Analyse non asymptotique des algorithmes stochastiques adaptatifs biaisés

Sobihan Surendran [1,2], Adeline Fermanian [2], Antoine Godichon-Baggioni [1]
Sylvain Le Corff [1]

[1] *LPSM, Sorbonne Université, France*
{*sobihan.surendran, antoine.godichon_baggioni, sylvain.le_corff*}*@sorbonne-universite.fr*
[2] *LOPF, Califrais, Machine Learning Lab, Paris, France*
{*sobihan.surendran, adeline.fermanian*}*@califrais.fr*

Résumé. La Descente de Gradient Stochastique (SGD) avec pas adaptatifs est désormais largement utilisée, en particulier pour l'apprentissage des réseaux neuronaux profonds. Cependant, la plupart des résultats théoriques supposent l'accès à des estimateurs du gradient non biaisés, ce qui n'est pas le cas dans de nombreuses applications récentes en apprentissage profond et en apprentissage par renforcement utilisant des méthodes de Monte Carlo. Nous proposons dans cette présentation une analyse non asymptotique de l'algorithme SGD utilisant des estimateurs biasés du gradient ainsi que des pas adaptatifs pour les fonctions convexes et non convexes. Notre étude intègre un biais dépendant du temps et met l'accent sur l'importance de contrôler le biais et l'erreur quadratique moyenne de l'estimateur du gradient. En particulier, nous établissons que les algorithmes Adagrad et RMSProp avec gradients biaisés convergent vers des points critiques à une vitesse de convergence similaire aux résultats existants dans la littérature pour le cadre non biaisé. Enfin, nous fournissons des résultats expérimentaux utilisant des Autoencodeurs Variationnels qui illustrent nos résultats de convergence et montrent comment l'effet du biais peut être réduit par un réglage approprié des hyperparamètres.

Mots-clés. Optimisation Stochastique, Approximation Stochastique Biaisée, Méthodes de Monte Carlo, Autoencodeurs Variationnels

Abstract. Stochastic Gradient Descent (SGD) with adaptive steps is now widely used for training deep neural networks. Most theoretical results assume access to unbiased gradient estimators, which is not the case in several recent deep learning and reinforcement learning applications that use Monte Carlo methods. We provide a comprehensive non-asymptotic analysis of SGD with biased gradients and adaptive steps for convex and non-convex smooth functions. Our study incorporates time-dependent bias and emphasizes the importance of controlling the bias and Mean Squared Error of the gradient estimator. In particular, we establish that Adagrad and RMSProp with biased gradients converge to critical points for smooth non-convex functions at a rate similar to existing results in the literature for the unbiased case. Finally, we provide experimental results using Variational Autoenconders (VAE) that illustrate our convergence results and show how the effect of bias can be reduced by appropriate hyperparameter tuning.

Keywords. Stochastic Optimization, Biased Stochastic Approximation, Monte Carlo Methods, Variational Autoenconders

1 Introduction

Les algorithmes de Descente de Gradient Stochastique (SGD) sont des méthodes classiques pour entraîner des modèles statistiques basés sur des architectures profondes. Considérons le problème d'optimisation :

$$\theta_* \in \operatorname*{argmin}_{\theta \in \mathbb{R}^d} V(\theta), \tag{1}$$

où V est la fonction objectif, supposée différentiable. La descente de gradient stochastique est définie par $\theta_0 \in \mathbb{R}^d$ et pour tout $n \geq 1$,

