Stochastic Gradient Descent

Optimization for Machine Learning — Exercise #4

Monday 22nd May, 2023

Part I: Theory

A nice reference on Stochastic Gradient Descent is [1]. We will see some highlights from its §4.

I.1. Useful inequalities

Exercise I.1 (Inequality of L-smooth functions)**.** Recall that, given $L > 0$, a function $E\colon \mathbb{R}^d \supseteq \Omega \to \mathbb{R}$ is called L-smooth if

$$\forall (x, y) \in \Omega^2, \quad \|\nabla E(x) - \nabla E(y)\| \leqslant L\|x - y\|,$$

i.e. if ∇E is L-Lipschitz.

Show that if E is L-smooth, then

$$\forall (x, y) \in \Omega^2, \quad E(y) \leqslant E(x) + \langle \nabla E(x), y - x\rangle + \frac{L}{2}\|y - x\|_2^2$$

Hint: Express $E(y)$ with the integral formula $E(y) = E(x) + \int_0^1 \frac{\partial}{\partial t} E(x + t(y - x))\,\mathrm{d}t$.

Exercise I.2 (Inequality of c-strongly convex functions)**.** Recall that, given $c > 0$, a function $E\colon \mathbb{R}^d \supseteq \Omega \to \mathbb{R}$ is called c-strongly convex if Ω is convex and

$$\forall (x, y) \in \Omega^2, \quad E(y) \geqslant E(x) + \langle \nabla E(x), y - x\rangle + \frac{c}{2}\|y - x\|_4^2$$

A strongly convex function has a unique minimizer $x_* = \operatorname{argmin}_{x \in \Omega} E(x)$.

Show that if E is c-strongly convex, it has a unique minimizer x_* so that, with $E_* := E(x_*)$,

$$\forall y \in \Omega, \quad 2c(E(y) - E_*) \leqslant \|\nabla E(y)\|_2^2$$

Hint: Study the function $q(y) = E(x) + \langle \nabla E(x), y - x\rangle + \frac{c}{2}\|y - x\|_2^2$.

I.2. Convergence of SGD

Notation We are concerned with learning a supervised task on $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$. The parameter space is $\mathcal{W} \subseteq \mathbb{R}^d$. Let $h\colon \mathcal{X} \times \mathcal{W} \to \mathcal{Y}$ be the *prediction function*, and $\ell\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ the *loss function*. Denote by $f\colon \mathcal{W} \times \mathcal{X} \times \mathcal{Y}$ the composition of ℓ and h.

With ξ a random variable selecting samples from $\mathcal{X} \times \mathcal{Y}$, the **expected risk** R can be written as $R(w) = \mathbb{E}_\xi[f(w;\xi)]$. The **empirical risk** R_n can be obtained when ξ takes n realizations $\{\xi(i)\}_{i\in[n]}$ corresponding to n training samples $\{(x_i, y_i)\}_{i\in[n]}$. Denoting $f_i(w) := f(w, \xi(i))$, one has $R_n(w) = \frac{1}{n}\sum_{i=1}^n f_i(w)$.

The objective function $F\colon \mathbb{R}^d \to \mathbb{R}$ is either

$$F(w) = \begin{cases} R(w) \\ \text{or} \\ R_n(w) \end{cases}$$

We assume to be able to compute the realization of a random variable ξ_k. Given an iteration w_k and the realization of ξ_k, we assume to be able to compute a stochastic vector $g(w_k, \xi_k) \in \mathbb{R}^d$ (the stochastic gradient).

Algorithm 1 Stochastic Gradient Descent algorithm [1, Algorithm 4.1].

Choose an initial iterate w_1
for $k = 1, 2, \ldots$ **do**
 Generate a realization of the random variable ξ_k
 Compute a stochastic vector $g(w_k, \xi_k)$
 Choose a step size $\alpha_k > 0$
 Set the new iterate as $w_{k+1} \leftarrow w_k - \alpha_k g(w_k, \xi_k)$.

