UMONS
Université de Mons

INTELLIGENCE ARTIFICIELLE

CHAPITRE 5 : APPRENTISSAGE AUTOMATIQUE

MACHINE LEARNING

Sidi Ahmed Mahmoudi

POLYTECH MONS

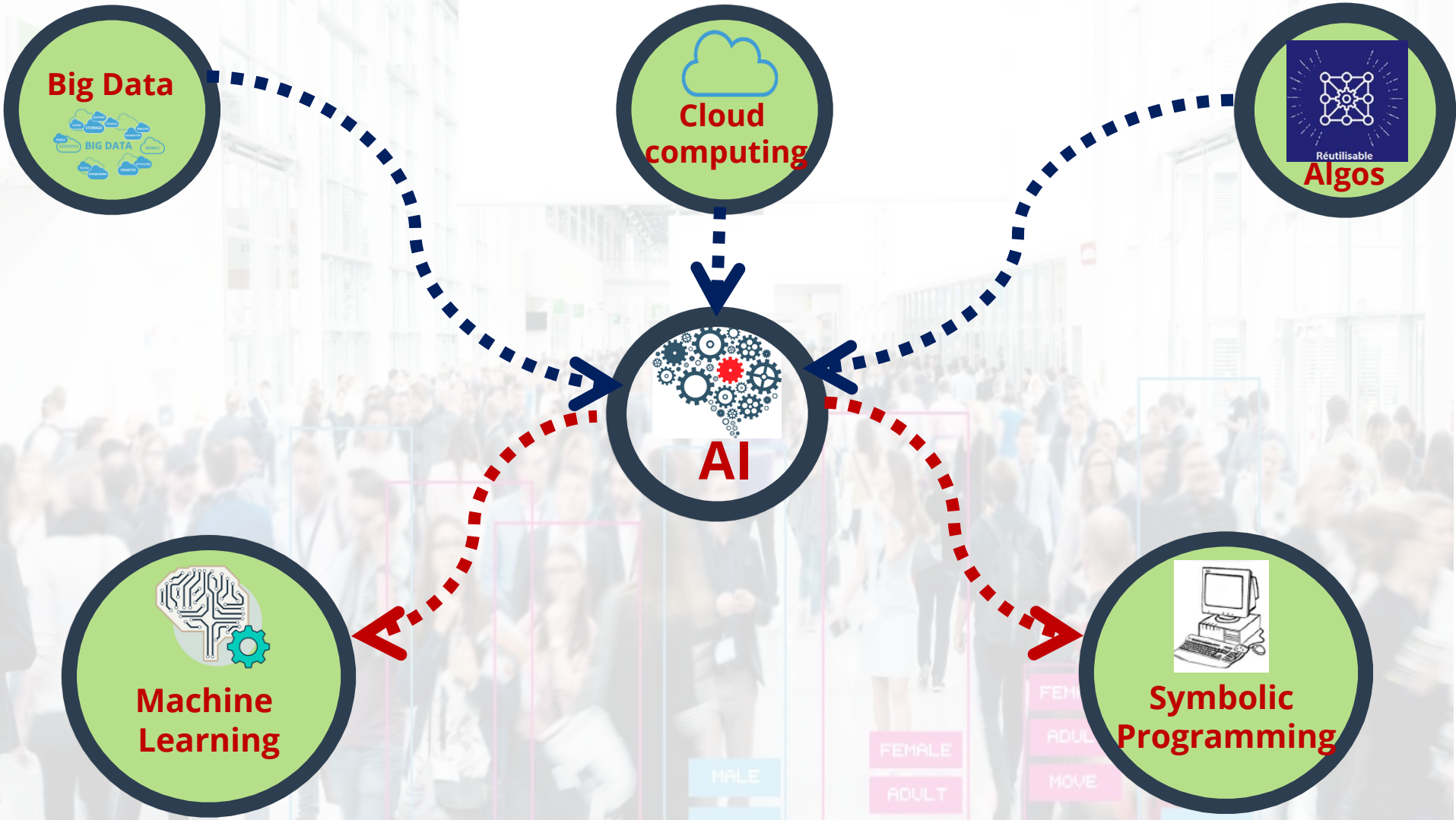
sidi.mahmoudi@umons.ac.be

18 Fév 2025

PLAN

PLAN

Prérequis de l'IA

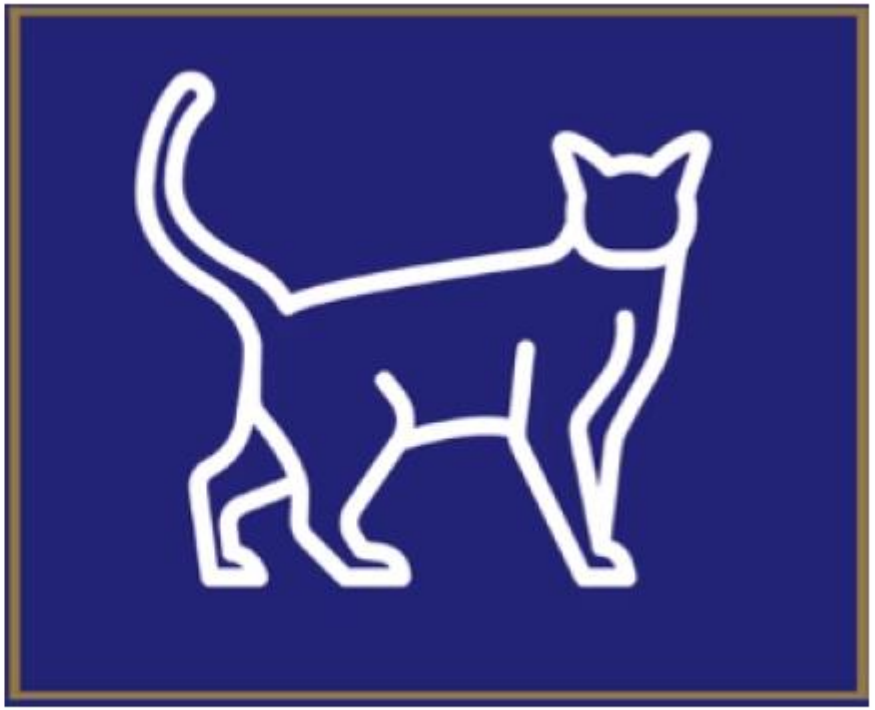


Approches de l'IA

Programmation symbolique

- Coder pour résoudre un problème (suite de relations)
- « si –alors + si-alors + si-alors » = solution

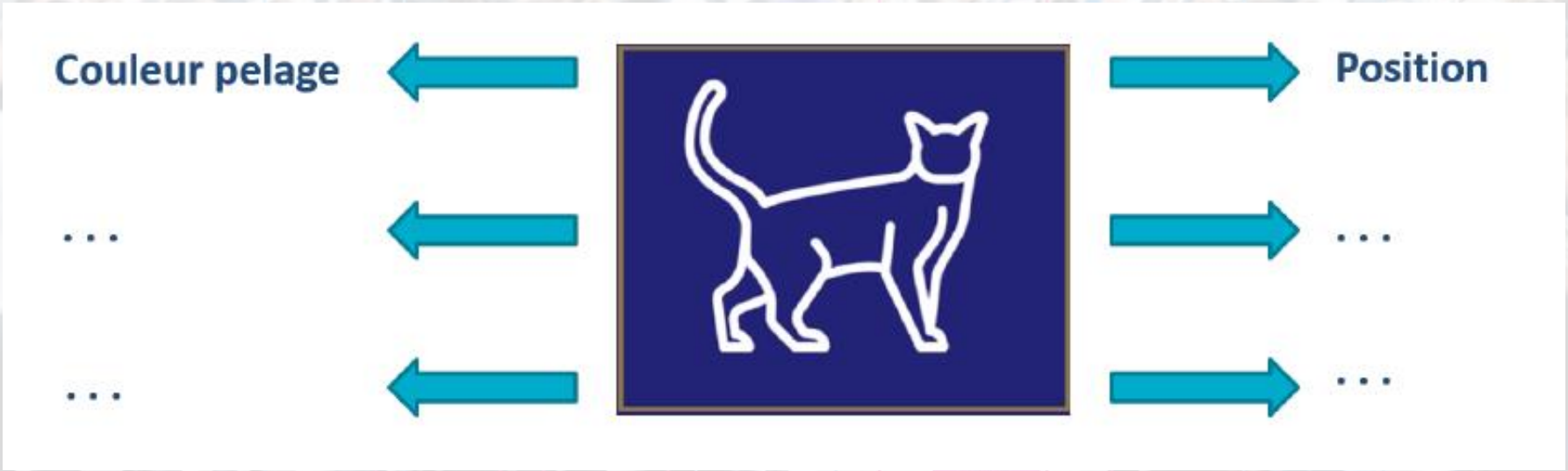
(a) 4 pattes + queue = animal



(b) animal + oreilles = chat

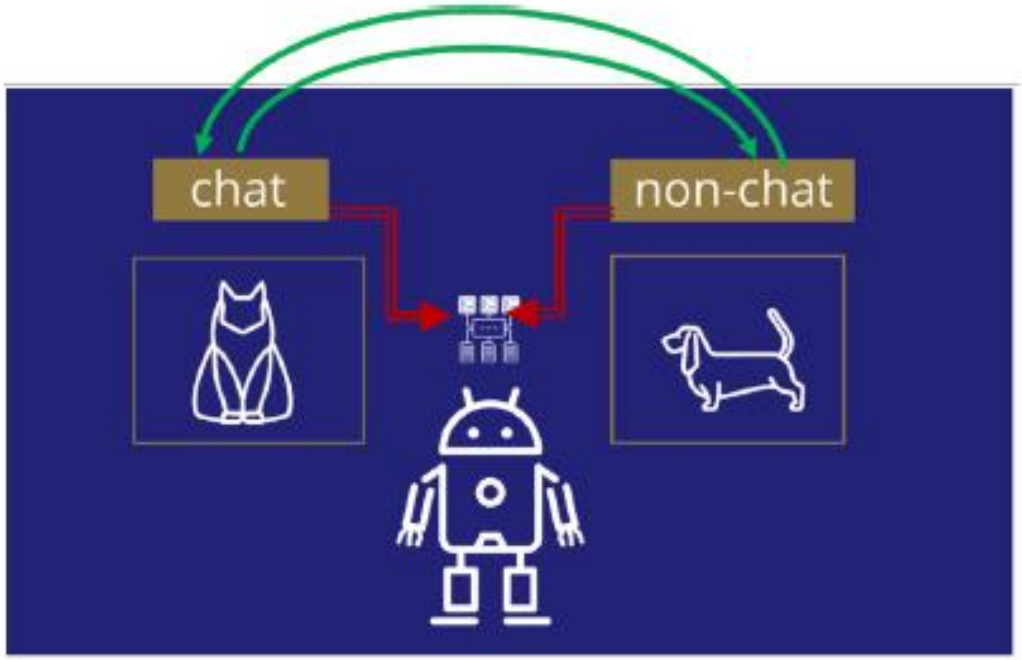
Programmation symbolique

- Difficulté de s'adapter à toutes les situations
- Besoin d'envisager toutes les situations possibles

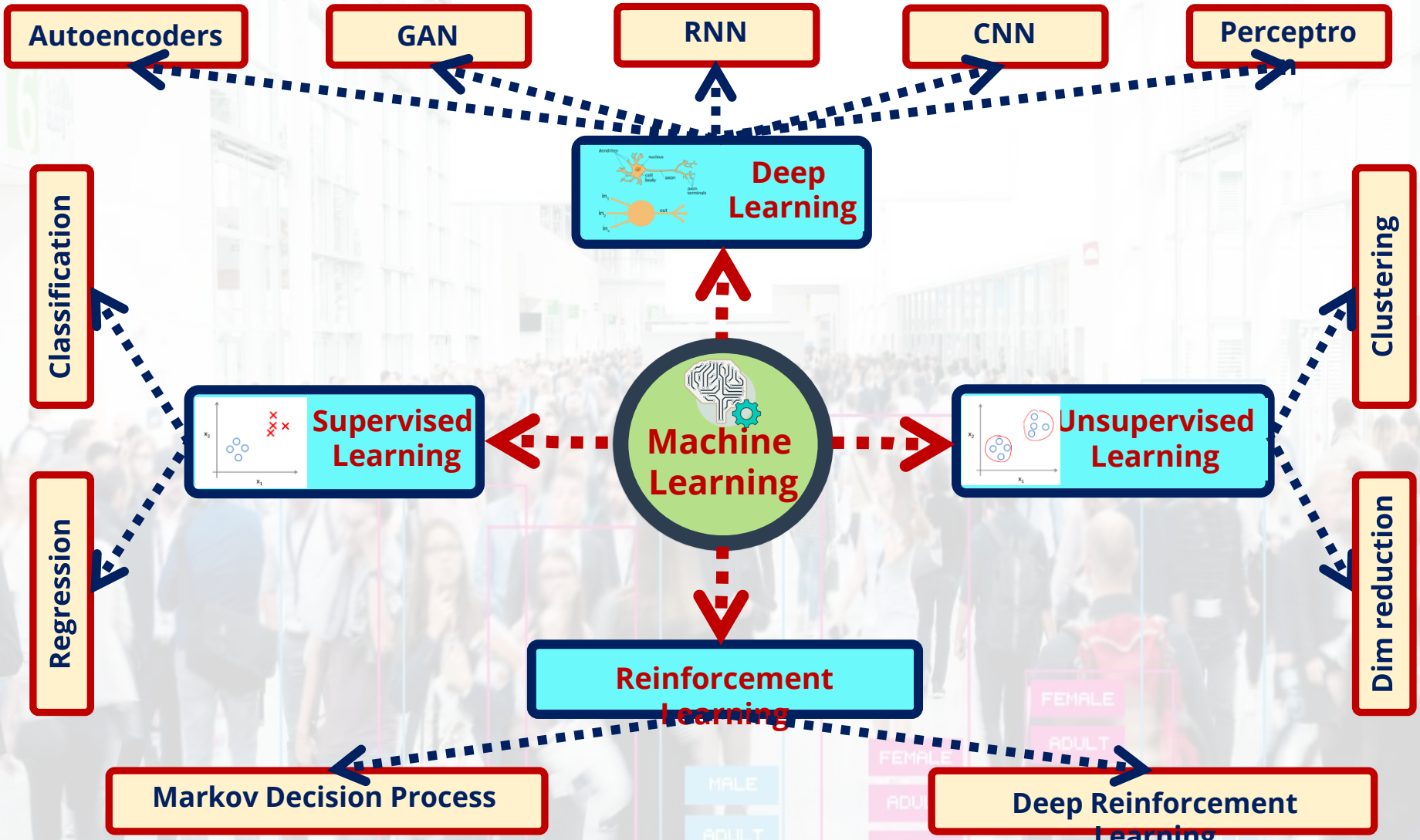


Machine Learning

- Difficulté de s'adapter à toutes les situations
- Besoin d'envisager toutes les situations possibles



Machine Learning



PLAN

Introduction

I. Définition

II. Types d'apprentissage

III. Terminologie

IV. Réduction de la perte

V. Généralisation et représentation des données

Conclusion

Définition

- **Machine Learning:** méthodes automatisables permettant à une machine d'évoluer grâce à un processus d'apprentissage

- Les systèmes de Machine Learning apprennent à
  - combiner des entrées pour formuler des prédictions utiles
  - afin de les appliquer sur des données **non observées**

- Apprendre est une capacité importante de l'humain, mais difficile pour un ordinateur

- Mémoriser par cœur ne veut pas dire apprendre

Définition

- **Informatique traditionnelle :** l'ordinateur résout les problèmes à partir d'instructions fournies par l'utilisateur

- **Apprentissage machine :** l'ordinateur résout les problèmes à partir d'exemples (entrées/sorties) formant les données d'apprentissage