$$\theta_{n+1} = \theta_n - \gamma_{n+1} \widehat{\nabla V}(\theta_n),$$

où $\widehat{\nabla V}(\theta_n)$ est un estimateur du gradient de V. En apprentissage profond, la stochasticité émerge par exemple par l'utilisation de mini-batchs, étant donné qu'il n'est pas possible de calculer les gradients sur l'ensemble des données. Bien que ces algorithmes aient fait l'objet de nombreuses études, tant sur le plan théorique que pratique [Bottou et al., 2018], de nombreuses questions restent ouvertes. En particulier, la plupart des résultats théoriques reposent sur l'hypothèse que l'estimateur de gradient est non biaisé. Cependant, cette hypothèse n'est pas vérifiée pour de nombreuses applications courantes. Par exemple, dans les algorithmes d'apprentissage par renforcement, les estimateurs du gradient sont souvent obtenus à partir d'une chaîne de Markov, ce qui conduit à des estimateurs biaisés [Sun et al., 2018, Doan et al., 2020]. D'autres exemples de gradients biaisés peuvent être trouvés dans le domaine des modèles génératifs utilisant des méthodes de Monte Carlo par chaînes de Markov (MCMC) et des méthodes de Monte Carlo séquentielles (SMC) [Gloaguen et al., 2022, Cardoso et al., 2023]. En particulier, l'approche IWAE proposée par Burda et al. [2015], qui est une variante de l'Autoencodeur variationnel standard (VAE) [Kingma and Welling, 2013], produit des estimateurs de gradient biaisés.

Dans les applications pratiques, l'algorithme SGD standard rencontre des difficultés pour calibrer les suites de pas. Par conséquent, les variantes modernes utilisent des pas adaptatifs basés sur les gradients ou la matrice Hessienne pour éviter les points selles et traiter les problèmes mal conditionnés. L'idée des pas adaptatifs a d'abord été proposée dans la littérature portant sur l'apprentissage en ligne par Auer et al. [2002] et ensuite adoptée pour l'optimisation stochastique, avec l'algorithme Adagrad de Duchi et al. [2011].

À notre connaissance, aucune analyse non asymptotique tenant compte à la fois de l'estimateur biaisé du gradient et des pas adaptatifs n'a été menée à ce jour. Nous présentons des garanties de convergence pour l'algorithme SGD avec des gradients biaisés et des pas adaptatifs, sous des hypothèses faibles portant sur le biais et l'erreur quadratique moyenne de l'estimateur. Dans de nombreux scénarios, il est en effet possible de contrôler ces quantités, comme cela a été récemment montré, par exemple, pour l'Échantillonnage d'Importance et les méthodes de Monte Carlo Séquentielles. En particulier, nous établissons qu'Adagrad et RMSProp avec un gradient biaisé convergent vers un point critique pour les fonctions non convexes avec une vitesse de convergence en $\mathcal{O}(\log n/\sqrt{n} + b_n)$, où b_n est lié au biais à l'itération n. Pour les fonctions convexes, nous obtenons une vitesse de convergence

amélioréoe en $\mathcal{O}(1/\sqrt{n} + b_n)$. Nos résultats théoriques nous fournissent des procédures de réglage des hyperparamètres pour éliminer efficacement le terme de biais, ce qui se traduit par de meilleures vitesses de convergence de l'ordre de $\mathcal{O}(\log n/\sqrt{n})$ et $\mathcal{O}(1/\sqrt{n})$ respectivement.

2 Algorithmes Stochastiques Adaptatifs Biasés

2.1 Cadre

Considérons le problème d'optimisation (1) et l'algorithme de gradient stochastique à pas adaptatifs biaisés suivant : $\theta_0 \in \mathbb{R}^d$ et pour tout $n \geq 0$,

$$\theta_{n+1} = \theta_n - \gamma_{n+1} A_n H_{\theta_n}\left(X_{n+1}\right) \ , \tag{2}$$

où $\gamma_{n+1} > 0$ et A_n est une suite de matrices symétriques et définies positives. Soit $(\mathcal{F}_n)_{n\geq 0}$ la filtration générée par les variables aléatoires $(\theta_0, \{X_k\}_{k\leq n})$ et supposons que pour tout $n \geq 0, A_n$ est $\mathcal{F}_n$-mesurable. Dans un contexte d'estimations de gradient biaisées, le choix de

