Machine Learning

5A – 3DJV

Bienvenue !

- Contact : pro@nicolasvidal.fr

Modalités

- Semaine thématique
- ≈ 12h de cours
- ≈ 15h de projet
- Parlons du projet!

Outils à installer

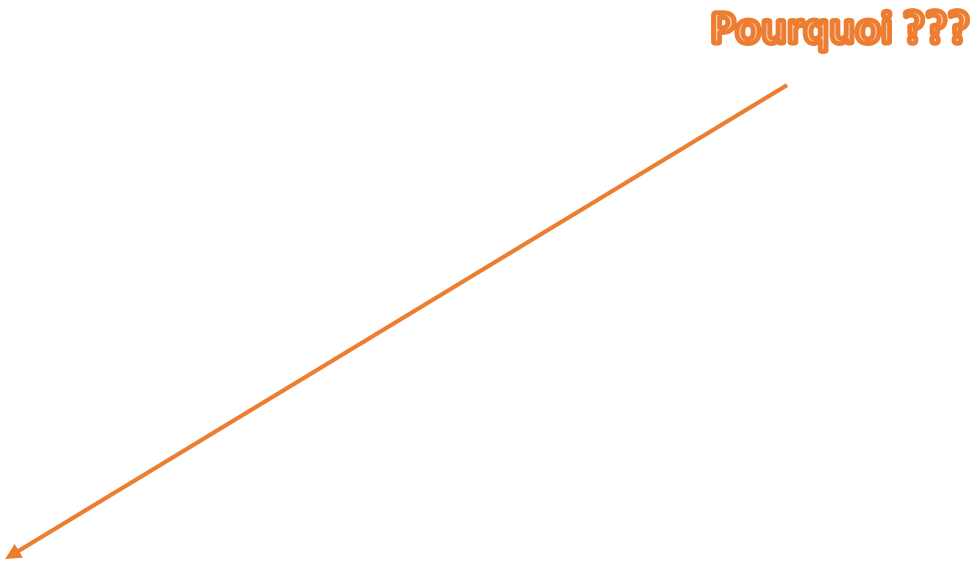
- Un environnement de développement de JV:
  - Unity
  - Unreal Engine
  - … ?

- Contraintes :
  - Visualisation 3D simple
  - Pouvoir importer une dll C/C++
  - .NET/Java dans le pire des cas (mais déconseillé)

Outils à installer

- Un environnement de développement de JV:
  - Unity
  - Unreal Engine
  - … ?

- Contraintes :
  - Visualisation 3D simple
  - Pouvoir importer une dll C/C++
  - .NET/Java dans le pire des cas (mais déconseillé)



Outils à installer

- Un enviro
  - Unity
  - Unreal E
  - … ?
- Contraintes
  - Visualisat
  - Pouvoir im
  - .NET/Java

Pourquoi ???

Outils à installer

- Un environnement de développement de JV:
  - Exemples
    - Unity
    - Unreal Engine
    - … ?
  - Contraintes :
    - Visualisation 3D simple
    - Pouvoir importer une dll C/C++
    - .NET/Java dans le pire des cas (mais déconseillé)

Outils à installer

- Un environnement de calcul scientifique et modélisation
  - Exemples
    - Mathematica (trial 15 days)
    - Anaconda (Python ! ☺)
    - Octave (Matlab-like)
    - … ?
  - Contraintes :
    - Pouvoir importer une dll C/C++
    - .NET/Java dans le pire des cas (mais déconseillé)
    - Plot / Génération de data aisée

Outils à installer

- Un environnement de développement en C/C++
  - Exemples
    - Visual Studio
    - Build Essentials
    - … ?
  - Contraintes :
    - Pouvoir créer une dll C/C++
    - .NET/Java dans le pire des cas (mais déconseillé)

Le projet !

- Le cours était un dilemme …

VS

Le projet !

- Le cours était un dilemme …

Old School Machine Learning / Statistical Learning

VS

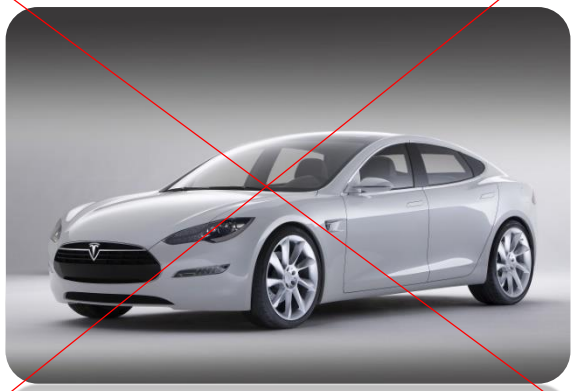
Deep Learning

Le projet !

- Le cours était un dilemme …

Old School Machine Learning / Statistical Learning

VS



Deep Learning

Le projet !

- Vous faire implémenter votre propre toolbox de Machine Learning !

Le projet !

- Vous faire implémenter votre propre toolbox de Machine Learning !
- Vous faire implémenter votre propre toolbox de Machine Learning !

Le projet !

