Adaptive stochastic optimizers, Euler-Krylov's polygonal approximations and the training of neural nets

Sotirios Sabanis

The University of Edinburgh & **The Alan Turing Institute** & **National Technical University of Athens**

30/06/2023

Many **ML problems**, **including the training of (artificial) neural networks**, can be described through the formulation of a non-convex optimization problem such as

$$\text{minimize} \quad u(\theta) := \mathbb{E}[U(\theta, X_0)],$$

where $\theta \in \mathbb{R}^d$ and X_0 is a random element with some unknown probability law. One then needs to estimate a $\hat{\theta}$, more precisely its law, such that the **expected excess risk**

$$\mathbb{E}[u(\hat{\theta})] - \inf_{\theta \in \mathbb{R}^d} u(\theta)$$

is **minimized**.

This optimization problem can thus be decomposed into sub-problems, one of which is a problem of sampling from the target distribution $\pi_\beta(\theta) \propto \exp(-\beta u(\theta))$ with $\beta > 0$.
In fact, one observes that if the n-th iterate of our **favourite sampling algorithm** (with fixed step-size λ), say θ_n^λ, is used in place of $\hat{\theta}$, then the expected excess risk can be estimated as follows

$$\mathbb{E}\left[u\left(\theta_n^\lambda\right)\right] - u_\star = \underbrace{\mathbb{E}\left[u\left(\theta_n^\lambda\right)\right] - \mathbb{E}\left[u\left(\theta_\infty\right)\right]}_{\mathcal{T}_1} + \underbrace{\mathbb{E}\left[u\left(\theta_\infty\right)\right] - u_\star}_{\mathcal{T}_2} \quad (1)$$

where $u_\star := \inf_{\theta \in \mathbb{R}^d} u(\theta)$.

Description of ADAM framework

ADAM algorithm - 148401 citations! (29/06/2023)
Adam frameowrk - i.e. adaptive stochastic gradient methods can be written as follows, for $n \in \mathbb{N}$,

$$\begin{aligned} m_n &= \phi_n(G_1, \cdots, G_n), \\ V_n &= \psi_n(G_1, \cdots, G_n), \\ \theta_{n+1} &= \theta_n - \lambda_n \frac{m_n}{\varepsilon + \sqrt{V_n}} \end{aligned} \tag{2}$$

where $G_i := G(\theta_i, X_i)$ is the **stochastic gradient** evaluated at the i-th iteration, λ_n is the **step size** and all operations are applied element-wise.

The Table below provides details of the different ϕ_n's and ψ_n's.

Table: Summary of stochastic optimization methods within the general framework. Note that $\widehat{v}_n = \max\{\widehat{v}_{n-1}, v_n\}$ is defined as $v_n = (1-\beta_2)v_{n-1} + \beta_2 G_n^2$.

	SGD	RMSPROP	ADAM	AMSGRAD
$\phi_n :=$	G_n	G_n	$(1-\beta_1)\sum_{i=1}^n \beta_1^{n-i} G_i$	$(1-\beta_1)\sum_{i=1}^n \beta_1^{n-i} G_i$
$\psi_n :=$	$\mathbb{I}_n$	$(1-\beta_2)diag(\sum_{i=1}^n \beta_2^{n-i} G_i^2)$	$(1-\beta_2)diag(\sum_{i=1}^n \beta_2^{n-i} G_i^2)$	$diag(\widehat{v}_n)$

A new class of *Langevin*-based, adaptive stochastic optimizers

POLYGONAL UNADJUSTED LANGEVIN ALGORITHMS

is compared with ADAM and variants of ADAM

- in key ML/AI tasks (e.g. image classification) and shows superb performance

- while full theoretical guarantees are provided for these new Langevin-based algorithms' convergent properties and non asymptotic error bounds.

Link between sampling and optimization: OU example

In the general setting, one would like to sample from a target distribution which is defined by

$$\pi_\beta(A) := \int_A \mathrm{e}^{-\beta u(\theta)}\,\mathrm{d}\theta \Big/ \int_{\mathbb{R}^d} \mathrm{e}^{-\beta u(\theta)}\,\mathrm{d}\theta,\ A \in \mathcal{B}(\mathbb{R}^d),$$

where $\beta \in \mathbb{R}_+$ is the 'inverse temperature parameter', $\mathcal{B}(\mathbb{R}^d)$ denotes the Borel sets of $\mathbb{R}^d$ and the function $u : \mathbb{R}^d \to \mathbb{R}_+$ is continuously differentiable with $h := \nabla u$.