$$A_n = \left[\delta I_d + \Big(\frac{1}{n}\sum_{k=0}^{n-1} H_{\theta_k}\left(X_{k+1}\right) H_{\theta_k}\left(X_{k+1}\right)^\top\Big)\right]^{-1/2}$$

peut être assimilé à l'algorithme full Adagrad [Duchi et al., 2011]. Cependant, calculer la racine carrée de l'inverse devient coûteux en grande dimension, donc en pratique, Adagrad est souvent utilisé avec des matrices diagonales. En notant $\mathrm{Diag}(A)$ la matrice formée avec les termes diagonaux de A et en annulant tous les autres termes, Adagrad est défini dans notre contexte par :

$$A_n = \left[\delta I_d + \mathrm{Diag}(\bar{H}_n(X_{0:n}, \theta_{0:n-1}))\right]^{-1/2}, \tag{3}$$

où

$$\bar{H}_n(X_{0:n}, \theta_{0:n-1}) = \frac{1}{n}\sum_{k=0}^{n-1} H_{\theta_k}(X_{k+1}) H_{\theta_k}(X_{k+1})^\top.$$

Dans RMSProp [Tieleman et al., 2012], $\bar{H}_n(X_{0:n}, \theta_{0:n-1})$ dans (3) est une moyenne mobile exponentielle des carrés des gradients, définie par :

$$(1-\rho)\sum_{k=0}^{n-1} \rho^{n-k} H_{\theta_k}(X_{k+1}) H_{\theta_k}(X_{k+1})^\top,$$

où ρ est le paramètre de la moyenne mobile. De plus, lorsque A_n est une estimation récursive de l'inverse de la Hessienne, cela correspond à l'algorithme Newton Stochastique [Boyer and Godichon-Baggioni, 2023].

2.2 Hypothèses

Nous énonçons ci-dessous les principales hypothèses nécessaires à nos résultats théoriques.

Hypothèse 1. *La fonction objectif V est convexe et il existe une constante $\mu > 0$ telle que :*

$$2\mu \left(V(\theta) - V\left(\theta^*\right)\right) \leq \|\nabla V(\theta)\|^2, \quad \forall \theta \in \mathbb{R}^d.$$

La deuxième condition de l'Hypothèse 1 correspond à la condition de Polyak-Łojasiewicz. De plus, l'Hypothèse 1 est une hypothèse plus faible par rapport à la forte convexité de la fonction.

Hypothèse 2. *Le gradient de la fonction objectif V est Lipschitz. Pour tout $(\theta, \theta') \in \mathbb{R}^d \times \mathbb{R}^d$,*

$$\|\nabla V(\theta) - \nabla V\left(\theta'\right)\| \leq L \left\|\theta - \theta'\right\|.$$

Cette hypothèse est cruciale pour obtenir notre vitesse de convergence et est très courante [Moulines and Bach, 2011, Bottou et al., 2018].

Hypothèse 3. *Pour tout $n \in \mathbb{N}$, il existe $r_n \geq 0$ et $\sigma_n^2 \geq 0$ tels que :*

(i) Biais : $\left\|\mathbb{E}[H_{\theta_n}(X_{n+1}) \mid \mathcal{F}_n] - \nabla V(\theta_n)\right\| \leq r_n$.

(ii) Erreur quadratique moyenne :

$$\mathbb{E}\left[\left\|H_{\theta_n}\left(X_{n+1}\right) - \nabla V\left(\theta_n\right)\right\|^2 \mid \mathcal{F}_n\right] \leq \sigma_n^2.$$

Dans cette hypothèse, les suites r_n et σ_n^2 contrôlent le biais et l'erreur quadratique moyenne sans aucune hypothèse spécifique sur leur dépendance par rapport à n, ce qui rend notre cadre très général. Cette hypothèse peut être vérifiée dans diverses applications utilisant des méthodes de Monte Carlo. Si $H_{\theta_n}\left(X_{n+1}\right)$ est un estimateur non biaisé du gradient, le deuxième point est équivalent à garantir que la variance du terme de bruit est bornée. Dans ce contexte, cette hypothèse est standard, voir par exemple Moulines and Bach [2011], Ghadimi and Lan [2013].

