



**RÉPUBLIQUE
FRANÇAISE**

*Liberté
Égalité
Fraternité*



**METEO
FRANCE**

À VOS CÔTÉS, DANS UN
CLIMAT QUI CHANGE

Machine Learning – Introduction

Pierre Lepetit
ENM, le 18/10/2024

Qu'est-ce que le Machine Learning ?

Intelligence artificielle :

- 1943 – Neurone formel
- 1956 – Conférence de Dartmouth.
- 1957 – Perceptron
- 1966 – Eliza (IA symbolique)
- 1986 – Back-prop → Connexionisme
- 1998 – LeNet – 5 / MNIST
→ succès en 'pattern recognition'

Statistiques :

- 1918 – Fisher GLM
 $Y = X \beta + \varepsilon$ avec $\varepsilon \sim N(0, \sigma)$
- 1922 – Fisher Max. Vraisemblance
- 1956 – Paradoxe de Stein
- 1957 – K-moyennes
- 1970s – 1980s : régularisation,
sélection de modèle,
validation croisée
- 1995 – Théorie de l'apprentissage
statistique

Qu'est-ce que le Machine Learning ?

Les années 1990 - 2000 : domination des approches data driven

- 1997 – Définition formelle du Machine Learning (Tom Mitchell) :

« A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E »

- Diversité des « programmes » : ANN, LSTM, SVM;kkk, Random Forests, etc
- Diversification des domaines/tâches : pattern recognition, bioinformatique, NLP, recommandations, etc
- Diversification des contextes :
Apprentissage supervisé / Faiblement-supervisé / Semi-supervisé / Unsupervised
Dataset déséquilibré/ Apprentissage par transfert, par Curriculum, etc

Qu'est-ce que le Machine Learning ?

Les années 2010 - 2020 : Deep Learning era

- 2012 – AlexNet (Dataset $\sim 10^6$ samples, GPU) → I
- Définition formelle Goodfellow, Bengio & Courville 2016

Sous classe du machine learning. Artificial neural network + multiple processing layers + hierarchical representations with multiple levels of abstraction.

- Abandons des « descripteurs » et post-traitements → entraînements « end-to-end »
- Diversifications des Réseaux : ResNet (2014), GCN (2016), Transformers (2017), etc
- Radiation méthodologique :
 - Optimisation + DL ? Régularisation + DL ?
 - Transfer learning + DL ? Clustering + DL ? Semi/non supervisé + DL ?
- Nouveautés : GAN, Diffusion, Autoregressive LM

Qu'est-ce que le Machine Learning ?

Les années 2020 - 2025 :

- Avancées significatives :
 - Méthodes propres aux LLM (Pretraining, SFT, Alignement, Prompt engineering, CoT)
 - Traitement de longues séries temporelles (S4, Mamba, RWKV, etc)
 - IA multimodale (Clip, Perceiver), IA agentique (alphaevolve)
- Impact sur l'industrie/la société
 - Datasets 10 TB – 10 PB, réseaux très lourds
 - Mort de paperwithcodes, émergence de HuggingFace
 - Risques environnementaux/sociaux croissants
 - Percée dans le quotidien (plantnet, Dall-e, chatGPT, vo3, suno, etc)

Quel est le cadre de ce cours ?

1. Apprentissage machine **supervisé** :

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 .

Quel est le cadre de ce cours ?

1. Apprentissage machine **supervisé**

2. « Modèles » du **Deep Learning** : réseaux de neurones profonds → TPs 1 - 4

- Perceptrons multicouches
- Réseaux de convolutions
- Transformers
- GNN
- ...

- Expressivité ?
- Entraînement ?
- Performances en généralisation ?

Quel est le cadre de ce cours ?

1. Apprentissage machine **supervisé**
2. « Modèles » du **Deep Learning** : réseaux de neurones profonds → TPs 1 - 4
3. Tâches tirées de la « **vision par ordinateur** »
 - Classification
 - Segmentation
 - Débruitage
 - Génération d'images
 - Learning to rank
 - Clustering ou Traitement de séries temporelles
 - ...