Thus one considers the so-called (overdamped) **Langevin** stochastic differential equation (SDE)

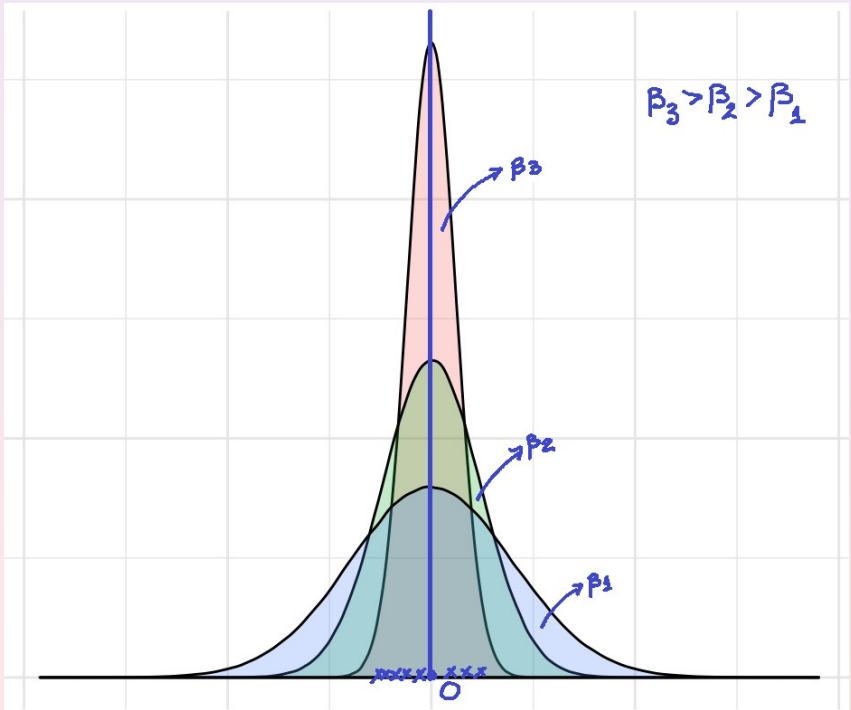
$$dL_t = -h(L_t)dt + \sqrt{2\beta^{-1}}dB_t, \tag{3}$$

where B is the standard Brownian motion in $\mathbb{R}^d$. Under suitable assumptions on h, it can be shown that the target measure (Gibbs measure) $\pi_\beta(\mathrm{d}\theta) \propto \exp(-\beta u(\theta))\mathrm{d}\theta$ is the unique invariant distribution of (3).

In particular, if one considers the following (linear) Langevin SDE (which is known also as OU process)

$$dL_t = -L_t dt + \sqrt{2\beta^{-1}} dB_t, \tag{4}$$

one knows that its limiting distribution is $\mathcal{N}(0, 1/\beta)$ with density function proportional to $e^{-\beta \frac{\theta^2}{2}}$. In other words, $u(\theta) = \frac{\theta^2}{2}$ and the (gradient) $h(\theta) = \theta$.



This is also true in the non-convex $u(\theta)$ case, i.e. π_β concentrates **around the minimizers** of u for sufficiently large β, see

C.-R. Hwang.
Laplace's method revisited: weak convergence of probability measures. The Annals of Probability. 8(6):1177–1182, 1980.

This will essentially take care of $\mathcal{T}_2$ in the expected excess risk calculations

$$\mathbb{E}\left[u\left(\theta_n^{\lambda}\right)\right] - u_\star = \underbrace{\mathbb{E}\left[u\left(\theta_n^{\lambda}\right)\right] - \mathbb{E}\left[u\left(\theta_\infty\right)\right]}_{\mathcal{T}_1} + \underbrace{\mathbb{E}\left[u\left(\theta_\infty\right)\right] - u_\star}_{\mathcal{T}_2} \quad (5)$$

where $u_\star := \inf_{\theta \in \mathbb{R}^d} u(\theta)$.
Even, with a polynomial u one obtains

$$\mathcal{T}_2 \leq \frac{\frac{d}{2}\log\left(\frac{Ke}{A}\left(\frac{B}{d}\beta + 1\right)\right) + \log 2}{\beta}$$

THE EXPLODING AND VANISHING GRADIENT ISSUES

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. Proceedings of the 30th International Conference on Machine Learning, 2013.

