Stochastic grandient descent and Langevin-simulated annealing algorithms

Pierre BRAS

Sorbonne Université

May 24, 2022

Optimization problem

Let $V : \mathbb{R}^d \to \mathbb{R}$ be $\mathcal{C}^1$, coercive (i.e. $V(x) \to +\infty$ as $|x| \to \infty$) and let $\text{argmin}(V) := \{x \in \mathbb{R}^d : \ V(x) = \min_{\mathbb{R}^d} V\}$.

Objective : find $\text{argmin}(V)$.

Example : Regression as an optimization problem

– Data $(u_i, v_i)_{1\leq i\leq N}$ with N large; we want to find some function Φ which can predict v from u i.e. such that for all i, $\Phi(u_i) \approx v_i$ i.e. such that

$$\frac{1}{N}\sum_{i=1}^{N} |\Phi(u_i) - v_i|^2 \quad \text{is small.}$$

– We reduce to a finite-dimensional problem: Φ is parametrized by a finite-dimensional parameter: $\{\phi_x,\ x \in \mathbb{R}^d\}$.
– A good choice of family of functions is neural functions thanks to their good approximation properties:

Neural functions

$$\Phi_x(u) = \varphi_{\alpha_R,\beta_R} \circ \ldots \circ \varphi_{\alpha_1,\beta_1}(u), \qquad \alpha_k \in \mathcal{M}_{d_k,d_{k-1}}(\mathbb{R}),\ \ \beta_k \in \mathbb{R}^{d_k},$$

$$\varphi_{\alpha_k,\beta_k} : \mathbb{R}^{d_{k-1}} \to \mathbb{R}^{d_k}, \qquad u \mapsto \varphi(\alpha_k \cdot u + \beta_k)$$

where $\varphi : \mathbb{R} \to \mathbb{R}$ is a non-linear function, applied coordinate by coordinate and where the parameter $x = (\alpha_1, \beta_1, \ldots, \alpha_R, \beta_R)$.
– The objective becomes

$$\min_{x\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^{N} |\Phi_x(u_i) - v_i|^2 =: V(x).$$

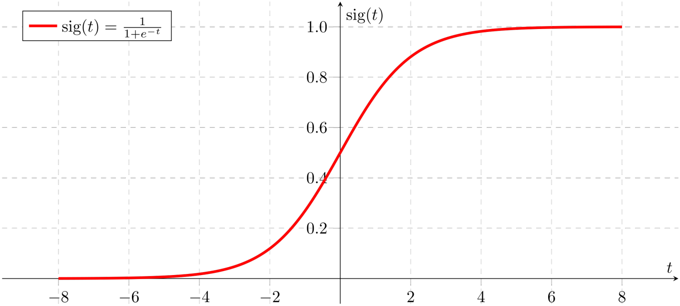


Figure: The sigmoid function

• Gradient descent algorithm : compute the gradient and "go down" the gradient with decreasing step sequence (γ_k):

$$x_0 \in \mathbb{R}^d$$
$$x_{n+1} = x_n - \gamma_{n+1} \nabla V(x_n).$$

• The continuous version is $dX_s = -\nabla V(X_s) ds$.
• With a a data regression problem, this would give

$$x_{n+1} = x_n - \gamma_{n+1} \sum_{i=1}^{N} \nabla_x \left(|\Phi_x(u_i) - v_i|^2 \right),$$

implying to compute all the gradients over the dataset at every iteration n. Instead we do the Stochastic Gradient Descent (SGD) algorithm:

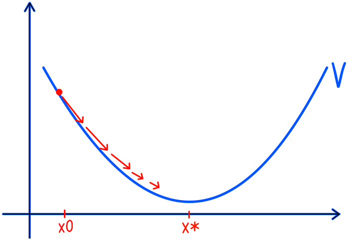
$$x_{n+1} = x_n - \gamma_{n+1} \nabla_x \left(|\Phi_x(u_{i_{n+1}}) - v_{i_{n+1}}|^2 \right),$$

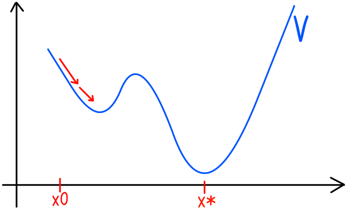
where i_{n+1} is chosen uniformly at random at every iteration.