Nous considérons également une hypothèse supplémentaire sur A_n. Soit $\|A\|$ la norme spectrale d'une matrice A.

Hypothèse 4. *Pour tout $n \in \mathbb{N}$, il existe $\beta_n, \lambda_n > 0$ tels que :*

$$\|A_n\| := \lambda_{\max}(A_n) \leq \beta_{n+1} \quad \textit{et} \quad \lambda_{\min}\left(A_n\right) \geq \lambda_{n+1}.$$

Dans notre cadre, puisque A_n est supposée être une matrice symétrique, la norme spectrale est égale à la plus grande valeur propre.

Hypothèse 5. *Le gradient est borné, c'est-à-dire qu'il existe $M \geq 0$ tel que pour tout $n \in \mathbb{N}$,*

$$\|H_{\theta_n}\left(X_{n+1}\right)\| \leq M.$$

Il est important de noter que sous l'Hypothèse 3, ceci équivaut à borner le gradient stochastique de la fonction objectif.

2.3 Résultats de Convergence

2.3.1 Cas Convexe

Dans cette section, nous étudions les résultats de convergence pour l'algorithme SGD avec des gradients biaisés et des pas adaptatifs dans le cas convexe. Nous donnons ci-dessous une version simplifiée de la borne que nous avons obtenue sur le risque sans aucune constante explicite.

Théorème 2.1. *Soit $\theta_n \in \mathbb{R}^d$ la n-ème itération de la récursion (2) et $\gamma_n = C_\gamma n^{-\gamma}$, $\beta_n = C_\beta n^{\beta}$, $\lambda_n = C_\lambda n^{-\lambda}$ avec $C_\gamma > 0$, $C_\beta > 0$ et $C_\lambda > 0$. Supposons que $\gamma, \beta, \lambda \geq 0$ et $\gamma + \lambda < 1$. Alors, sous les hypothèses $1-4$, nous avons :*

$$\mathbb{E}\left[V\left(\theta_n\right) - V(\theta^*)\right] = \mathcal{O}\left(n^{-\gamma+2\beta+\lambda} + b_n\right), \tag{4}$$

où le terme de biais b_n peut être constant ou décroissant. Dans le dernier cas, en écrivant $r_n = n^{-\alpha}$, nous avons :

$$b_n = \mathcal{O}\left(n^{-2\alpha+2\beta+2\lambda}\right).$$

Le résultat obtenu est classique et montre le compromis entre un terme provenant des pas adaptatifs (avec une dépendance en γ, β, λ) et un terme b_n qui dépend du contrôle du biais r_n. Pour minimiser le membre de droite de (4), nous aimerions avoir $\beta = \lambda = 0$. Cependant, cela nécessiterait des hypothèses beaucoup plus fortes. Par exemple, dans le cas d'Adagrad et de RMSProp, les gradients devraient être bornés. Dans le cas de SGD avec échantillonnage de coordonnées mais sans pas adaptatifs, le résultat analogue peut être trouvé dans Leluc and Portier [2022].

2.3.2 Cas non convexe à gradient Lipschitz

Dans le cas non convexe à gradient Lipschitz, nous avons obtenu la vitesse de convergence sans faire l'hypothèse d'un gradient borné, cependant, nous présentons uniquement les résultats pour le cas d'un gradient borné dans le cadre de l'Approximation Stochastique Adaptative Randomisée. Le théorème suivant fournit une borne en espérance sur le gradient de la fonction objectif V, ce qui est le meilleur résultat que nous puissions obtenir étant donné qu'aucune hypothèse n'est faite sur l'existence d'un minimum global de V.

