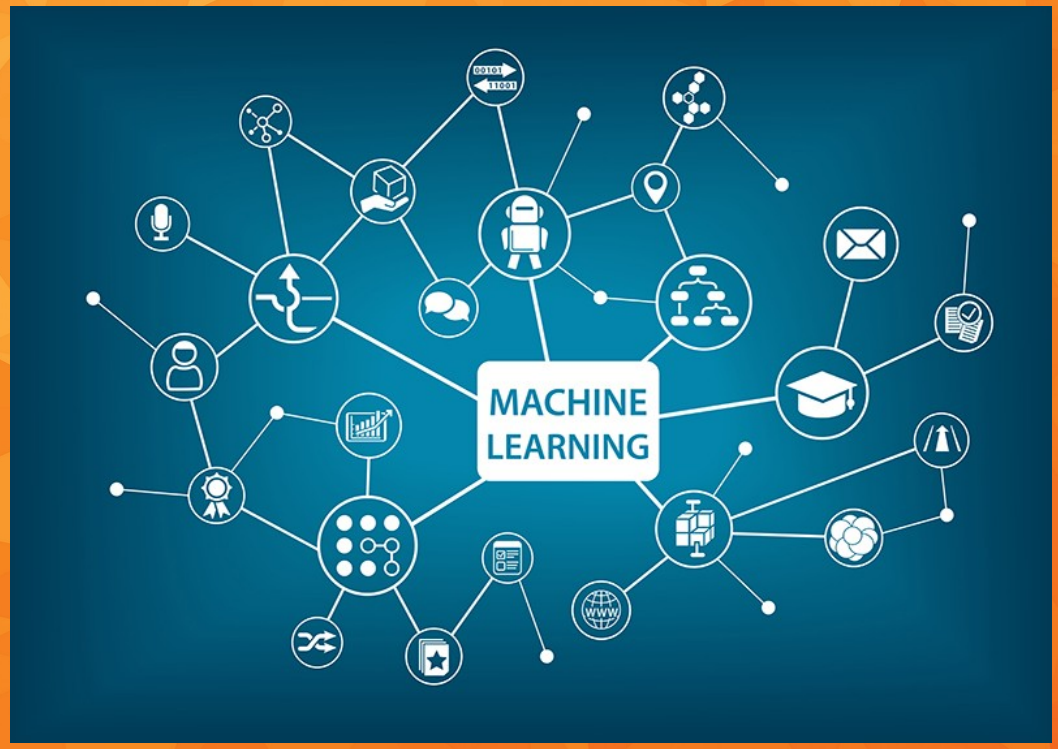
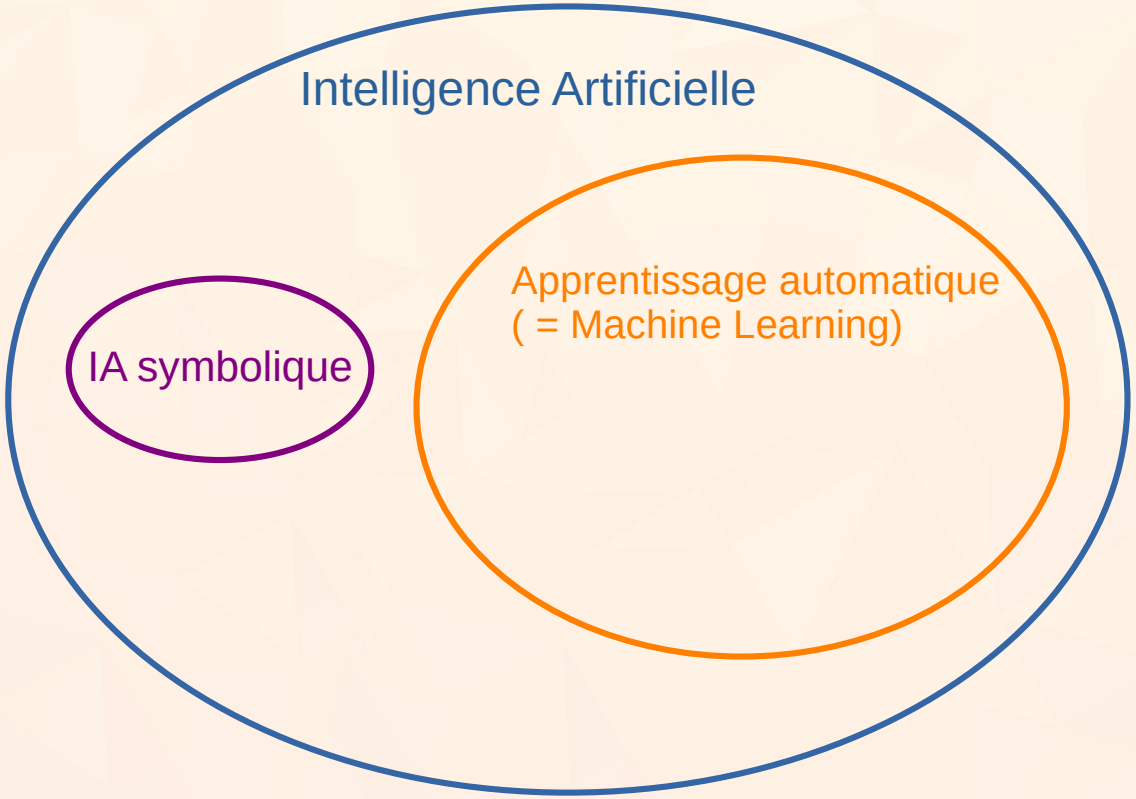
Machine Learning