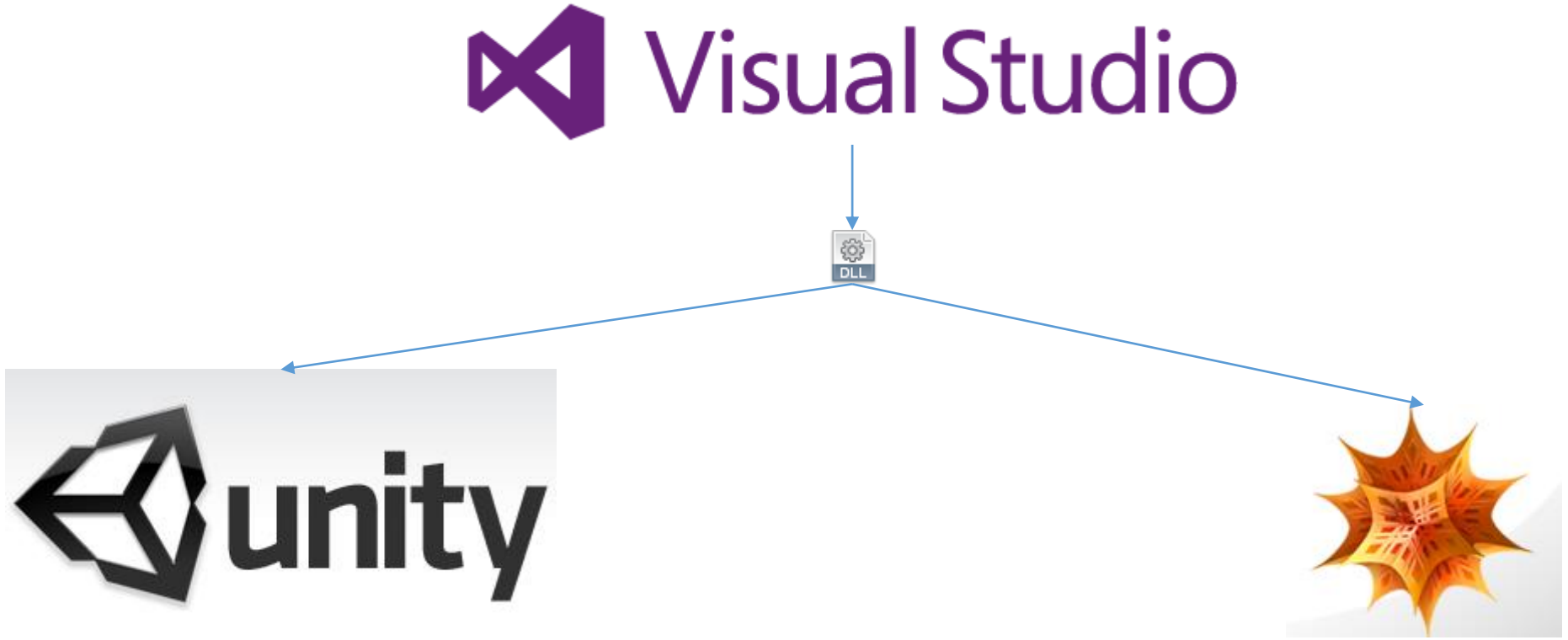
- Vous faire implémenter votre propre toolbox de Machine Learning !
- Vous faire implémenter votre propre toolbox de Machine Learning !

- Utiliser votre toolbox dans votre environnement de développement de JV préféré

- Acquérir une sensibilité à la problématique de l'apprentissage artificiel supervisé

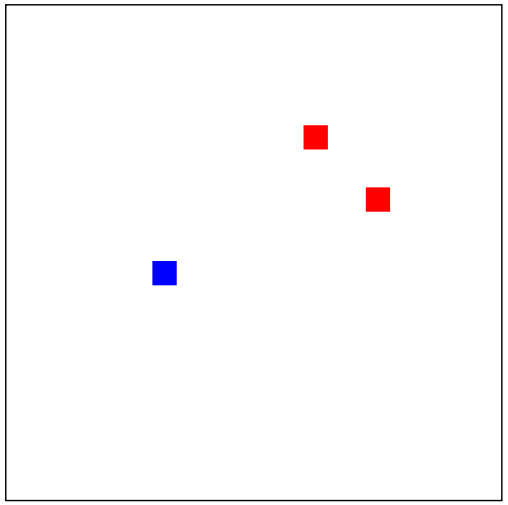
Le projet !

1. Mettre en place le pipeline de développement

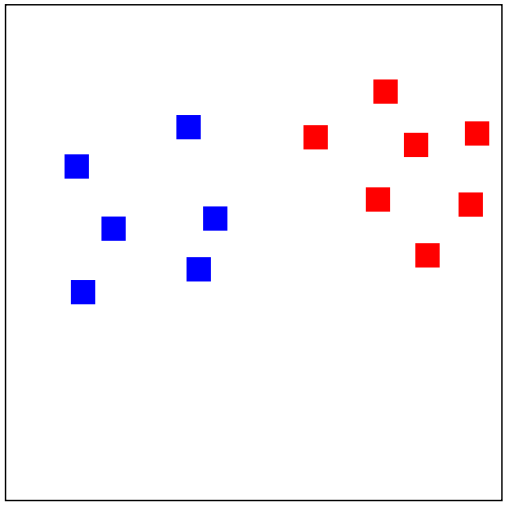


Le projet !