[6164 citations] “There are two widely known issues with properly training Recurrent Neural Networks, the **vanishing** and the **exploding** gradient problems detailed in Bengio et al. (1994).”

First attempt: **Introduction of 'taming' technology in stochastic gradient (Langevin based) algorithms**

Lovas, A., Lytras, I., Rásonyi, M. and Sabanis, S.: Taming neural networks with TUSLA: Non-convex learning via adaptive stochastic gradient Langevin algorithms. SIAM Journal on Mathematics of Data Science 5(2):323-345 (2023).

Back to

POLYGONAL UNADJUSTED LANGEVIN ALGORITHMS

Mathematically, it is described as follows: Given an i.i.d. sequence of random variables $\{X_n\}_{n\geq 0}$ of interest, which typically represent available data, the algorithm follows

$$\theta^{\lambda}_{n+1} := \theta^{\lambda}_{n} - \lambda H_{\lambda}(\theta^{\lambda}_{n}, X_{n+1}) + \sqrt{2\lambda\beta^{-1}}\xi_{n+1}, \tag{6}$$

where $n \in \mathbb{N}$, $\theta^{\lambda}_{0} := \theta_0$, θ_0 is an $\mathbb{R}^d$-valued random variable, $\lambda > 0$ denotes the step size of the algorithm, $\beta > 0$ is the so-called inverse temperature, $(\xi_n)_{n\in\mathbb{N}}$ is an $\mathbb{R}^d$-valued Gaussian process with i.i.d. components and $H_{\lambda} : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ satisfies the following three properties:

1. For every $\lambda > 0$, there exist constants $K_\lambda > 0$ and $\rho_1 \geq 0$ such that

$$|H_\lambda(\theta, x)| \leq K_\lambda(1 + |x|)^{\rho_1}(1 + |\theta|), \qquad \forall \theta \in \mathbb{R}^d \text{ and } x \in \mathbb{R}^m.$$

2. There exist constants $\gamma \geq 1/2$, $K_2 > 0$ and $\rho_2, \rho_3 \geq 0$ such that for all $\lambda > 0$,

$$|H_\lambda(\theta, x) - H(\theta, x)| \leq \lambda^\gamma K_2(1+|x|)^{\rho_2}(1+|\theta|)^{\rho_3}, \quad \forall \theta \in \mathbb{R}^d, x \in \mathbb{R}^m,$$

where H is the (unbiased) stochastic gradient of the objective function of the optimization problem.

3. There exist constants λ_{max} and $\delta \in \{1, 2\}$ such that for any $\lambda \leq \lambda_{max}$,

$$\liminf_{|\theta| \to \infty} \mathbb{E}\left[\left\langle \frac{\theta}{|\theta|^\delta}, H_\lambda(\theta, X_0)\right\rangle - \frac{2\lambda}{|\theta|^\delta}|H_\lambda(\theta, X_0)|^2\right] > 0.$$

N. Brosse, A. Durmus, É. Moulines, and S. Sabanis. *The tamed unadjusted Langevin algorithm. Stochastic Processes and their Applications.* 129(10):3638–3663, 2019.

$$X_{k+1} = X_k - \gamma G_\gamma(X_k) + \sqrt{2\gamma} Z_{k+1} , \ X_0 = x_0 . \tag{3}$$

We suggest two different explicit choices for the family $(G_\gamma)_{\gamma>0}$ based on previous studies on the tamed Euler scheme [18,20,35]. Define for all $\gamma > 0$, H_γ, $H_{\gamma,c} : \mathbb{R}^d \to \mathbb{R}^d$ for all $x \in \mathbb{R}^d$ by

$$H_\gamma(x) = \frac{\nabla U(x)}{1 + \gamma \|\nabla U(x)\|} \quad \text{and} \quad H_{\gamma,c}(x) = \left(\frac{\partial_i U(x)}{1 + \gamma |\partial_i U(x)|} \right)_{i \in \{1,\dots,d\}} , \tag{4}$$

where $\partial_i U$ is the ith-coordinate of ∇U. The Euler scheme (3) with $G_\gamma = H_\gamma$, respectively $G_\gamma = H_{\gamma,c}$, is referred to as the Tamed Unadjusted Langevin Algorithm (TULA), respectively the coordinate-wise Tamed Unadjusted Langevin Algorithm (TULAc).