- **Le but** est que l'ordinateur puisse généraliser ce qu'il a appris à de nouveaux jeux de données non encore rencontrés

PLAN

Introduction

I. Définition

II. Types d'apprentissage

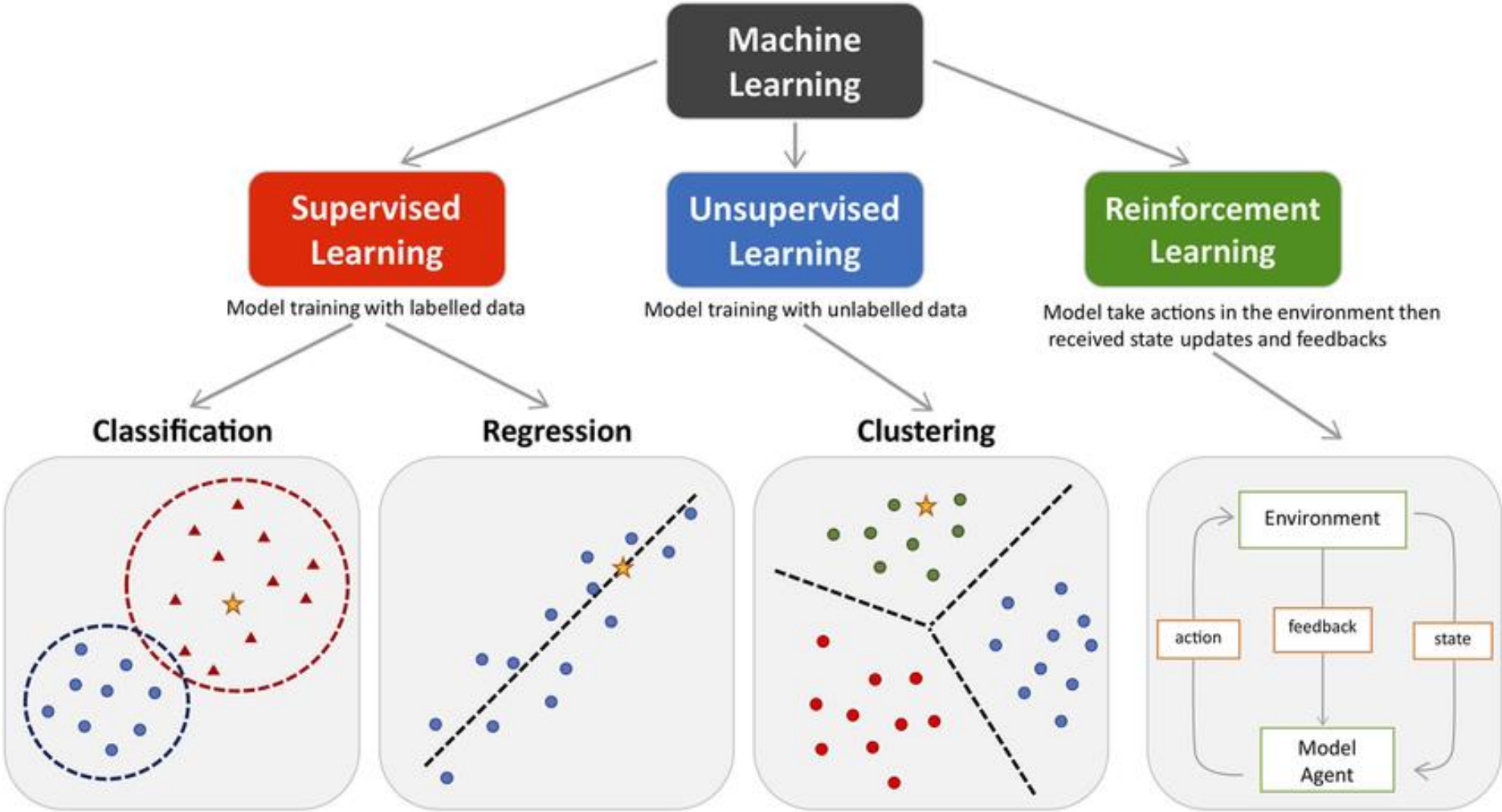
III. Terminologie

IV. Réduction de la perte

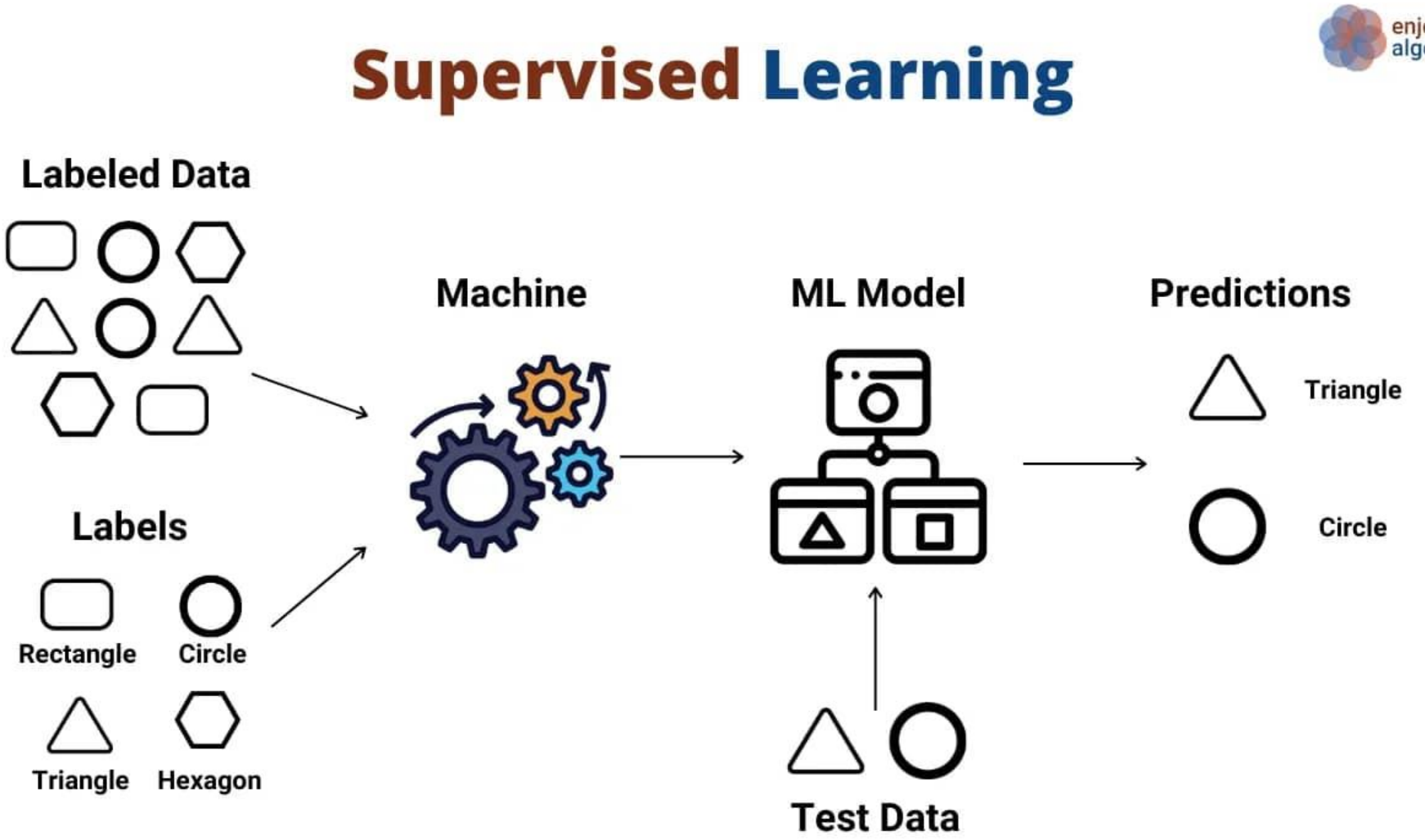
V. Généralisation et représentation des données