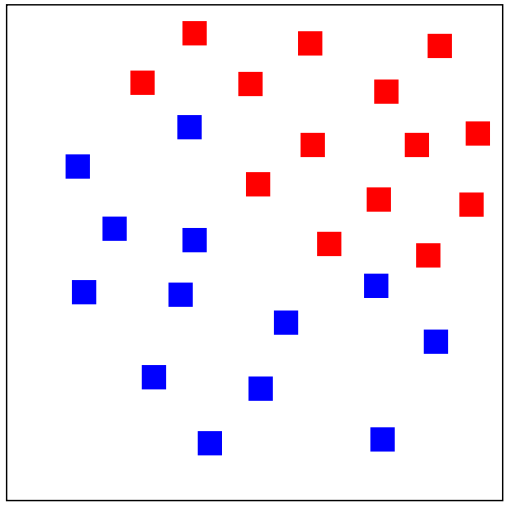
2. Création des cas de tests



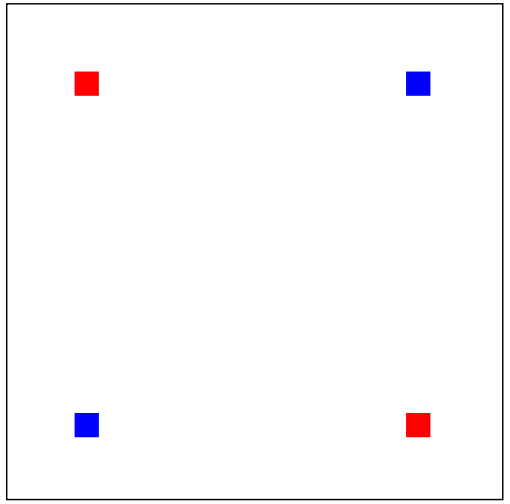
Simple, linéairement séparable



Réel, linéairement séparable



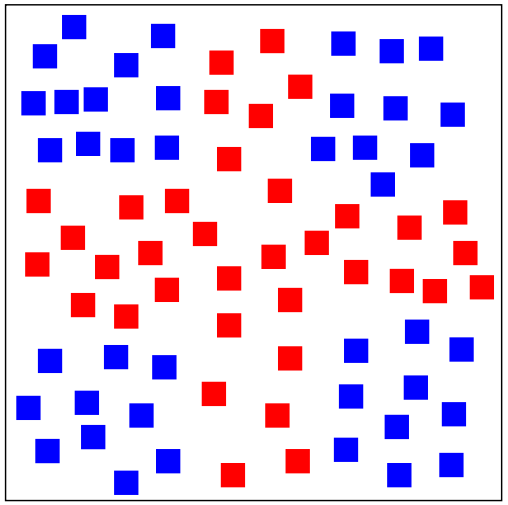
Soft, non linéairement séparable



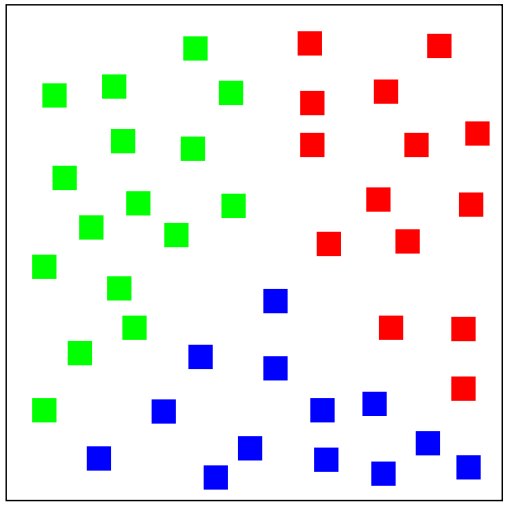
XOR

Le projet !

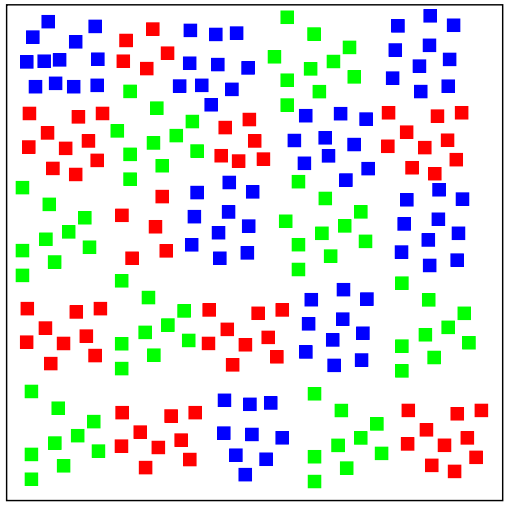
2. Création des cas de tests



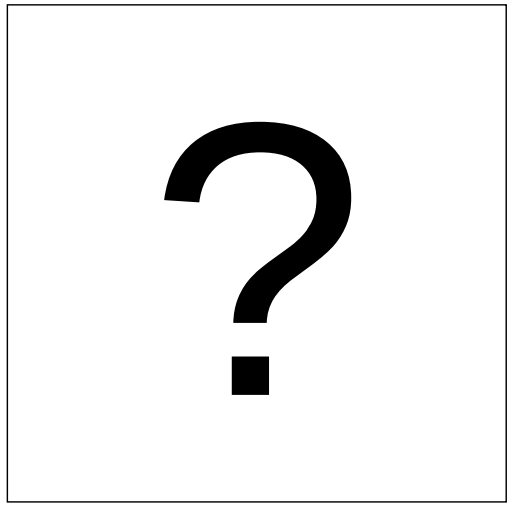
Cross



Multi Class Soft



Multi Class Hard

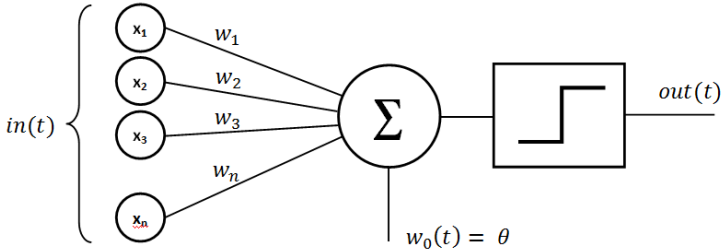


Real Dataset

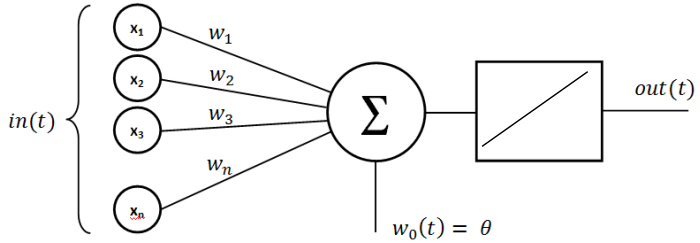
Le projet !