Lovas, A., Lytras, I., Rásonyi, M. and Sabanis, S.: Taming neural networks with TUSLA: Non-convex learning via adaptive stochastic gradient Langevin algorithms. SIAM Journal on Mathematics of Data Science 5(2):323-345 (2023)

To address both the exploding and vanishing gradient issues we introduce

TAMED HYBRID ε-ORDER POLYGONAL UNADJUSTED LANGEVIN ALGORITHM

or, as an acronym, **THεO POULA**,

which has superior empirical performance (in given tasks below) than popular adaptive optimizers such as **AdaGrad**, **RMSProp**, **Adam** (with more than 140k citations!) and some of its most popular variants.

Diederik P. Kingma and Jimmy Ba.
Adam: A Method for Stochastic Optimization.
arXiv:1412.6980 [ICLR (Poster) 2015]

THEOPOULA (Daughter of God) vs ADAM (First Human)

Let $G : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ be a measurable function such that $\mathbb{E}[G(\theta, X_n)] = h(\theta),\ \theta \in \mathbb{R}^d,\ n \in \mathbb{N}$ (**unbiased estimator**).

STOCHASTIC GRADIENT DESCENT (SGD):

$$\theta_0^\lambda := \theta_0, \qquad \theta_{n+1}^\lambda := \theta_n^\lambda - \lambda G(\theta_n^\lambda, X_{n+1}), \quad n \in \mathbb{N},$$

STOCHASTIC GRADIENT LANGEVIN DYNAMICS (SGLD):

$$\theta_0^\lambda := \theta_0, \qquad \theta_{n+1}^\lambda := \theta_n^\lambda - \lambda G(\theta_n^\lambda, X_{n+1}) + \sqrt{2\lambda\beta^{-1}}\xi_{n+1}, \quad n \in \mathbb{N},$$

where appropriate scaled noise is added, i.e. $\{\xi_n\}_{n\geq 1}$ is a sequence of independent standard d-dimensional Gaussian random variables.

TAMED UNADJUSTED STOCHASTIC LANGEVIN ALGORITHM (TUSLA):

$$\theta_0^\lambda := \theta_0, \qquad \theta_{n+1}^\lambda := \theta_n^\lambda - \lambda \frac{G(\theta_n^\lambda, X_{n+1})}{1 + \sqrt{\lambda}|\theta_n^\lambda|^{2r}} + \sqrt{2\lambda\beta^{-1}}\xi_{n+1}, \quad n \in \mathbb{N},$$

A very recent advance in this direction (Euler-Krylov polygonal schemes) produced a **new algorithm** THεO POULA (Tamed Hybrid ϵ-Order POlygonal Unadjusted Langevin Algorithm), which iterately updates as follows:

$$\theta_0^\lambda := \theta_0, \qquad \theta_{n+1}^\lambda := \theta_n^\lambda - \lambda H_\lambda(\theta_n^\lambda, X_{n+1}) + \sqrt{2\lambda\beta^{-1}}\xi_{n+1}, \quad n \in \mathbb{N}, \tag{7}$$

where $H_\lambda := (H_\lambda^{(1)}(\theta, x), \cdots, H_\lambda^{(d)}(\theta, x))^T$ is given by

$$H_\lambda^{(i)}(\theta, x) = \frac{G^{(i)}(\theta, x)}{1 + \sqrt{\lambda}|G^{(i)}(\theta, x)|}\left(1 + \frac{\sqrt{\lambda}}{\epsilon + |G^{(i)}(\theta, x)|}\right) + \eta\frac{\theta^{(i)}|\theta|^{2r}}{1 + \sqrt{\lambda}|\theta|^{2r}},$$

and $\{\xi_n\}_{n\geq 1}$ is a sequence of independent standard d-dimensional Gaussian random variables.

Taming and boosting functions

- How to prevent the vanishing and exploding gradient problems?

	taming function	boosting function	stepsiz
region ①	≈ 1	amplify stepsize by up to $\frac{\sqrt{\lambda}}{\epsilon}$	↑
region ②	proportional to G	≈ 1	↓

- Therefore, THεO POULA takes a desirable stepsize depending on the size of the gradient.