Théorème 2.2. *Soient $\gamma_n = C_\gamma n^{-\gamma}, \beta_n = C_\beta n^{\beta}, \lambda_n = C_\lambda n^{-\lambda}$ avec $C_\gamma > 0, C_\beta > 0$, et $C_\lambda > 0$. Supposons que $\gamma, \beta, \lambda \geq 0$ et $\gamma + \lambda < 1$. Pour tout $n \geq 1$, soit $R \in \{0, \ldots, n\}$ une variable aléatoire uniformément distribuée. Alors, sous les hypothèses $2-4, 5$, nous avons :*

$$\mathbb{E}\left[\left\|\nabla V\left(\theta_R\right)\right\|^2\right] \leq \frac{2V^* + \alpha_{1,n} + LM^2\alpha_{2,n}}{\sqrt{n}},$$

où $V^* = \mathbb{E}[V(\theta_0) - V(\theta^*)]$, $\alpha_{1,n} = \sum_{k=0}^{n} \gamma_{k+1}\beta_{k+1}^2 r_k^2/\lambda_{k+1}$ et $\alpha_{2,n} = \sum_{k=0}^{n} \gamma_{k+1}^2\beta_{k+1}^2$.

2.3.3 Application à Adagrad et RMSProp

Dans cette section, nous fournissons l'analyse de convergence d'Adagrad et de RMSProp avec un estimateur de gradient biaisé.

Corollaire 2.1. *Soient $\gamma_n = c_\gamma n^{-1/2}$ et A_n la matrice adaptative dans Adagrad ou RMSProp. Pour tout $n \geq 1$, soit $R \in \{0, \ldots, n\}$ une variable aléatoire uniformément distribuée. Alors, sous les Hypothèses $2, 3, 5$, nous avons :*

$$\mathbb{E}\left[\|\nabla V(\theta_R)\|^2\right] = \begin{cases} \mathcal{O}\left(n^{-2\alpha}\right) & \text{if } \alpha < 1/4\,, \\ \mathcal{O}\left(\frac{\log n}{\sqrt{n}}\right) & \text{if } \alpha \geq 1/4\,. \end{cases}$$

Le Corollaire 2.1 établit la vitesse de convergence d'Adagrad et de RMSProp avec un gradient biaisé vers un point critique pour les fonctions non convexes à gradient Lipschitz. Dans le cas d'un gradient non biaisé, nous obtenons la même borne que dans Zou et al. [2018], sous les mêmes hypothèses :

$$\mathbb{E}\left[\|\nabla V(\theta_R)\|^2\right] = \mathcal{O}\left(\frac{\log n}{\sqrt{n}}\right).$$

Si le biais est de l'ordre $\mathcal{O}(n^{-1/4})$, l'algorithme atteint le même vitesse de convergence que dans le cas d'un gradient non biaisé.

3 Illustrations numériques

Dans cette section, nous illustrons nos résultats théoriques dans le contexte des Variational Autoencoders (VAE) profonds. Dans les modèles génératifs, l'objectif est de maximiser la vraisemblance marginale définie comme suit :

$$\log p_\theta(x) = \log \mathbb{E}_{p_\theta(\cdot|x)}\left[\frac{p_\theta(x, Z)}{p_\theta(Z|x)}\right],$$

où $(x, z) \mapsto p_\theta(x, z)$ est la vraisemblance complète, x sont les observations et Z est la variable latente. Sous certaines hypothèses techniques, selon l'identité de Fisher, nous avons :

$$\nabla_\theta \log p_\theta(x) = \int \nabla_\theta \log p_\theta(x, z) p_\theta(z \mid x) \mathrm{d}z. \tag{5}$$

Cependant, dans la plupart des cas, la densité conditionnelle $z \mapsto p_\theta(z \mid x)$ est intractable et ne peut être échantillonnée directement. Les autoencodeurs variationnels introduisent un paramètre supplémentaire ϕ et une famille de distributions variationnelles $z \mapsto q_\phi(z \mid x)$ pour approcher la vraie distribution postérieure. Les paramètres sont estimés en maximisant la borne inférieure de la log-vraisemblance (ELBO) :

