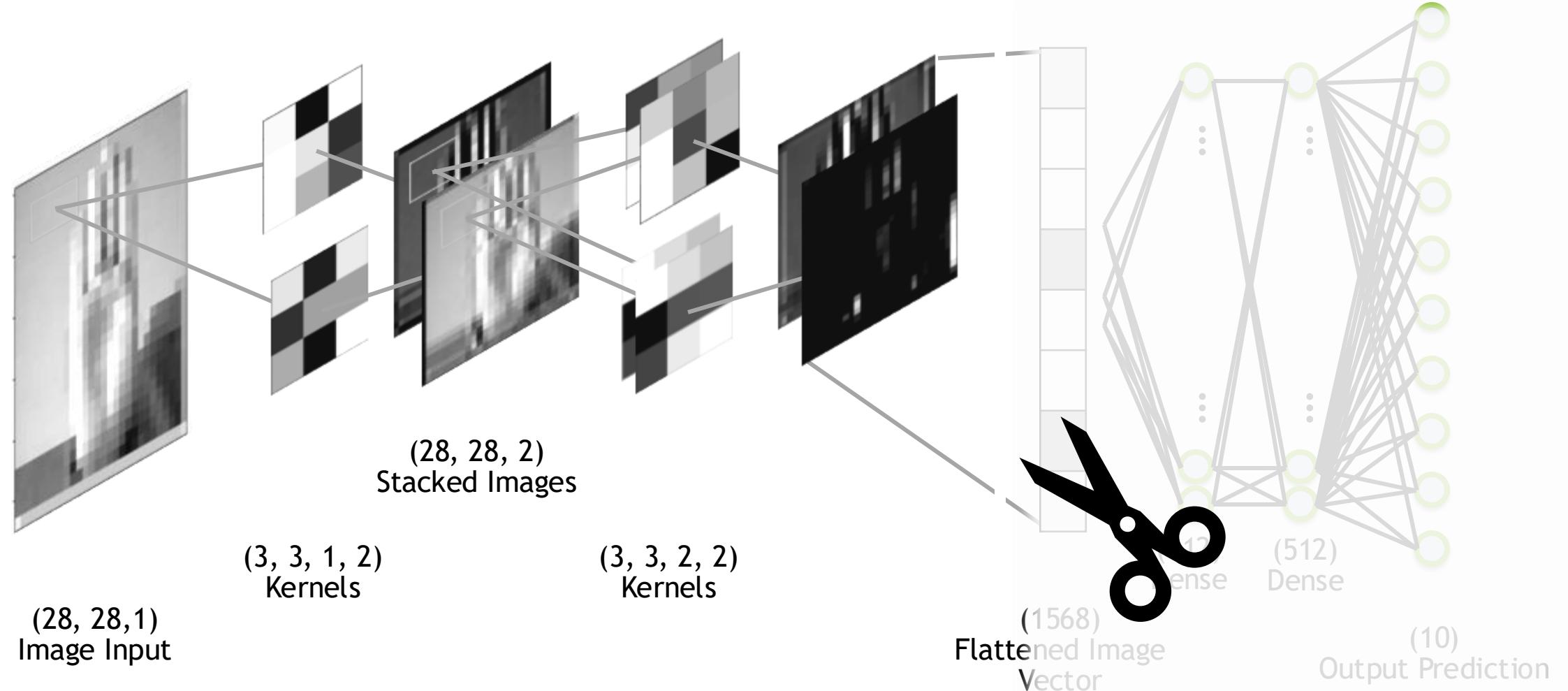
OPTION 2 - TRANSFER LEARNING

- Les avantages d'une organisation hiérarchique des features





TRANSFER-LEARNING À PARTIR DE MODÈLES PRÉ-ENTRAÎNÉS



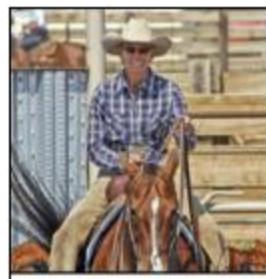