Conclusion

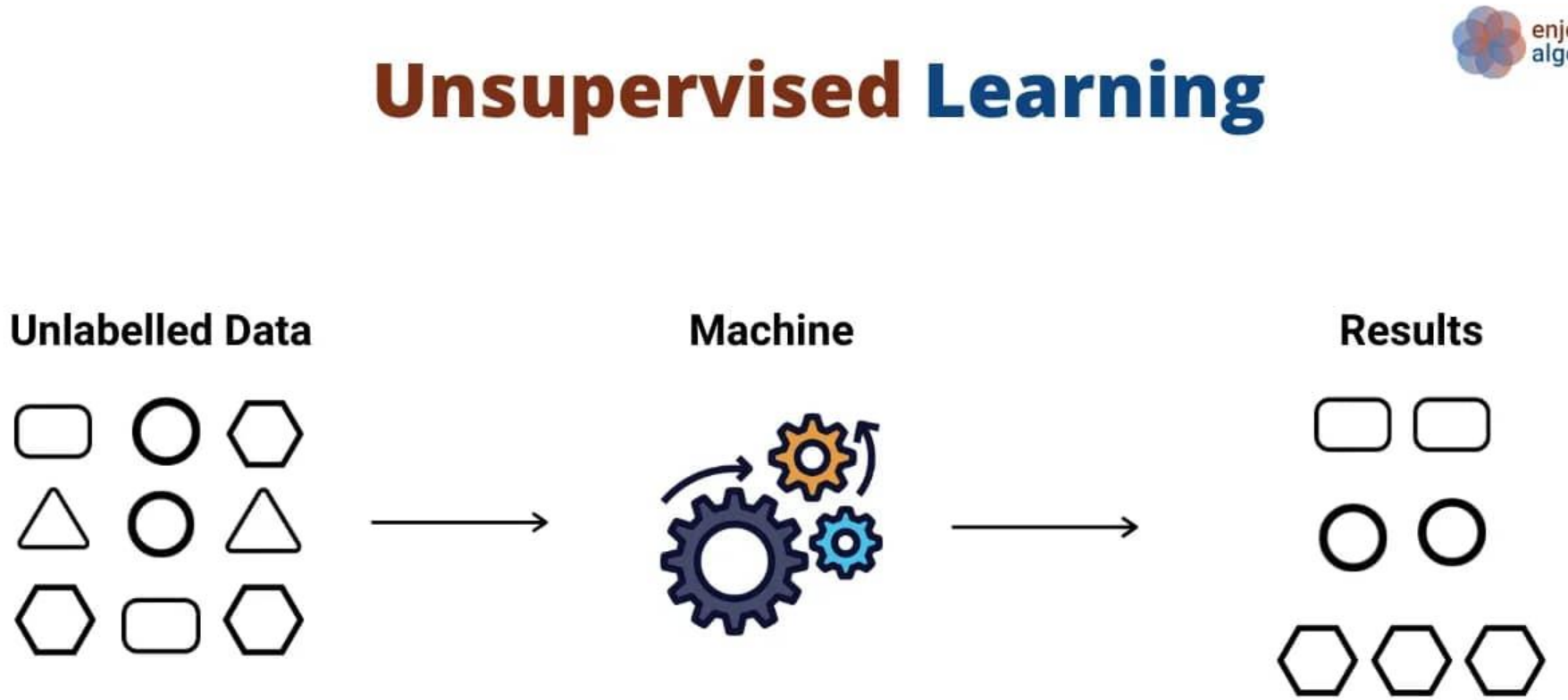
Types d'apprentissage



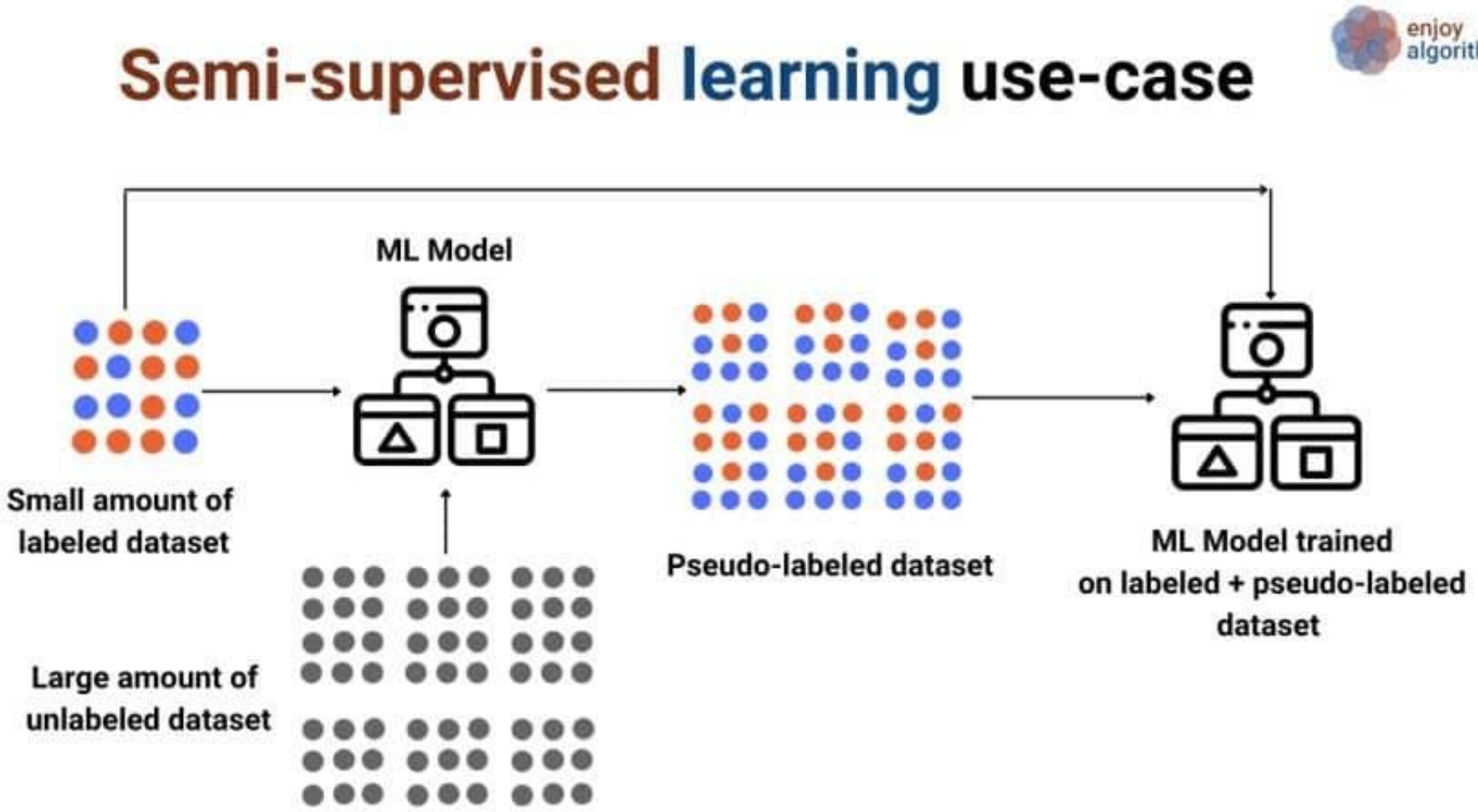
Apprentissage supervisé



Apprentissage non supervisé



Apprentissage semi supervisé

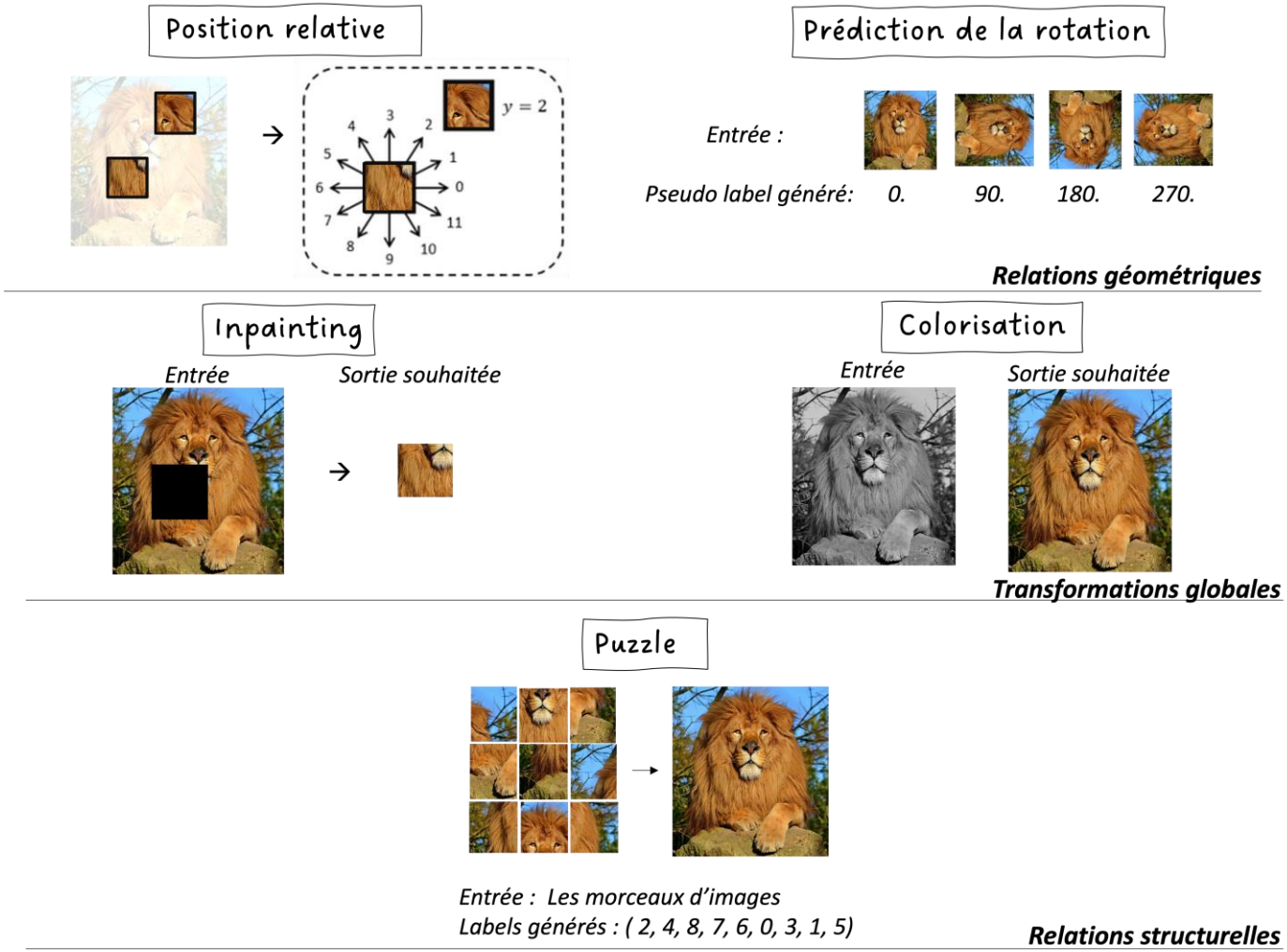


Apprentissage auto supervisé

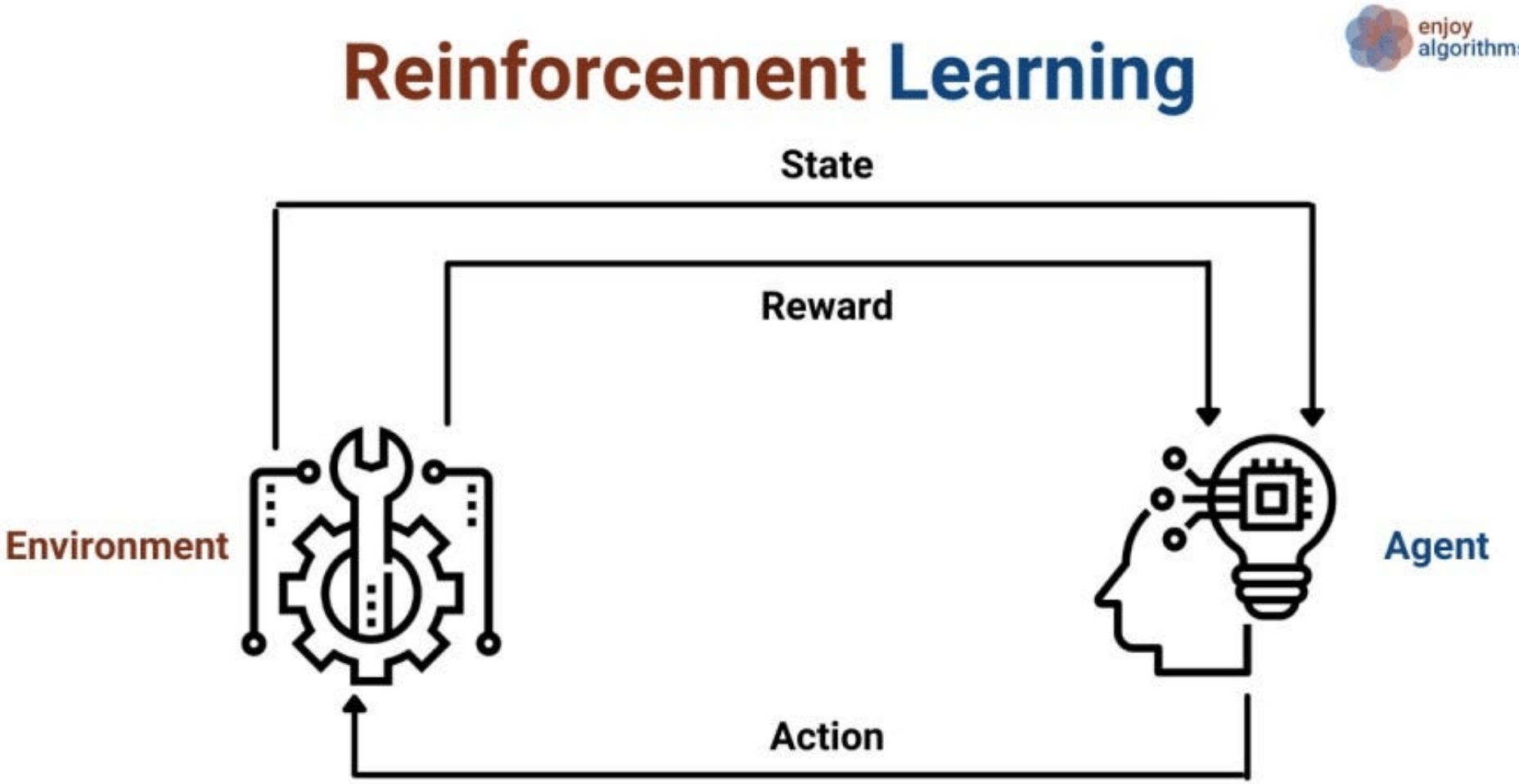
- **Tâche prétexte**

- **Objectif :**

Obtenir les caractéristiques les plus pertinentes des images



Apprentissage par renforcement



Apprentissage Par renforcement

Reinforcement Learning with Online Interactions

Offline Reinforcement Learning

Formulation mathématique

- **Apprentissage non supervisé :** on reçoit des observations brutes de variables aléatoires : $x_1, x_2, x_3, x_4, \ldots$ *et on espère découvrir la relation avec des variables latentes structurelles :* $x_i \rightarrow y_i$

- **Apprentissage supervisé :** on reçoit des exemples annotés : $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), \ldots$ *et on espère prédire la sortie sur de nouvelles observations :* $x^* \rightarrow y^*$

PLAN

Introduction

I. Définition

II. Types d'apprentissage

III. Terminologie

IV. Réduction de la perte

V. Généralisation et représentation des données

Conclusion

Terminologie

- Un exemple est une instance particulière de donnée: **x**
- Un exemple étiqueté comporte {caractéristiques, étiquette} : **(x, y)**
- Les exemples étiquetés sont utilisés pour **entraîner le modèle**
- Un exemple sans étiquette comporte {caractéristiques, ?} : **(x, ?)**
- Exemples non étiquetés : prédictions sur les nouvelles données
- **Modèle :** correspondre des exemples à des étiquettes prédites : **y'**

Exemple

- 5 exemples étiquetés

housingMedianAge (caractéristique)	totalRooms (caractéristique)	totalBedrooms (caractéristique)	medianHouseValue (étiquette)
15	5612	1283	66900
19	7650	1901	80100
17	720	174	85700
14	1501	337	73400
20	1454	326	65500

- Générer le modèle à partir des exemples étiquetés
- Exemple sans étiquette présente des caractéristiques mais pas d'étiquette
- Reste à prédire l'étiquette sur de nouveaux exemples

Modèles d'apprentissage

- **Modèle d'apprentissage :** relation entre les caractéristiques et étiquettes
- **Exemple :** Un modèle de détection de spam peut associer certaines caractéristiques à un spam
- Un modèle d'apprentissage s'appuie sur deux phases :
  - **Apprentissage :** créer et entraîner le modèle à partir des données étiquetées
  - **Inférence :** appliquer le modèle à des exemples non étiquetés
  - **Exemple :** prédire *mediaHouseValue* pour des nouveaux exemples

Régression et classification

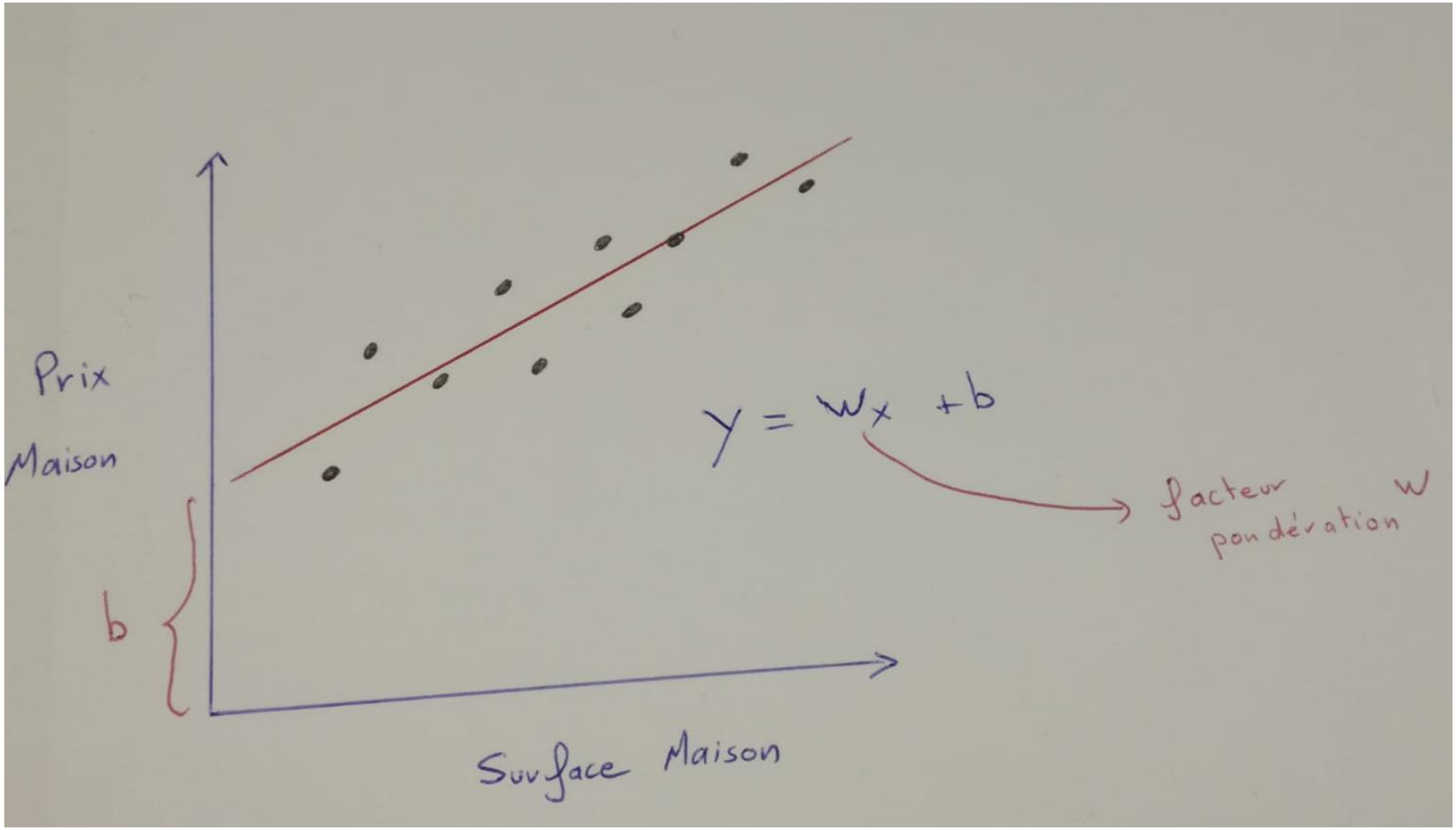
- **Modèle de régression :** prédire des valeurs **continues**
  - Quelle est la valeur d'un logement a New York ?
  - Quelle est la probabilité qu'un utilisateur clique sur cette annonce ?

- **Modèle de classification :** prédire des valeurs **discrètes**
  - Un e-mail représente t-il un spam ou non ?
  - Cette image représente-t-elle une voiture ou un camion ?