Introduction à la théorie et bonnes pratiques



Sommaire

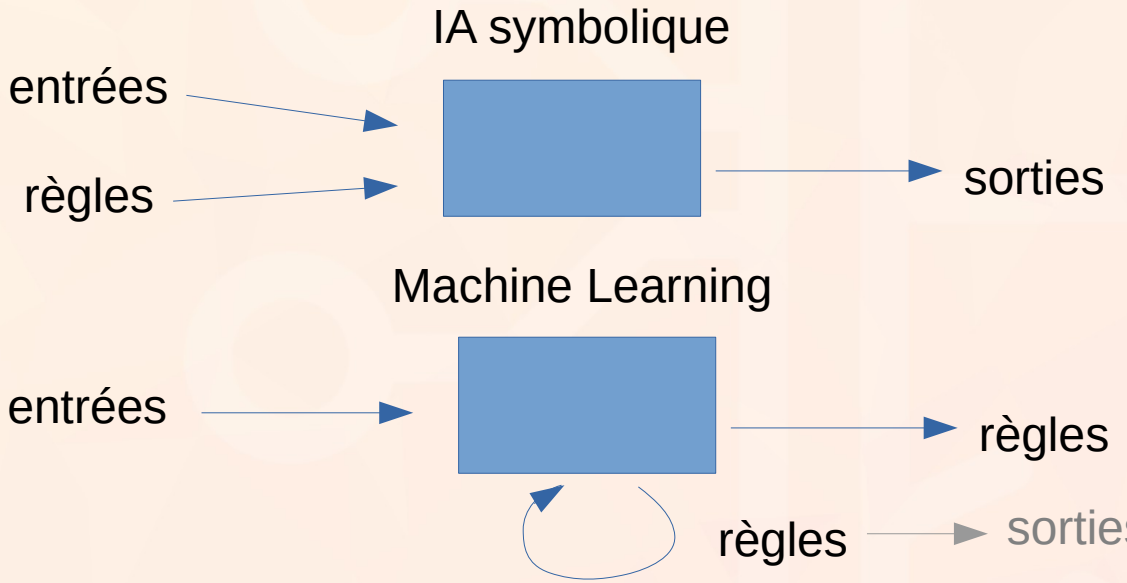
I - Définition



Intelligence artificielle : Science des théories et techniques visant à créer des machines/programmes qui font preuve d'intelligence (vague)

IA symbolique : résolution par manipulation logique de symboles

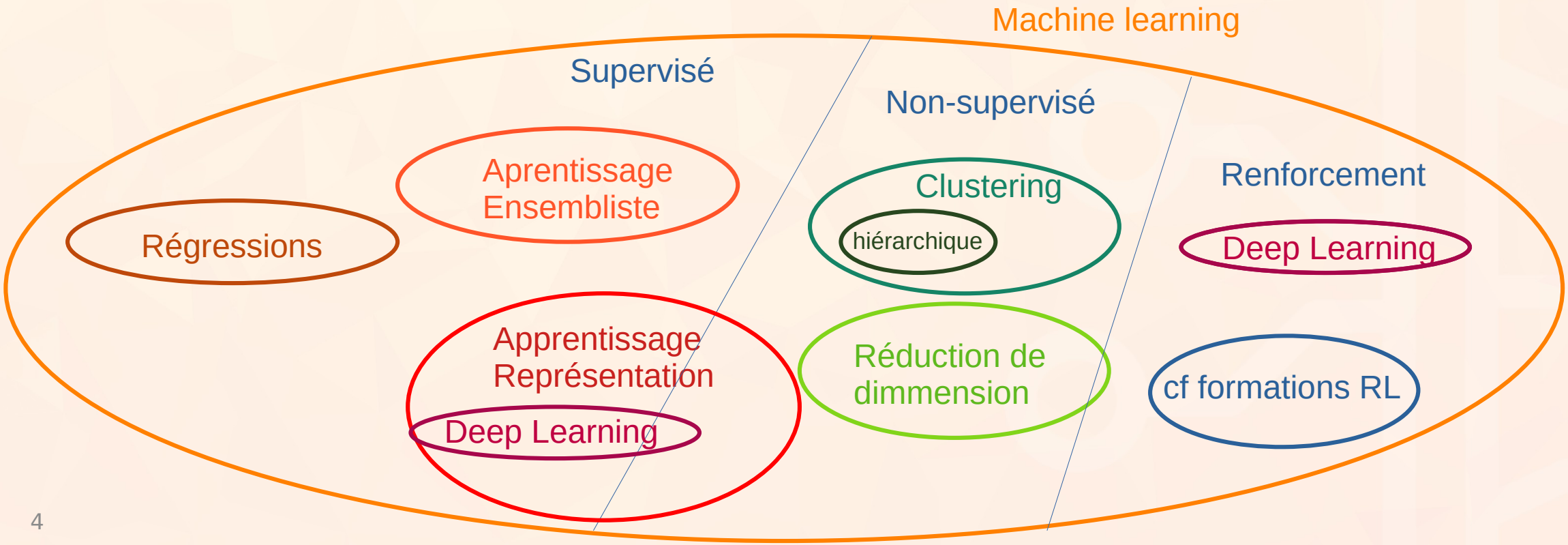
Machine Learning : résolution par confrontation et amélioration avec des données



II - Types et sous-catégories

3 types de problèmes :

- Supervisé : entrées, labels
- Non-supervisé : entrées, pas de labels
- Renforcement : récompenses situationnelles