We replace:

$$x_{n+1} = x_n - \gamma_{n+1} \left(\nabla V(x_n) + \zeta_{n+1} \right),$$

where $\mathbb{E}[\zeta_{n+1} | x_n] = 0$ (martingale increments).





- **Problem** : x_n can be "trapped" !

- We add a white noise to x_n, hoping to escape traps :

$$x_{n+1} = x_n - \gamma_{n+1} \left(\nabla V(x_n) + \zeta_{n+1} \right) + \sqrt{\gamma_{n+1}} \sigma \xi_{n+1}, \quad \xi_{n+1} \sim \mathcal{N}(0, I_d).$$

$\implies$ called SGLD algorithms (Stochastic Gradient Langevin Dynamics)

- The continuous version becomes:

$$dX_s = -\nabla V(X_s) ds + \sigma dW_s \qquad \textbf{(Langevin Equation)}$$

where (W_s) is a Brownian motion and $\sigma > 0$.

- It is invariant measure is the **Gibbs measure**

$$\nu_\sigma(x) dx = C_\sigma e^{-2V(x)/\sigma^2} dx, \quad C_\sigma := \left(\int_{\mathbb{R}^d} e^{-2V(x)/\sigma^2} dx \right)^{-1}.$$

- Exogenous noise σdW_t added to escape local minima ('traps') and explore the state space.
- For small σ, ν_σ is concentrated around argmin(V):

Solve the Langevin equation $\implies$ approximation of ν_σ $\implies$ approximation of argmin(V).

- We have $\nu_\sigma \underset{\sigma\to 0}{\longrightarrow} \text{argmin}(V)$ in law.
- One possibility : solve the Langevin equation for small σ
- Another possibility : make $\sigma \to 0$ while iterating the algorithm :

$$x_{n+1} = x_n - \gamma_{n+1}\nabla V(x_n) + a(\gamma_1 + \cdots + \gamma_{n+1})\sigma\sqrt{\gamma_{n+1}}\xi_{n+1}, \quad \xi_{n+1} \sim \mathcal{N}(0, I_d),$$

where $a(t)$ is decreasing and $a(t) \underset{t\to 0}{\longrightarrow} 0$.

The continuous version becomes :

Langevin-Simulated Annealing Equation

$$dX_t = -\nabla V(X_t)dt + a(t)\sigma dW_t,$$

- The 'instantaneous' invariant measure $\nu_{a(t)\sigma}(dx) \propto \exp\left(-2V(x)/(a^2(t)\sigma^2)\right)$ converges itself to argmin(V)
- Schedule $a(t) = A\log^{-1/2}(t)$ then $X_t \underset{t\to\infty}{\longrightarrow} \text{argmin}(V)$ in law [Chiang-Hwang 1987], [Miclo 1992]

Multiplicative noise

- Noise $\sigma > 0 \implies$ isotropic, homogeneous noise $\implies$ not adapted to V
- Instead : $\sigma(X_t) \in \mathcal{M}_d(\mathbb{R})$ is a matrix depending on the position
- In Machine Learning literature, a good choice is $\sigma(x)\sigma(x)^\top \simeq (\nabla^2 V(x))^{-1}$ as in the Newton algorithm.
- SGLD often used in ML literature, but no general theoretical guarantee of convergence.

$$dY_t = -(\sigma\sigma^\top \nabla V)(Y_t)dt + a(t)\sigma(Y_t)dW_t + \underbrace{\left(a^2(t)\left[\sum_{j=1}^{d} \partial_i(\sigma\sigma^\top)(Y_t)_{ij}\right]_{1\leq i\leq d}\right)}_{\text{correction term } a^2(t)\Upsilon(Y_t)} dt$$

$$a(t) = \frac{A}{\sqrt{\log(t)}},$$

• Correction term so that $\nu_{a(t)} \propto \exp\left(-2V(x)/a^2(t)\right)$ is still the "instantaneous" invariant measure