3. Implémentation des algorithmes

1. Modèles linéaires



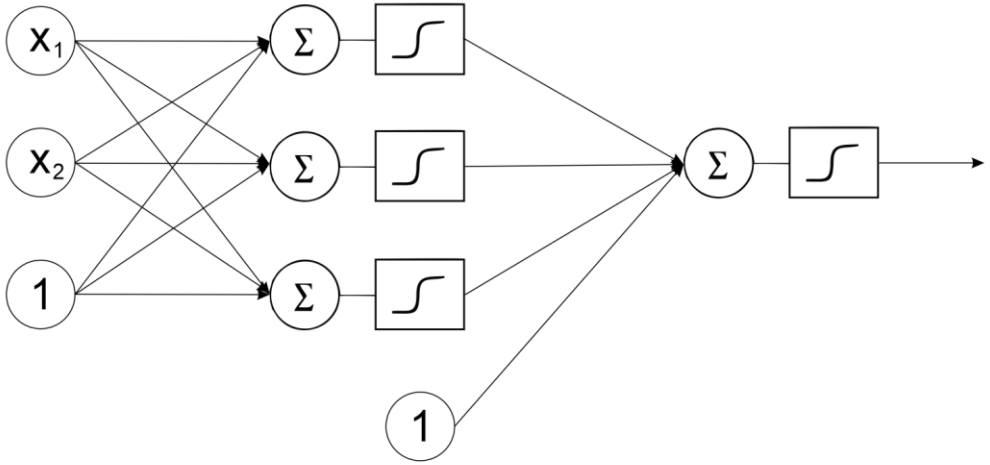
Perceptron pour la Classification



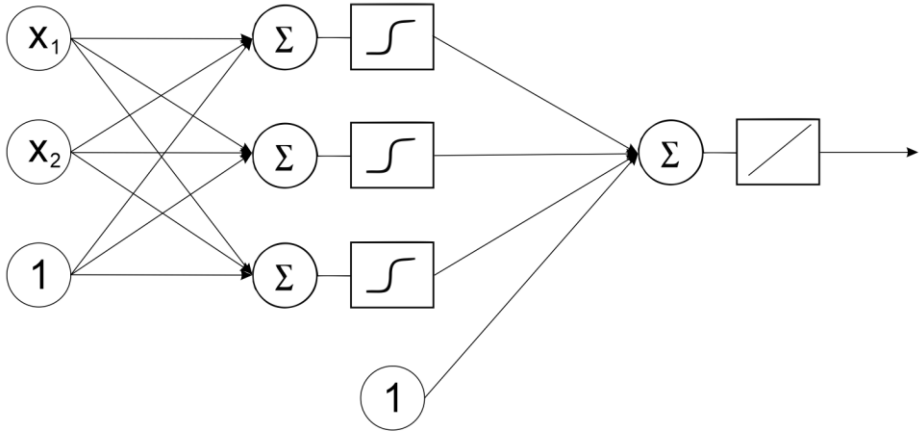
Perceptron pour la Régression

Le projet !

3. Implémentation des algorithmes
    2. Perceptron Multi Couches



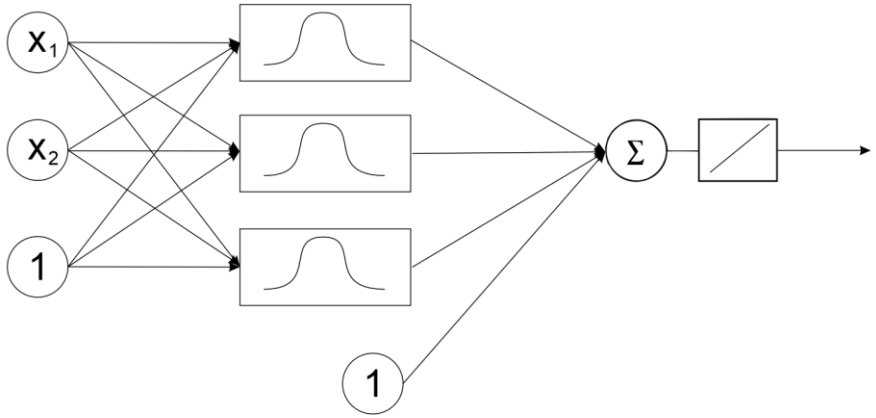
Perceptron multi couches pour la classification



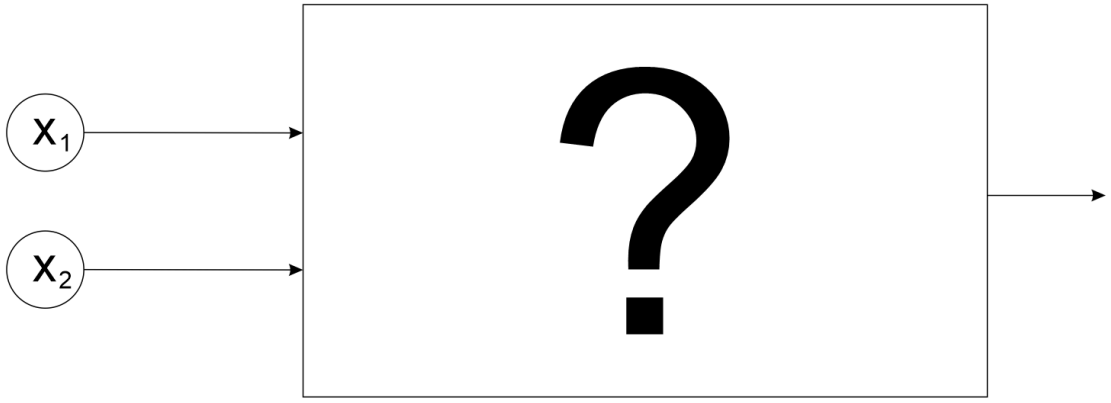
Perceptron multi couches pour la régression

Le projet !

3. Implémentation des algorithmes
    3. Modèles non linéaires



RBF (s)



Autre

Le projet !

4. Application a un dataset réel
    1. Trouver un dataset réel (ne soyez pas trop ambitieux ! ☺)
        - https://archive.ics.uci.edu/ml/datasets/
        - http://grouplens.org/datasets/movielens
        - http://yann.lecun.com/exdb/mnist/
        - https://www.kaggle.com/
        - … ?
    2. Etablir un protocole de test
    3. Entrainement du/des modèles
    4. Présentation et analyse des résultats

Le projet !

- Modalités pratiques
  - Groupes de 4/5
    - Répartition des tâches est à éviter pour l'implémentation
  - Evaluation intermédiaire Mercredi
  - Soutenance : 20 minutes (15 présentation + 5 questions)
  - Amusez-vous !

Qu’est-ce qu’apprendre ?

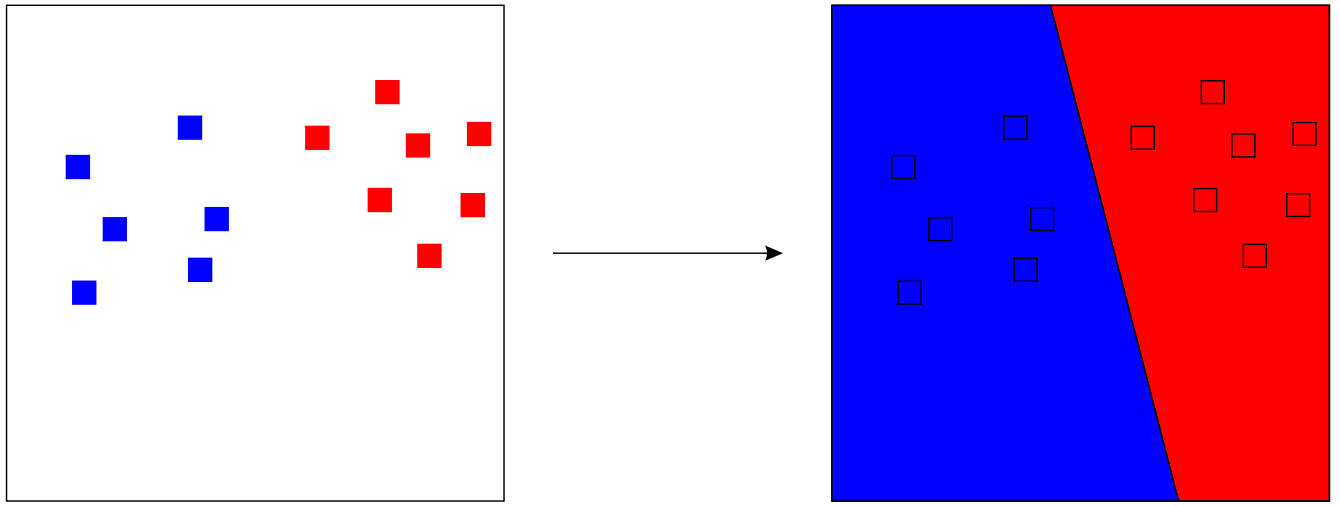
Qu’est-ce qu’apprendre ?