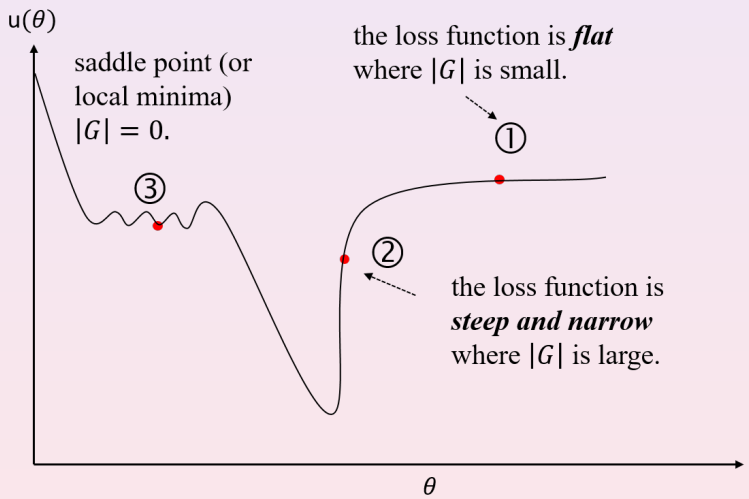
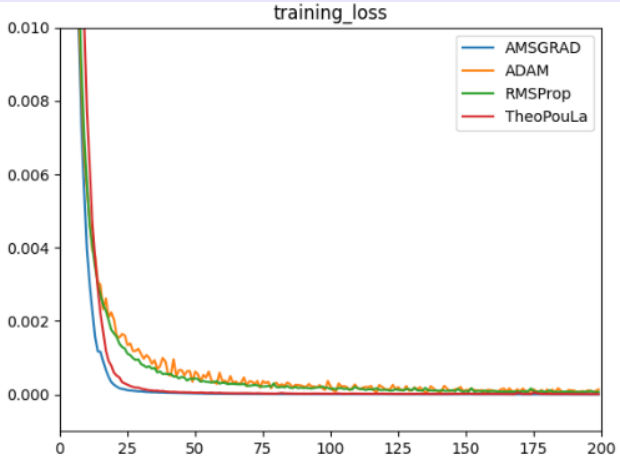


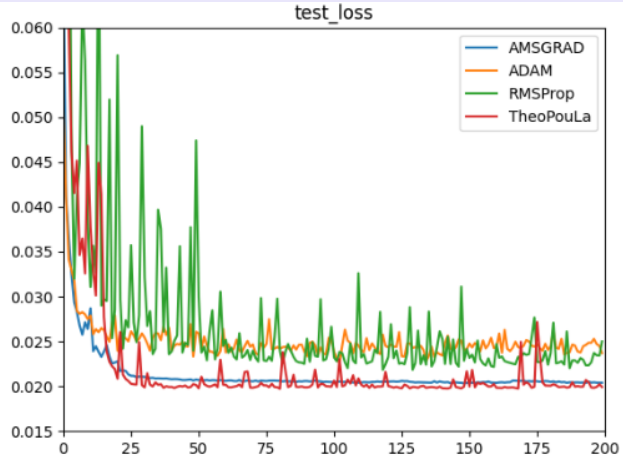
Figure: An Illustration of the behavior of THεO POULA.

[1] stepsize $:= |\Delta\theta_n^\lambda| = |\theta_{n+1}^\lambda - \theta_n^\lambda|$