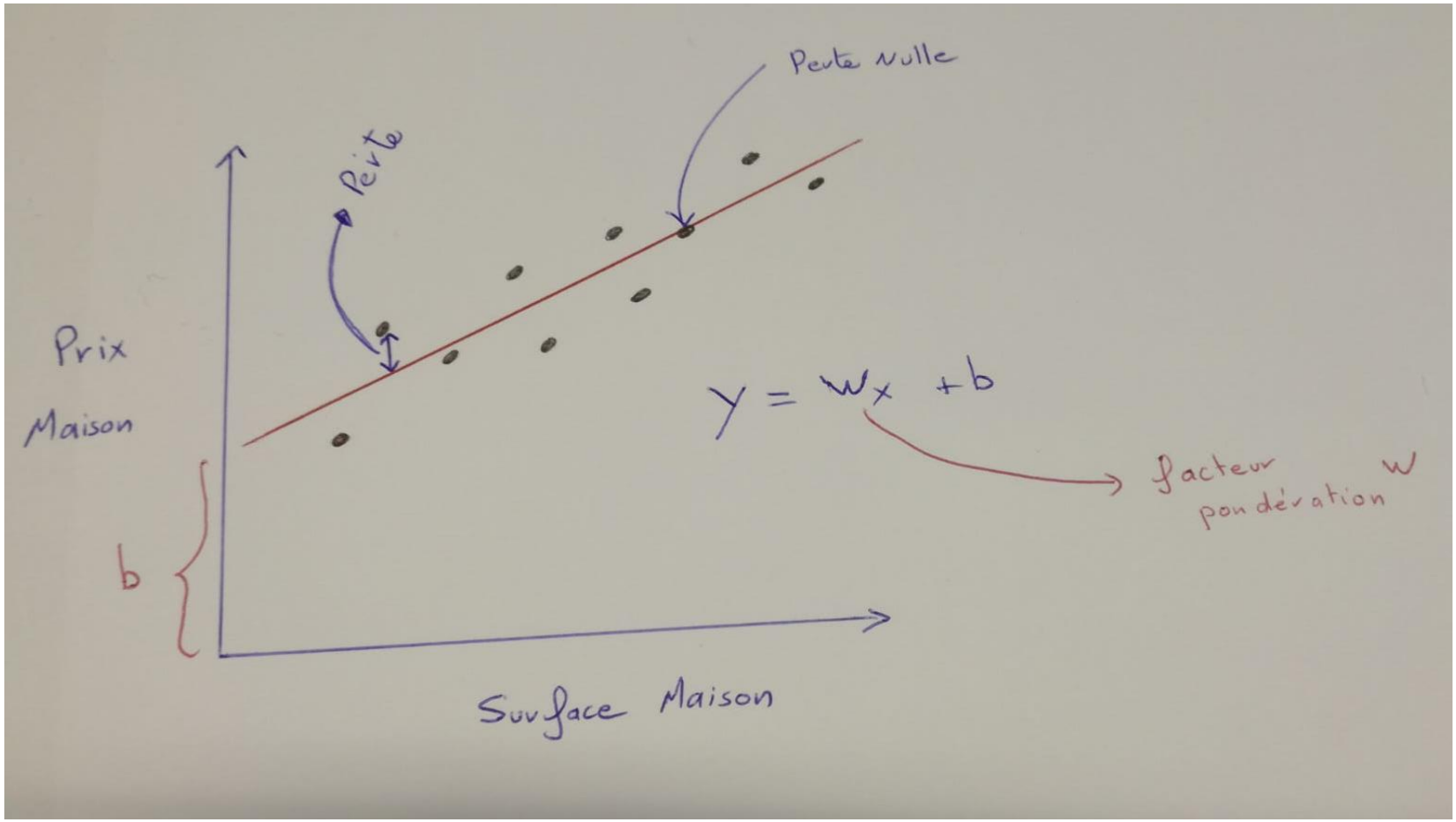
Exemple de régression

Exemple de régression

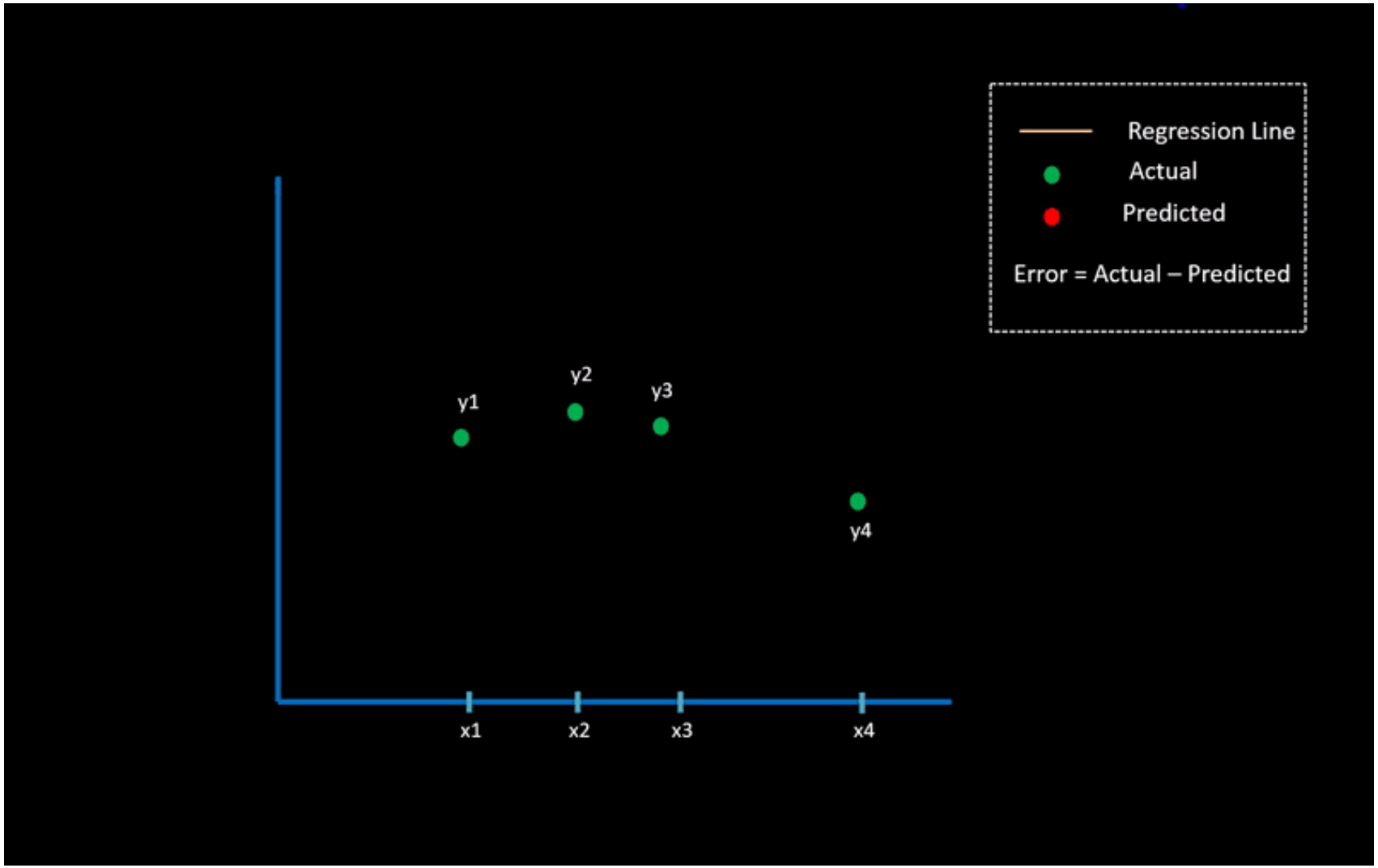
Exemple de régression



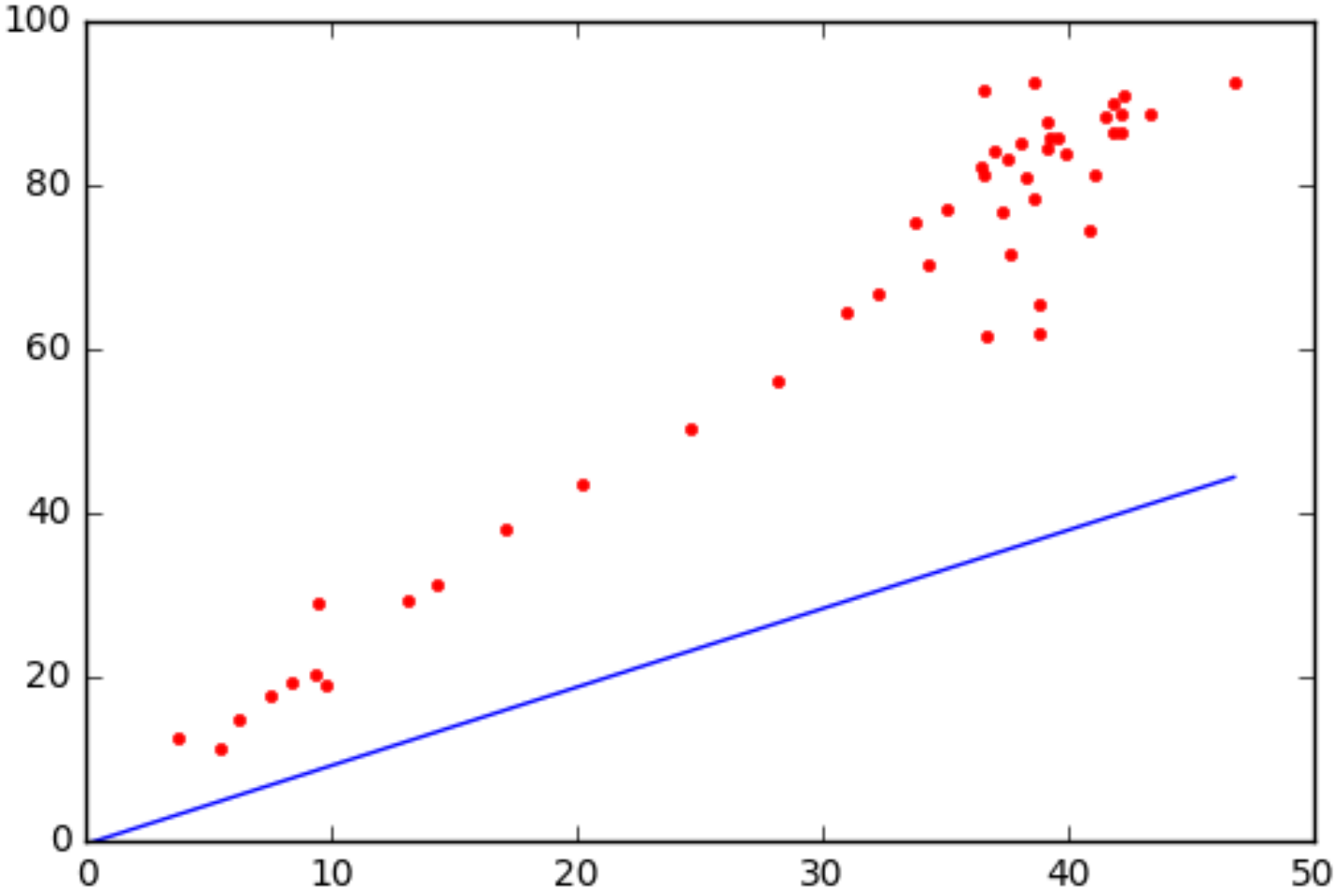
Exemple de régression



Exemple de régression



Exemple de régression



Perte de la régression

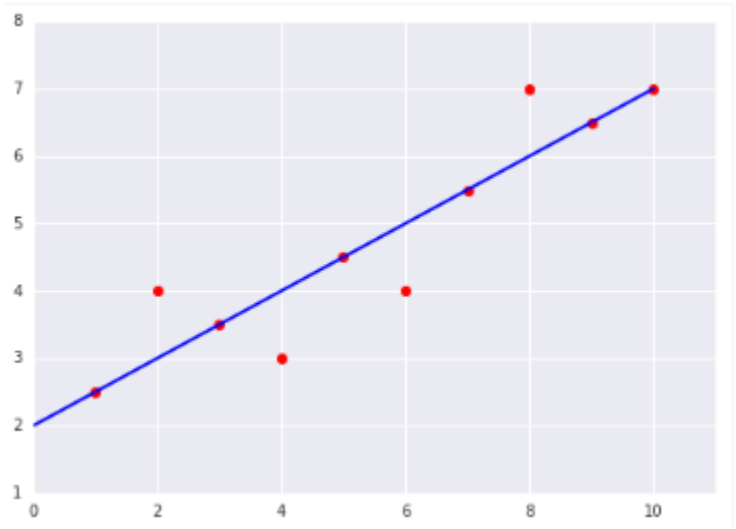
- **Perte L_2 : perte quadratique**
  - (observation – prédiction)^2
  - (y –y')^2

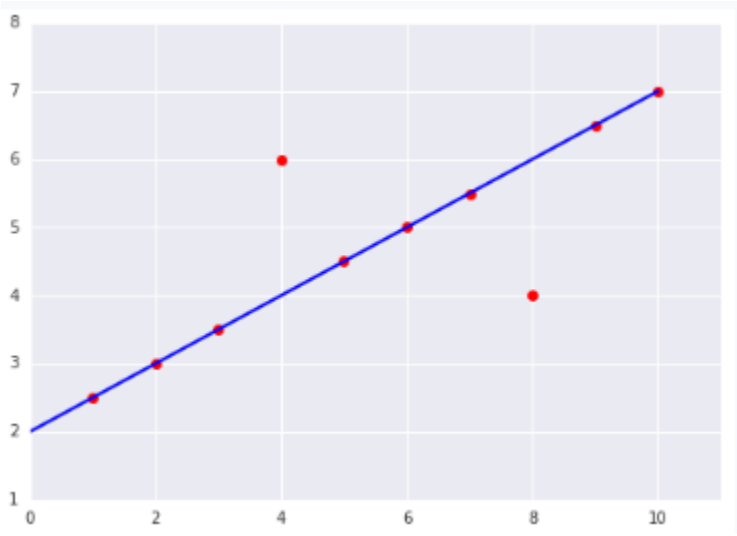
$$L_2 Loss = \sum_{(x,y)\in D} (y - prédiction(x))^2$$

$$MSE = \frac{1}{N} \sum_{(x,y)\in D} (y - prediction(x))^2$$

Question

- Lequel des deux ensembles de données présente l'erreur quadratique moyenne la plus élevée ?





Question

$$MSE = \frac{0^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2 + 0^2 + 0^2}{10} = 0.4$$

$$MSE = \frac{0^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2}{10} = 0.8$$

Réduction de la perte

- Comment réduire la perte ?
  - La dérivée de (y - y') par rapport aux pondérations et au biais nous informe sur la variation de la perte pour un exemple donné
  - Simple à calculer et convexe

- Des petits pas répétés dans la direction permettant de réduire la perte:
  - Pas de gradient (pas de gradient négatif)
  - Stratégie d'optimisation appelée descente de gradient