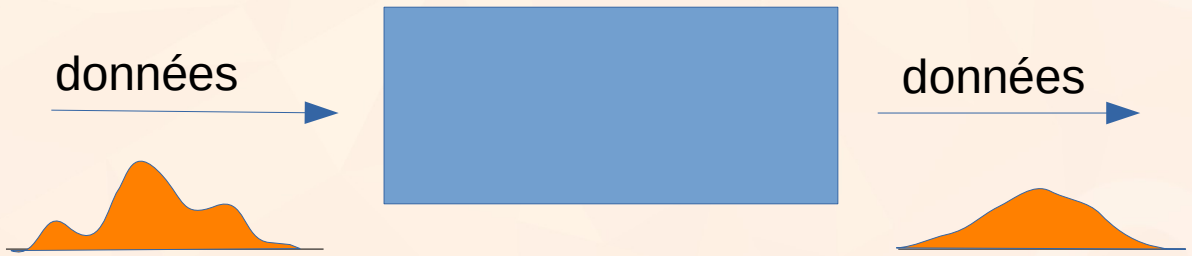
III - Données – Loi de Proba

Les données

Censées représenter la réalité :

- Modélisation incomplète
- Observabilité incomplète

→ données = variables aléatoires, lois de probabilités

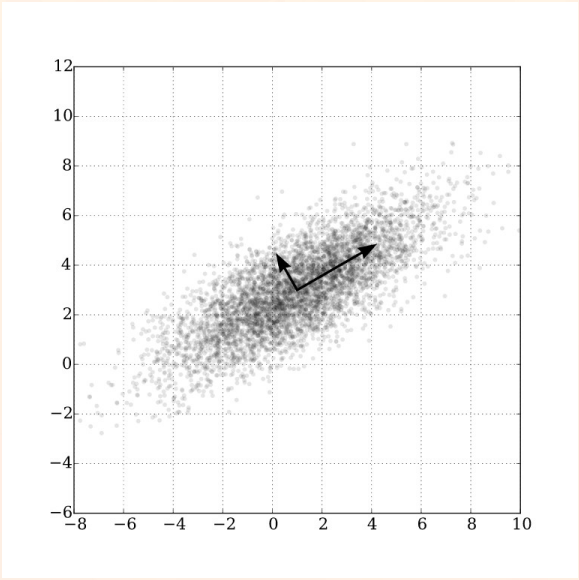


IV - Non-supervisé

Objectif : modifier la représentation des données

3 types :

- Représentation par réduction dimension : projection sans perte
- Représentation indépendante : éliminer les dépendances des axes
- Représentation parcimonieuse : données clairsemées (0 & 1)



PCA (Scikit-learn)

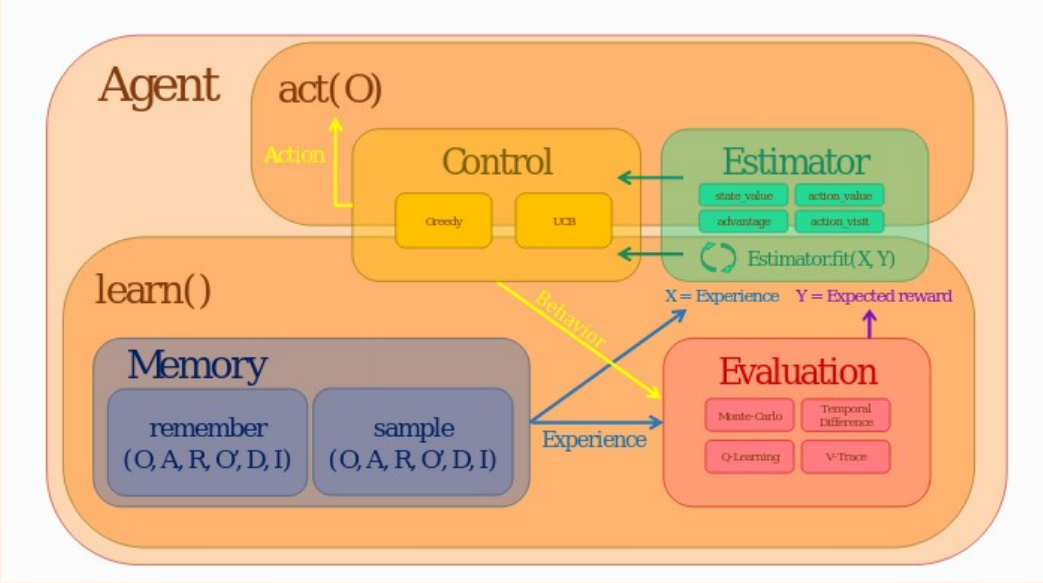
A comparison of the clustering algorithms in scikit-learn

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters or distance threshold	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters or distance threshold, linkage type, distance	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
OPTICS	minimum cluster membership	Very large `n_samples`, large `n_clusters`	Non-flat geometry, uneven cluster sizes, variable cluster density	Distances between points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large `n_clusters` and `n_samples`	Large dataset, outlier removal, data reduction.	Euclidean distance between points

Clustering (Scikit-learn)

V - Apprentissage Renforcement

- Implémentez votre environement (états, actions, récompenses, observations) avec Gym par exemple
- Implémentez votre agent avec une bibliothèque (LearnRL, Horizon, Dopamine)
- Tester les différentes possibilités !



VI – Supervisé

Données : entrées & labels

Entrées

Labels

$$X = \begin{matrix} x_{1,1} & x_{1,2} & - & x_{1,d} \\ x_{2,1} & x_{2,2} & - & x_{2,d} \\ | & | & & | \\ x_{n,1} & x_{n,2} & - & x_{n,d} \end{matrix} \quad Y = \begin{matrix} y_{1,1} & y_{1,2} & - & y_{1,r} \\ y_{2,1} & y_{2,2} & - & y_{2,r} \\ | & | & & | \\ y_{n,1} & y_{n,2} & - & y_{n,r} \end{matrix}$$

Supervisé – Lois usuelles

Entrées

$$X = \begin{matrix} x_{1,1} & x_{1,2} & - & x_{1,d} \\ x_{2,1} & x_{2,2} & - & x_{2,d} \\ | & | & & | \\ x_{n,1} & x_{n,2} & - & x_{n,d} \end{matrix}$$

Labels

$Y \sim$ 

Famille paramétrique de lois

$\mathcal{N}(\mu, I)$
$\mathcal{N}(0, \Gamma)$
$\mathcal{B}(p)$
$\mathcal{B}(p_1, ..., p_{k-1})$
$\mathcal{E}(\lambda)$
...