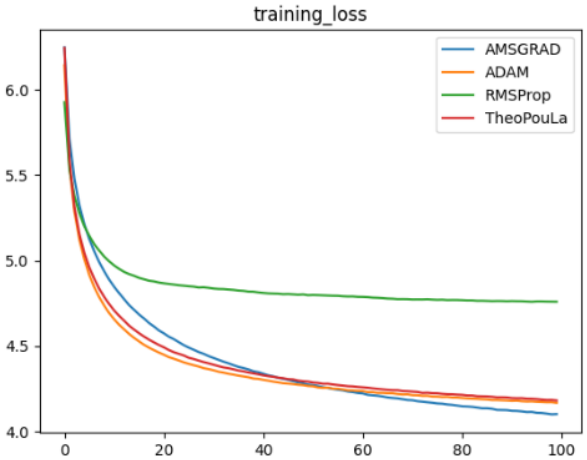
THϵO POULA vs ADAM



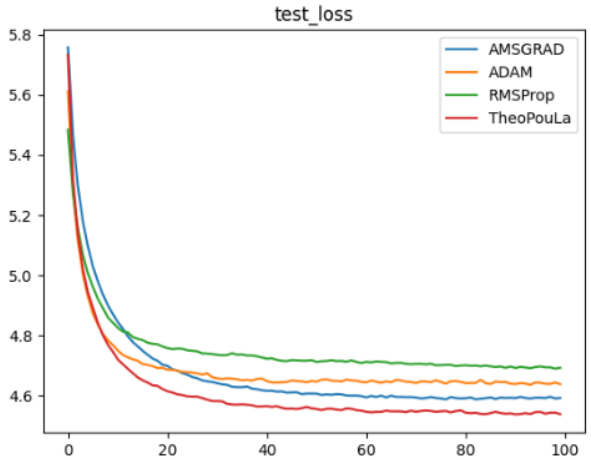
(a) CIFAR10 (Training Set)



(b) CIFAR10 (Test Set)



(c) Penn Treebank (Training Set)



(d) Penn Treebank (Test Set)

Further results on more advanced architecture of neural nets (e.g. deep convolutional networks):

VGG11, **ResNet34** and **DenseNet121** on CIFAR10.

THϵO POULA outperforms all aforementioned stochastic optimizers including SGD (with momentum) and AdaBound.

Image classification

Table: Test accuracy for VGG11, ResNet34 and DenseNet121 on CIFAR 10 and CIFAR 100.

dataset	CIFAR-10			CIFAR-100		
model	VGG	ResNet	DenseNet	VGG	ResNet	DenseN
THεO POULA†	**92.30** (0.0231)	**95.43** (0.095)	**95.66** (0.066)	**70.31** (0.117)	**77.60** (0.144)	**79.90** (0.133)
AdaBelief	92.17 (0.035)	95.29 (0.196)	95.58 (0.095)	69.50 (0.111)	77.33 (0.172)	79.12 (0.382)
Adam	90.79 (0.075)	93.11 (0.184)	93.21 (0.240)	67.30 (0.137)	73.02 (0.231)	74.03 (0.334)
AdamP	91.68 (0.162)	95.18 (0.116)	95.17 (0.079)	69.41 (0.297)	76.14 (0.347)	77.58 (0.091)
AdaBound	91.81 (0.272)	94.83 (0.131)	95.05 (0.176)	68.61 (0.312)	76.27 (0.256)	77.56 (0.120)
AMSGrad	91.24 (0.115)	93.76 (0.108)	93.74 (0.236)	67.71 (0.291)	73.51 (0.692)	74.50 (0.416)
RMSProp	90.82 (0.201)	93.06 (0.120)	92.89 (0.310)	65.45 (0.394)	71.79 (0.287)	71.75 (0.632)

Assumptions

Assumption 1

There exists positive constant L_1, ρ and $q \geq 1$ such that

$$|G(\theta, x) - G(\theta', x)| \leq L_1(1 + |x|)^{\rho}(1 + |\theta| + |\theta'|)^{q-1}|\theta - \theta'|.$$

for all $x \in \mathbb{R}^m$ and $\theta, \theta' \in \mathbb{R}^d$. Moreover, $g(\theta) := \mathbb{E}[G(\theta, X_0)]$ and $h(\theta) := \mathbb{E}[H(\theta, X_0)]$ for every $\theta \in \mathbb{R}^d$.

Moreover, we impose conditions on the initial value θ_0 and on the data process $(X_n)_{n\in\mathbb{N}}$.

Assumption 2

The process $(X_n)_{n\in\mathbb{N}}$ is a sequence of i.i.d. random variables with $\mathbb{E}[X_0]^{16\rho(2r+1)} < \infty$ where ρ is given in Assumption 1. In addition, the initial condition is such that $\mathbb{E}[\theta_0]^{16(2r+1)} < \infty$.

