OPTIMISATION POUR L'APPRENTISSAGE

23 janvier 2025

Après-midi :

Exercices de révision

Démonstration momentum (si temps)

Exam 2023-2024

Exo 2

$$\underset{w \in \mathbb{R}^d}{\text{minimiser}} \ \frac{1}{n}\sum_{i=1}^{n} f_i(w) \qquad f_i(w) = \frac{1}{2}(x_i^T w - y_i)^2$$

a) $w_{k+1} = w_k - \alpha \boxed{\nabla f_{i_k}(w_k)}$ → estimateur de $\nabla f(w) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w)$

$\alpha > 0$ longueur de pas constante

i_k tiré aléatoirement dans $\{1, \dots, n\}$

b) Unité de calcul : accès à un x_i

- 1 itération de descente de gradient : coût de n

$$w_{k+1} = w_k - \frac{\alpha}{n}\sum_{i=1}^{n} \nabla f_i(w_k)$$

- 1 itération de gradient stochastique : coût de 1

c) 1 epoch (dans ce cours)

= 1 unité de coût qui correspond à n accès à 1 point de jeu de données

On fait tourner la descente de gradient pendant K epochs ⟹ K itérations

On fait tourner le gradient stochastique pendant K epochs ⟹ nK itérations

Aparté : Pb convexe

K epochs $\Rightarrow$ K itérations de DG

$$f(w_K) - \min_{w\in\mathbb{R}^d} f(w) \leq O\left(\frac{1}{K}\right)$$

$\Rightarrow$ nK itérations de GS

$$\mathbb{E}\left[f(w_{nK}) - \min_{w\in\mathbb{R}^d} f(w)\right] \leq O\left(\frac{1}{\sqrt{nK}}\right) + O(1)$$

Constante qui dépend de la longueur de pas et de la variance des $\nabla f_{i_k}(w_k)$

Si $n \gg 1$, $\frac{1}{\sqrt{nK}} \ll \frac{1}{K}$

Batch de taille n_b

d)

$$w_{k+1} = w_k - \frac{\alpha}{n_b} \sum_{i\in S_k} \nabla f_i(w_k)$$

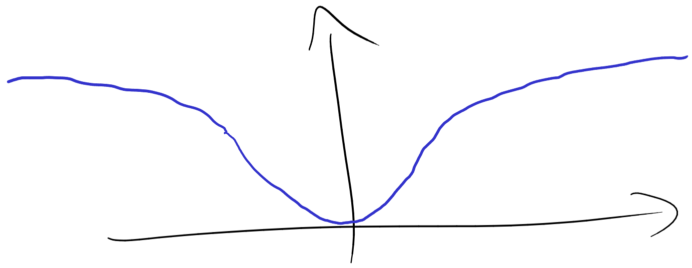
$\frac{1}{|S_k|} \sum_{i\in S_k} \nabla f_i(w_k)$ est un estimateur de $\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w_k)$

$\alpha > 0$

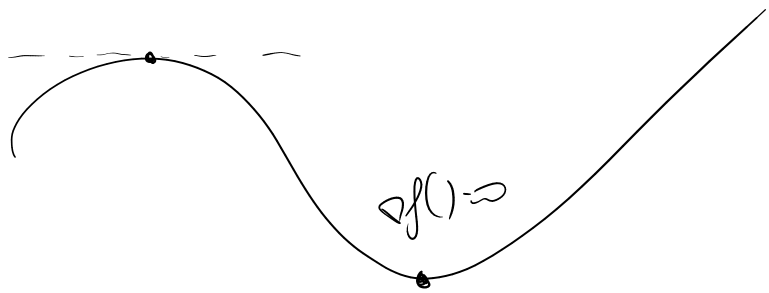
avec S_k ensemble de n_b indices tirés aléatoirement dans $\{1, \ldots, n\}$ avec ou sans remise

e) GS : $n_b = 1$, DG : $n_b = n$ + tirage sans remise

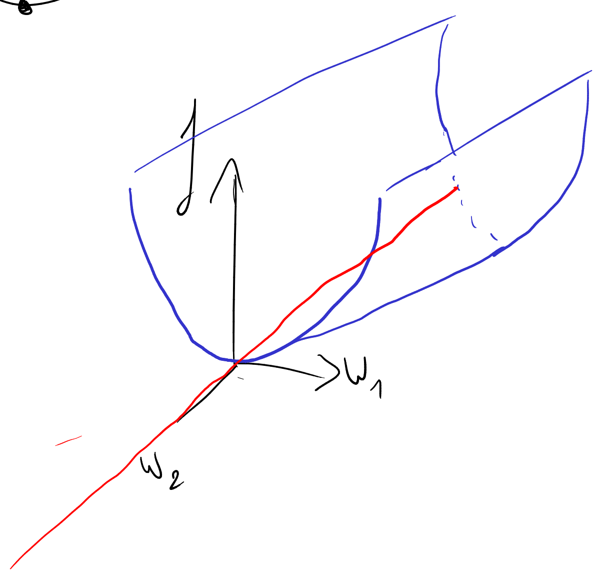
Ex 3



b) <u>Théorème</u> : Si on lance la descente de gradient en partant de w_0 choisi aléatoirement dans $\mathbb{R}^d$, la probabilité de converger vers un point selle ou un maximum local est de 0.