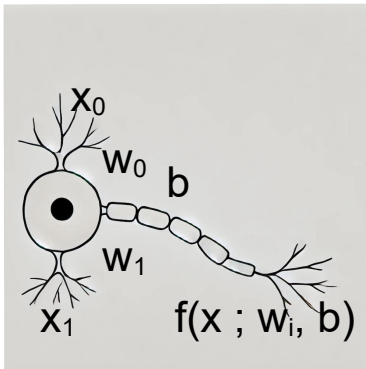
Aspects pratiques

- Cours 20 % - TPs 80 % :
 - Cours en début de séance
 - 1 TP = 1h30 (sous colab, en anglais)
 - Matériel sur GitHub
- Exercices disponibles :
 - Fiches d'exercice
 - Annales depuis 2022
- Evaluation :
 - Tests de 15 minutes en début de cours
 - TP final (3h) sous pytorch

Accès libre aux **LLM**
sauf pour les tests
en début de cours.
Mais ...

TP n°1 : perceptron + classification + SGD

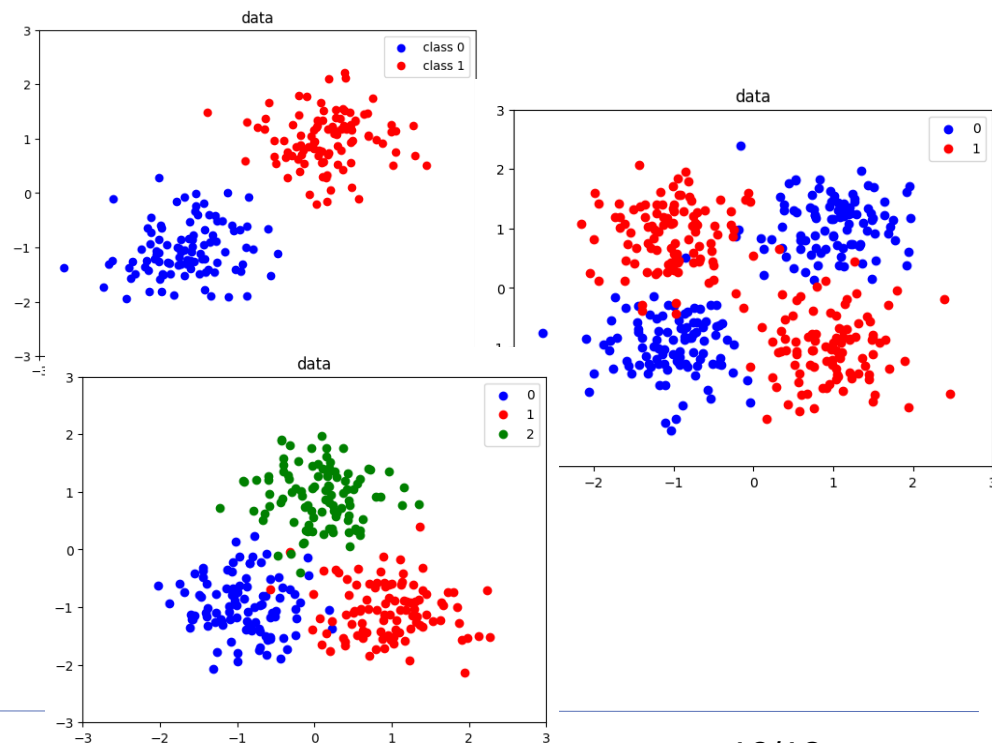
Neurone formel :



Perceptron sous pytorch :

```
class Perceptron(nn.Module):  
    def __init__(self):  
        super(P1, self).__init__()  
        self.fc = nn.Linear(2, 1, bias=True)  
  
    def forward(self, x):  
        x = self.fc(x)  
        x = x.sigmoid()  
        return torch.cat((x, 1 - x), dim=1)
```

Tâches de classification :



TP n°1 : perceptron + classification + SGD

Entraînement d'un perceptron par **descente de gradient stochastique (SGD)**

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

Répéter jusqu'à
« convergence »

Répéter jusqu'à
épuisement du jeu

- Echantillon aléatoire d'une dizaine d'éléments x^k et c^k ; $k = 0..9$ (batch)
- Calcul d'une sortie (vecteur de probabilités) : $p^k = f(x^k; w_i, b)$
- Evaluation de la fonction de coût : $\text{Loss} = 1/10 \times \sum -\ln(p^k[c^k])$ (negative log-likelihood)
- 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

TP n°1 : perceptron + classification + SGD

Entraînement d'un perceptron par **descente de gradient stochastique (SGD)**

- Etape 1 : Initialisation des paramètres
- 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
```