- We proved the convergence of Y_t and $\bar{Y}_t$ to $\nu^\star = \delta_{\text{argmin}(V)}$ for the L^1-Wasserstein distance, where $\bar{Y}$ is the discretization of Y:

$$\bar{Y}_{t_{k+1}} = \bar{Y}_{t_k} + \gamma_{k+1}\left(-\sigma\sigma^\top \nabla V(\bar{Y}_{t_{k+1}}) + a^2(t)\Upsilon(\bar{Y}_{t_k}) + \zeta_{k+1}\right) + a(t_{k+1})\sigma(\bar{Y}_{t_{k+1}})\sqrt{\gamma_{k+1}}\xi_{k+1},$$

$\xi_{k+1} \sim \mathcal{N}(0, I_d)$.

- We use the L^1-Wasserstein distance:

$$\mathcal{W}_1(\pi_1, \pi_2) = \sup\left\{\int_{\mathbb{R}^d} f(x)(\pi_1 - \pi_2)(dx) : \ f : \mathbb{R}^d \to \mathbb{R},\ [f]_{\text{Lip}} = 1\right\}.$$

and we show that $\mathcal{W}_1(Y_t, \nu^\star) \to 0$ and $\mathcal{W}_1(\bar{Y}_t, \nu^\star) \to 0$. We have

$$\mathcal{W}_1(Y_t, \nu^\star) \le \mathcal{W}_1(Y_t, \nu_{a(t)}) + \mathcal{W}_1(\nu_{a(t)}, \nu^\star)$$

The convergence is limited by the slowness of $a(t)$ as $\mathcal{W}_1(\nu_{a(t)}, \nu^\star) \asymp a(t) \asymp \log^{-1/2}(t)$. In fact we also prove for every $\alpha \in (0,1)$:

$$\mathcal{W}_1(Y_t^{x_0}, \nu_{a(t)}) \le C_\alpha \max(1 + |x_0|, V(X_0))t^{-\alpha}$$
$$\mathcal{W}_1(\bar{Y}_t^{x_0}, \nu_{a(t)}) \le C_\alpha \max(1 + |x_0|, V^2(X_0))t^{-\alpha}.$$

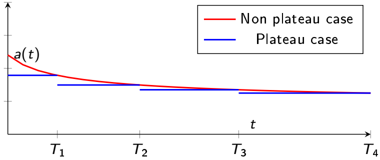
Assumptions:

1. V is strongly convex outside some compact set and ∇V is Lipschitz
2. σ is bounded and elliptic: $\sigma\sigma^\top \ge \sigma_0 I_d$, $\sigma_0 > 0$.
3. Decreasing steps (γ_n) for the Euler scheme, with $\sum_n \gamma_n = \infty$, $\sum_n \gamma_n^2 < \infty$, $\Gamma_n := \gamma_1 + \cdots + \gamma_n$.

• To apply ergodicity properties, we require σ to be elliptic however the ellipticity of $a(t)\sigma(Y_t) \longrightarrow 0$ as $t \to \infty$.
• Instead, we consider the plateau SDE where a is piecewise constant:

$$dX_t = -\sigma\sigma^\top \nabla V(X_t)dt + a_{n+1}\sigma(X_t)dW_t + a_{n+1}^2 \Upsilon(X_t)dt, \quad t \in [T_n, T_{n+1}),$$
$$a_n = A \log^{-1/2}(T_n)$$

And we apply the ergodicity properties on each plateau, giving a recurrence relation.
• In the proof, we investigate the dependence in a_n and the factor $e^{-\rho_{a_n}(T_n - T_{n-1})}$, $\rho_{a_n} = e^{-C_2/a_n^2}$ appears, so we need to choose $a_n = A \log^{-1/2}(T_n)$.



Thank you for your attention !