$f\left(\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}\right) = w_1^2$



$v^T \nabla^2 f(w) v \geq 0 \quad \forall v$

$\nabla^2 f(w) \succeq 0 \quad \forall w$

pas de min
nb fini ou infini de minima $\Big[$ f convexe $\quad C^2$

au plus 1 minimum $\Big[$ f strictement convexe

$\nabla^2 f(w) \succ 0$

$v^T \nabla^2 f(w) v > 0 \quad \forall v \neq 0$

exactement 1 minimum $\Big[$ f μ-fortement convexe, $\mu > 0$

$\nabla^2 f(w) - \mu I \succeq 0$

$v^T \nabla^2 f(w) v \geq \mu \; \forall w \, \forall v$

---

c) Complexité de la descente de gradient pour un problème non convexe

Etant donné $\varepsilon > 0$, sur un problème non convexe de fonction objectif f^{bw}, la descente de gradient calcule un point tel que $\|\nabla f^{bw}(w_K)\| \leq \varepsilon$ en au plus $O(\varepsilon^{-2})$ itérations

"Borne de complexité"

Critère de convergence (approchée)

Dans le cas convexe : $\to$ Borne : $O(\varepsilon^{-1})$

$\to$ Critère de convergence : $f(w_K) - \min_{w \in \mathbb{R}^d} f(w) \leq \varepsilon$

Dans le cas fortement convexe $\to$ Borne $O(\ln(\varepsilon^{-1}))$

$\to$ Critère $f(w_K) - f(w^*) \leq \varepsilon$

ou $\|w_K - w^*\| \leq \varepsilon$

f μ-fortement convexe

$f(w_K) - f(w^*) \geq C \|w_K - w^*\|^2$

Ex 1

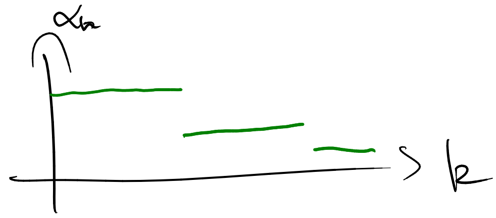
b) $\underset{w \in \mathbb{R}^d}{\operatorname{argmin}} f^{lin}(w) = \{ w \in \mathbb{R}^d \mid \nabla f^{lin}(w) = 0_{\mathbb{R}^d} \}$

$$\nabla f^{lin}(w) = X^T(Xw - y)$$

c) $w_{k+1} = w_k - \alpha_k \nabla f^{lin}(w_k)$ avec $\alpha_k > 0$

Choix possibles

- $\alpha_k = \alpha > 0$ (constante)
- $\alpha_k \searrow 0$ (décroissante)
- hybride



plutôt courant en optimisation pour le deep learning ("learning rate scheduler")

- adaptative (recherche linéaire)

Ex) choisir la plus grande valeur $\alpha_k \in \{1, \frac{1}{2}, 1/4, 1/8, \dots\}$ telle que $f^{lin}(w_k - \alpha_k \nabla f^{lin}(w_k)) < f(w_k)$

plutôt courant en optimisation hors deep learning

d) DG, pb convexe, vitesse de convergence

Après $K \geq 1$ itération(s) (budget), l'itéré de la descente de gradient vérifie

$$0 \leq f(w_K) - \min_{w \in \mathbb{R}^d} f(w) \leq O\left(\frac{1}{K}\right)$$

vitesse de convergence

f) ii) f μ-fortement convexe et $C_L^{1,1}$ $(\Rightarrow 0 < \mu/L \leq 1)$

Après $K \geq 1$ itérations, on a

$$f(w_K) - \min_{w \in \mathbb{R}^d} f(w) \leq O\Big(\underbrace{\left(1-\frac{\mu}{L}\right)}_{\in [0,1)}^K\Big)$$

Gradient accéléré (ou gradient accéléré de Nesterov) 1983

↳ Algorithme qui a les vitesses de convergence

Mieux que la descente de gradient

$$\begin{cases} O\left(\frac{1}{K^2}\right) & \text{pour } f \text{ convexe} \\ O\left(\left(1-\sqrt{\frac{\mu}{L}}\right)^K\right) & \text{pour } f \text{ } \mu\text{-fortement convexe } C_L^{1,1} \end{cases}$$

"SGD with momentum"

$$w_{k+1} = w_k - \alpha_k \nabla f_{i_k}(w_k) + \beta(w_k - w_{k-1})$$

Dans PyTorch / JAX : $\beta = 0$ (défaut)

$\beta = 0.9$

Adam

$$w_{k+1} = w_k - \alpha_k \, m_k \oslash v_k$$

↑ division composante à composante

$$\forall j = 1..d, \quad [v_k]_j = \sqrt{\frac{1-\beta_2}{1-\beta_2^{k+1}} \sum_{l=0}^{k} \beta_2^{k-l} [\nabla f_{i_l}(w_l)]_j^2}$$

→ Moyenne géométrique des valeurs des coordonnées des gradients

→ $\beta_2 \in (0,1)$:

→ Plus d'importance pour les itérations les + récentes

→ Similaire à Adagrad

$$m_k = \frac{1-\beta_1}{1-\beta_1^{k+1}} \sum_{l=0}^{k} \beta_1^{k-l} \nabla f_{i_l}(w_l)$$

⇒ Équivalent à faire du momentum

$\beta_1 = 0.9$, $\beta_2 = 0.999$

Adam - Kingma & Ba (2015)