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x, Z)}{q_\phi(Z|x)}\right] =: \mathcal{L}_{\mathrm{ELBO}}(\theta, \phi; x).$$

L'algorithme IWAE [Burda et al., 2015] est une variante du VAE qui incorpore des poids d'importance pour obtenir une ELBO plus précise. L'objectif IWAE peut être écrit comme suit :

$$\mathcal{L}_k^{\text{IWAE}}(\theta, \phi; x) = \mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)} \left[\log \frac{1}{k} \sum_{\ell=1}^{k} \frac{p_\theta(x, Z^{(\ell)})}{q_\phi(Z^{(\ell)}|x)} \right],$$

où k correspond au nombre d'échantillons tirés sous la distribution variationnelle. L'estimateur du gradient de la ELBO dans IWAE est biaisé et le biais ainsi que l'erreur quadratique moyenne sont d'ordre $\mathcal{O}(1/k)$.

Nous menons nos expériences sur l'ensemble de données CIFAR-10 en utilisant Adagrad et RMSProp. Pour illustrer nos résultats, nous choisissons d'incorporer un biais d'ordre $\mathcal{O}(n^{-\alpha})$ à l'itération n. Comme le biais de l'estimateur du gradient dans IWAE est de l'ordre $\mathcal{O}(1/k)$, choisir un biais d'ordre $\mathcal{O}(n^{-\alpha})$ équivaut à utiliser n^α échantillons à l'itération n pour estimer le gradient. Nous faisons varier α pour IWAE et nous affichons les quantités suivantes.

- Dans la Figure 1, la norme au carré du gradient $\|\nabla V(\theta_n)\|^2$ pour illustrer la vitesse de convergence.
- Dans la Figure 2, la vraisemblance négative en fonction des itérations.

La Figure 1 illustre nos résultats, tandis que les autres figures visent à confirmer le comportement de la perte de test avec différentes valeurs de α. Toutes les figures sont tracées sur une échelle logarithmique pour une meilleure visualisation. Il est important de noter que toutes les figures sont par rapport aux époques, alors qu'ici, n représente l'itération (nombre de mises à jour du gradient).

Nous pouvons clairement observer qu'une convergence rapide est atteinte dans chacun des cas lorsque n est suffisamment grand. Il existe plusieurs explications possibles à cette convergence rapide. Par exemple, nous pourrions être en mesure d'améliorer la borne supérieure en obtenant une meilleure estimation du biais. Nos expériences montrent des résultats similaires pour Adagrad et RMSProp en termes de la vitesse de convergence, bien que RMSProp semble se comporter légèrement mieux.

Il est clair qu'avec un α plus grand, la convergence à la fois de la norme au carré du gradient et de la vraisemblance négative est plus rapide. Cependant, au-delà d'un certain seuil pour α, nous observons que la vitesse de convergence ne change pas significativement. Comme choisir un α plus grand induit un coût computationnel supplémentaire, il est crucial de sélectionner une valeur appropriée de α, qui atteint une convergence rapide sans être trop coûteuse en termes de temps de calcul.

References

P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75, 2002.