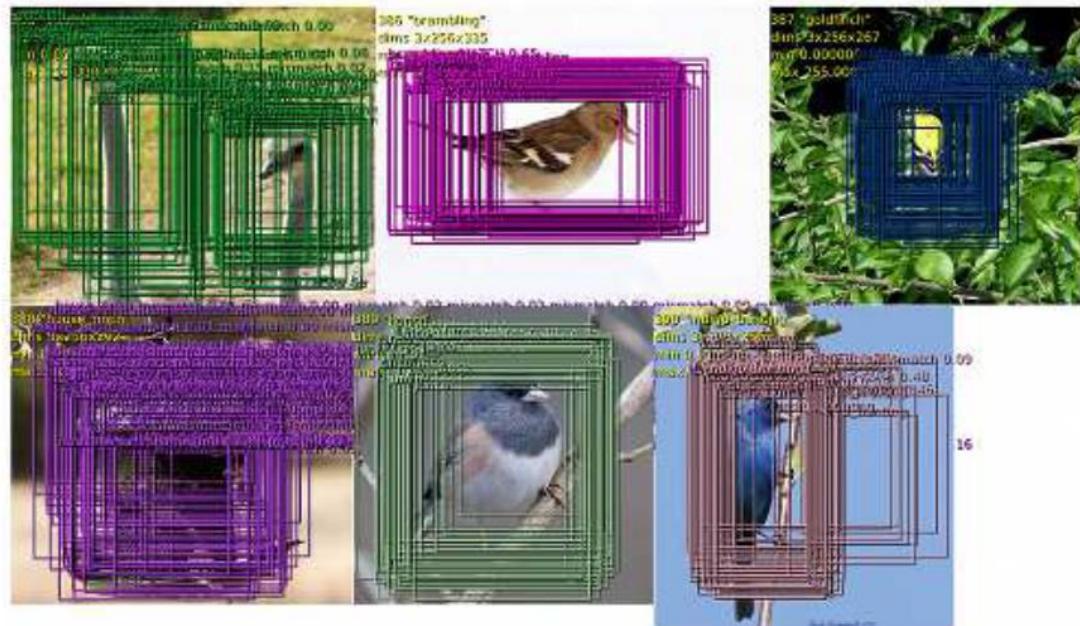
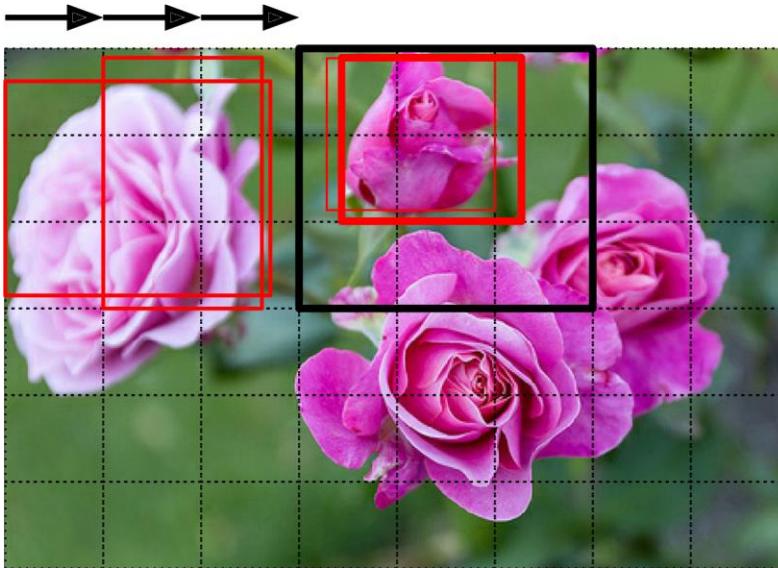
EXERCICE 5

- Mieux qu'identifier Chiens vs Chats, détecter un chien spécifique !!
 - Bo, le chien de Barack Obama
- Accéder le site <https://t.ly/axam->

