CMPT 409/981: Optimization for Machine Learning

Lecture 2

Sharan Vaswani

September 12, 2022

Recap

Smooth functions: f is L-smooth if its gradient is Lipschitz continuous, and does not change arbitrarily fast i.e. $\forall x, y$, $\|\nabla f(x) - \nabla f(y)\| \leq L \, \|x - y\|$.

If f is L-smooth, then for all $x, y \in \mathcal{D}$, $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$.

Objective: Find an ϵ-approximate stationary point $\hat{w}$ i.e. $\|\nabla f(\hat{w})\|^2 \leq \epsilon$ with access to a *first-order oracle* that returns $\{f(w), \nabla f(w)\}$ at any point $w \in \mathcal{D}$.

Minimizing the above upper-bound iteratively recovers gradient descent (GD) with $\eta = 1/L$.

Starting from an *initialization* equal to w_0, at iteration k, GD computes the gradient $\nabla f(w_k)$ at iterate w_k (call to the first-order oracle).

- If $\|\nabla f(w_k)\|^2 \leq \epsilon$, terminate and return $\hat{w} := w_k$.
- Else, update the iterate as: $w_{k+1} = w_k - \frac{1}{L} \nabla f(w_k)$.

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.

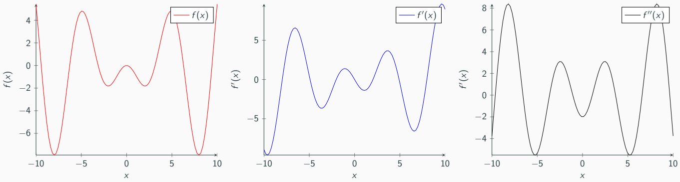
Hence, we need to run the algorithm for $T \geq \frac{2L\,[f(w_0) - f^*]}{\epsilon}$ iterations. This is also referred to as an $O\left(\frac{1}{\epsilon}\right)$ convergence rate.

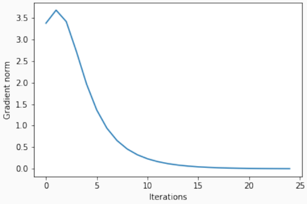
Lower-Bound: When minimizing a smooth function (without additional assumptions), any *first-order* algorithm requires $\Omega\left(\frac{1}{\epsilon}\right)$ oracle calls to return a point $\hat{w}$ such that $\|\nabla f(\hat{w})\|^2 \leq \epsilon$.

Hence, gradient descent is optimal for minimizing smooth functions!

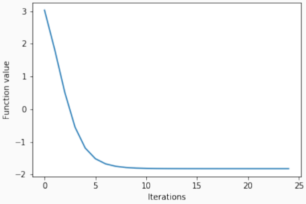
Gradient Descent – Example

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





(a) Gradient norm



(b) Function value

Questions?

Gradient Descent

We have seen that we can reach a stationary point of a smooth function in $O\left(\frac{1}{\epsilon}\right)$ iterations of GD with step-size $\eta = \frac{1}{L}$.

Problems with this approach:

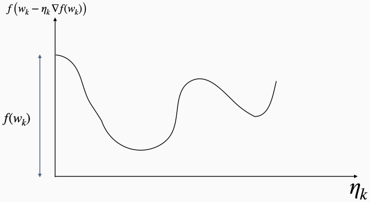
- Computing L in closed-form can be difficult as the functions get complicated.
- Theoretically computed L is global (the "local" L might be much smaller) and often loose in practice (typically we tend to overestimate L resulting in a smaller step-size).

Gradient Descent with Line-search

Instead of setting η according to L, we can "search" for a good step-size η_k in each iteration k.

Exact line-search: At iteration k, solve the following sub-problem:

$$\eta_k = \arg\min_{\eta} f(w_k - \eta \nabla f(w_k)).$$



After computing η_k, do the usual GD update: $w_{k+1} = w_k - \eta_k \nabla f(w_k)$.

- Can adapt to the "local" L, resulting in larger step-sizes and better performance.
- Can solve the sub-problem approximately by doing gradient descent w.r.t η (expensive).
- Can compute η_k analytically (only in special cases).

Gradient Descent with Line-search – Example

Recall linear regression: $\min_{w \in \mathbb{R}^d} f(w) := \frac{1}{2} \|Xw - y\|^2 = \frac{1}{2} [w^\top (X^\top X) w - 2 w^\top X^\top y + y^\top y]$.

For the exact line-search, we need to $\min_\eta h(\eta) := f(w_k - \eta \nabla f(w_k))$.

Since f is a quadratic, we can directly use the second-order Taylor series expansion.

$$
\begin{aligned}
h(\eta) &= f(w_k - \eta \nabla f(w_k)) \\
&= f(w_k) + \langle \nabla f(w_k), -\eta \nabla f(w_k) \rangle + \frac{1}{2} [-\eta \nabla f(w_k)]^\top \nabla^2 f(w_k) [-\eta \nabla f(w_k)]
\end{aligned}
$$

$$
\nabla h(\eta_k) = - \|\nabla f(w_k)\|^2 + \eta [\nabla f(w_k)]^\top \nabla^2 f(w_k) [\nabla f(w_k)] = 0 \implies \eta_k = \frac{\|\nabla f(w_k)\|^2}{\|\nabla f(w_k)\|^2_{\nabla^2 f(w_k)}}
$$

For linear regression, $\nabla^2 f(w_k) = X^\top X$ and $\nabla f(w_k) = X^\top (X w_k - y)$. With exact line-search, the GD update for linear regression is:

$$
w_{k+1} = w_k - \frac{\|X^\top (X w_k - y)\|^2}{\|X^\top (X w_k - y)\|^2_{X^\top X}} [X^\top (X w_k - y)]
$$