PLAN

Introduction

I. Définition

II. Types d'apprentissage

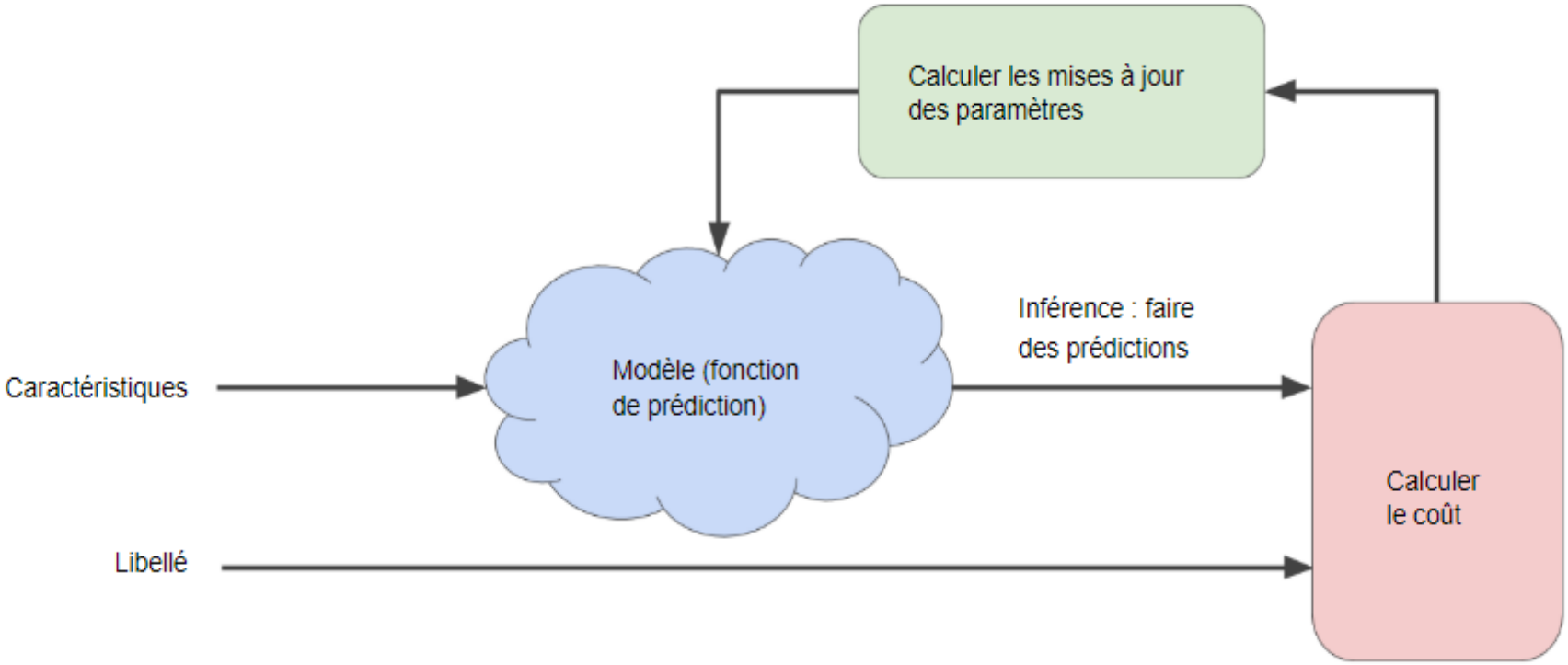
III. Terminologie

IV. **Réduction de la perte**

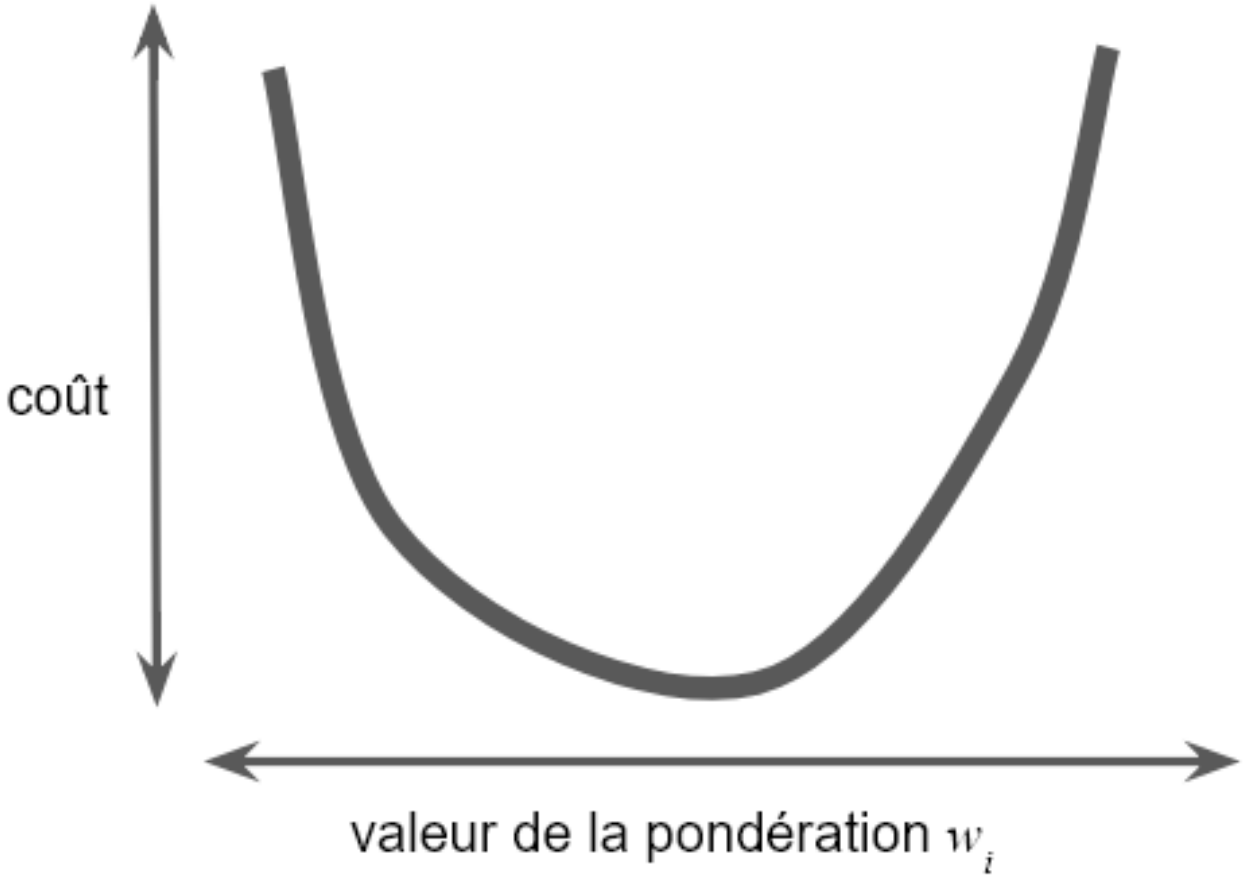
V. Généralisation et représentation des données

Conclusion

Réduction de la perte : approche itérative



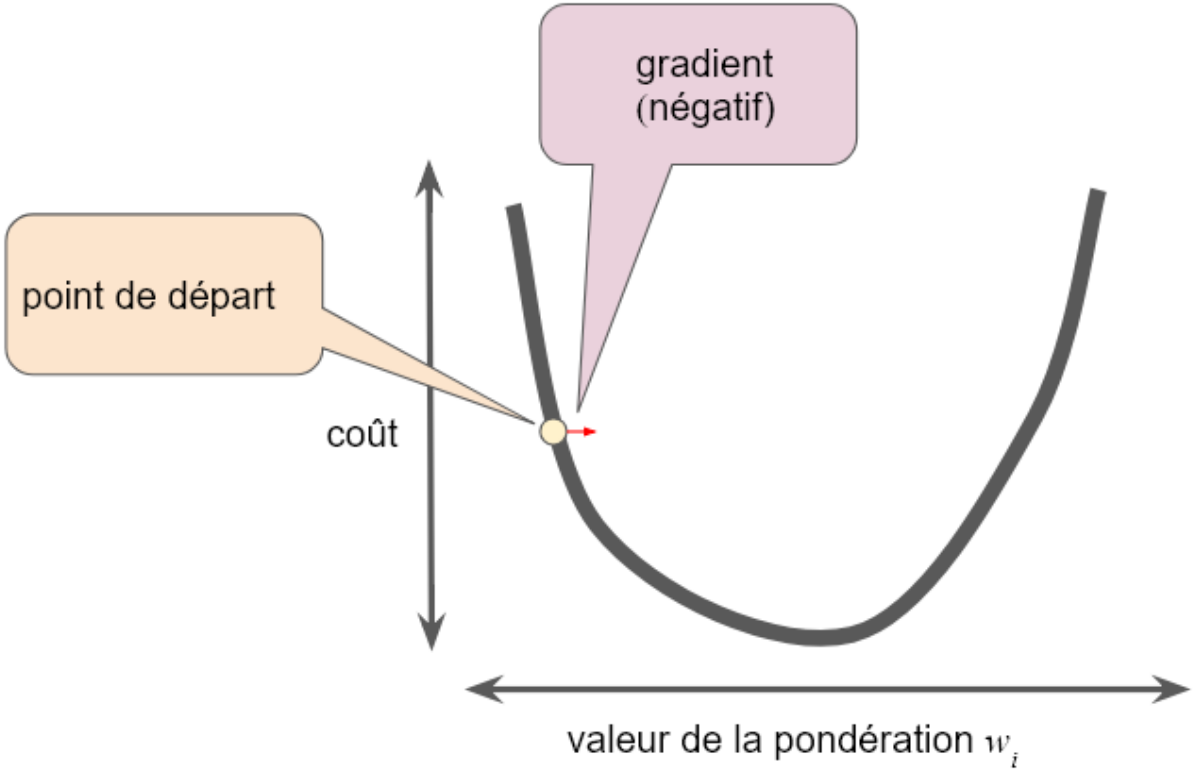
Réduction de la perte : descende du gradient



Réduction de la perte : descende du gradient

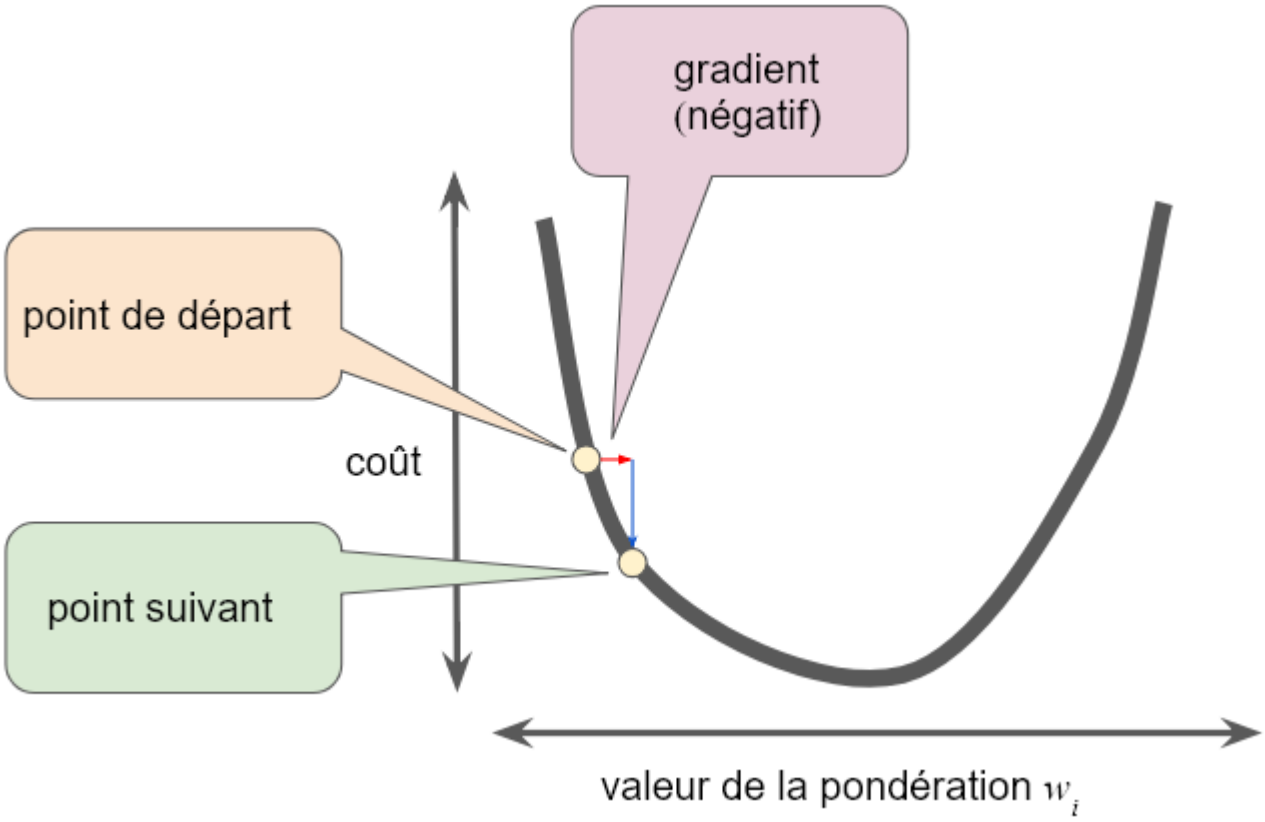
- **Gradient de perte :** dérivée pour chaque valeur de pondération

Réduction de la perte : descende du gradient



- **Gradient :** vecteur ayant deux caractéristiques : direction et magnitude
- Il indique la direction de la croissance maximale de la fonction de perte
- L'algorithme de descente de gradient fait **un pas dans le sens inverse** afin de **réduire la perte** aussi rapidement que possible.

Réduction de la perte : descende du gradient

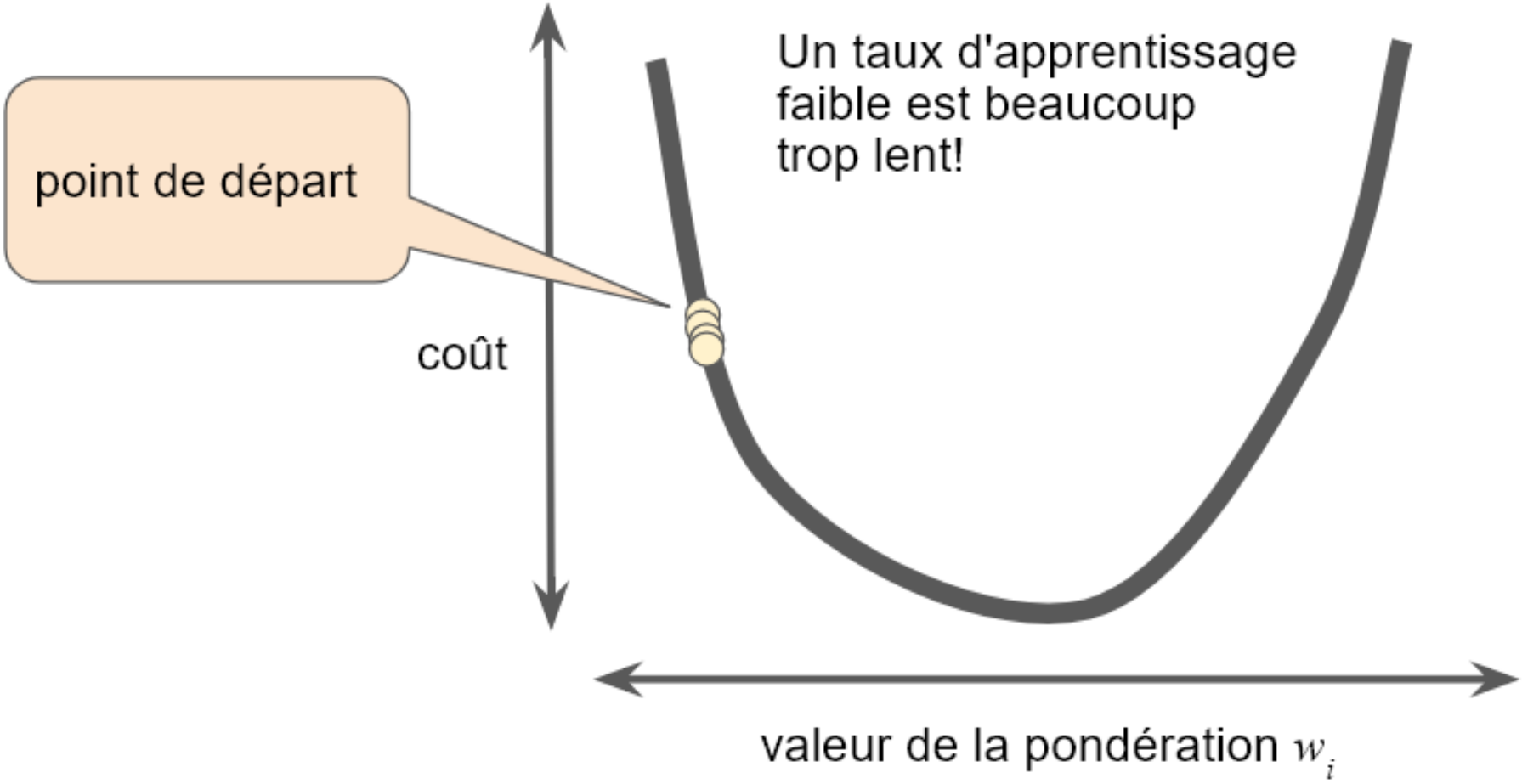


- Pour déterminer le point suivant, l'algorithme de descente de gradient ajoute une fraction de la magnitude du gradient au point de départ
- Processus répété jusqu'à l'arrivée au minimum