Supervisé – Lois usuelles

Entrées

x_i ⟶

	Famille paramétrique de lois
$\mu(x_i)$	$\mathcal{N}(\mu(x_i), I)$
$\Gamma(x_i)$	$\mathcal{N}(0, \Gamma(x_i))$
$p(x_i)$	$\mathcal{B}(p(x_i))$
$\mathbf{p}(x_i)$	$\mathcal{B}(\mathbf{p}(x_i))$
$\lambda(x_i)$	$\mathcal{E}(\lambda(x_i))$
...	...

$y_i \sim$

Supervisé - Principe

$$y_i \sim \mathcal{N}(\mu(x_i), I)$$

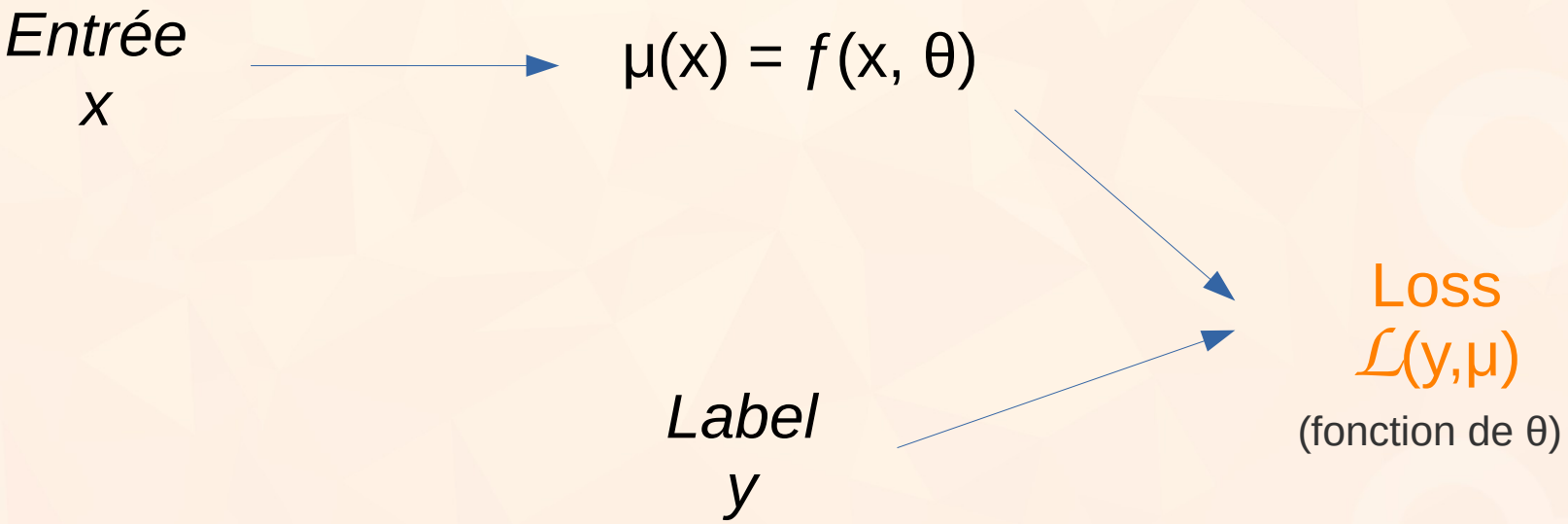
Entrée
x

$$\mu(x) = f(x, \theta)$$

f := model
θ:= paramètres (poids)

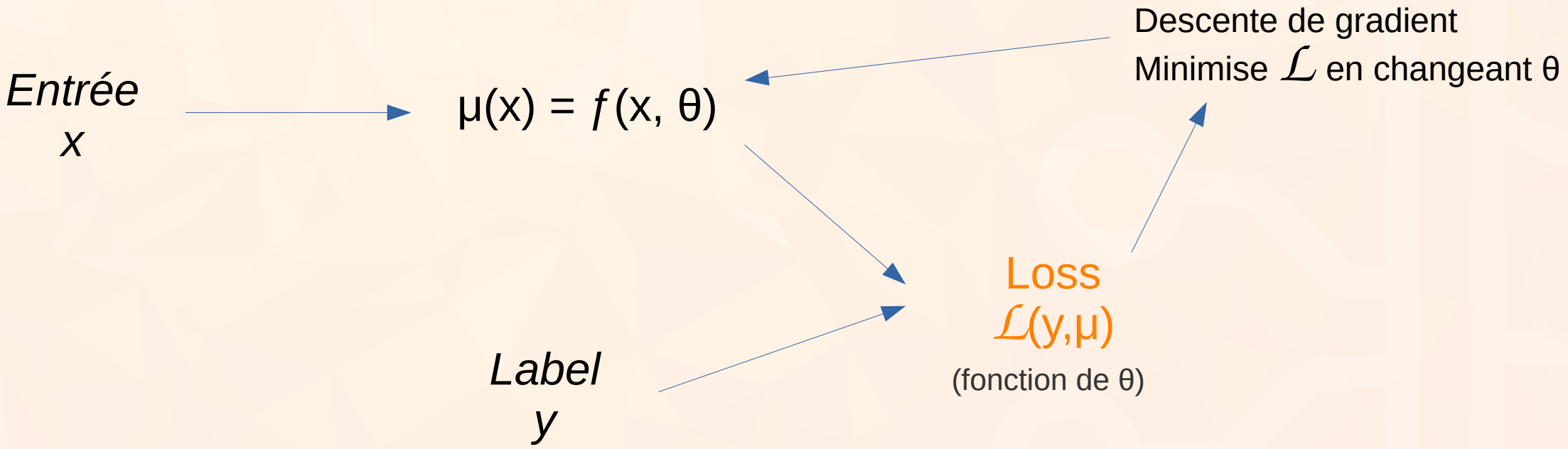
Supervisé - Principe

$$y_i \sim \mathcal{N}(\mu(x_i), I)$$



Supervisé - Principe

$y_i \sim \mathcal{N}(\mu(x_i), I)$



Supervisé – Fonction de perte

Fonction de perte (de cout, d'erreur) :

