

# III-La descente de gradient stochastique

Module : Outils mathématiques pour l'ingénieur



1<sup>ère</sup> année cycle d'ingénieur

Fadwa EL KIHAL

## La descente de gradient stochastique

- La Descente de Gradient stochastique est une approximation de la Descente de Gradient (**GD**).
- L'algorithme de la descente de gradient peut être infaisable lorsque la taille des données d'entraînement est énorme.
- Contexte big data : la taille des bases à traiter devient un enjeu essentiel pour les algorithmes de machine learning.
- Il faut développer des stratégies d'apprentissage qui permettent d'appréhender les grandes bases (en particulier en nombre de descripteurs), en réduisant notamment la taille des structures de données à maintenir en mémoire
- Tout en obtenant des résultats de qualité satisfaisante (comparables à ceux produits par les algorithmes standards ex. régression logistique)

- Descente de gradient (stochastique) n'est pas un concept nouveau, mais elle connaît un très grand intérêt aujourd'hui, en particulier pour l'entraînement des réseaux de neurones profonds (deep learning). Elle permet aussi de revisiter des approches statistiques existantes.
- Dans la descente de gradient (**Batch Gradient Descent**), nous avons examiné tous les exemples pour chaque étape de la descente de gradient.

## Mais que faire si notre ensemble de données est très vaste?

- Les modèles d'apprentissage profond ont besoin de données. Plus les données sont nombreuses, plus un modèle a plus de chances d'être bon.
- Supposons que notre dataset comporte 5 millions d'exemples, alors pour faire un pas, le modèle devra calculer les gradients de tous les 5 millions exemples. Cela ne semble pas être une méthode efficace.
- Pour résoudre ce problème, nous disposons de la méthode de la descente de gradient stochastique.

- La descente de gradient s'effectue de manière globale (**batch gradient**), la descente de gradient stochastique s'effectue par des lots (**mini batch gradient**), soit de façon unitaire (**Stochastic gradient descent « Online »**).
- La première (comme nous avons déjà vu) consiste à traiter la totalité des données d'un seul trait, et de faire ensuite le calcul du gradient ainsi que la correction des coefficients.
- Alors que la seconde consiste à traiter, les données par petit groupe d'une taille définit par l'utilisateur.
- La dernière quant à elle, traite une donnée à la fois.

## Mini batch gradient descent

Pour motiver l'utilisation d'algorithmes d'optimisation stochastique, il convient de noter que lors de la formation de modèles d'apprentissage profond, nous considérons souvent la fonction objectif comme la somme d'un nombre fini de fonctions :

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n J_i(\theta),$$

où  $J_i(\theta)$  est une fonction de perte basée sur l'instance de données d'apprentissage indexée par  $i$ . Il est important de souligner que le coût de calcul par itération dans la descente de gradient s'échelonne linéairement avec la taille  $n$  de l'ensemble de données d'apprentissage.

Dans ce contexte, la descente de gradient stochastique offre une solution plus légère. À chaque itération, plutôt que de calculer le gradient  $\nabla J(\theta)$ , la descente stochastique du gradient échantillonne  $i$  de façon aléatoire et uniforme et calcule  $\nabla J_i(\theta)$  à la place.

L'idée est que la descente stochastique de gradient utilise  $\nabla J_i(\theta)$  comme un estimateur non biaisé puisque

$$\mathbb{E}_i \nabla J_i(\theta) = \frac{1}{n} \sum_{i=1}^n \nabla J_i(\theta) = \nabla J(\theta).$$

Dans un cas généralisé, à chaque itération, un mini lot  $\mathcal{J}$  composé d'indices pour les instances de données d'apprentissage peut être échantillonné à l'uniforme avec remplacement. De même, on peut utiliser

$$\nabla J_{\mathcal{J}}(\theta) = \frac{1}{|\mathcal{J}|} \sum_{i \in \mathcal{J}} \nabla J_i(\theta)$$

pour mettre à jour  $\theta$  en tant que

$$\theta = \theta - \eta \nabla J_{\mathcal{J}}(\theta),$$

où  $|J|$  indique la cardinalité du mini lot et le scalaire positif  $\eta$  est le taux d'apprentissage. De même, le gradient stochastique du mini lot  $\nabla J_J(\theta)$  est un estimateur non biaisé pour le gradient  $\nabla J(\theta)$  :

$$\mathbb{E}_i \nabla J_J(\theta) = \nabla J(\theta)$$

Cet algorithme stochastique généralisé est également appelé descente de gradient stochastique **mini-batch**.

Le coût de calcul par itération est  $\mathcal{O}(|J|)$ . Ainsi, lorsque la taille du mini lot est petite, le coût de calcul à chaque itération est faible.

Il existe d'autres raisons pratiques qui peuvent rendre la descente par mini batch plus attrayante que la descente de gradient. Si l'ensemble de données d'apprentissage comporte de nombreuses instances de données redondantes, les gradients stochastique peuvent être si proches du véritable gradient  $\nabla J(\theta)$  qu'un petit nombre d'itérations permettra de trouver des solutions utiles au problème d'optimisation.

En fait, lorsque l'ensemble de données d'apprentissage est suffisamment important, la descente par mini batch ne nécessite qu'un petit nombre d'itérations pour trouver des solutions utiles, de sorte que le coût total de calcul est inférieur à celui de la descente de gradient, même pour une seule itération. En outre, la descente mini batch peut être considérée comme offrant un effet de régularisation, en particulier lorsque la taille du mini lot est petite en raison du caractère aléatoire et du bruit de l'échantillonnage du mini lot.