Intuition :

Découvrir (ou estimer) une fonction (ou une distribution) inconnue à partir d’un ensemble d’exemples

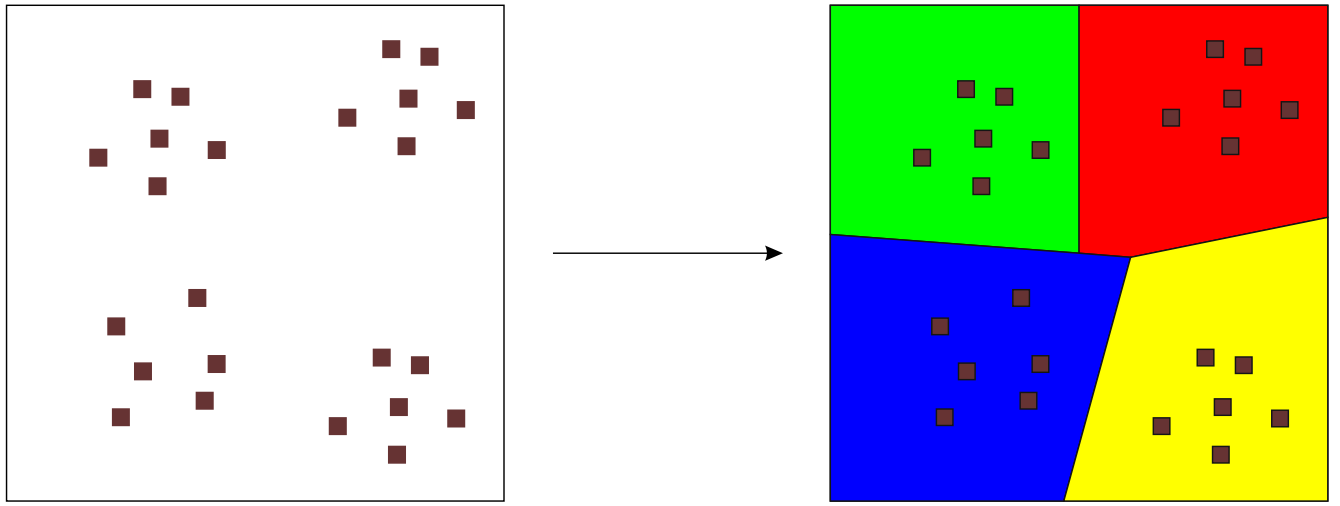
Qu’est-ce qu’apprendre ?

Apprentissage supervisé :



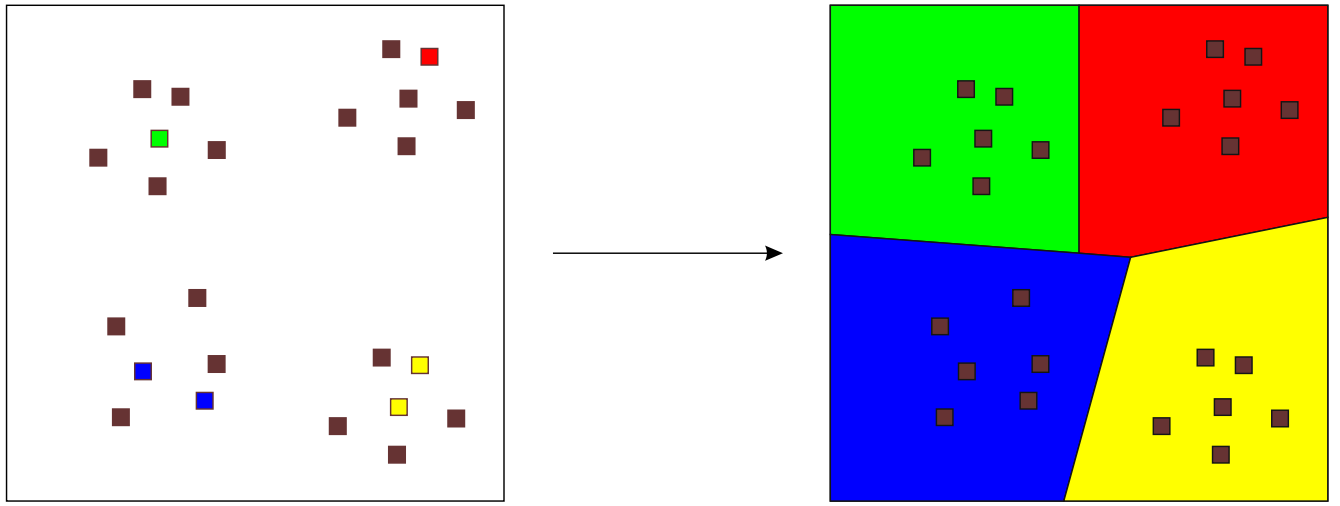
Qu’est-ce qu’apprendre ?

Apprentissage non supervisé :



Qu’est-ce qu’apprendre ?

Apprentissage semi supervisé :



Qu’est-ce qu’

Apprentissage

Apprentissage supervisé

Apprentissage supervisé

Apprendre …

Apprentissage supervisé

Apprendre …

- Apprendre par cœur ?

Apprentissage supervisé

Apprendre …

- Exemples (Input => Output)
  - {1 , 2} => {3}
  - {4 , 2} => {6}
  - {2 , 2} => {4}
  - {8 ,13} => {21}

Apprentissage supervisé

Apprendre …

- Exemples (Input => Output)
  - {1 , 2} => {3}
  - {4 , 2} => {6}
  - {2 , 2} => {4}
  - {8 ,13} => {21}
- Apprendre par cœur ?
  - Dictionnaire ?