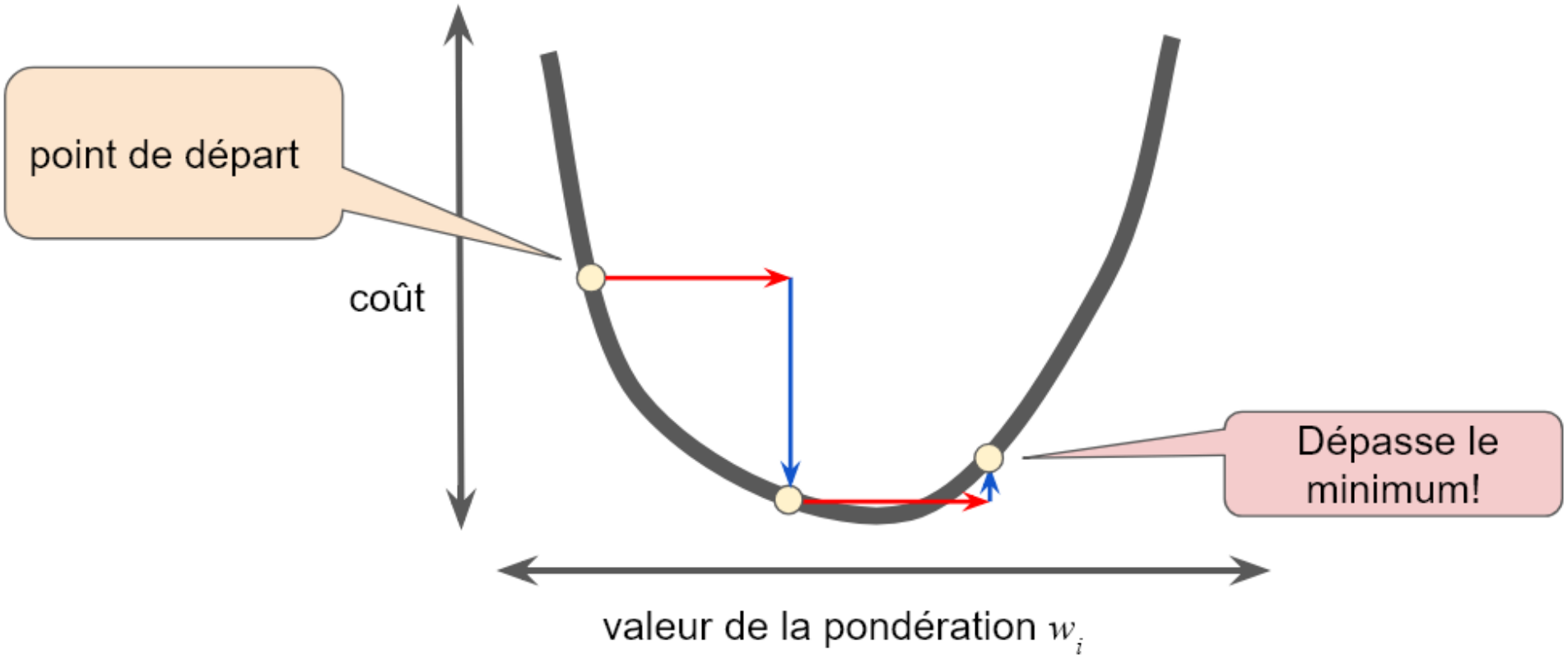
Réduction de la perte : taux d'apprentissage

- En général, le gradient est multiplié par une valeur scalaire appelée taux d'apprentissage (ou pas d'apprentissage) pour déterminer le point suivant
- **Exemple :** magnitude du gradient = 2,5, le taux d'apprentissage = 0,01
- Position du point suivant à 0,025 du point précédent.
- **Hyper paramètres :** variables pouvant être ajustées par les programmateurs dans les algorithmes de Machine Learning

Réduction de la perte : taux d'apprentissage

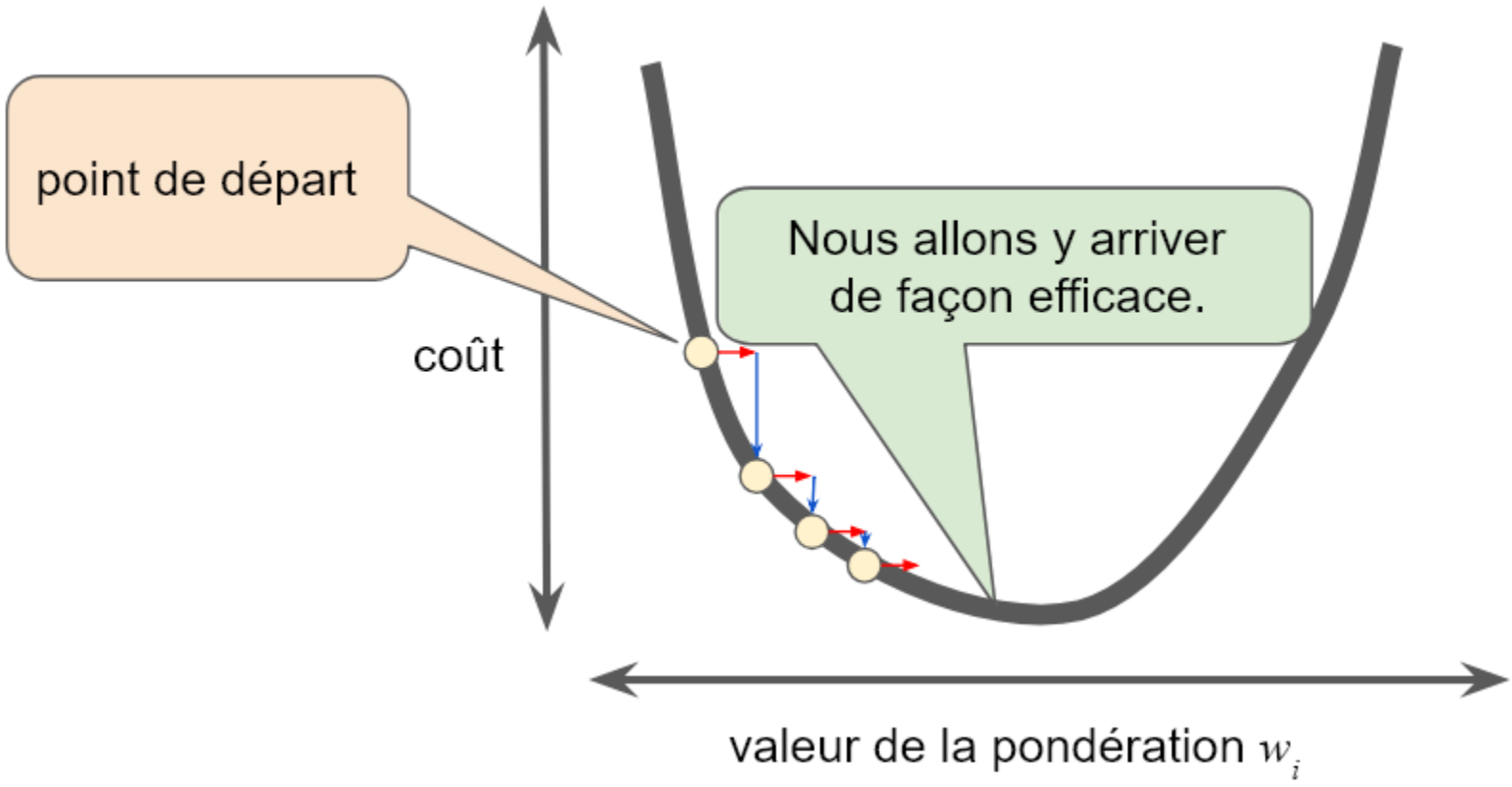


Réduction de la perte : taux d'apprentissage



- **Taux d'apprentissage trop élevé :** minimum dépassé

Réduction de la perte : taux d'apprentissage



- **Taux d'apprentissage efficace**
  - **Exemple :** si gradient faible, essayer un taux élevé

Réduction de la perte point de départ

- **Pour les problèmes convexes :** pondérations commençant de n'importe quel point
- Plus complexe pour les problèmes non convexes : plus d'un minimum
- Les valeurs initiales sont déterminantes pour les problèmes non convexes

Problème convexe

Problème non convexe

Exemple : pratiquement

```
if __name__ == '__main__':

    # Fonction a minimiser
    fc = lambda x,y: (3*x**2) + (x*y) + (5*y**2)
```

$$f_{(x,y)} = (3 * x^2) + (x * y) + (5 * y^2)$$

```
 # Calcul des dérivées partielles
 D_x = lambda x,y : 6*x  + y
 D_y = lambda x,y : 10*y + x
```

```
# Initialisation des variables
x = 10
y = -13
# Pas d'apprentissage
lr = 0.1
print (" *** Valeur initial avant DSG ***")
print ("      Fc= %s" % (fc(x,y)))
print ("\n *** Nouvelles valeurs calculées lors de l'entrainement *** ")
```

```
# epoch : période de minimisation
  for epoch in range(0,20):
      # Calcul des gradients
      G_x = D_x(x,y)
      G_y = D_y(x,y)
      # Appliquer la descente de gradients
      x = x - lr*G_x
      y = y - lr*G_y

      # Vérifier la nouvelle valeur
      print ("Fc= %s" % (fc(x,y)))
```

Descente de gradient pour DNN

- Problème d'optimisation pouvant être représenté par la formule suivante :

$$\min_{w \in \mathcal{W}} f(w) = \frac{1}{n} \sum_{i=1}^{n} \ell\big(h(x_i, w), y_i\big).$$

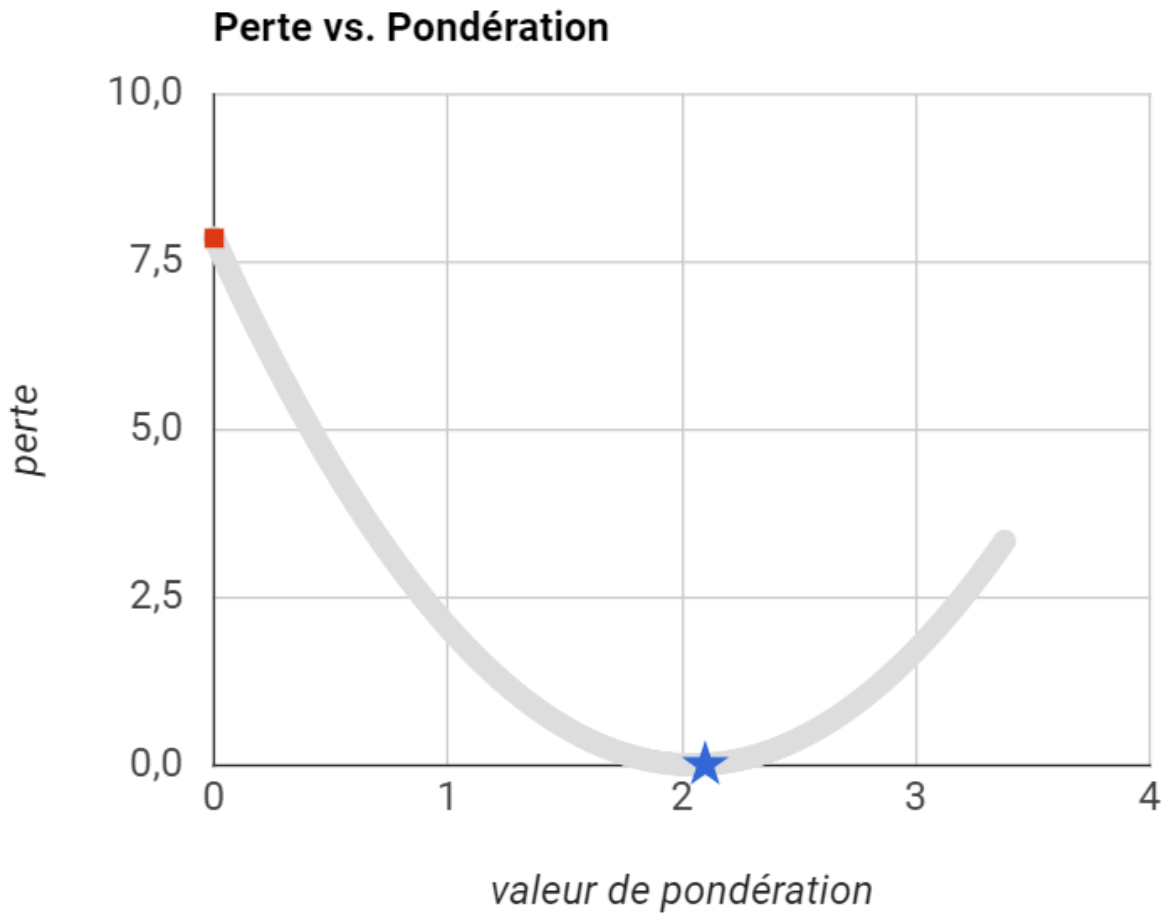
- Ou n : nombre de données,

  Xi : données d'entrainement et Y : labels réels

- Les poids d'un réseau de neurones peuvent être mise à jour comme ceci :

$$w_{t+1} = w_t - \alpha \frac{\partial L}{\partial w_t}$$

Petit exercice



https://developers.google.com/machine-learning/crash-course/fitter/graph

Descente de gradient classique

- **Données :** (X_i, y_i) $i = 1 \ldots N$

  $X_i \in R^I$: **Entrée** $y_i \in R$: **Sortie attendue**

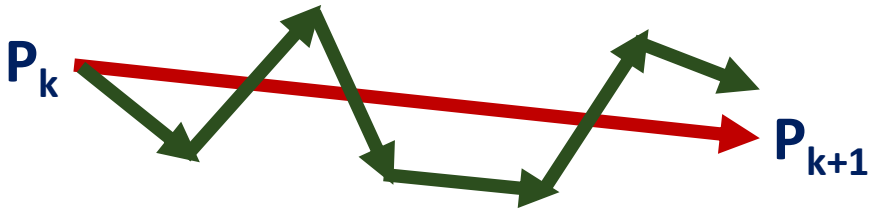
- **Objectif :** $F : R^I \rightarrow R$ $F(X_i) = y_i$
- **$F(X_i)$:** sortie prédite
- F dépend des paramètres $a_1, a_2, \ldots, a_N$
- **Erreur locale :** $E_i = (y_i - F(X_i))^2$
- **Erreur totale :** $E = \sum Ei(a_1, a_2, \ldots, a_N)$ $(i= 1, \ldots, N)$
- But : **minimiser l'erreur en utilisant le gradient**
- **Gradient classique :** grad $E = \sum grad\ Ei$ $(i= 1, \ldots, N)$

Descente de gradient classique

- **Descente de gradient classique**
  - $P_0 = (a_1, a_2, \ldots, a_N)$
  - $P_1 = P_0 - \sigma\ grad\ E(P_0)$
  - $P_2 = P_1 - \sigma\ grad\ E(P_1)$
  - E dépend de toutes les données
  - Problèmes de mémoire
  - Problèmes de temps de calcul

- **Descente de gradient stochastique**
  - $P_0 = (a_1, a_2, \ldots, a_N)$
  - $P_1 = P_0 - \sigma\ grad\ E_1(P_0)$
  - $P_2 = P_1 - \sigma\ grad\ E_2(P_1)$

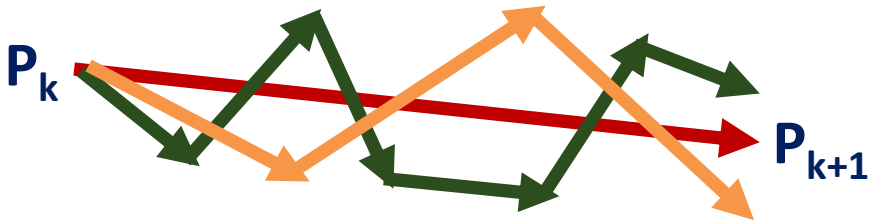
  Mais :
  - Erreur locale
  - Moins intensif et progressif