Expected excess risk result

$$\mathbb{E}[u(\theta_n^\lambda)] - u(\theta^*) = \underbrace{\mathbb{E}\left[u\left(\theta_n^\lambda\right)\right] - \mathbb{E}\left[u\left(\theta_\infty\right)\right]}_{\mathcal{T}_1} + \underbrace{\mathbb{E}\left[u\left(\theta_\infty\right)\right] - u_\star}_{\mathcal{T}_2}$$

$$\leq \underbrace{C W_2(\mathcal{L}(\theta_n^\lambda), \pi_\beta)}_{\mathcal{T}_1} + \mathcal{T}_2,$$

Preliminary result

Proposition 1

([1]) Let Assumptions 1 and 2 hold. Then, for every $\theta, \theta' \in \mathbb{R}^d$,

$$\langle \theta - \theta', h(\theta) - h(\theta') \rangle \geq -a|\theta - \theta'|^2,$$

where $a = L_1 \mathbb{E}[(1 + |X_0|)^\rho](1 + 2|R|)^{q-1}$ *and* R *is given by*

$$R = \max\left\{ \left(\frac{2^{3(q-1)+1} L_1 \mathbb{E}[(1 + |X_0|)^\rho]}{\eta} \right)^{\frac{1}{2r-1}}, \left(\frac{2^q L_1 \mathbb{E}[(1 + |X_0|)^\rho]}{\eta} \right)^{\frac{1}{2r}} \right\}.$$

For each $m \geq 1$, define the Lyapunov function V_m by

$$V_m(\theta) := (1 + |\theta|^2)^{\frac{m}{2}}, \quad \theta \in \mathbb{R}^d \tag{9}$$

and similarly $v_m(x) = (1 + x^2)^{\frac{m}{2}}$ for any real $x \geq 0$.

Let $T := 1/\lambda$. Then, for any $n \in \mathbb{N}$, there exists a unique integer m such as $n \in [mT, (m+1)T)$.

Theorem 1

Let Assumptions 1 and 2 hold. Then, there exist constants C_1, C_2, C_3, $\hat{c}$, $\dot{c}$ and z_1 such that, for every $0 < \lambda \leq \lambda_{\max}$ and $n \in \mathbb{N}$,

$$W_1\left(\mathcal{L}\left(\theta_n^\lambda\right), \pi_\beta\right) \leq \sqrt{\lambda}\left(z_1 + \sqrt{e^{3a}(C_1 + C_2 + C_3)}\right) + \hat{c}e^{-\dot{c}m}\left[1 + \mathbb{E}\left[V_2\left(\theta_0\right)\right] + \int_{\mathbb{R}^d} V_2(\theta)\pi_\beta(d\theta)\right],$$

where V_2 is defined in (9) and a is defined in Proposition 1. The explicit form of C_1, C_2, C_3, $\hat{c}$, $\dot{c}$ and z_1 are given in Table 2 (in the article).

Corollary 2

Let Assumptions 1 and 2 hold. Then, there exists a constant z_2 such that, for every $0 < \lambda \leq \lambda_{max}$ and $n \in \mathbb{N}$,

$$W_2\left(\mathcal{L}\left(\theta_n^\lambda\right), \pi_\beta\right) \leq \sqrt{e^{3a}(C_1 + C_2 + C_3)}\sqrt{\lambda} + z_2\lambda^{\frac{1}{4}} + \sqrt{2\hat{c}e^{-\dot{c}m}\left(1 + \mathbb{E}\left[V_2\left(\theta_0\right)\right] + \int_{\mathbb{R}^d} V_2(\theta)\pi_\beta(d\theta)\right)},$$

where V_2 is defined in (9). The explicit form of z_2 is given in Table 2 (in the article).

Theorem 3

Let Assumptions 1 and 2 hold and $\beta \geq \frac{2}{A}$. For any $n \in \mathbb{N}$, the expected excess risk of the n-th iterate of THεO POULA (7) is upper bounded by

$$\begin{aligned}\mathbb{E}[u(\theta_n^\lambda)] - u(\theta^*) \quad &\leq \quad \left(\frac{a_1}{l+1}\sqrt{\mathbb{E}|\theta_0|^{2l} + \frac{A_l}{\eta^2}} + \frac{a_1}{l+1}\sqrt{\mathbb{E}|\theta_\infty|^{2l}} + 2\mathbb{E}[K(X_0)]\right) W_2(\mathcal{L}(\theta_n^\lambda, \pi_\beta)) \\ &+ \quad \frac{1}{\beta}\left[\frac{d}{2}\log\left(\frac{eK}{A}\left(\frac{B\beta}{d}+1\right)\right) - \log\left(1 - e^{-(R_0\sqrt{K\beta}-\sqrt{d})^2}\right)\right]\end{aligned}$$

where $l = 2r + 1$, $W_2(\mathcal{L}(\theta_n^\lambda), \pi_\beta)$ is given in Corollary 2 and a_1, K, A_l, R_0 are given in Table 2 (in the article).

