



**RÉPUBLIQUE
FRANÇAISE**

*Liberté
Égalité
Fraternité*



**METEO
FRANCE**

À VOS CÔTÉS, DANS UN
CLIMAT QUI CHANGE

Machine Learning – Recap' n°1

Pierre Lepetit
ENM, le 18/10/2024

Qu'est-ce que le Machine Learning ?

C'est une classe de problèmes où l'on dispose :

- D'entrées x (ex : images, bandes son, texte, variable gaussienne)
- De cibles y (ex : classes, classes par pixel, images)

Et où l'un cherche un « modèle » (i.e. une fonction paramétrée) $z = f(x ; w)$ qui « fasse le lien » entre les x et les y .

complément

Cette fonction doit être construite de manière à pouvoir :

- « Exprimer » le lien – densité/approximation
- Se prêter à un « entraînement » – optimisation
- Généraliser en dehors du jeu d'entraînement – régularisation

Qu'est-ce que le Machine Learning ?

C'est une classe de problèmes où l'on dispose :

- D'entrées x (ex : images, bandes son, texte, variable gaussienne)
- De cibles y (ex : classes, classes par pixel, images)

Et où l'on cherche un « modèle » (i.e. une fonction paramétrée) $z = f(x ; w)$ qui « fasse le lien » entre les x et les y .

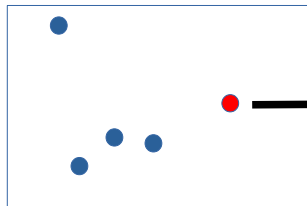
exemples

pb d'IA
(en 2010)



→ classe « chat »

pb de ML
(en 1950)



→ classe 0

Qu'est-ce que le Machine Learning ?

C'est une classe de problèmes où l'on dispose :

- D'entrées x (ex : images, bandes son, texte, variable gaussienne)
- De cibles y (ex : classes, classes par pixel, images)

Et où l'on cherche un « modèle » (i.e. une fonction paramétrée) $z = f(x ; w)$ qui « fasse le lien » entre les x et les y .

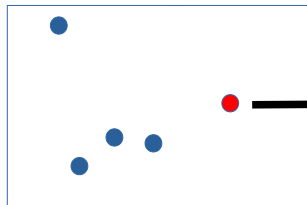
exemples

pb d'IA
(en 2010)



→ classe « chat »

pb de ML
(en 1950)



→ classe 0

pb d'IA
Aujourd'hui

Qu'est-ce que le Machine Learning ?

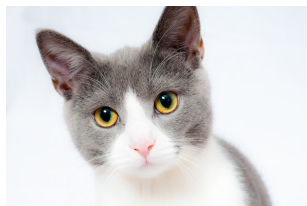
C'est une classe de problèmes où l'on dispose :

- D'entrées x (ex : images, bandes son, texte, variable gaussienne)
- De cibles y (ex : classes, classes par pixel, images)

Et où l'un cherche un « modèle » (i.e. une fonction paramétrée) $z = f(x ; w)$ qui « fasse le lien » entre les x et les y .

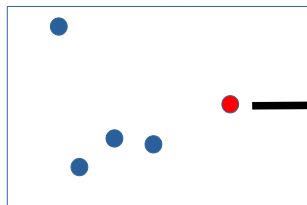
exemples

pb d'IA
(en 2010)



→ classe « chat »

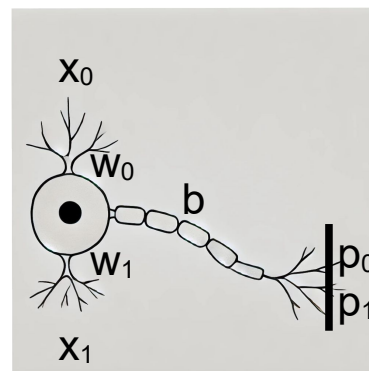
pb de ML
(en 1950)



→ classe 0

TP1

PERCEPTRON à un neurone



Qu'est-ce que le Machine Learning ?

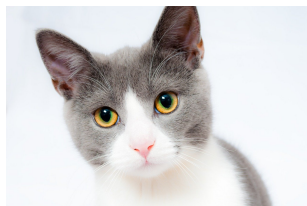
C'est une classe de problèmes où l'on dispose :

- D'entrées x (ex : images, bandes son, texte, variable gaussienne)
- De cibles y (ex : classes, classes par pixel, images)

Et où l'un cherche un « modèle » (i.e. une fonction paramétrée) $z = f(x ; w)$ qui « fasse le lien » entre les x et les y .