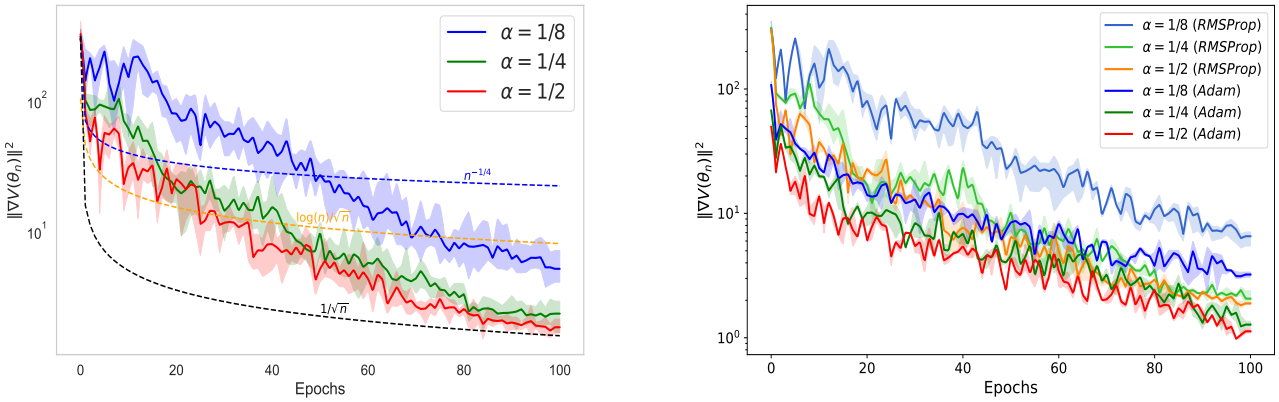

Figure 1: Valeur de $\|\nabla V(\theta_n)\|^2$ dans IWAE avec Adagrad (à gauche), RMSProp et Adam (à droite) pour différentes valeurs de α. La courbe en pointillés correspond à la vitesse de convergence attendue $\mathcal{O}(n^{-1/4})$ pour $\alpha = 1/8$ et $\mathcal{O}(\log n/\sqrt{n})$ pour $\alpha = 1/4$ et pour $\alpha = 1/2$. Les lignes en gras représentent la moyenne sur 5 exécutions indépendantes.

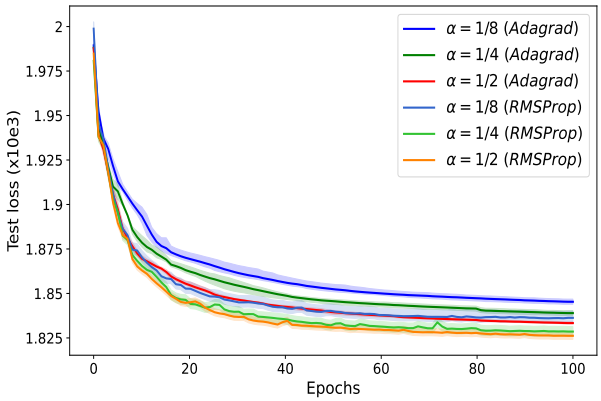

Figure 2: Vraisemblance négative sur l'ensemble de test de CIFAR-10 pour IWAE avec Adagrad and RMSProp en fonction des itérations pour différentes valeurs de α.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

C. Boyer and A. Godichon-Baggioni. On the asymptotic rate of convergence of stochastic newton algorithms and their weighted averaged versions. *Computational Optimization and Applications*, 84(3):921–972, 2023.

Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

G. Cardoso, Y. J. E. Idrissi, S. Le Corff, É. Moulines, and J. Olsson. State and parameter learning with PaRIS particle Gibbs. *arXiv preprint arXiv:2301.00900*, 2023.

T. T. Doan, L. M. Nguyen, N. H. Pham, and J. Romberg. Finite-time analysis of stochastic gradient descent under markov randomness. *arXiv preprint arXiv:2003.10973*, 2020.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

S. Ghadimi and G. Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

P. Gloaguen, S. Le Corff, and J. Olsson. A pseudo-marginal sequential Monte Carlo online smoothing algorithm. *Bernoulli*, 28(4):2606–2633, 2022.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

R. Leluc and F. Portier. Sgd with coordinate sampling: Theory and practice. *The Journal of Machine Learning Research*, 23(1):15470–15516, 2022.

E. Moulines and F. Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in Neural Information Processing Systems*, 24, 2011.

T. Sun, Y. Sun, and W. Yin. On Markov chain gradient descent. *Advances in Neural Information Processing Systems*, 31, 2018.

T. Tieleman, G. Hinton, et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2):26–31, 2012.

F. Zou, L. Shen, Z. Jie, J. Sun, and W. Liu. Weighted adagrad with unified momentum. *arXiv preprint arXiv:1808.03408*, 2018.