$\mathcal{L}(y, \mu(x, \theta))$ petit ⟶ $y \sim \mathcal{N}(\mu(x, \theta), I)$

$$p(y|x,\theta) = \frac{1}{\sqrt{2\pi}} \exp -\frac{||y - \mu(x,\theta)||^2}{2}$$

1ère idée : estimateur du maximum de vraissemblance

$$\begin{aligned}
\theta_{MV} &= \underset{\theta}{\operatorname{argmax}}\ p(Y|X,\theta) \\
&= \underset{\theta}{\operatorname{argmax}}\ \log p(Y|X,\theta) \\
&= \underset{\theta}{\operatorname{argmax}} \sum_i \log p(y_i|x_i,\theta) \\
&= \underset{\theta}{\operatorname{argmax}} \frac{1}{2} \sum_i (-\log 2\pi - (y_i - \mu(x_i,\theta))^2) \\
&= \underset{\theta}{\operatorname{argmin}} \sum_i (y_i - \mu(x_i,\theta))^2
\end{aligned}$$

$$\begin{aligned}
\mathcal{L}(Y, \mu(X,\theta)) &= \sum (y - \mu(x,\theta))^2 \\
&= \sum (y - f(x,\theta))^2
\end{aligned}$$

Autre exemple pour $y_i \sim \mathcal{B}(p(x_i))$:

$$\mathcal{L}(Y, p(X,\theta)) = -\sum y \log f(x,\theta) + (1-y) \log (1 - f(x,\theta))$$

Supervisé – Fonction de perte

2ème idée : loi de Bayes et maximum *a posteriori*

$$p(A|B) = \frac{p(B|A)P(A)}{P(B)}$$

$$\begin{aligned}
\theta_{MAP} &= \underset{\theta}{\operatorname{argmax}}\ p(\theta|Y,X) \\
&= \underset{\theta}{\operatorname{argmax}}\ \log p(\theta|Y,X) \\
&= \underset{\theta}{\operatorname{argmax}}\ \log p(Y|\theta,X) + \log p(\theta) \\
\theta &\sim \mathcal{N}\left(0, \frac{1}{\lambda} I\right) \\
\theta_{MAP} &= \underset{\theta}{\operatorname{argmax}}\ \log p(Y|\theta,X) + \lambda\|\theta\|^2 \\
\mathcal{L}_{MAP}(\theta) &= \mathcal{L}_{MV}(\theta) + \lambda\|\theta\|^2 \\
\mathcal{L}_{MAP}(\theta) &= \sum (y - f(x,\theta))^2 + \lambda\|\theta\|^2
\end{aligned}$$

→ Rasoir d'Ockham !

Supervisé – Choix distribution

Quelle distribution choisir ?
Quelques idées (souvent bonnes !) :

- Régression (prédire fonction dans IR) :
  $\mathcal{N}(\mu, I)$ → MSE + version multivariée
  $\mathcal{N}(0, \Gamma)$ si on s'interesse aux valeurs proches de 0
  $\mathcal{E}(\lambda)$ si évènement temporel sans mémoire

- Classification (prédire la classe)
  $\mathcal{B}(p)$ → 2 labels (ou 1 label)
  $\mathcal{B}(p_1, ..., p_{k-1})$ → k labels

- Mélange ...

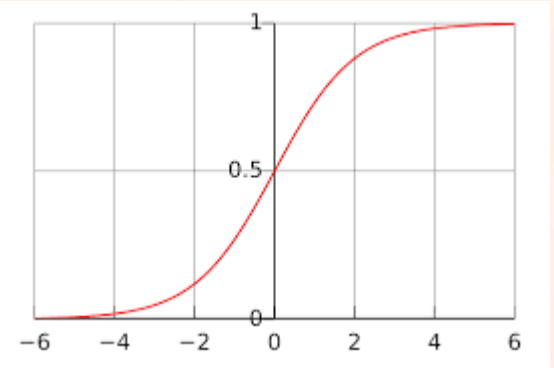
Deep Learning

Quelles fonctions d'activation choisir à la fin d'un NN ?

→ Quelle domaine pour la cible ?

-]-∞, +∞[: sortie linéaire
-]0, +∞[: prédire le **log** + sortie linéaire
-]0, 1[: sortie sigmoïdale
-]0, 1[^{k} de somme 1 : sortie softmax
- Autre : se ramener linéairement à un cas précédent
  → 1 sortie par prédiction

Sigmoide



Softmax

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ pour tout } j \in \{1, \dots, K\}.$$