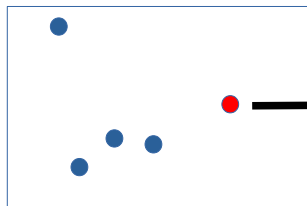
exemples

pb d'IA
(en 2010)



→ classe « chat »

pb de ML
(en 1950)



→ classe 0

TP1

$$f(x ; w, b) = \text{Sigmoid}(\sum w_i x_i + b)$$

```
class P1(nn.Module):  
    def __init__(self):  
        super(P1, self).__init__()  
        self.fc = nn.Linear(2, 1)  
  
    def forward(self, x):  
        x = self.fc(x)  
        x = x.sigmoid()  
        return x
```

Un exemple d'entraînement

Entraînement d'un perceptron par descente de gradient stochastique

- Etape 1 : Initialisation des paramètres (voir TP 5)
- Etape 2 : boucle d'apprentissage (incomplète):

Répéter jusqu'à
« convergence »

Répéter jusqu'à
épuisement du jeu

- Echantillon aléatoire sans remise d'une dizaine d'éléments x^k et c^k ; $k = 0..10$ (**batch**)
- Calcul des sorties : $(p^k[0], p^k[1]) = f(x^k; w)$
- Evaluation de la fonction de coût : $\text{Loss} = 1/10 \times \sum -\ln(p^k[c^k])$
- Calcul des gradients : $\partial_w \text{Loss}$
- Mise à jour des poids (SGD) : $w := w - \text{lr} \times \partial_w \text{Loss}$ (**lr** : **learning rate**)

$$N_{\text{batch}} = |\text{Jeu}| / |\text{Batch}|$$

N**époques**

Un exemple d'entraînement

Entraînement d'un perceptron par descente de gradient stochastique

- Etape 1 : Initialisation des paramètres (voir TP 5)
- Etape 2 : boucle d'apprentissage (incomplète):

Répéter jusqu'à
« convergence »

Répéter jusqu'à
épuisement du jeu

- Echantillon aléatoire sans remise d'une dizaine d'éléments x^k et c^k ; $k = 0..10$ (**batch**)
- Calcul des sorties : $(p^k[0], p^k[1]) = f(x^k; w)$
- Evaluation de la fonction de coût : $\text{Loss} = 1/10 \times \sum -\ln(p^k[c^k])$
- Calcul des gradients : $\partial_w \text{Loss}$
- Mise à jour des poids (SGD) : $w := w - \text{lr} \times \partial_w \text{Loss}$ (**lr** : **learning rate**)

$$N_{\text{batch}} = |\text{Jeu}| / |\text{Batch}|$$

N**époques**

Un exemple d'entraînement

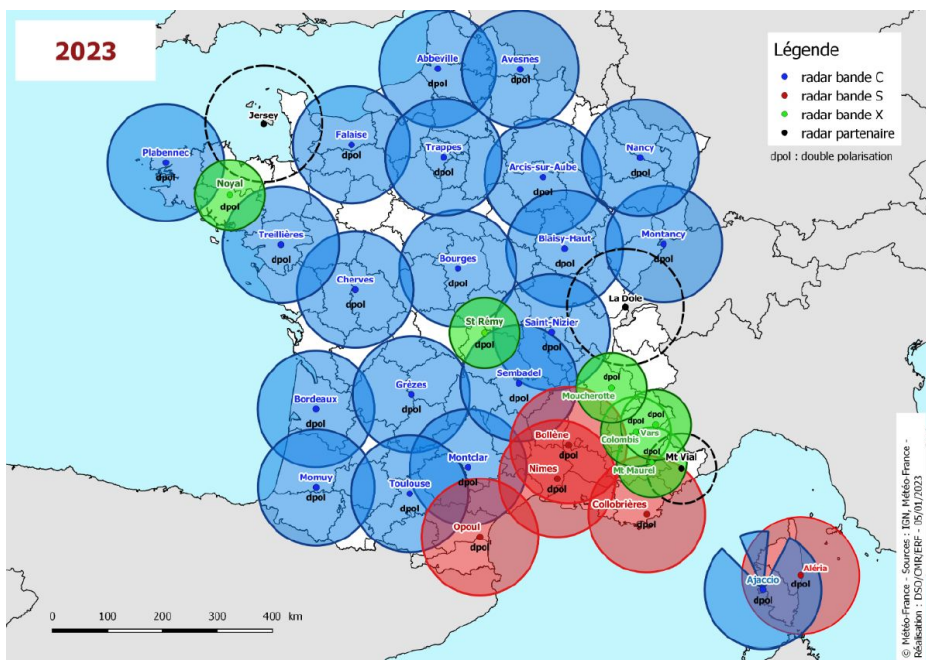
Entraînement d'un perceptron par descente de gradient stochastique