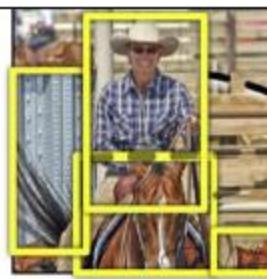


CLASSIFICATION, MAIS PAS SEULEMENT

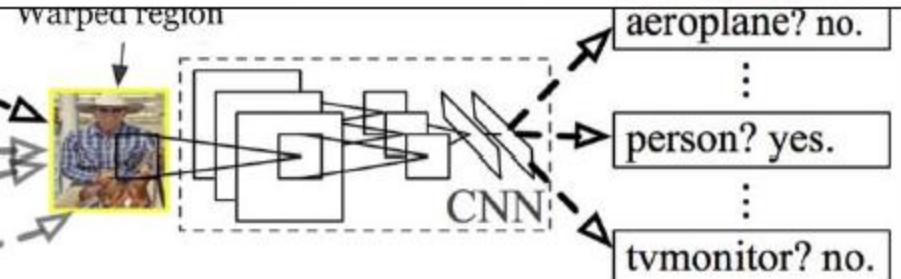
- Si les CNN excellent dans la classification d'images, on peut les détourner à d'autres usages
- Ex : La détection d'objets



1. Input images



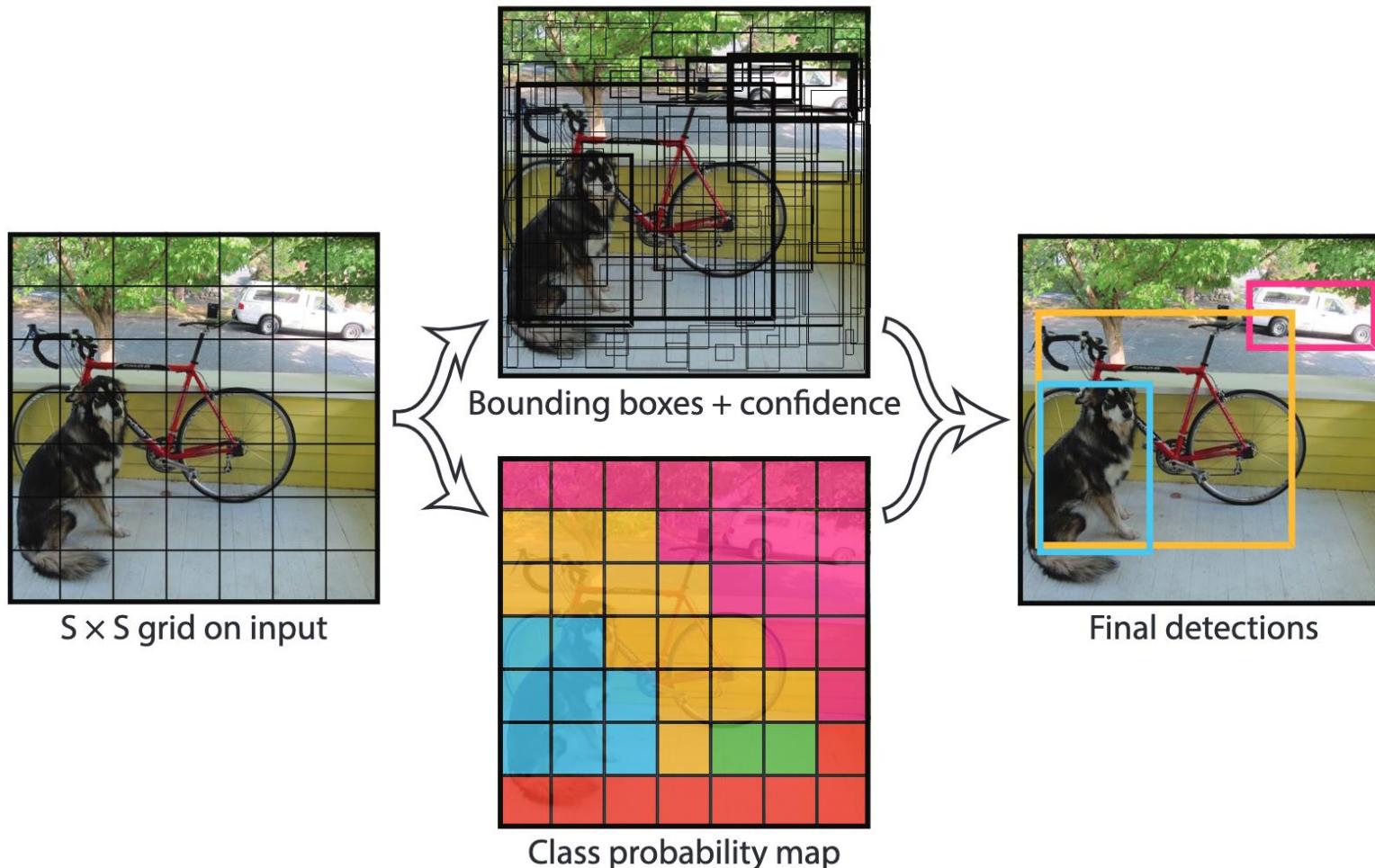
2. Extract region proposals (~2k)



