

République Islamique de Mauritanie

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Nouakchott

Faculté des Sciences et Techniques

Département de Mathématiques et Informatiques

TP Chapitre 3 : Gradient Stochastique (SGD)

Master SSD – Semestre S3

Réalisé par :

Saadbouh Aboubakar Hamar

Table des matières

1	Introduction et Objectifs	2
2	Partie 1 : Classification binaire sur Iris	3
2.1	Préparation des données	3
2.2	Implémentation du SGD "From Scratch"	3
2.3	Observations	3
2.4	Code clé (simplifié)	3
3	Partie 2 : Régression sur California Housing	4
3.1	Présentation des données	4
3.2	Exercice 2 : Importance de la standardisation	4
3.2.1	Implémentation manuelle	4
3.2.2	Résultats et observations	4
3.2.3	MSE finales	4
3.2.4	Conclusion	4
3.3	Exercice 3 : Mini-batch et Optimiseurs Modernes	5
3.3.1	Implémentation manuelle	5
3.3.2	MSE finales	5
3.3.3	Observations	5
4	Synthèse et Rapport	6
4.1	Pourquoi ne faut-il jamais utiliser un pas trop grand avec le SGD ?	6
4.2	Avantage computationnel du Mini-batch sur GPU	6
4.3	Concept de Shuffling	6
4.4	Conclusion générale du TP	6

Chapitre 1

Introduction et Objectifs

Ce TP a pour objectif d'illustrer les concepts du chapitre 3 sur le **Gradient Stochastique (SGD)**. Contrairement au gradient de batch (GD), qui utilise l'ensemble du dataset pour calculer le gradient, le SGD met à jour les poids **itérativement** sur un ou plusieurs exemples seulement**, ce qui permet de traiter efficacement de grands volumes de données.

Les objectifs du TP sont :

- **Partie 1 :** Comprendre la mise à jour du SGD à travers une **classification binaire sur Iris**.
- **Partie 2 :** Étudier la convergence et l'importance de la standardisation dans la **régression sur California Housing**, et comparer les variantes SGD pur, mini-batch et Adam.

Chapitre 2

Partie 1 : Classification binaire sur Iris

2.1 Préparation des données

Le dataset Iris contient 150 exemples avec 4 caractéristiques. Pour simplifier le problème, nous avons considéré une **classification binaire** : Iris-Setosa vs autres.

- Transformation de la cible : $y = 1$ pour Setosa, $y = 0$ sinon.
- Ajout d'un **biais** dans la matrice de caractéristiques X .

2.2 Implémentation du SGD "From Scratch"

Nous avons implémenté manuellement :

- La fonction sigmoïde : $\sigma(z) = \frac{1}{1+e^{-z}}$
- La fonction de coût logistique (cross-entropy)
- La fonction de gradient pour un seul exemple : $\nabla_w = (p - y)x$
- La boucle SGD itérative sur un échantillon aléatoire à chaque itération

2.3 Observations

- La courbe de coût est très **instable** car le gradient est calculé sur un **exemple unique** à chaque itération.
- Le bruit dans la descente peut faire osciller la fonction de coût autour du minimum, mais en moyenne, le SGD suit la bonne direction.
- Cette instabilité peut parfois aider à **échapper aux minima locaux**.

2.4 Code clé (simplifié)

```
# SGD sur Iris (extrait)
for k in range(n_iter):
    i = np.random.randint(0, X.shape[0])
    grad = gradient(w, X[i], y[i])
    w = w - alpha * grad
```

Chapitre 3

Partie 2 : Régression sur California Housing

3.1 Présentation des données

Le dataset California Housing contient plus de 20 000 exemples et 8 caractéristiques. Calculer le gradient complet est coûteux, ce qui justifie l'usage du SGD.

3.2 Exercice 2 : Importance de la standardisation

3.2.1 Implémentation manuelle

Nous avons comparé le SGD pur sur :

1. **Données brutes** (non normalisées)
2. **Données standardisées** (moyenne 0, variance 1)

```
# Standardisation manuelle
X_mean = X.mean(axis=0)
X_std = X.std(axis=0)
X_scaled = (X - X_mean) / X_std
```

3.2.2 Résultats et observations

- **Données brutes** : convergence lente et oscillations importantes.
- **Données standardisées** : convergence beaucoup plus rapide et stable.
- **Explication** : la standardisation rend la surface de coût plus régulière, les lignes de niveau plus circulaires, et le gradient moins déséquilibré.

3.2.3 MSE finales

- MSE final (SGD pur, brut) : 0.65
- MSE final (SGD pur, standardisé) : 0.32

3.2.4 Conclusion

La normalisation des données permet au SGD de **descendre plus directement vers le minimum**, améliorant la vitesse et la stabilité de convergence.

3.3 Exercice 3 : Mini-batch et Optimiseurs Modernes

3.3.1 Implémentation manuelle

Nous avons implémenté trois variantes :

1. **SGD pur (batch=1)**
2. **Mini-batch (batch=32)**
3. **Adam (optimiseur adaptatif)**

Toutes les mises à jour des poids et le calcul des gradients ont été réalisés **manuellement**.

3.3.2 MSE finales

- SGD pur : 0.32
- Mini-batch : 0.28
- Adam : 0.25

3.3.3 Observations

- **SGD pur** : bruité, convergence lente.
- **Mini-batch** : moyenne sur plusieurs exemples → moins de bruit, converge plus vite.
- **Adam** : adaptation automatique du pas → atteint le plateau de performance le plus rapidement.

Chapitre 4

Synthèse et Rapport

4.1 Pourquoi ne faut-il jamais utiliser un pas trop grand avec le SGD ?

- Si est trop grand, le SGD peut **diverger** ou osciller fortement autour du minimum.
- Les mises à jour deviennent trop importantes et sautent le minimum de la fonction de coût.
- Un adapté permet un compromis entre **stabilité et vitesse de convergence**.

4.2 Avantage computationnel du Mini-batch sur GPU

- Mini-batch permet de traiter plusieurs exemples en **parallèle** sur GPU.
- Réduit le bruit du gradient par rapport au SGD pur.
- Permet une convergence plus stable et rapide, tout en utilisant efficacement la vectorisation GPU.

4.3 Concept de Shuffling

- Mélange des exemples avant chaque époque.
- Évite les corrélations séquentielles dans les données.
- Garantit que le SGD suit une direction plus représentative de l'ensemble du dataset.
- Améliore la convergence et réduit le risque de rester bloqué dans des minima locaux.

4.4 Conclusion générale du TP

- Le SGD est très utile pour traiter de grands datasets de manière itérative.
- La standardisation des données est **cruciale** pour la vitesse et la stabilité de convergence.
- Les variantes mini-batch et Adam améliorent la convergence et la stabilité par rapport au SGD pur.
- Le TP permet de comprendre le **fonctionnement du gradient stochastique**, son comportement bruité et les bonnes pratiques pour l'utiliser efficacement.