- Etape 1 : Initialisation des paramètres (voir TP 5)
- Etape 2 : boucle d'apprentissage (incomplète):

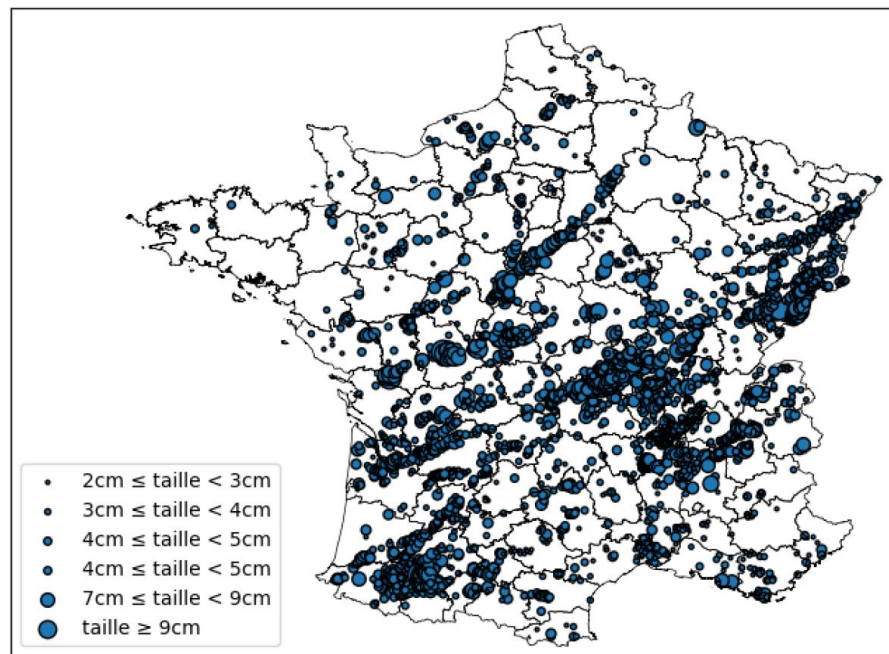
```
loader = DataLoader(dataset, bs=10, shuffle=True)
for epoch in range(Nepochs):
    for x, targets in loader:
        optimizer.zero_grad() # zeroing gradients
        output = model(x) # compute outputs
        l = loss_fn(output, targets) # compute loss
        l.backward() # calculation of gradients
        optimizer.step() # weight update
```

Stage « détection de la grêle »

Entrées : données radar (+données modèles)

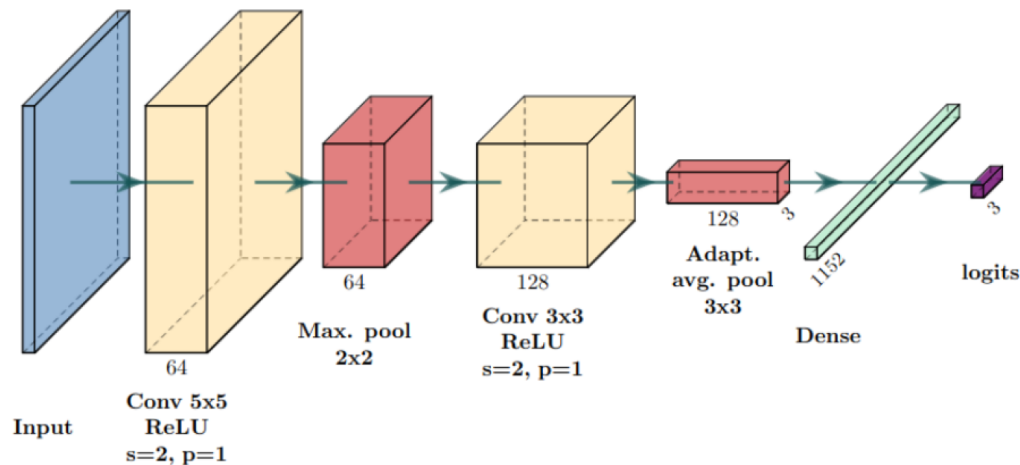


Cibles : données grêle (taille)



Stage « détection de la grêle »

Méthode à appliquer : (Vincent Forcadell)



Condition pour postuler

- 1) Avoir un background « météo » ou ...
- 2) ... aimer la « spatialisation » (TP3)