Big-Data & ML - 5A INFO APP

26/01/2021 à 14h00

Big Data & ML (Machine Learning).

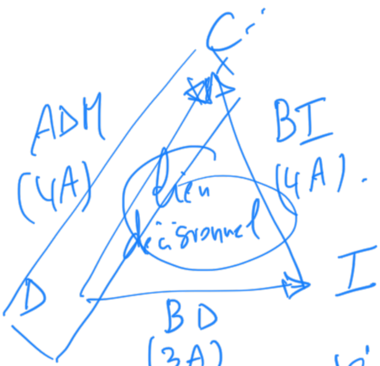
30h (BELGHAZER).

15h : CM + TD + TP (Python).

Big : - V : Volume (Quantité).
- V : Vélocité (flux).
- V : Variété : types complexes.
- V : Véracité : la fiabilité.
- V : Valeur : la pertinence.

Data : Données.

Big-Data → BD
→ ML (Analyse de données)



Préparation du terrain :
+ Prétraitement et Codage
+ Sélection de variables
  + Eliminer les variables avec la même valeur pour tous les instances.
  + Détecter les fortes corrélations.
  + Traiter les données manquantes.
  * ... les données.

Normaliser ...

→ Apprentissage automatique (Machine Learning) :

Une discipline de l'IA qui consiste à modéliser une fonction statistique pour résoudre les problèmes complexes linéairement / non linéairement séparables.

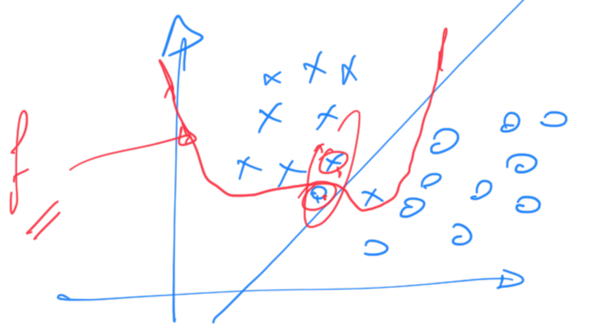
$$y = f(x) \in \mathbb{R}^p$$

Cible (Target) / Variable à expliquer

mono-label : bi-classes, multi-classes

multi-label

$\mathbb{R}^p$ (variables explicatives).



$$\| f - \tilde{f} \|$$

→ Types d'apprentissage :

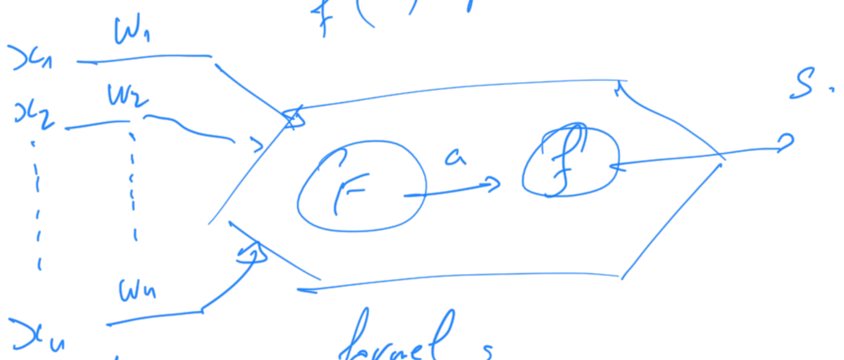
Apprentissage

- Symbolique (Math discrètes).
- Numérique (Maths appli).
  - génératif.
  - déterministe
    - Non supervisé : y est totalement inconnu.
    - Semi-supervisé : y est partiellement connu.
    - Supervisé : y est totalement connu.
      - à base de distances.
      - Connexionniste. NN.
      - à base de Noyeau
      - à base d'induction. DT + RF
      - à base de régression

... des

Le neurone formel : une unité de traitement qui reçoit des données en entrée, $x = (x_1, x_2, \dots, x_n)$ et produit une sortie réelle s S en fonction d'un vecteur de poids de connexions $w = (w_1, w_2, \dots, w_n)$.

⟶ le neurone formel effectue 2 transformations. $F(x, w)$ pour calculer la 1ère est linéaire son activation a. La 2ème fct est non-linéaire $f(a)$ pour calculer la décision S.



Types du neurone formel :

- Le neurone produit scalaire.

$$F(x, w) = \langle x, w \rangle = \sum_{i=1}^{n} w_i x_i$$

- Le neurone distance.

$$F(x, w) = \|x - w\|_2^2 .$$

Si les poids sont normalisés.

$$\|x - w\|^2 = \|x\|^2 + \|w\|^2 - 2(\langle x, w \rangle)$$

$< 0.$ $> 0.$ $> 0.$ $< 0.$

1er Modèle : ADALINE (Adaptive Linear Element).

On suppose des données en entrée : $x^1, x^2, \dots, x^N$ de $\mathbb{R}^n$ avec des sorties désirées : $d^1, d^2, \dots, d^N$.

⟶ Un modèle adaptatif, linéaire et bi-classes qui consiste à calculer une sortie y en fct de poids de connexion w :

$$\boxed{y = w_0 + \langle x, w \rangle .}$$

unité de

tolais.

Algo :

1) A l'instant $t=0$, initialiser les poids w de manière aléatoire.

2) Présenter une donnée x^k avec sa sortie désirée d^k.

3) Calculer l'écart

$$e = d^k - \langle x^k, w \rangle$$ scalaire

4) Calculer l'approximation.

$$\Delta = -2 e x^k$$ vecteur.