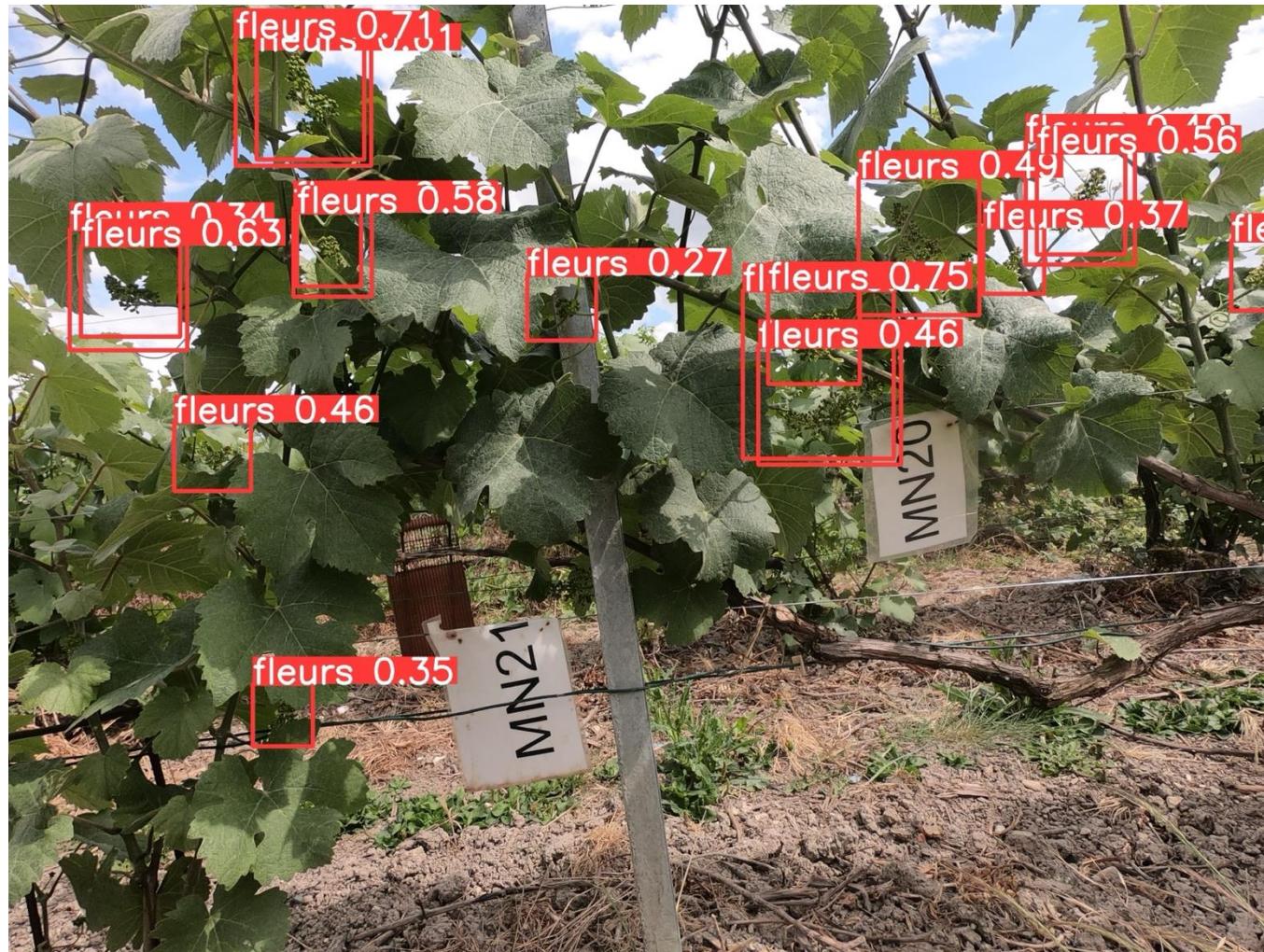
3. Compute CNN features

4. Classify regions

Ex : YOLO (YOU ONLY LOOK ONCE)



(Redmon et al., 2015)