Because of the stochastic nature of $g(w_k, \xi_k)$, we are not assured to decrease the objective function at every step. But, we can carry an expectation analysis and show that, in expectation (over ξ_k), we do make progress in the minimization problem.

We first require the objective F to be L-smooth.

Assumption 1 (F is L-smooth). There exists a constant $L > 0$ such that F is L-smooth.

Exercise I.3 (Descent update with L-smooth function)**.** Under Assumption 1, show that the iterates of SGD (Algorithm 1) satisfy the following inequality for all $k \in \mathbb{N}$:

$$\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) \leqslant -\alpha_k \langle \nabla F(w_k), \mathbb{E}_{\xi_k}[g(w_k, \xi_k)]\rangle + \frac{\alpha_k^2 L}{2}\mathbb{E}_{\xi_k}[\|g(w_k, \xi_k)\|_2^2]$$

We need some more assumptions on the stochastic estimation $g(w_k, \xi_k)$ in order to control $\mathbb{E}_{\xi_k}[\|g(w_k, \xi_k)\|^2]$. More specifically, we require bounding the first and second moment of $g(w_k, \xi_k)$ like so:

Assumption 2. 1. There exist scalars $\mu_G \geqslant \mu > 0$ such that, for all $k \in \mathbb{N}$,

$$\begin{aligned} \langle \nabla F(w_k), \mathbb{E}_{\xi_k}[g(w_k, \xi_k)]\rangle &\geqslant \mu \|\nabla F(w_k)\|_2^2 \\ \|\mathbb{E}_{\xi_k}[g(w_k, \xi_k)]\| &\leqslant \mu_G \|\nabla F(w_k)\|_2 \end{aligned}$$

2. The second moment of g is bounded: there exist $M, M_G \geqslant 0$, such that

$$\mathbb{E}_{\xi_k}[\|g(w_k, \xi_k)\|_2^2] \leqslant M + M_G \|\nabla F(w_k)\|_2^2$$

Exercise I.4. Show that, under Assumptions 1 and 2, the iterates of SGD in Algorithm 1 satisfy the following inequalities for all $k \in \mathbb{N}$:

$$\begin{aligned}\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) &\leqslant -\mu\alpha_k \|\nabla F(w_k)\|_2^2 + \frac{1}{2}\alpha_k^2 L \mathbb{E}_{\xi_k}[\|g(w_k, \xi_k)\|_2^2] \\ &\leqslant -(\mu - \frac{1}{2}\alpha_k L M_G)\alpha_k \|\nabla F(w_k)\|_2^2 + \frac{1}{2}\alpha_k^2 L M\end{aligned}$$

A last assumption is helpful to make: strong convexity of the objective function F. This will allow to obtain a linear rate of convergence to a neighbourhood of a solution, if the step size is not too big.

Assumption 3 (F is c-strongly convex). There exists a constant $c > 0$ such that F is c-strongly convex.

Exercise I.5. Under Assumptions 1 to 3, suppose that Algorithm 1 is run with a fixed step size $\alpha_k = \bar{\alpha}$ for all $k \in \mathbb{N}$, satisfying

$$0 < \bar{\alpha} \leqslant \frac{\mu}{L M_G}.$$

Denote $F_* = \min_w F(w)$ (exists and is unique by Assumption 3). Show that the expected optimality gap satisfies the following inequality for all $k \in \mathbb{N}$:

$$\begin{aligned}\mathbb{E}[F(w_k) - F_*] &\leqslant \frac{\bar{\alpha} L M}{2c\mu} + (1 - \bar{\alpha} c\mu)^{k-1} \left(F(w_1) - F_* - \frac{\bar{\alpha} L M}{2c\mu} \right) \\ &\xrightarrow[k \to \infty]{} \frac{\bar{\alpha} L M}{2c\mu}.\end{aligned}$$

References

[1] Léon Bottou, Frank E. Curtis and Jorge Nocedal. "Optimization Methods for Large-Scale Machine Learning". 2016. DOI: `10.1137/16M1080173`. arXiv: `1606.04838`.