Lim, D.-Y., and Sabanis, S.: Polygonal Unadjusted Langevin Algorithms: Creating stable and efficient adaptive algorithms for neural networks. *Preprint*, 2021. arXiv:2105.13937

Discontinuity

Let $H : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ be an unbiased estimator of h, i.e., $h(\theta) = \mathbb{E}[H(\theta, X_0)]$, for all $\theta \in \mathbb{R}^d$, which takes the following form: for all $\theta \in \mathbb{R}^d, x \in \mathbb{R}^m$,

$$H(\theta, x) := G(\theta, x) + F(\theta, x), \tag{10}$$

where $G = (G^{(1)}, \ldots, G^{(d)}) : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ and $F = (F^{(1)}, \ldots, F^{(d)}) : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ are Borel measurable functions.

Remark: We consider H taking the form of (10) with G containing discontinuities and F being locally Lipschitz continuous as it is satisfied by a wide range of real-world applications including *quantile estimation*, *vector quantization*, *CVaR minimization*, and *regularized optimization problems involving ReLU neural networks*.

Algorithm We define

$$\theta_0^\lambda := \theta_0, \quad \theta_{n+1}^\lambda := \theta_n^\lambda - \lambda H_\lambda(\theta_n^\lambda, X_{n+1}) + \sqrt{2\lambda\beta^{-1}}\xi_{n+1}, \quad n \in \mathbb{N}_0, \tag{11}$$

where $\lambda > 0$ is the stepsize, $\beta > 0$ is the inverse temperature parameter, and where $H_\lambda(\theta, x)$ is defined, for all $\theta \in \mathbb{R}^d, x \in \mathbb{R}^m$, by

$$H_\lambda(\theta, x) := G_\lambda(\theta, x) + F_\lambda(\theta, x), \tag{12}$$

with $G_\lambda(\theta, x) = (G_\lambda^{(1)}(\theta, x), \ldots, G_\lambda^{(d)}(\theta, x))$
and $F_\lambda(\theta, x) = (F_\lambda^{(1)}(\theta, x), \ldots, F_\lambda^{(d)}(\theta, x))$ given by

$$\begin{aligned} G_\lambda^{(i)}(\theta, x) &:= \frac{G^{(i)}(\theta, x)}{1 + \sqrt{\lambda}|G^{(i)}(\theta, x)|}\left(1 + \frac{\sqrt{\lambda}}{\varepsilon + |G^{(i)}(\theta, x)|}\right), \\ F_\lambda^{(i)}(\theta, x) &:= \frac{F^{(i)}(\theta, x)}{1 + \sqrt{\lambda}|\theta|^{2r}}, \end{aligned} \tag{13}$$

for any $i = 1, \ldots, d$ with fixed $0 < \varepsilon < 1$, $r > 0$.

Let $q \in [1, \infty)$ and $\rho \in [1, \infty)$ be fixed.
Note that G satisfies a "continuity in average" condition.

Assumption 3

There exists a constant $L_G > 0$ such that, for all $\theta, \bar{\theta} \in \mathbb{R}^d$,

$$\mathbb{E}[|G(\theta, X_0) - G(\bar{\theta}, X_0)|] \leq L_G(1 + |\theta| + |\bar{\theta}|)^{q-1}|\theta - \bar{\theta}|.$$

In addition, there exists a constant $K_G > 1$, such that for all $\theta \in \mathbb{R}^d, x \in \mathbb{R}^m$,

$$|G(\theta, x)| \leq K_G(1 + |x|)^{\rho}(1 + |\theta|)^{q}.$$


Lim, D.-Y., Neufeld, A., Sabanis, S. and Zhang, Y.: PLangevin dynamics based algorithm e-THϵO POULA for stochastic optimization problems with discontinuous stochastic gradient. *Preprint*, 2022. arXiv:2210.13193

Thank you!