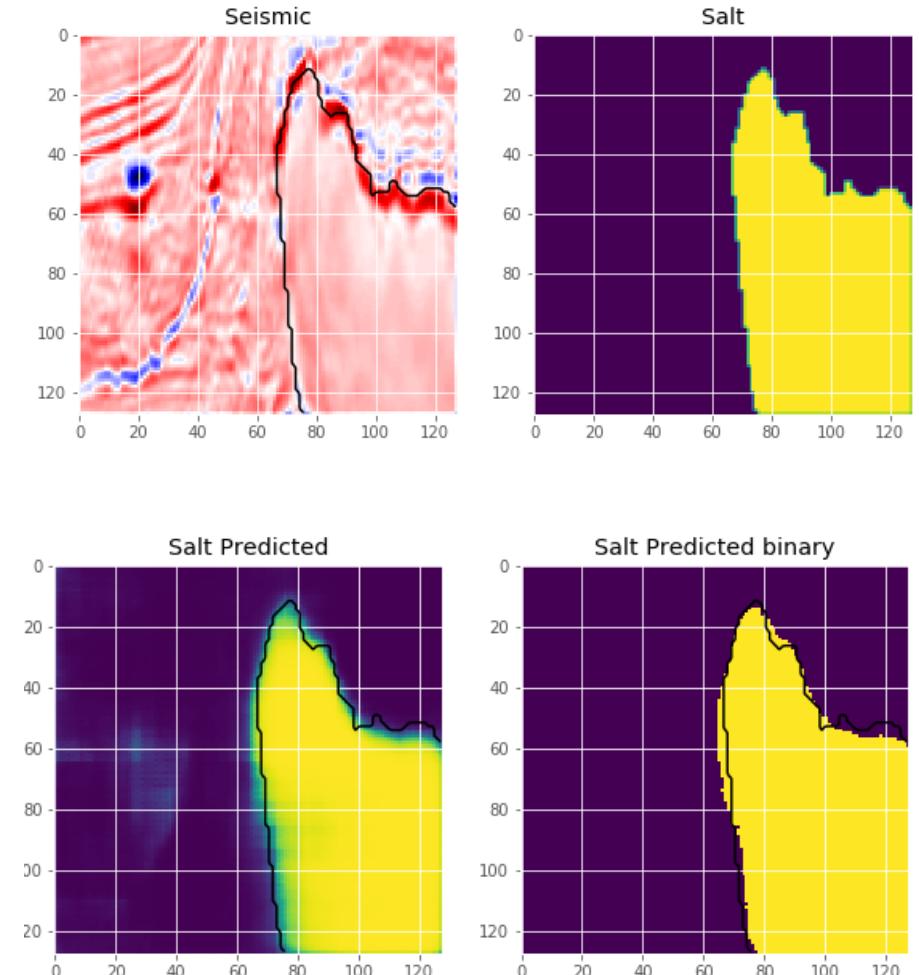






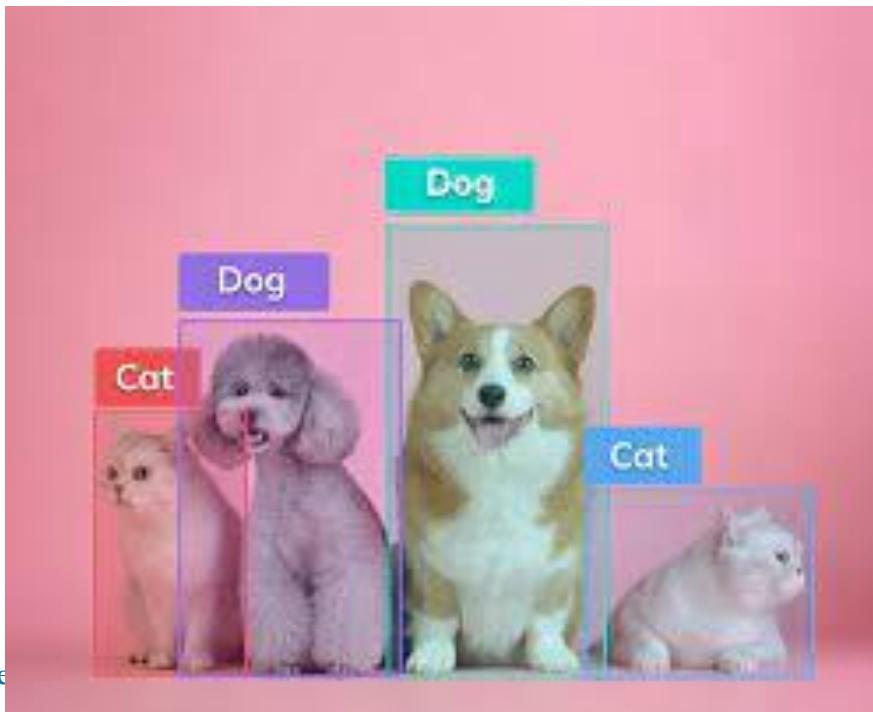
SEGMENTATION SÉMANTIQUE

- Chaque pixel est classé individuellement
- Les labels sont des images avec masques (classes détournées)



EXERCICE 6

- Faire de la détection d'objets et la segmentation d'images avec YOLO
- Accéder le site <https://t.ly/4Zg9i>



EXERCICE FINAL (DEVOIR ?)

- Aider à organiser la défense civile
- Grâce à des images aériennes, identifier des zones touchées par un ouragan
- Comparer la performance de différents modèles de classification
- Accéder le site <https://t.ly/M5Bs1>



C'EST LA FIN POUR AUJOURD'HUI

- On espère que cela vous a plu
- Il y encore beaucoup de choses qu'on n'a pas couvert
 - Séries temporelles (prévisions)
 - Génération de données (GAN, diffusion stable)
 - Modèles linguistiques (ChatGPT)
- N'hésitez pas à nous contacter si vous voulez essayer quelque chose avec vos données de thèse
 - Le LICIIS a des chercheurs qui peuvent vous aiguiller
 - ROMEO a tout le matériel nécessaire pour tourner vos expériences
- Tout le matériel de ce cours restera disponible sur :
 - <https://github.com/lsteffenel/ED-SNI-IntroDL>
- Contact :
 - Angelo.steffenel@univ-reims.fr

