Optimisation pour le Machine Learning

PROJET D'EXAMEN FINAL

Filière : Master SSD **Date :** Janvier 2026

Note : *Ce projet est à réaliser individuellement. La rigueur mathématique dans la rédaction en LaTeX, la qualité des implémentations Python sur des datasets de grande taille et l'analyse critique des résultats seront les principaux critères d'évaluation.*

Objectifs :

Ce projet vise à évaluer la maîtrise des quatre piliers du cours : la modélisation (Chap. 1), les méthodes de gradient déterministes (Chap. 2), le passage à l'échelle via la stochasticité (Chap. 3) et l'optimisation non lisse par algorithmes proximaux (Chap. 4).

Exercice 1: Modélisation et Étude Théorique (6 points)

On considère un problème de régression sur le dataset *YearPredictionMSD* ($n \approx 515\,000$ exemples, $d = 90$) visant à prédire l'année de sortie d'une chanson. Soit X la matrice des données et y le vecteur cible.

a) **Modélisation :** Formuler le problème de minimisation sous la forme d'une somme finie $f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$. Justifiez l'ajout d'une pénalité $\frac{\mu}{2}\|w\|^2$ pour garantir l'unicité du minimum global en citant le **Théorème 1.2.9** du cours.

b) **Étude du Gradient :** Calculer analytiquement le gradient $\nabla f(w)$ et la matrice Hessienne $\nabla^2 f(w)$.

c) **Lipschitz et SVD :** En utilisant la Décomposition en Valeurs Singulières (SVD) de X vue au Chapitre 2, déterminer l'expression de la constante de Lipschitz L du gradient. Quel est l'impact de cette constante sur la stabilité de la descente de gradient ?

Exercice 2: Stochasticité et Passage à l'Échelle (7 points)

Compte tenu de la taille du dataset ($n > 500\,000$), le calcul du gradient complet est proscrit.

a) **Implémentation SGD :** Coder l'algorithme de Descente de Gradient Stochastique *from scratch*. Prouvez mathématiquement que $\mathbb{E}[\nabla f_i(w)] = \nabla f(w)$ (estimateur sans biais).

b) **Analyse Comparative :** Comparer la convergence (MSE vs Temps CPU) entre le Gradient de Batch (sur un sous-échantillon) et le SGD. Illustrer le phénomène de "bruit de gradient" via des graphiques.

c) **Mini-batch et Adam :** Implémenter une variante Mini-batch et comparez ses performances avec l'algorithme **Adam**. Discutez de l'importance de la standardisation des données pour la condition de la Hessienne.

Exercice 3: Parcimonie et Algorithmes Proximaux (7 points)

On s'intéresse à la classification de documents sur le dataset *Reuters RCV1* où la dimension est très élevée et les variables redondantes.

a) **Analyse Géométrique :** Expliquer, schémas à l'appui, pourquoi la régularisation L_1 (Lasso) est préférable à la régularisation L_2 pour obtenir un modèle interprétable en haute dimension.

b) **ISTA :** Définir et implémenter l'opérateur proximal de la norme L_1 (*Soft-thresholding*). Utilisez-le pour coder l'algorithme ISTA.

c) **Accélération :** Implémenter l'algorithme accéléré **FISTA**. Comparez le taux de convergence théorique $O(1/k)$ d'ISTA par rapport au $O(1/k^2)$ de FISTA.

d) **Sélection de variables :** Identifier les mots (caractéristiques) les plus significatifs pour la classification en faisant varier le paramètre de régularisation λ.

Livrables attendus

— Un rapport rédigé en LaTeX présentant de manière rigoureuse les justifications théoriques (comme indiqué en cours, l'aspect théorique est privilégié par rapport à l'implémentation pratique).

— Un notebook (Python) contenant les implémentations et les visualisations demandées.

— Une analyse comparative des temps de calcul et des taux de convergence pour chaque méthode.