

Mini-Projet d'Optimisation ML

Modélisation, SGD et Méthodes Proximales

M2 SSD - FST

2025/2026

1 Introduction et Contexte

Ce projet vise à mettre en pratique l'ensemble des concepts du cours : de la formalisation mathématique (Chapitre 1) aux méthodes proximales pour la régularisation (Chapitre 4), en passant par les algorithmes de gradient stochastique (Chapitre 3). Nous étudierons un problème de classification binaire défini par un jeu de données $\{(x_i, y_i)\}_{i=1}^n$ avec $x_i \in \mathbb{R}^d$ et $y_i \in \{-1, 1\}$.

2 Phase 1 : Fondements et Gradient Déterministe (Chap. 1 & 2)

On considère la fonction de perte logistique régularisée (Ridge) :

$$\min_{w \in \mathbb{R}^d} F(w) = \frac{1}{n} \sum_{i=1}^n \log(1 + e^{-y_i x_i^T w}) + \frac{\lambda}{2} \|w\|^2 \quad (1)$$

1. **Analyse (Chap. 1)** : Justifier que F est de classe C^2 , convexe, et λ -fortement convexe[cite : 612, 621, 383].
2. **Gradient (Chap. 2)** : Calculer $\nabla F(w)$ et montrer que le gradient est L -Lipschitzien. Exprimer L en fonction de la matrice des données X [cite : 351].
3. **Implémentation** : Coder la *Descente de Gradient* à pas fixe et comparer avec la *Méthode du Gradient Conjugué*[cite : 75, 87].

3 Phase 2 : Passage à l'Échelle Stochastique (Chap. 3)

Lorsque n devient très grand, le calcul du gradient complet est proscrit[cite : 200, 320].

1. **SGD** : Implémenter l'algorithme du *Gradient Stochastique* avec une règle de décroissance du pas α_k [cite : 204, 243].
2. **Optimiseurs Modernes** : Implémenter et comparer les performances de **RMSProp** et **Adam** sur les premières époques d'entraînement[cite : 310, 312].
3. **Analyse** : Discuter de l'impact du momentum sur la stabilité des itérations[cite : 256].

4 Phase 3 : Non-Lissé, Parcimonie et Proximal (Chap. 4)

On remplace la régularisation L_2 par une norme L_1 pour favoriser la sélection de variables (parcimonie)[cite : 385, 386] :

$$\min_{w \in \mathbb{R}^d} \Phi(w) = f(w) + \lambda \|w\|_1 \quad \text{où } f \text{ est la perte logistique.} \quad (2)$$

1. **Théorie** : Expliquer pourquoi ce problème est non lisse et définir l'opérateur proximal associé à la norme L_1 (seuil doux)[cite : 47, 51].
2. **Algorithmes** :
 - Implémenter l'algorithme **ISTA** (Iterative Soft-Thresholding Algorithm)[cite : 51].
 - Implémenter sa version accélérée **FISTA** et observer le gain de convergence[cite : 151].
3. **Interprétation** : Analyser le nombre de coefficients nuls dans w^* en fonction de λ .

5 Livrables

- Un rapport PDF contenant les preuves mathématiques et les courbes de convergence (Perte vs Temps).
- Un Notebook Python (Scikit-learn autorisé uniquement pour charger les données).