-5,1	-1,2	0,3	2,0

→

< 1e-3	0,03	0,15	0,82

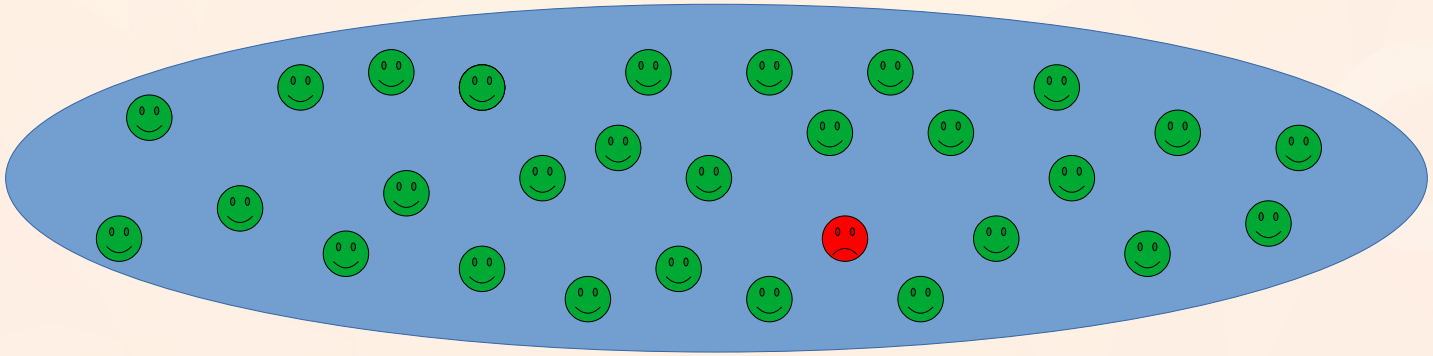
Supervisé - Métrique

Choisir sa métrique

→ Plein de possibilités !

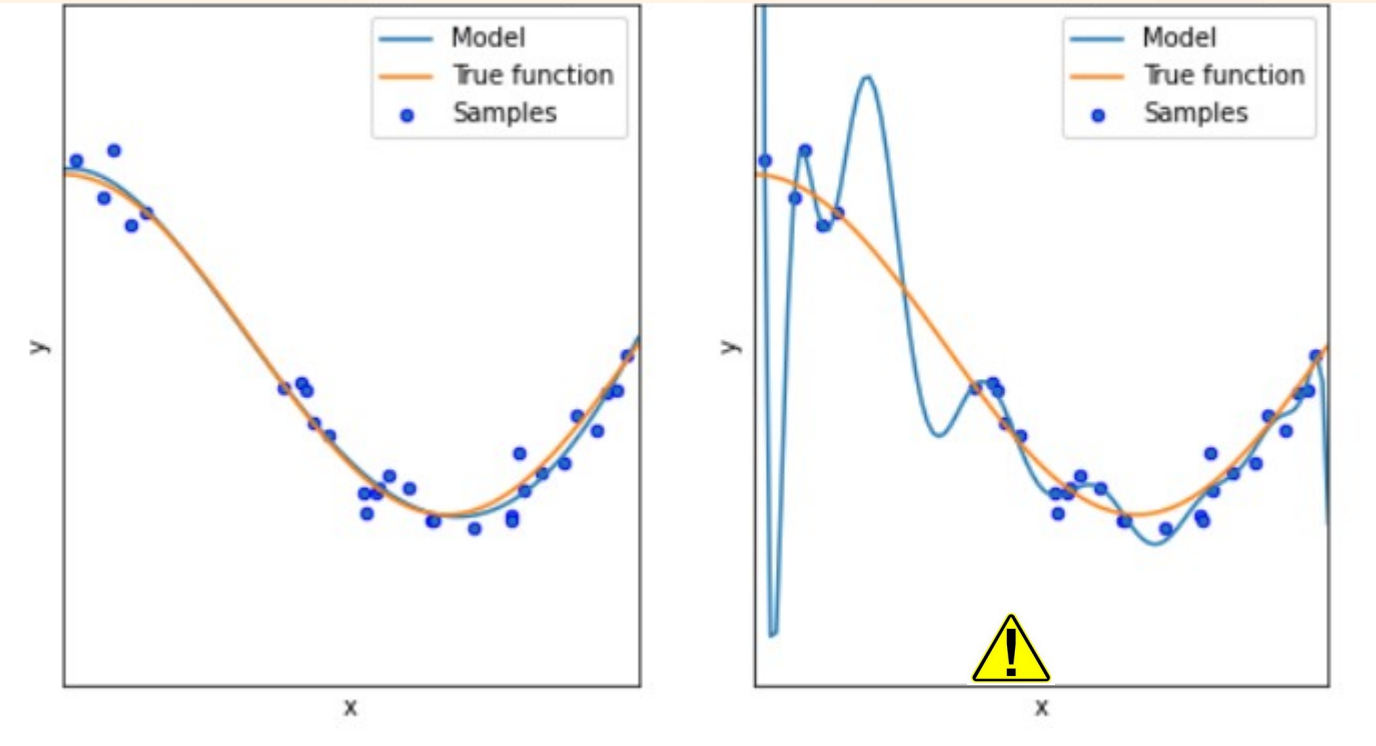
- Lien direct avec l'objectif informel
- Interprétable « physiquement » ou intuitivement
- Facile à calculer
- Attention aux pièges !

Exemple :
Accuracy en classification binaire = (TP+TN)/N



Supervisé - overfitting

Un mot sur l'overfitting ...



Supervisé - Train/Validation/Test

Le modèle

- Transformations
- Structure
- Régularisation
- Hyper-paramètres

Λ

- Paramètres entraînables

θ

Supervisé - Train/Validation/Test

Le modèle

- Transformations
- Structure
- Régularisation
- Hyper-paramètres

Λ

- Paramètres entraînables

θ

Données x_{train} + Label y_{train} → (descente de gradient) → Hyper-paramètres et caractéristiques $\Lambda_0 \rightarrow \Lambda$ / Paramètres $\theta_0 \rightarrow \theta$