## Online stochastique graient descent

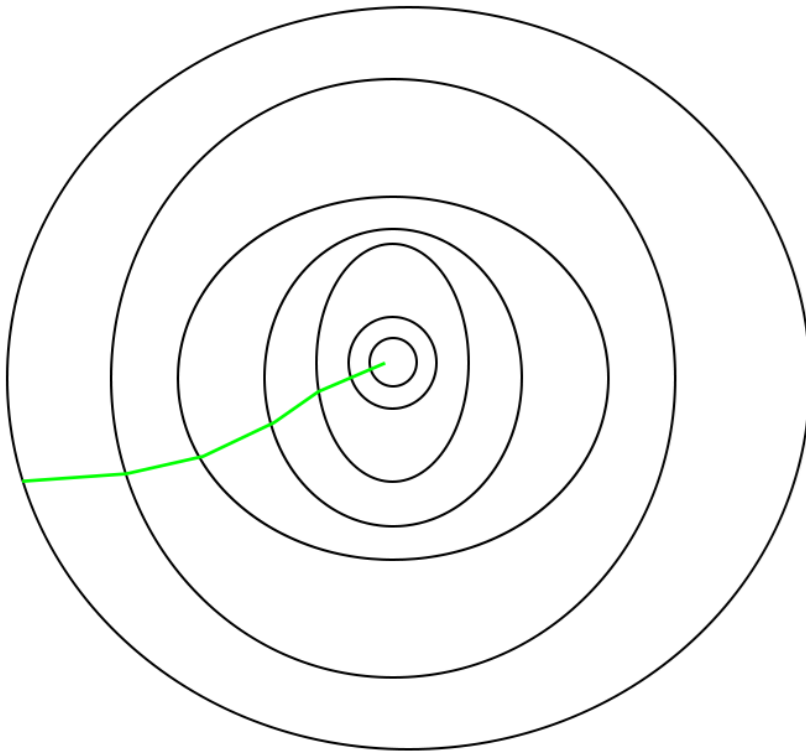
Dans le cas de la descente de gradient stochastique (**SGD : Online descent**), nous considérons un seul exemple à la fois pour faire un seul pas. Pour la **SGD**, nous effectuons les étapes suivantes en une seule fois :

1. Prendre un exemple
2. Calculer son gradient
3. Utilisez le gradient que nous avons calculé pour mettre à jour les paramètres

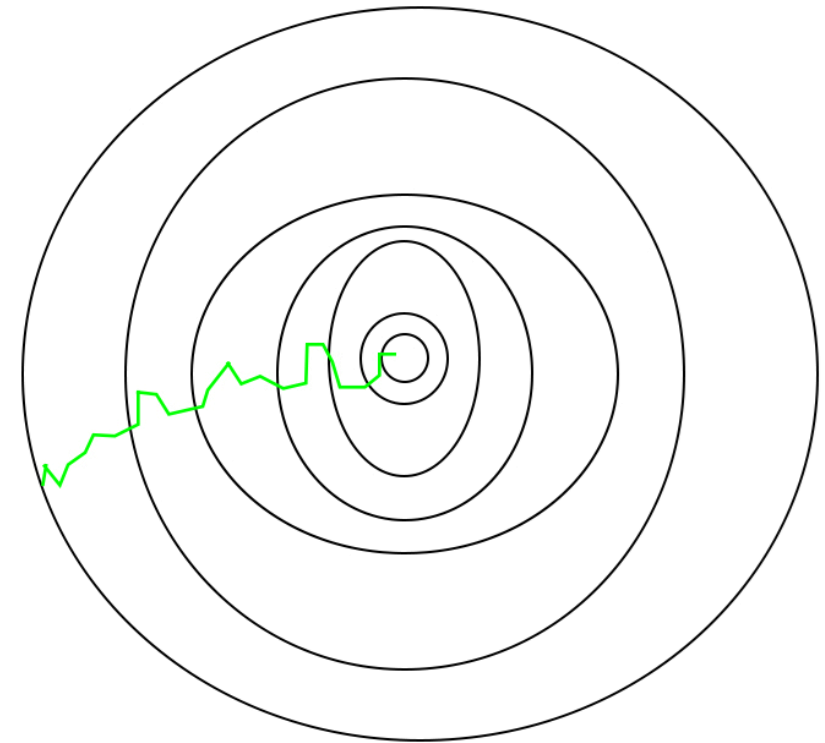
Répétez les étapes 1 à 3 pour tous les exemples de l'ensemble des données d'apprentissage



Comme un seul échantillon de l'ensemble de données est choisi au hasard pour chaque itération, le chemin suivi par l'algorithme pour atteindre les minima est généralement plus bruyant que votre algorithme typique de Gradient Descent. Mais cela n'a pas beaucoup d'importance car le chemin pris par l'algorithme n'a pas d'importance, tant que nous atteignons les minima et avec un temps d'entraînement significativement plus court.



Chemin suivi par l'algorithme **Batch Gradient Descent**



Chemin suivi par la **descente de gradient stochastique**

SGD peut être utilisé pour des ensembles de données plus importants. Il converge plus rapidement lorsque l'ensemble de données est important, car il entraîne des mises à jour plus fréquentes des paramètres.

## Algorithme

L'algorithme de la **SGD** réalise une mise à jour après chaque exemple.

choix de  $\theta_0$  et de  $\eta$

For i=1 : n do

    choisir aléatoirement un exemple noté  $l(j)$

$$\theta_{i+1} = \theta_i - \eta J_{l(j)}(\theta)$$

Fin do