CMPT 409/981: Optimization for Machine Learning

Lecture 2

Sharan Vaswani

September 10, 2024

Recap

- Machine learning tasks involve optimizing some (potentially complicated) function of the model parameters.
- Minimizing generic functions is hard, and we need to make assumptions on the structure.
- **Lipschitz continuous functions**: f is G-Lipschitz continuous if $\forall x, y \in \mathcal{D}$, $|f(x) - f(y)| \leq G \, \|x - y\|$.
- Global minimization of Lipschitz continuous functions using a *zero-order oracle* requires $\Omega\left(\left(\frac{G}{\epsilon}\right)^d\right)$ oracle calls. The naive algorithm of forming an ϵ-net is near-optimal.
- **Smooth functions**: f is L-smooth if its gradient is Lipschitz continuous i.e. $\forall x, y \in \mathcal{D}$, $\|\nabla f(x) - \nabla f(y)\| \leq L \, \|x - y\|$.
- If f is twice-differentiable and L-smooth, $\nabla^2 f(w) \preceq L \, I_d$.
- For linear regression, $f(w) = \frac{1}{2} \|Xw - y\|^2 = \sum_{i=n} (\langle x_i, w \rangle - y_i)^2$ is $\lambda_{\max}[X^T X]$-smooth.

Smooth functions

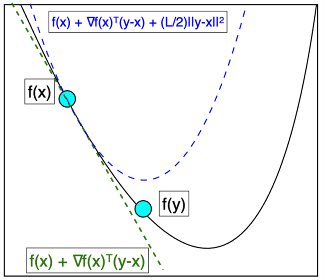
Claim: For an L-smooth function, $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$ for all $x, y \in \mathcal{D}$.

Proof:

$$
\begin{aligned}
f(y) &= f(x) + \int_{t=0}^{1} \left[\nabla f(x + t(y - x))\right] (y - x)^\intercal \, dt && \text{(Fundamental theorem of calculus)} \\
&= f(x) + \langle \nabla f(x), y - x \rangle + \int_{t=0}^{1} \left[\nabla f(x + t(y - x))\right] (y - x)^\intercal \, dt - \left[\nabla f(x)\right](y - x)^\intercal \\
&= f(x) + \langle \nabla f(x), y - x \rangle + \int_{t=0}^{1} \left[\nabla f(x + t(y - x)) - \nabla f(x)\right] (y - x)^\intercal \, dt \\
&\leq f(x) + \langle \nabla f(x), y - x \rangle + \int_{t=0}^{1} \|\nabla f(x + t(y - x)) - \nabla f(x)\| \, \|y - x\| \, dt \\
& && \text{(Cauchy–Schwarz)} \\
&\leq f(x) + \langle \nabla f(x), y - x \rangle + L \int_{t=0}^{1} \|x + t(y - x) - x\| \, \|y - x\| \, dt && \text{(Lipschitz continuity)} \\
&= f(x) + \langle \nabla f(x), y - x \rangle + L \, \|y - x\|^2 \int_{t=0}^{1} t \, dt = f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2
\end{aligned}
$$

Smooth functions

The inequality $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$ can be interpreted as a *global* quadratic upper-bound on f at point x i.e. the bound holds for all $y \in \mathcal{D}$.



There are other related ways to state the L-smoothness of f (prove these in Assignment 1).

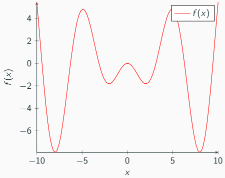
$$f(y) \geq f(x) + \langle f(x), y - x \rangle + \frac{1}{2L} \|\nabla f(y) - \nabla f(x)\|^2$$

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \leq L \|x - y\|^2$$

Questions?

Local Minimization

Smooth functions can include functions with multiple local/global minimum and stationary points. Eg: $f(x) = -x\sin(x)$.



Consider minimizing a smooth function over $\mathbb{R}^d$ (unconstrained minimization)

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

Since we have seen that global minimization can be impossible (without Lipschitz assumption on f) or the number of oracle calls can be exponential in d, let us aim to solve an easier problem.

- Access to a **first-order oracle** – query the oracle at point w and it returns $f(w)$ and $\nabla f(w)$.
- **Objective**: For a target accuracy of $\epsilon > 0$, return a point $\hat{w}$ s.t. $\|\nabla f(\hat{w})\|^2 \leq \epsilon$? Characterize the required number of oracle calls.

We only care about making the gradient small and finding an approximate stationary point.

Local Minimization

Recall that L-smoothness of f implies that $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$.

Idea: Since the RHS is a global upper-bound on the true function, instead of minimizing the function directly, let us minimize the upper-bound at x w.r.t y.

Setting the gradient of the RHS w.r.t y to zero, we obtain $\hat{y}$ as:

$$\nabla f(x) + L [\hat{y} - x] = 0 \implies \hat{y} = x - \frac{1}{L} \nabla f(x)$$

This is exactly the gradient descent update at x!

