

Aide-mémoire : Apprentissage Paramétrisé

Ce document regroupe les concepts clés et les étapes principales de l'apprentissage paramétrisé à partir du notebook d'exemple.

Concepts clés

1. Paramétrisation :

- Définir et ajuster les paramètres d'un modèle (poids et biais).

2. Fonction de perte (Loss Function) :

- Mesure la différence entre les prédictions et les étiquettes réelles.

3. Gradient et descente de gradient :

- Le *gradient* guide la direction d'optimisation.
- La *descente de gradient* ajuste progressivement les paramètres pour minimiser la perte.

4. Sigmoid (fonction d'activation) :

- Formule : $\sigma(x) = 1 / (1 + \exp(-x))$
 - Aplatit les valeurs d'entrée dans un intervalle entre 0 et 1.
-

Etapes principales du modèle

1. Initialisation

- Les poids (w) sont initialisés avec des petites valeurs aléatoires proches de 0.
- Les biais (b) sont initialisés à 0.

2. Propagation avant (Forward Pass)

- Calcul des sorties du modèle via une fonction linéaire puis activation (exemple : *sigmoïde*).

$$\text{Formule : } f(x, w, b) = \sigma(w \cdot x + b)$$

où :

- σ est la fonction sigmoïde,
- $W \cdot x$ désigne le produit scalaire entre les poids et les entrées.

Exemple de produit scalaire :

Soit deux vecteurs :

- $x = [2, 3]$
- $w = [0.5, -1]$

Le produit scalaire $w \cdot x$ est calculé comme : $(2 * 0.5) + (3 * -1) = 1 - 3 = -2$.

3. Rétropropagation (Backward Pass)

- Adapte les poids et biais selon :
 - Le gradient (dérivée partielle par rapport à la perte).
 - Un taux d'apprentissage.

Pseudo-code :

```
while True:
    Wgradient = evaluate_gradient(loss, data, W)
    W += -alpha * Wgradient
```

4. Boucle d'entraînement

- Répétition des étapes de propagation avant et arrière sur un certain nombre d'epochs.
 - Durée contrôlée par le taux d'apprentissage et la convergence de la perte.
-

Evaluation du modèle

1. Calcul des scores :

- Prédictions générées via `forward pass` avec les données de test.

2. Rapport de classification :

- Rapport généré via `classification_report` (de `scikit-learn`).

Exemple de code :

```
from sklearn.metrics import classification_report
print(classification_report(y_test, np.around(preds)))
```

Ressources supplémentaires

- Article recommandé sur la descente de gradient (<https://www.charlesbordet.com/fr/gradient-descent/#comment-ca-marche>).