Données x_{valid} + Label y_{valid} → (métriques) → θ ~~or θ~~

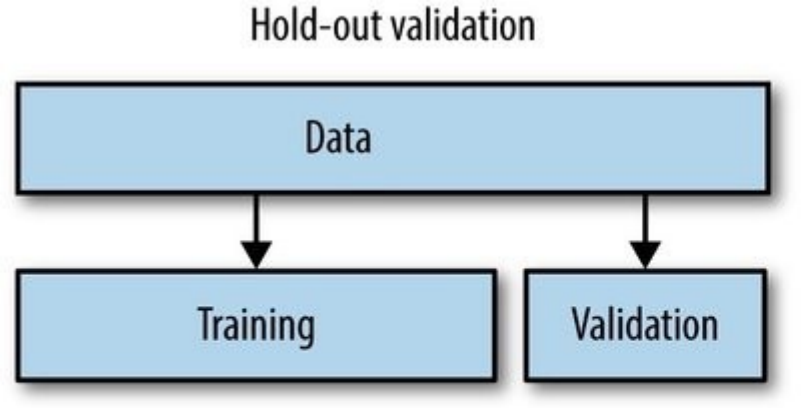
Données x_{test} + Label y_{test} → (métriques) → Λ ~~or Λ~~

⚠ D_{train}, D_{valid}, D_{test} disjoints !

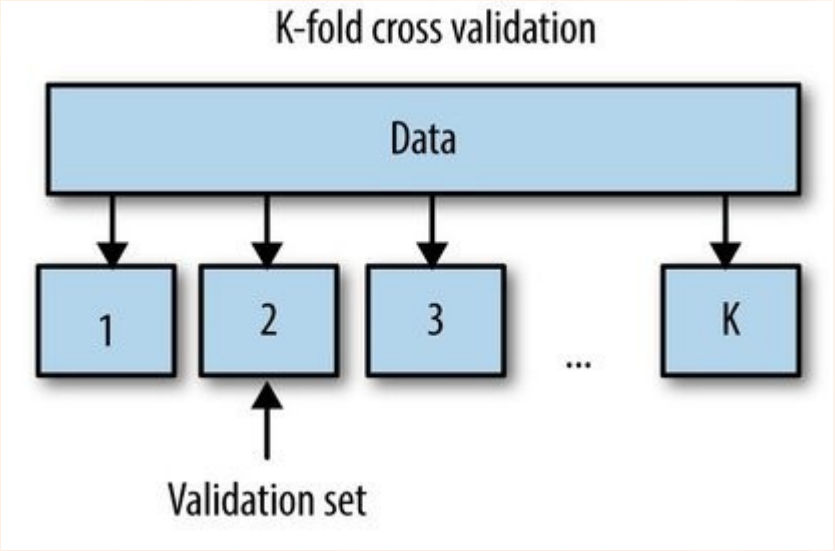
Supervisé – Stratégie Validation

Choix de la stratégie de validation

- *Hold-out validation* :
  → ensemble validation fixe
  performance = performance sur validation



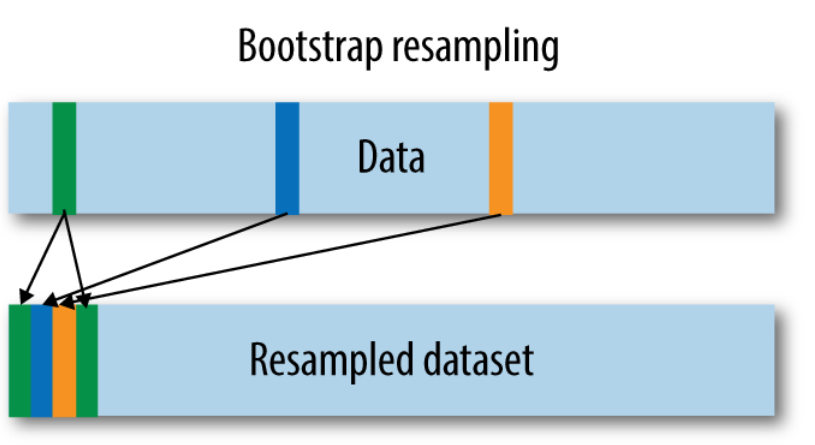
- Validation croisée K-fold
  → K entraînements sur K-1 folds
  le dernier est l'ensemble de validation
  performance = moyenne des performances



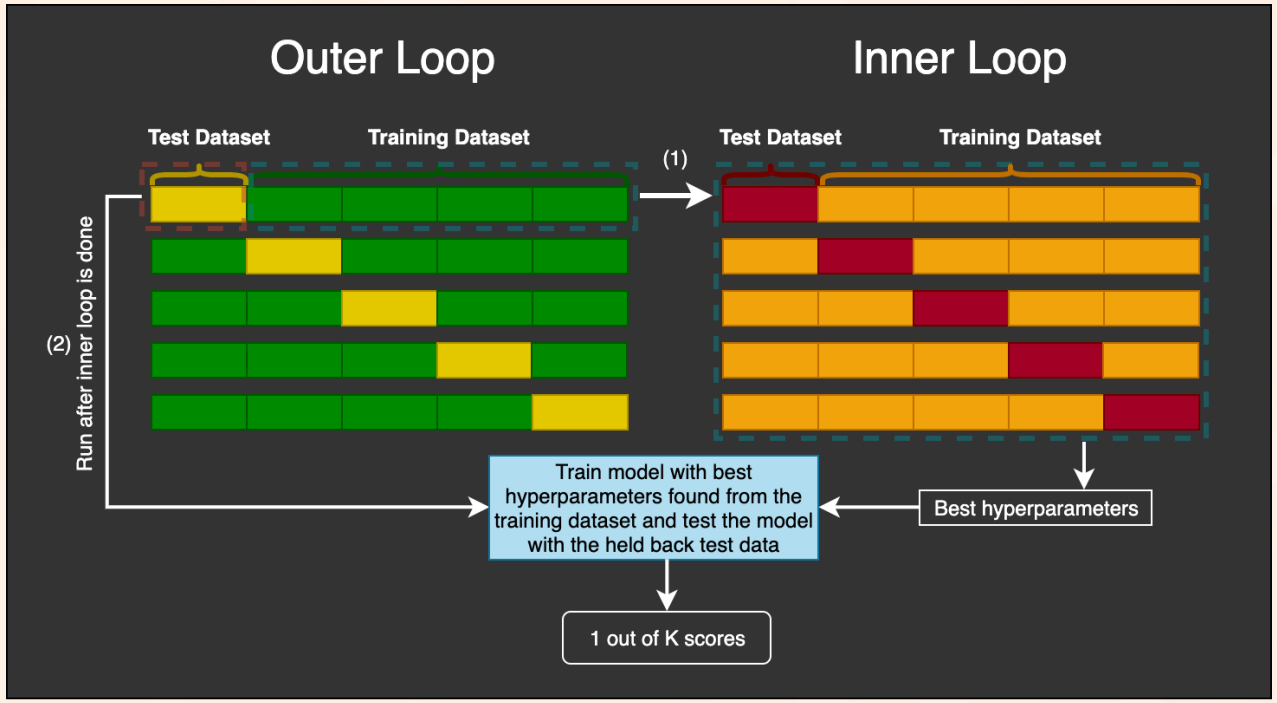
- Stratified K-fold
  → classes équilibrées sur chaque fold