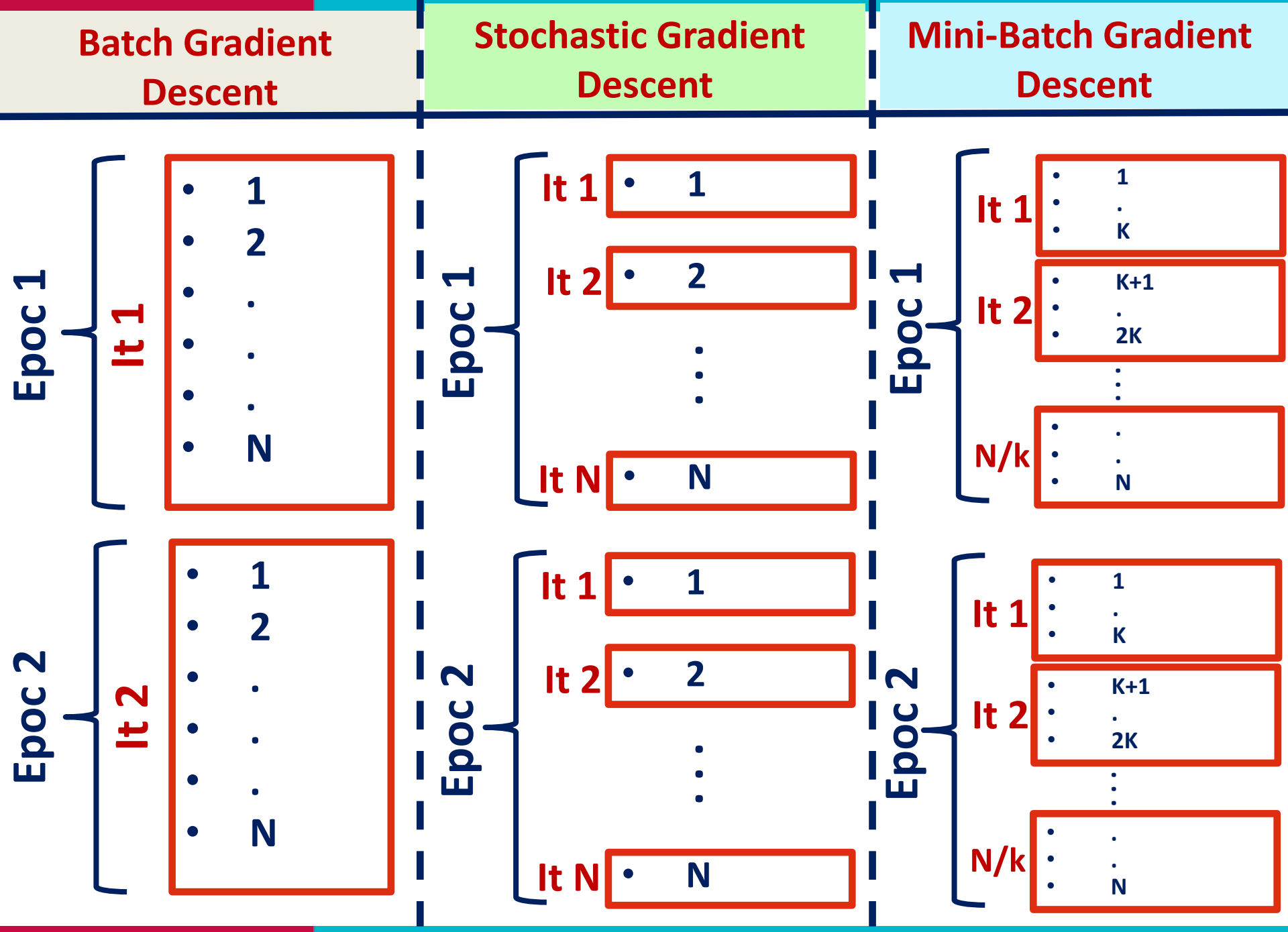


Descente de gradient par lot

- **Descente de gradient par lots**
  - Solution intermédiaire
  - N données
  - $P_0 = (a_1, a_2, \ldots, a_N)$
  - $P_1 = P_0 - \sigma\ grad\ (E_1 + E_2 + Ek)\ (P_0)$
  - $P_2 = P_1 - \sigma\ grad\ (E_k + E_{k+1} + E_{2k})\ (P_1)$

N/k itérations : toutes les données ont servi une fois : **1 époque**




Batch Gradient Descent
Stochastic Gradient Descent
Mini-Batch Gradient Descent
Epoc 1
It 1
1
2
.
.
.
N
Epoc 2
It 2
1
2
.
.
.
N
Epoc 1
It 1
1
It 2
2
It N
N
Epoc 2
It 1
1
It 2
2
It N
N
Epoc 1
It 1
1
.
K
It 2
K+1
.
2K
N/k
.
.
N
Epoc 2
It 1
1
.
K
It 2
K+1
.
2K
N/k
.
.
N

Exercice 1

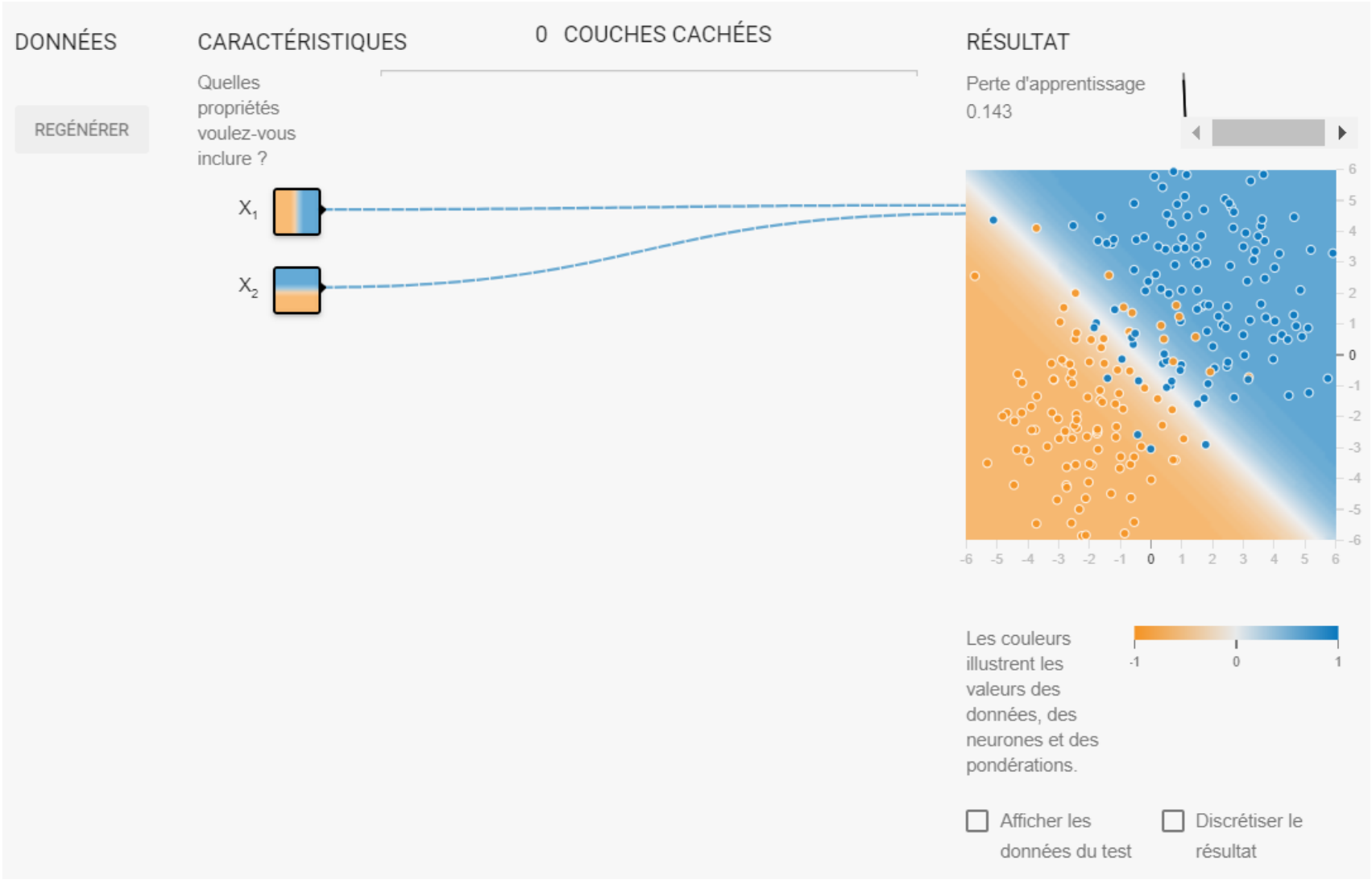
- Veuillez cliquer sur le **lien** suivant pour tester un exemple de régression

- **Question 1:** observez l'évolution du modèle en changeant la valeur du pas 10 ou 20 fois. Observerez-vous une instabilité du modèle ? Pourquoi ?

- Pour les lignes allant de x_1 et x_2 vers la visualisation du modèle. Les pondérations de lignes indiquent celles des caractéristiques dans le modèle. Par exemple, une ligne épaisse indique une pondération élevée

- **Question 2:** Quelle solution proposez vous pour améliorer le modèle ?

Exercice 1


DONNÉES
REGÉNÉRER
CARACTÉRISTIQUES
Quelles propriétés voulez-vous inclure ?
X1
X2
0 COUCHES CACHÉES
RÉSULTAT
Perte d'apprentissage
0.143
Les couleurs illustrent les valeurs des données, des neurones et des pondérations.
Afficher les données du test
Discrétiser le résultat

PLAN

Introduction

I. Définition

II. Types d'apprentissage

III. Terminologie

IV. Réduction de la perte

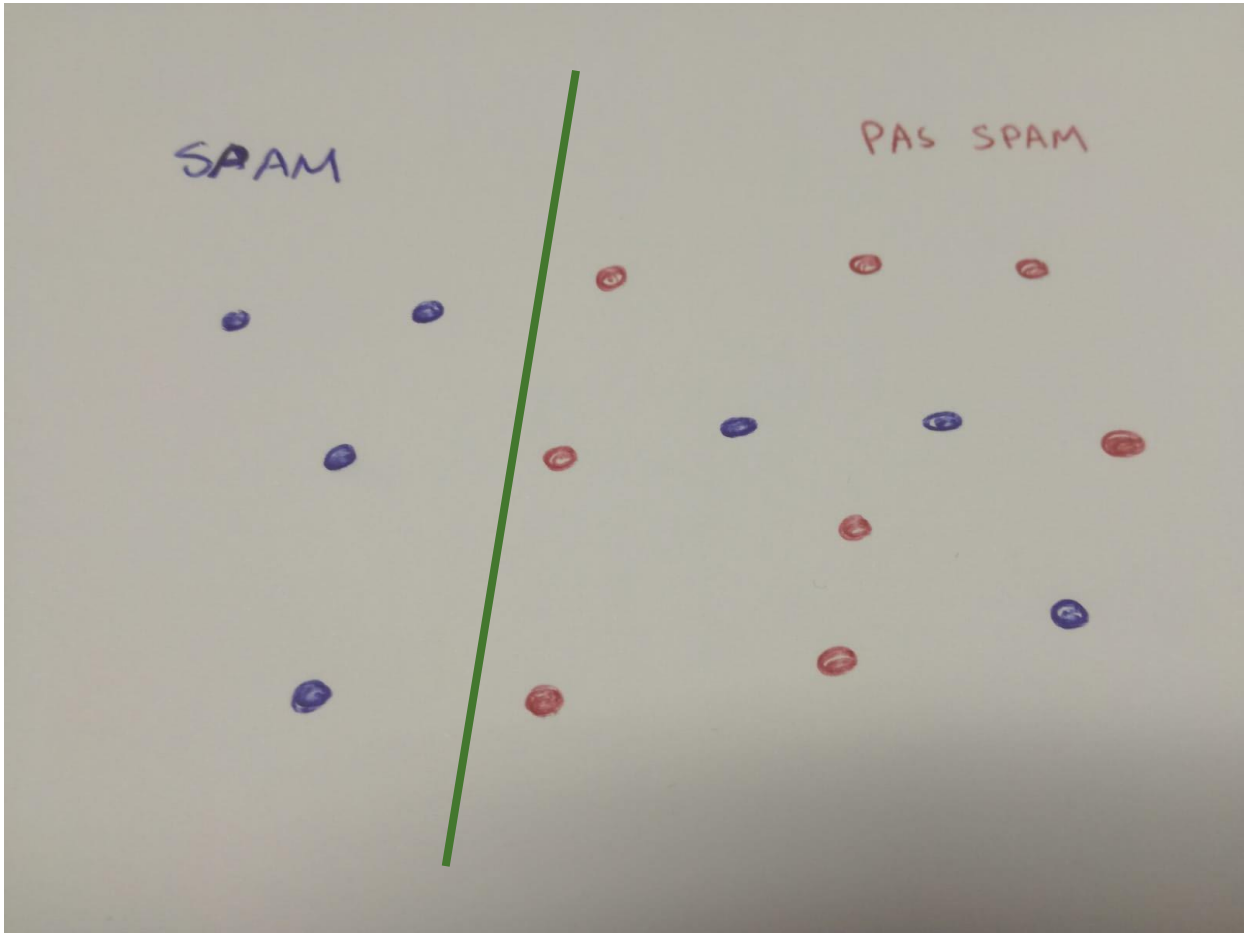
V. **Généralisation et représentation des données**