Apprentissage supervisé

Apprendre …

- Exemples (Input => Output)
  - {1 , 2} => {3}
  - {4 , 2} => {6}
  - {2 , 2} => {4}
  - {8 ,13} => {21}
- Apprendre par cœur ?
  - Dictionnaire ?
  - Aucune information sur le reste de l'espace d'entrée !

Apprentissage supervisé

Apprendre …

… n’a d’intérêt que si on généralise !

Apprentissage supervisé

Qu’est-ce que généraliser ?

Apprentissage supervisé

⇒Généraliser :

⇒Supposer qu'il existe une **fonction cible** qui a généré les exemples que nous avons à disposition.

⇒Essayer d'**approximer** les résultats de cette fonction cible à l'aide d'un modèle.

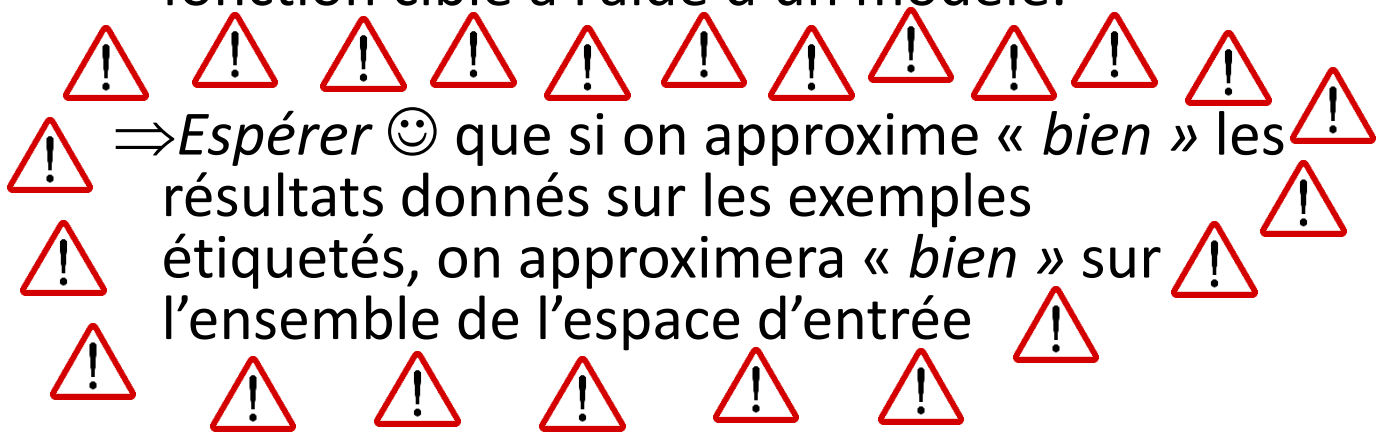
⇒*Espérer* ☺ que si on approxime « *bien* » les résultats donnés sur les exemples étiquetés, on approximera « *bien* » sur l'ensemble de l'espace d'entrée

Apprentissage supervisé

⇒Généraliser :

⇒Supposer qu'il existe une **fonction cible** qui a généré les exemples que nous avons à disposition.

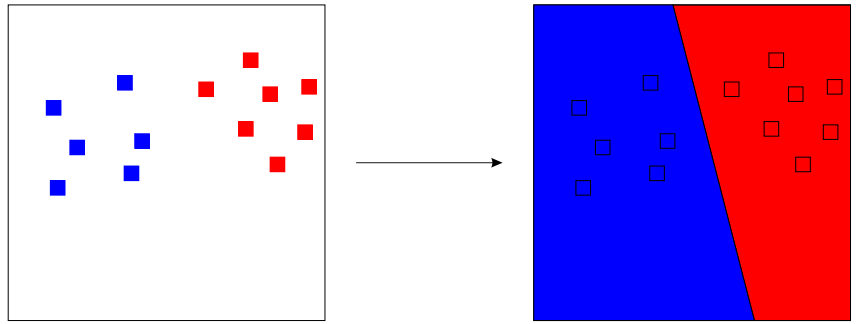
⇒Essayer d'**approximer** les résultats de cette fonction cible à l'aide d'un modèle.



⇒*Espérer* ☺ que si on approxime « *bien* » les résultats donnés sur les exemples étiquetés, on approximera « *bien* » sur l'ensemble de l'espace d'entrée

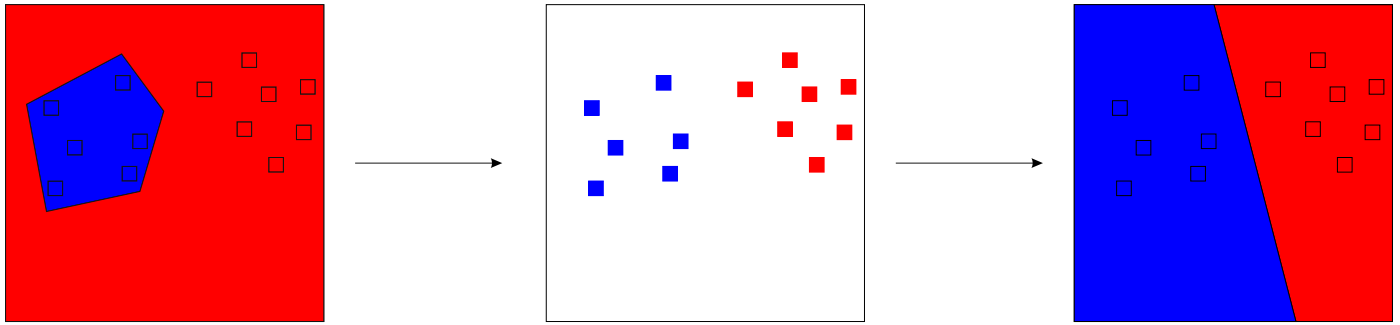
Apprentissage supervisé

⇒Contre exemple abstrait:



Apprentissage supervisé

⇒Contre exemple abstrait:



Apprentissage supervisé

⇒Contre exemple de l'arnaque à la prédiction

Apprentissage supervisé

Arnaque à la prédiction :

Apprentissage supervisé

Arnaque à la prédiction :

Apprentissage supervisé

Arnaque à la prédiction :

Apprentissage supervisé

Arnaque à la prédiction :