Supervisé – Stratégie Validation

- K-fold + itérations randomisées
  → P fois la stratégie K-folds
  on mêlange/divise les données aléatoirement
  performance = moyenne des performances



- Nested K-fold
  → K-fold + K-fold sur chaque itération en faisant une recherche d'hyper-paramètres
  performance = moyenne des performances



Supervisé - Modèle

Mise au point du modèle

→ étape la plus longue
→ équilibre entre underfitting et overfitting

Astuces :

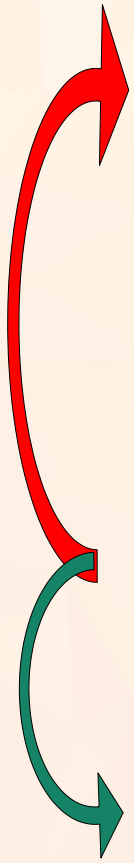
→ chercher des papiers de recherche sur des sujets similaires
→ commencer par un modèle pauvre qui bat le hasard ou des stratégies triviales
→ s'intéresser au preprocessing pour améliorer la prédiction sur D_{train}
→ essayer ensuite d'overfitter puis diminuer progressivement la taille
→ essayer des régularisations (loss, dropout, data augment, early stop, ...)
→ se fixer des objectifs sur les métriques sur D_{test}

Supervisé - Résumé

Résumé (to-do list) :

- Analyser données et objectif → type de ML, distributions, loss, FA de sortie
- Métriques prenant en compte les spécificités du problème et l'objectif
- Séparation des données et système de validation
- Mise au point du modèle
  - Preprocessing
  - Modèle simple
  - Overfitting
  - Diminution taille
  - Régularisation
- Test sur D_{test}

Supervisé - Résumé



Résumé (to-do list) :

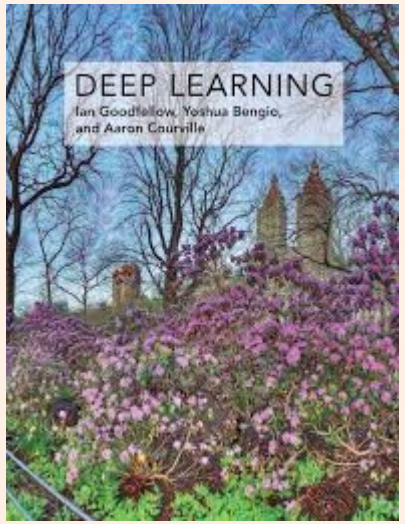
- Analyser données et objectif → type de ML, distributions, loss, FA de sortie
- Métriques prenant en compte les spécificités du problème et l'objectif
- Séparation des données et système de validation
- Mise au point du modèle
  - Preprocessing
  - Modèle simple
  - Overfitting
  - Diminution taille
  - Régularisation
- Test sur D_{test}

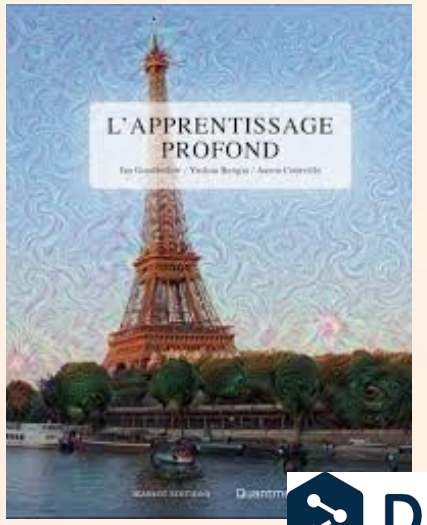
- Super, ça marche !

Sources

Sources :

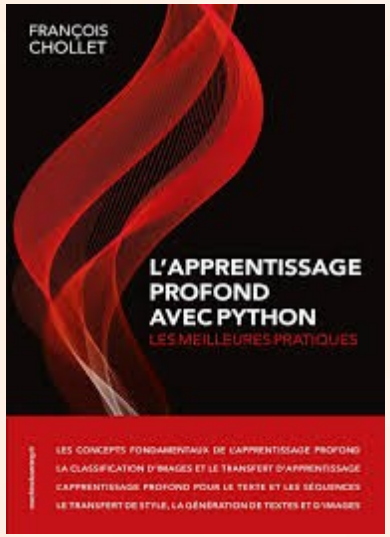
- *Deep Learning*, Ian Goodfellow
- *L'Apprentissage Profond avec Python*, François Chollet
- Types de cross-validation : https://towardsdatascience.com/understanding-8-types-of-cross-validation-80c935a4976d





70€

DeepL



40€

Vive le machine learning !

Merci de votre attention !

Valentin
Goldité