5) $w(t+1) = w(t) - \varepsilon \Delta$

ε : pas d'apprentissage

6) Répéter 2) à 5) jusqu'à stabilité.

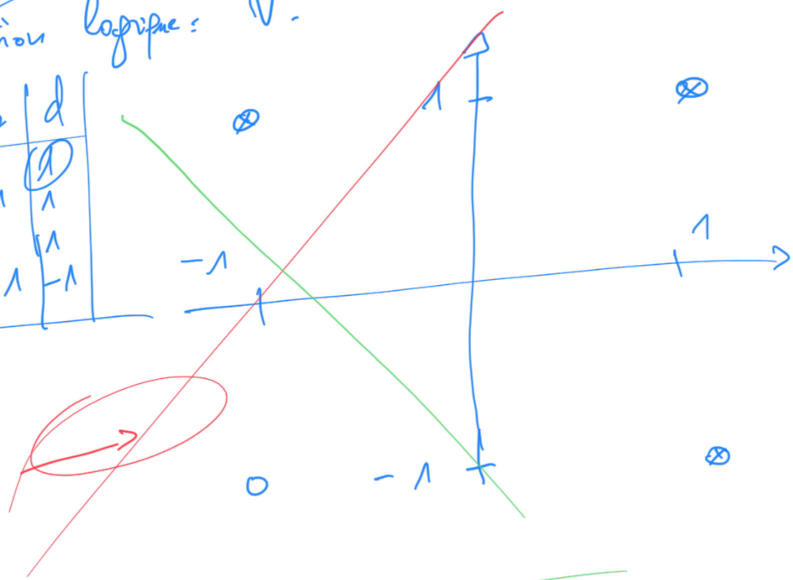
FIN

ex : la fonction logique : V.

⊗ : 1

O : −1

#	V_1	V_2	d
1	1	1	1
2	1	−1	1
3	−1	1	1
4	−1	−1	−1



TD

Ex 1 : ET logique.

$\varepsilon = 0.1$

#	V_0	V_1	V_2	d
1	1	1	1	1
2	1	1	−1	−1

$t=0$, $w = (0.3, 0.8, 0.4)$, w_0.

$$w_0 + w_1 V_1 + w_2 V_2 = 0.$$

$$\begin{array}{c|c|c|c|c} 3 & 1 & -1 & 1 & -1 \\ 4 & 1 & -1 & -1 & -1 \end{array}$$

$\#1 : (1, 1, 1) \; ; d = 1.$

$e = \underline{1} - (1,1,1)\begin{pmatrix} 0.3 \\ 0.8 \\ 0.4 \end{pmatrix} = -0.5$

$\Delta = -2(-0.5)(1, 1, 1)$

$= (1, 1, 1).$

$w(1) = (\underline{0.3}, 0.8, 0.4) - 0.1(1,1,1)$

$= (0.2, 0.7, 0.3).$

$\#2 : (1, 1, -1) \; ; \; d = -1$

e

Δ

$w(2) = w(1) - \varepsilon \Delta$

$$\boxed{w^* = (-0.5, 0.4286, 0.3571)}$$

Test :

$\#1 (1, 1, 1) \cdot \begin{pmatrix} -0.5 \\ 0.4286 \\ 0.3571 \end{pmatrix} = \cdot > 0 \Rightarrow d = \underline{1}.$

$\#2 : (1, 1, -1) \cdot w^* = \; < 0 \Rightarrow d = -1$

$\#3 : (1, -1, 1) \; w^* = \cdot \; < 0 \Rightarrow d = -1$

$\#4 : (1, -1, -1) \cdot w^* = \cdot \; < 0 \Rightarrow d = -1$

Exo3 :

Base d'apprentissage

$\varepsilon = 0.1$

$t = 0, \; w = (10, 3, -2).$

#	V_0	V_1	V_2	d
4	1	6	4	1
5	1	7	5	1
8	1	11	16	−1
9	1	14	11	−1
10	1	16	5	−1

$\#4: (1,6,4)\ ;\ d = 1.$

$e = 1 - (1,6,4)\begin{pmatrix}10\\3\\-2\end{pmatrix} = -19$

$\Delta = -2(-19)(1,6,4) = (38, 228, 152).$

$W(1) = (10,3,-2) - 0.1(38, 228, 152)$
$= (6.2, -19.8, -17.2)$

$$\boxed{W^* = (10, -0.8479, -0.5759)}$$

Test

#	V_0	V_1	V_2	d
1	1	1	2	1
2	1	2	16	1 ✗
3	1	4	9	1
6	1	8	15	−1
7	1	9	9	−1

$\#1: (1,1,2)\,W^* = .\ > 0 \Rightarrow d = 1$

$\#2: (1,2,16)\,W^* = .\ < 0 \Rightarrow d = -1$

$\#3: (1,4,9)\,W^* = .\ > 0 \Rightarrow d = 1$

$\#6: (1,8,15)\,W^* = .\ < 0 \Rightarrow d = -1$

$\#7: (1,9,9)\,W^* = .\ < 0 \Rightarrow d = -1$

Rappel

Réel \ Prédit	C_1	C_2	
C_1	2	1	3
C_2	0	2	2
	2	3	$4/5 = 80\%$.

Précision

Précision $C_1 = 2/3 = 67\%$

$C_2 = 2/2 = 100\%$

Rappel $C_1 = 2/2 = 100\%$

$C_2 = 2/3 = 67\%$.