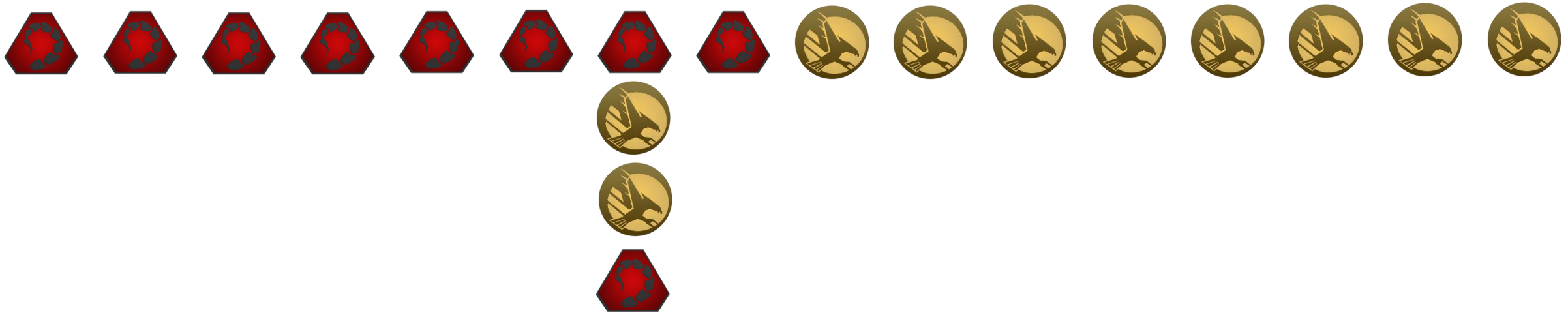
Apprentissage supervisé

Arnaque à la prédiction :

?

Apprentissage supervisé

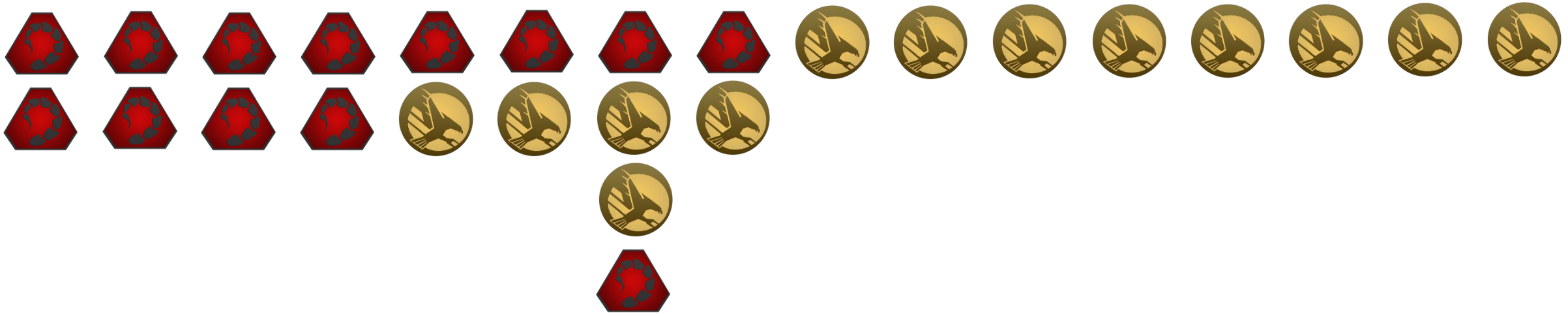
Arnaque à la prédiction :



?

Apprentissage supervisé

Arnaque à la prédiction :



?

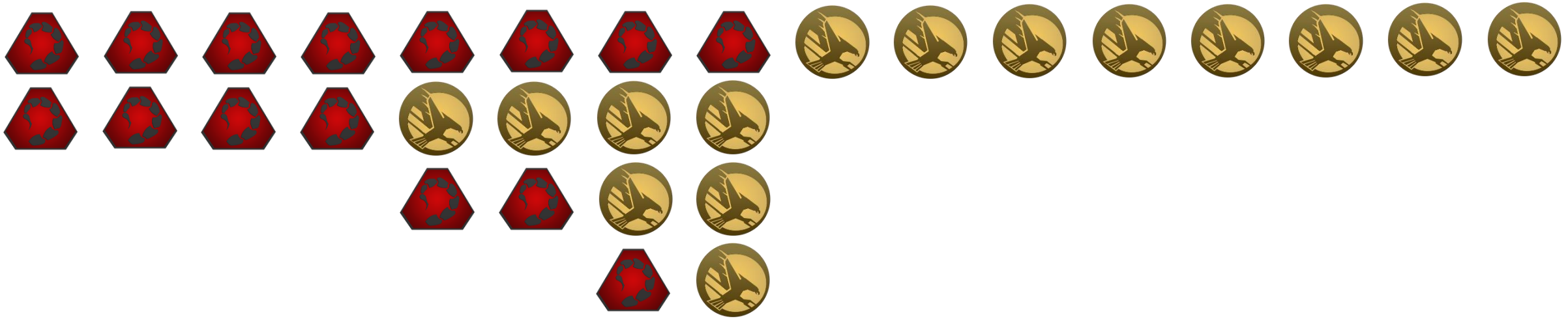
Apprentissage supervisé

Arnaque à la prédiction :

?

Apprentissage supervisé

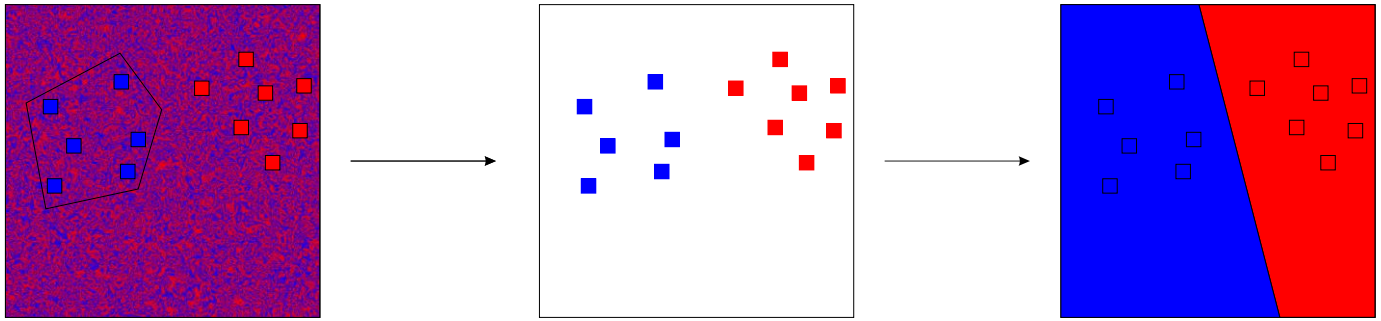
Arnaque à la prédiction :



?