We can do this iteratively i.e. starting at w_0, form the upper-bound at w_0, minimize it by setting $w_1 = w_0 - \frac{1}{L} \nabla f(w_0)$, then form the quadratic upper-bound at w_1 and repeat. Continue to do this until we find a point $\hat{w}$ s.t. $\|\nabla f(\hat{w})\|^2 \leq \epsilon$ and terminate.

This is exactly the gradient descent procedure – move in the direction of the negative gradient ("downhill") with *step-size* η equal to to $1/L$. Formally, at iteration k, the GD update is:

$$w_{k+1} = w_k - \eta \nabla f(w_k).$$

Gradient Descent

Is GD guaranteed to terminate? If so, can we characterize the number of iterations?

Claim: For L-smooth functions lower-bounded by f^*, gradient descent with $\eta = \frac{1}{L}$ returns $\hat{w}$ such that $\|\nabla f(\hat{w})\|^2 \leq \epsilon$ and requires $T = \frac{2L\,[f(w_0) - f^*]}{\epsilon}$ iterations (oracle calls).

Proof:
Using the L-smoothness of f with $x = w_k$ and $y = w_{k+1} = w_k - \frac{1}{L}\nabla f(w_k)$ in the quadratic bound (also referred to as the *descent lemma*),

$$
\begin{aligned}
f(w_{k+1}) &\leq f(w_k) + \langle \nabla f(w_k), -\frac{1}{L}\nabla f(w_k)\rangle + \frac{L}{2}\left\|\frac{1}{L}\nabla f(w_k)\right\|^2 \\
\implies f(w_{k+1}) &\leq f(w_k) - \frac{1}{2L}\|\nabla f(w_k)\|^2
\end{aligned}
$$

By moving from w_k to w_{k+1}, we have decreased the value of f since $f(w_{k+1}) \leq f(w_k)$.

Gradient Descent

Rearranging the inequality from the previous slide, for every iteration k,

$$\frac{1}{2L} \|\nabla f(w_k)\|^2 \leq f(w_k) - f(w_{k+1})$$

By running GD for T iterations, adding up $k = 0$ to $T - 1$,

$$\frac{1}{2L} \sum_{k=0}^{T-1} \|\nabla f(w_k)\|^2 \leq \sum_{k=0}^{T-1} [f(w_k) - f(w_{k+1})] = f(w_0) - f(w_T) \leq [f(w_0) - f^*]$$

(Since f is lower-bound by f^*)

$$\implies \frac{\sum_{k=0}^{T-1} \|\nabla f(w_k)\|^2}{T} \leq \frac{2L\,[f(w_0) - f^*]}{T}$$

The LHS is the average of the gradient norms over the T iterates. Let $\hat{w} := \arg\min_{k \in \{0,1,\ldots,T-1\}} \|\nabla f(w_k)\|^2$. Since the minimum is smaller than the average,

$$\|\nabla f(\hat{w})\|^2 \leq \frac{2L\,[f(w_0) - f^*]}{T}$$

Gradient Descent

Since $\|\nabla f(\hat{w})\|^2 \leq \frac{2L\,[f(w_0)-f^*]}{T}$, the *rate of convergence* is $O(1/T)$.

If the RHS equal to $\frac{2L\,[f(w_0)-f^*]}{T} \leq \epsilon$, this would guarantee that $\|\nabla f(\hat{w})\|^2 \leq \epsilon$ and we would achieve our objective.

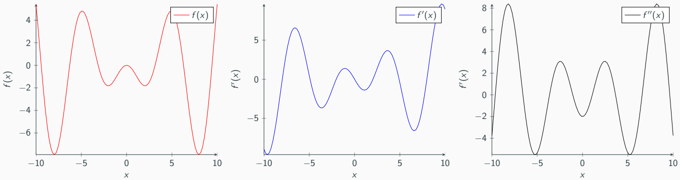
Hence, we need to run the algorithm for $T \geq \frac{2L\,[f(w_0)-f^*]}{\epsilon}$ iterations. This is also referred to as an $O\left(\frac{1}{\epsilon}\right)$ convergence rate.

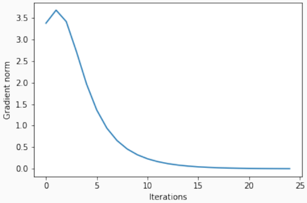
Lower-Bound: When minimizing a smooth function (without additional assumptions), any *first-order* algorithm requires $\Omega\left(\frac{1}{\epsilon}\right)$ oracle calls to return a point $\hat{w}$ such that $\|\nabla f(\hat{w})\|^2 \leq \epsilon$.

Hence, gradient descent is optimal for minimizing smooth functions!

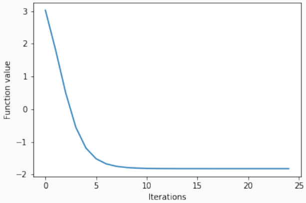
Gradient Descent – Example

$\min_{x\in[-10,10]} f(x) := -x\sin(x)$. Run GD with $\eta = 1/L \approx 0.1$ and $x_0 = 4$.





(a) Gradient norm



(b) Function value

Questions?