OUTILS D'OPTIMISATION POUR LES SCIENCES DES DONNÉES ET DE LA DÉCISION

Programme : Gradient stochastique 2/2 + TD avec notebook

Semaine prochaine :

- Jeudi 8h30-11h45
- Vendredi 8h30-11h45

GRADIENT STOCHASTIQUE ET RÉDUCTION DE VARIANCE

(1) Variance dans le gradient stochastique classique

Cadre : $f : \mathbb{R}^d \to \mathbb{R}$ μ-fortement convexe $C^{1,1}_L$

$\forall w \in \mathbb{R}^d$, $f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w)$

f_i dépend du $i$ème point d'un jeu de données à n éléments

Gradient stochastique

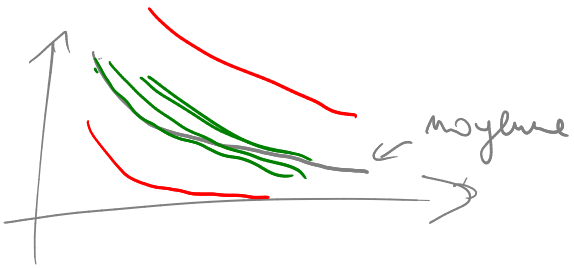
$w_{k+1} = w_k - \alpha_k \nabla f_{i_k}(w_k)$ $\quad i_k \in \{1, \dots, n\}$ tiré aléatoirement

↳ On suppose en général que

$\mathbb{E}_{i_k}\left[\nabla f_{i_k}(w_k)\right] = \nabla f(w_k)$

(en moyenne, on retrouve le vrai gradient)

⇒ Garanties sur le comportement moyen du gradient stochastique mais ne dit pas à quel point on peut dévier de ce comportement moyen



Q) Comment analyser l'écart entre les réalisations de l'algorithme et la moyenne ?

Hypothèse :

$$\mathbb{E}_{i_k}\left[\|\nabla f_{i_k}(w_k)\|^2\right] - \|\nabla f(w_k)\|^2 \leq \sigma^2 \qquad \sigma^2 \geq 0$$

- $\mathbb{E}_{i_k}$: Moyenne par rapport à i_k
- $\|\nabla f_{i_k}(w_k)\|^2$: Norme du gradient stochastique
- $\|\nabla f(w_k)\|^2$: Norme du vrai gradient

Si $\mathbb{E}_{i_k}\left[\nabla f_{i_k}(w_k)\right] = \nabla f(w_k)$, alors

$$\mathbb{E}_{i_k}\left[\|\nabla f_{i_k}(w_k)\|^2\right] - \|\nabla f(w_k)\|^2 = \underbrace{\mathbb{E}_{i_k}\left[\|\nabla f_{i_k}(w_k)\|^2 - \|\mathbb{E}_{i_k}\left[\nabla f_{i_k}(w_k)\right]\|^2\right]}_{\text{"Variance du gradient stochastique"}}$$

$\mathbb{E}[a]$: moyenne

$\mathrm{Var}[a] : \mathbb{E}[a^2] - (\mathbb{E}[a])^2$ Variance

σ^2 : Paramètre de variance (ou niveau de bruit) du gradient stochastique

- Dépend des données (f_i)
- Dépend de la façon dont on construit les gradients stochastiques (et notamment du tirage des i_k)

Théorème : Si on effectue K itérations de gradient stochastique avec $\alpha_k = \frac{1}{L}$, alors

$$\mathbb{E}_{i_1,\dots,i_K}\left[f(w_K) - f^*\right] \leq \frac{\sigma^2}{2\mu} + \left(1 - \frac{\mu}{L}\right)^K \left(f(w_0) - f^* - \frac{\sigma^2}{2\mu}\right)$$

avec $f^* = \min_{w \in \mathbb{R}^d} f(w)$

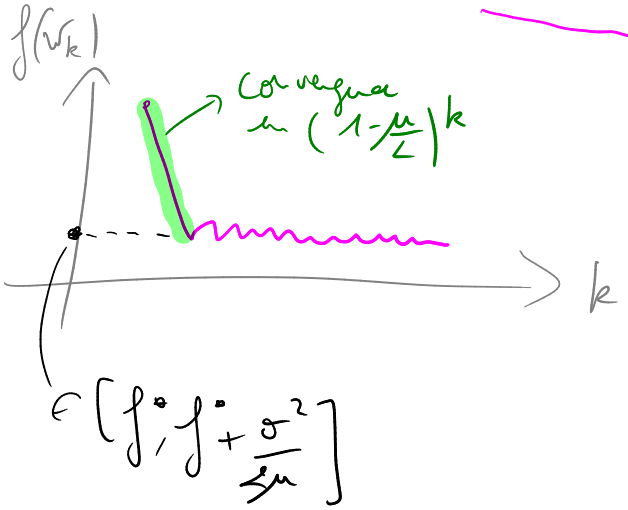
Rappel : Sur le même problème, la descente de gradient vérifie

$$\text{Or} \quad f(w_K) - f^* \leq \left(1 - \frac{\mu}{L}\right)^K (f(w_0) - f^*)$$

$\Rightarrow$ Garantit que $f(w_K) \to f^*$ en vitesse $\left(1 - \frac{\mu}{L}\right)^K$