Apprentissage supervisé

⇒Arnaque à la prédiction:



Apprentissage supervisé

Quelles validations théoriques ?

https://work.caltech.edu/telecourse.html

Apprentissage supervisé

Inégalité de Hoeffding :

soit μ: probabilité d'obtenir un échantillon bleu dans un ensemble

soit ν: proportion d'échantillons bleus dans un échantillonnage

Si N est mon nombre d'échantillons et ϵ un réel :

$$P[|\nu - \mu| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$

Apprentissage supervisé

Ce qui nous amène à :

soit E_{in}: l'erreur de classement d'une hypothèse sur les échantillons par rapport à la fonction cible.

soit E_{out}: l'erreur de classement d'une hypothèse l'ensemble des entrées possibles par rapport à la fonction cible.

soit g : mon hypothèse

soit M : L'ensemble des hypothèses possibles pour mon modèle

$$P[|E_{in}(g) - E_{out}(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}$$

Apprentissage supervisé

Inégalité de Vapnik-Chervonenkis :

soit m_H : le nombre maximum de dichotomies réalisables sur un ensemble d'exemples par une classe d'hypothèse H.

$$P[|E_{in}(g) - E_{out}(g)| > \epsilon] \leq 4m_H(2N)e^{-\frac{1}{8}\epsilon^2 N}$$

Apprentissage supervisé

Conclusion théorique :

- Généraliser est parfois possible
- Cela dépend :
  - Du nombre d'exemples étiquetés à disposition
  - De la qualité de la généralisation que l'on cherche
  - De la complexité du modèle utilisé pour générer nos hypothèses

- Règle générale, approximative mais pratique :
  - Ne pas espérer obtenir une bonne généralisation si le nombre d'exemple à disposition n'est pas supérieur à **10 fois** le nombre de paramètres du modèle utilisé.

Classification VS Régression

Classification :

- Appartenance d’un exemple à un ensemble fini :

Classification VS Régression

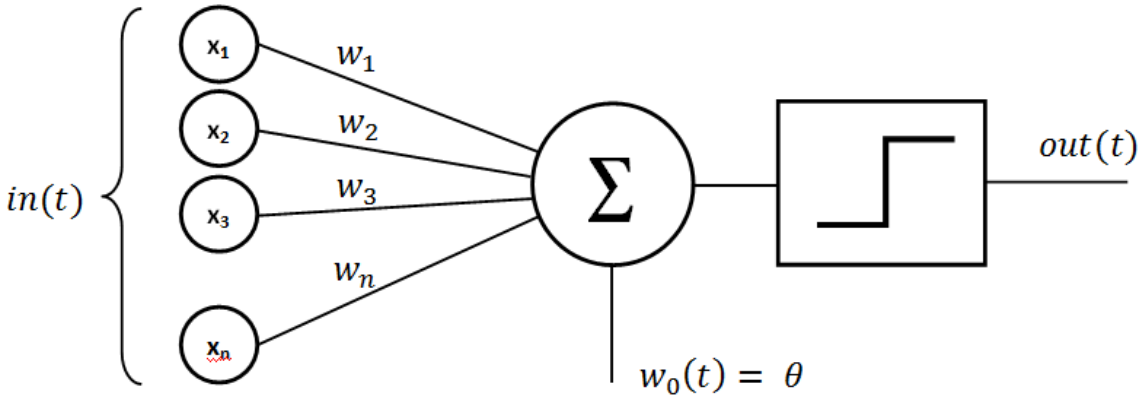
Régression :

- Prédire une (ou plusieurs) valeurs réelles :

Classification et séparations linéaires

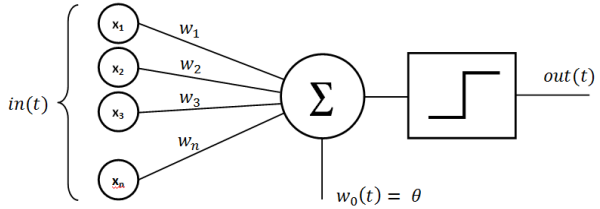
Classification et séparations linéaires

- Retours sur le Perceptron



Classification et séparations linéaires

- Retours sur le Perceptron



- Que l'on peut réécrire :
  - $out = Sign(\sum_{i=0}^{n} w_i x_i)$

- Ou sous forme matricielle :
  - $out = Sign(W^T X)$ en prenant soin d'ajouter le biais $(x_0 = 1)$

Classification et séparations linéaires

- Algorithmes d'apprentissages du perceptron pour la classification
- But du jeu : déterminer W
- Non supervisée
  - Règle de Hebb
- Supervisée
  - PLA ou Règle de Rosenblatt

Classification et séparations linéaires

- Perceptron Learning Algorithm (pour des sorties à -1 ou 1)
  - Initialiser W (randomf(-1,1) ou 0)
  - Répéter :
    - Prendre un exemple MAL classé (où $g(X^k) \neq Y^k$) au hasard et, mettre à jour W selon la règle :

$$W \leftarrow W + \alpha Y^k X^k$$

- Règle de Rosenblatt (pour des sorties à 0 ou 1)
  - Initialiser W (random(-1,1) ou 0)
  - Répéter :

$$W \leftarrow W + \alpha\left(Y^k - g(X^k)\right)X^k$$

Avec :

- α le pas d'apprentissage
- X^k les paramètres de l'exemples k et le biais $x_0^k = 1$.
- Y^k la sortie attendue pour l'exemple k.
- $g(X^k)$ la sortie obtenue par le perceptron pour l'exemple k.

Régression linéaire

- Minimiser le carré de l'erreur
- Notons $X = \begin{bmatrix} x_0^0 & \cdots & x_n^0 \\ \vdots & \ddots & \vdots \\ x_0^k & \cdots & x_n^N \end{bmatrix}$ et $Y = \begin{bmatrix} y_0^0 & \cdots & y_n^0 \\ \vdots & \ddots & \vdots \\ y_0^k & \cdots & y_n^N \end{bmatrix}$
- Supposons $n \leq N$
- Utilisation de la pseudo inverse pour calculer W en un coup :

$$W = ((X^T X)^{-1} X^T) Y$$

Exemples

Implémentation

- A vos claviers !