Conclusion

Généralisation

- Capacité du modèle à s'adapter correctement à de nouvelles données

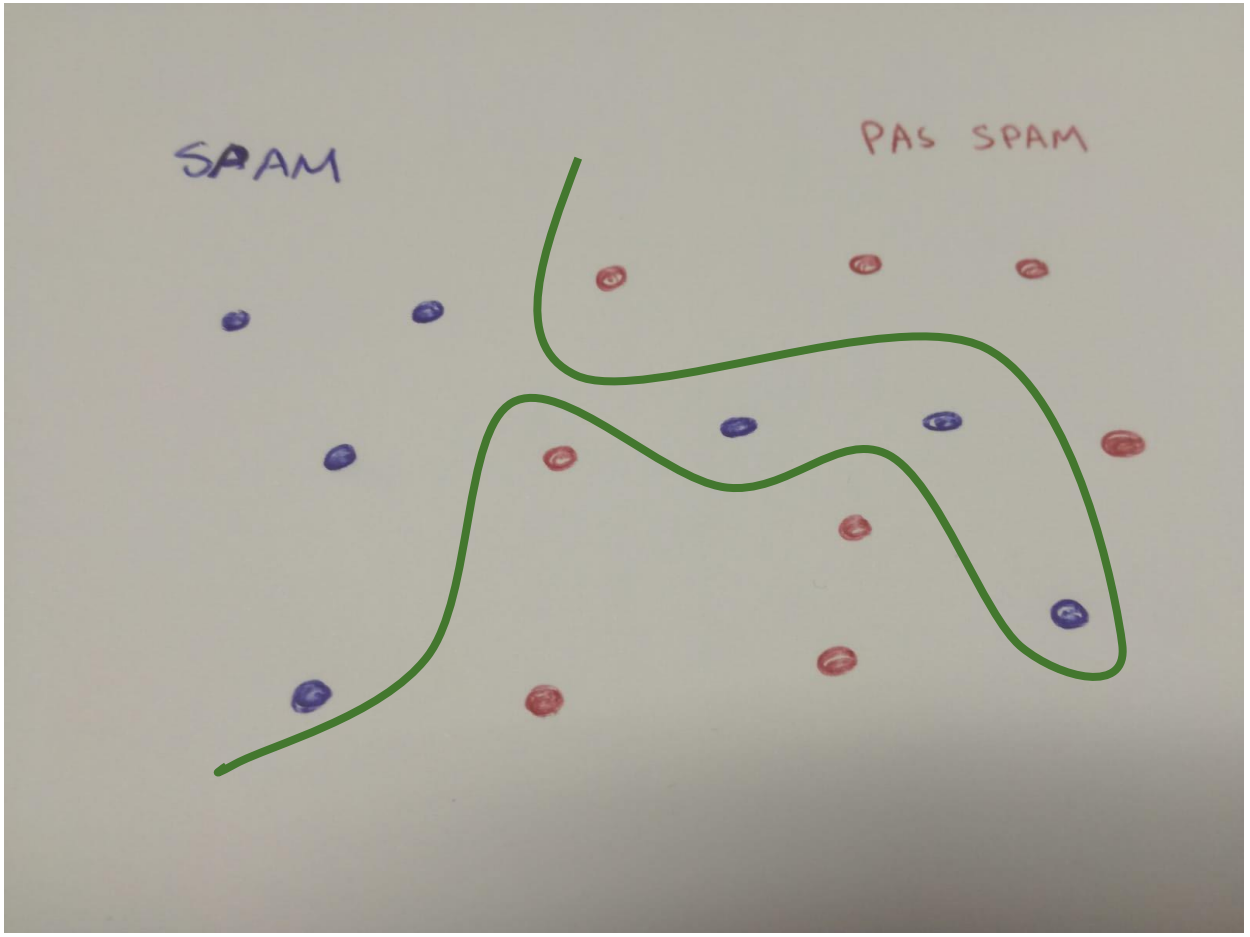
3 erreurs



Généralisation

- Capacité du modèle à s'adapter correctement à de nouvelles données

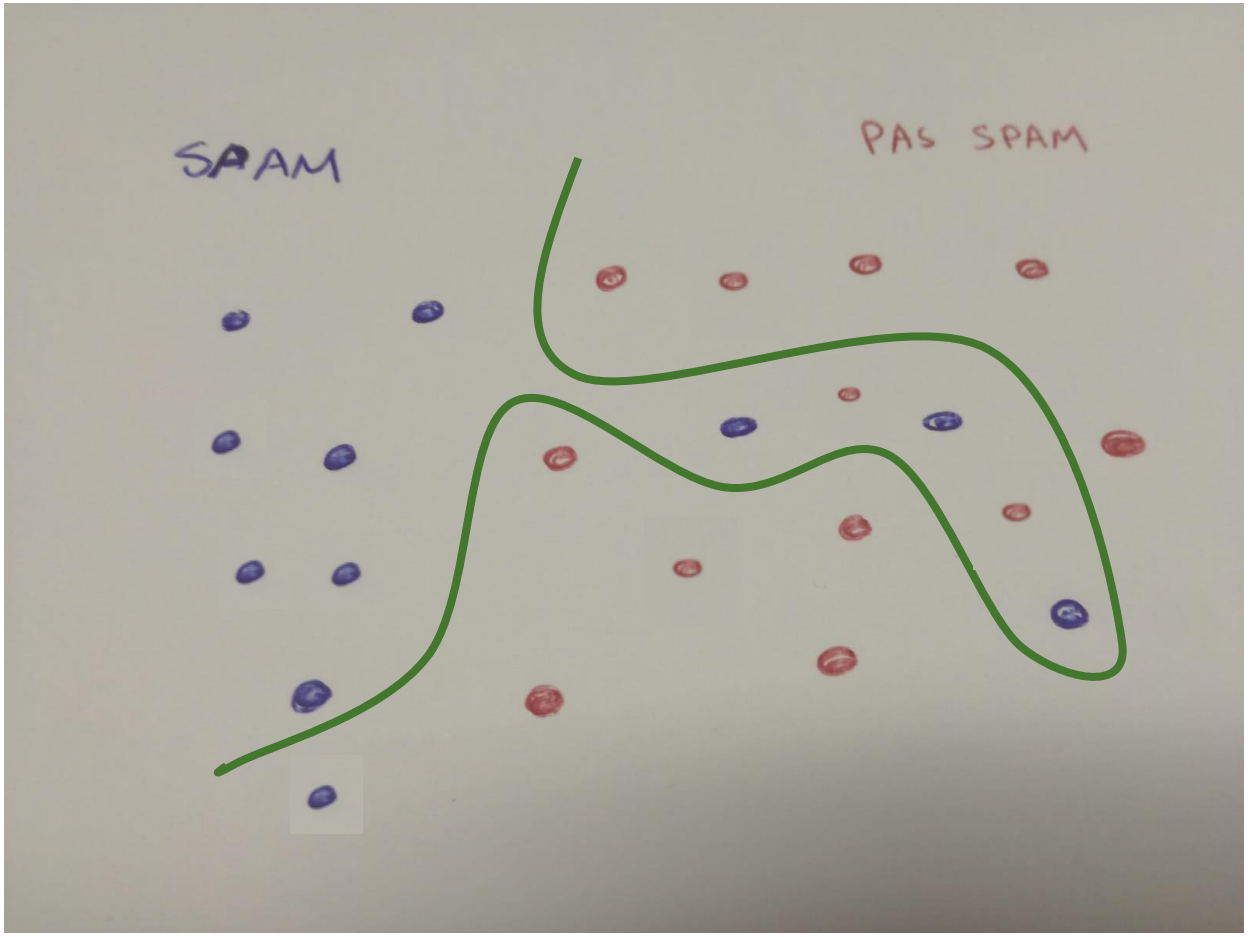
0 erreur



Généralisation

- Problème de surapprentissage : modèle plus complexe que nécessaire

Nouveaux exemples



Généralisation

- Le modèle prédit t-il correctement sur les nouvelles données ?
- Privilégier les théories simples au détriment des concepts compliqués
- **Modèle simple :** plus de chances d'avoir un résultat empirique correct, non dû aux particularités de l'échantillon
- **Ensemble d'apprentissage :** sous-ensemble destiné à l'apprentissage
- **Ensemble d'évaluation :** sous-ensemble destiné à l'évaluation
- Différentes métriques de choix d'ensemble d'évaluation

Ensembles d’apprentissage et d’évaluation

- Ensemble de données suffisamment volumineuses
- Pas trop de divergence entre données d’apprentissage et évaluation
- Evitez de faire l’apprentissage et l’évaluation avec les mêmes données (pas de doublons)

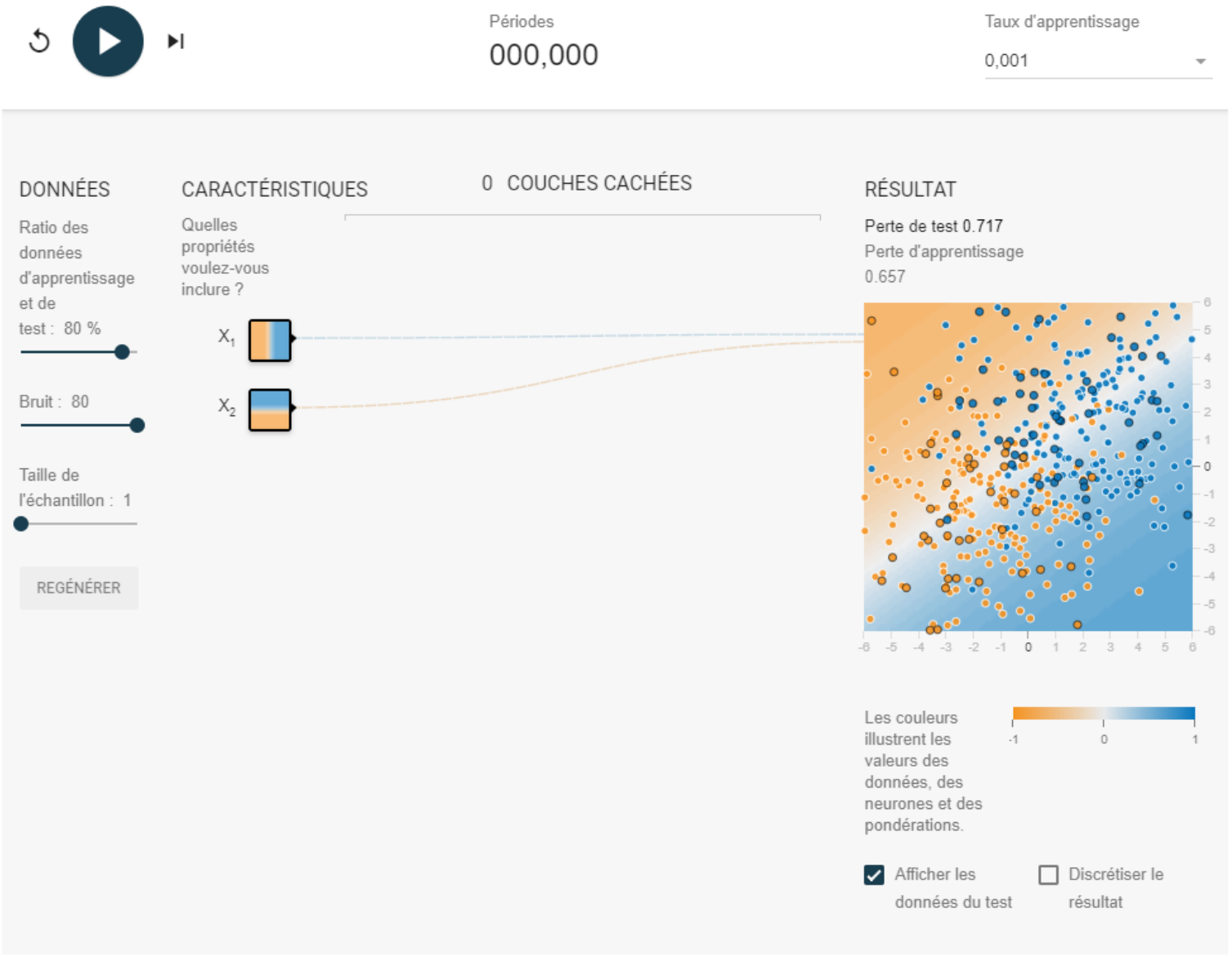
Exercice 2

- Veuillez cliquer sur ce **lien** pour afficher l'ensemble d'évaluation, cliquez sur la case Show test data

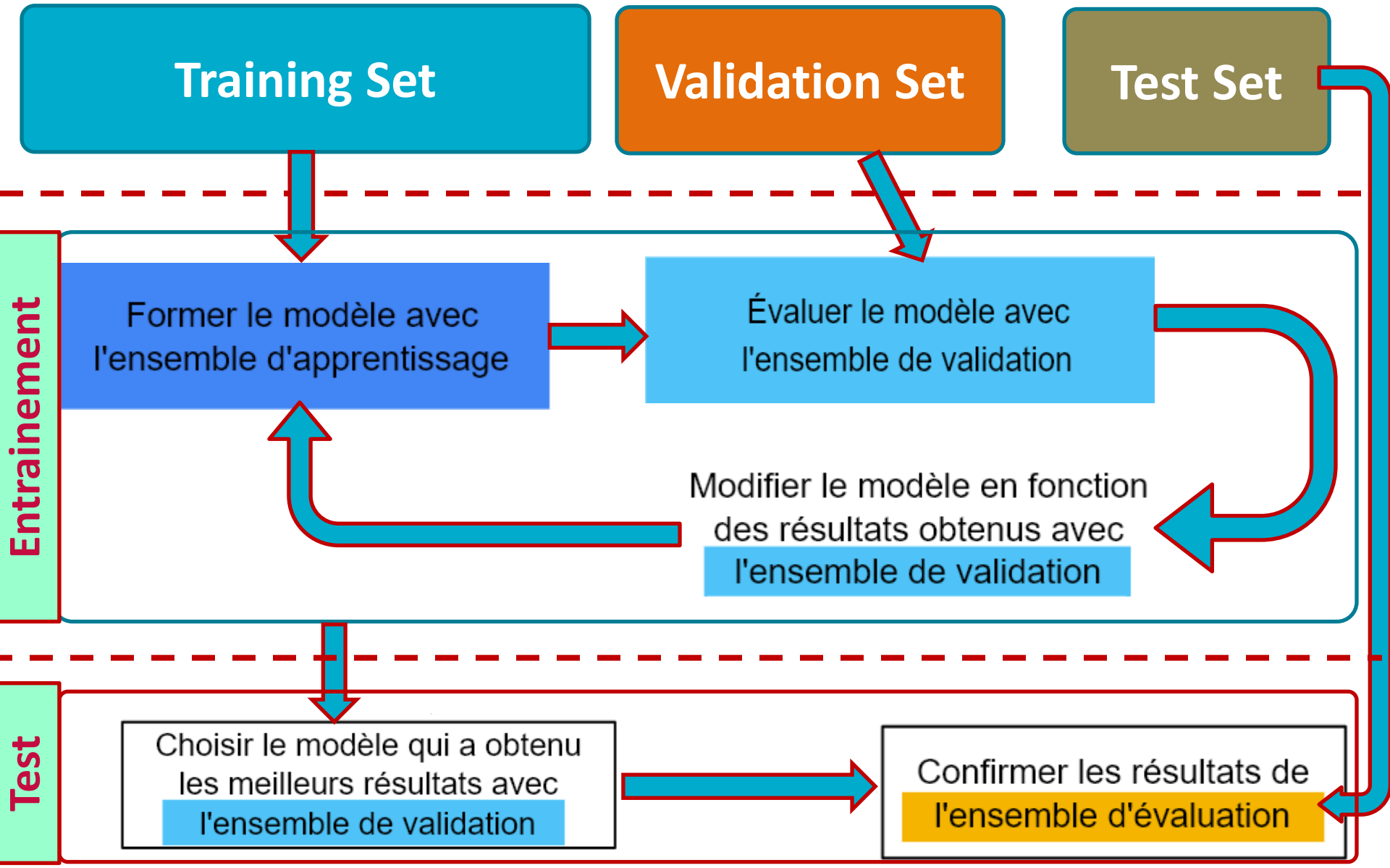
- Chaque type d'élément s'affiche différemment :
  - Les exemples d'apprentissage ont des contours blancs
  - Les exemples d'évaluation ont des contours noirs

- **Question 1:** Appuyez sur « play » et analysez la différence entre perte d'apprentissage et perte de test ?
- **Question 2:** Comment améliorer le modèle ?
- **Question 3:** Analysez l'influence du changement d'ensemble apprentissage/évaluation ?

Exercice 2


Périodes
000,000
Taux d'apprentissage
0,001
DONNÉES
Ratio des données d'apprentissage et de test : 80 %
Bruit : 80
Taille de l'échantillon : 1
REGÉNÉRER
CARACTÉRISTIQUES
Quelles propriétés voulez-vous inclure ?
X1
X2
0 COUCHES CACHÉES
RÉSULTAT
Perte de test 0.717
Perte d'apprentissage
0.657
Les couleurs illustrent les valeurs des données, des neurones et des pondérations.
Afficher les données du test
Discrétiser le résultat

Processus : entraînement/validation/test



PLAN

Introduction

I. Définition

II. Types d'apprentissage

III. Terminologie

IV. Réduction de la perte

V. Généralisation et représentation des données

Conclusion

Conclusion

- L'apprentissage est un élément clé de l'intelligence artificielle
- Apprentissage supervisé : modèle à partir de données annotées
- Apprentissage non supervisé : modèle à partir de données non annotées
- Réduction de la perte : descende de gradient, etc.
- Généralisation et validation
- Représentation des données

Références

[1] Russel, S. Et Norvig, P., **"Artificial Intelligence : A Modern Approach "** *3rd edition, Pearson.* 2010

[2] R. O. Duda et al, "**Pattern Classification**", *chapter : Unsupervised Learning and Clustering. Wiley Inter science* (2001)

[3] S. Kotsiantis. "**Supervised machine learning: A review of classification techniques**". *Informatica Journal, 31 :249–268* (2007)

[4] O. Chapelle, "**Semi-supervised Learning**". *MIT Press* (2006)

[5] R. Sutton et al, "**Reinforcement Learning - An Introduction**", *MIT Press* (2012)

[6] Google, "**Cours d'initiation au Machine Learning avec mes API Tensor Flow**", *https://developers.google.com/machine-learning/crash-course/*

MERCI