↳ En comparaison, le résultat sur le gradient stochastique garantit en moyenne

que $f(w_K) \longrightarrow \left[f^*, f^* + \frac{\sigma^2}{2\mu}\right]$



Convergence dans un intervalle de valeurs défini par le niveau de bruit

Interprétation

* Plus les gradients stochastiques diffèrent du vrai gradient, plus une exécution de l'algorithme peut varier par rapport à sa moyenne
* En pratique, cela signifie qu'on ne peut converger que jusqu'à un certain niveau de

bruit sur les gradients stochastiques

(2) Méthodes de réduction de variance

↳ Idée: Avec un algorithme de gradient stochastique classique, on a une certaine variance sur les gradients stochastiques

⇒ Comment modifier l'algorithme pour réduire cette variance ?

a) Utiliser une fournée de gradients stochastiques

Gradient stochastique par fournées

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

S_k ensemble d'indices tirés aléatoirement dans $\{1, \dots, m\}$ (avec ou sans remise)

$|S_k| = m_b$

$$\nabla f(w_k) \approx \frac{1}{m_b} \sum_{i \in S_k} \nabla f_i(w_k)$$

et si on a un niveau de bruit σ^2 lorsque $m_b = 1$ (GS de base), alors on a un niveau de bruit $\frac{\sigma^2}{m_b}$ pour $m_b > 1$ indices tirés avec remise

b) Moyenne des itérés

↳ Dans le cas convexe, on peut montrer que $\left\{\frac{1}{k+1}\sum_{l=0}^{k} w_l\right\}$ a une plus faible variance que $\{w_k\}$ et converge vers une meilleure valeur de l'optimum

↳ En pratique, peu utilisé à cause du coût de stockage de la moyenne qui peut être trop grand et des erreurs numériques possibles dans le calcul de la moyenne

c) Agrégation de gradients

↳ Combinent des pas de gradients stochastiques avec des vrais gradients issus d'itérations précédentes

⇒ Corrige la valeur du gradient stochastique

↳ Algorithmes : SVRG, SAGA (sklearn)

⇒ Très efficaces sur des problèmes de faible dimension et convexes

⇒ Le coût de calcul peut être trop élevé dans certains contextes (ex: apprentissage profond)

(3) Variantes du gradient stochastique pour l'apprentissage profond

Forme générale

- Calculer g_k
- Calculer

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

α : taille de pas constante $\alpha > 0$

$\oslash$: division composante à composante

$$a \oslash b = \left[\frac{[a]_i}{[b]_i} \right]$$

m_k : direction du pas $m_k \in \mathbb{R}^d$ (dépend de g_k)

v_k : vecteur de normalisation (de la taille de pas selon la coordonnée j est $\frac{\alpha}{[v_k]_j}$)

↳ A chaque itération k, le pas selon la

Cas de base

$$m_k = g_k$$

$$v_k = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

⇒ Gradient stochastique classique

(a) Gradient stochastique avec momentum

$$m_k = \beta_1 m_{k-1} + (1-\beta_1) g_k$$

$\beta_1 \in (0,1)$; m_{k-1} : direction précédente

$$v_k = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

↳ Adaptation de l'idée de momentum stochastique au cas d'un gradient pour entraîner les
↳ Méthode utilisée de l'apprentissage profond réseaux lors de l'essor en 2012

(b) Adagrad

$$m_k = g_k$$

$\forall j = 1..d$

$$[v_k]_j = \sqrt{\sum_{l=0}^{k} [g_l]_j^2}$$

=> taille de pas selon la coordonnée j

$$\frac{\alpha}{\sqrt{\sum_{l=0}^{k} [g_l]_j^2}}$$

→ Méthode efficace pour des problèmes à gradients parcimonieux (beaucoup de coordonnées nulles), typique des systèmes de recommandation

→ Peut cependant conduire à des tailles de pas très petites trop rapidement

→ Des variantes comme RMS Prop utilisent une autre formule :

$$[v_k]_j = \sqrt{\beta_2 [v_{k-1}]_j^2 + (1-\beta_2)[g_k]_j^2}$$

$\beta_2 \in (0,1)$

$[v_{k-1}]_j$: valeur à l'itération $k-1$

$[g_k]_j$: valeur courante

⑥ Adam (~2015)

↳ Combinaison pondérée des gradients stochastiques pour donner plus de poids aux gradients calculés le plus récemment de l'information fournie

$$m_k = \frac{(1-\beta_1)}{1-\beta_1^{k+1}} \sum_{l=0}^{k} \beta_1^{k-l} g_l$$

$$\forall j = 1..d, \quad [v_k]_j = \sqrt{\frac{1-\beta_2}{1-\beta_2^{k+1}} \sum_{l=0}^{k} \beta_2^{k-l} [g_l]_j^2}$$

↳ (en pratique, β_1 et β_2 (0.9, 0.99)) sont pas très proches de 1

↳ LA méthode par défaut pour entraîner la plupart des architectures neuronales actuellement

=> Très efficace dans les problèmes de traitement naturel des langues (NLP)

Remarque : La théorie de convergence initialement présentée pour Adam en 2015 s'est révélée fausse en 2018.

=> Toujours des zones d'ombre dans la théorie

=> Mais la méthode est extrêmement efficace en pratique, et elle reste donc très utilisée