Chapter 1: Gradient Descent

Ernest K. Ryu

MATH 164: Optimization
University of California, Los Angeles
Department of Mathematics

Last edited: January 17, 2025

Gradient descent

Consider the optimization problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x),$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is differentiable.[1]

Gradient descent (GD) has the form

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

for $k = 0, 1, \ldots$, where $x_0 \in \mathbb{R}^n$ is a suitably chosen starting point and $\alpha_0, \alpha_1, \ldots \in \mathbb{R}$ is a positive step size sequence.

Under suitable conditions, we hope $x_k \overset{?}{\to} x_\star$ for some solution $x_\star$.

---

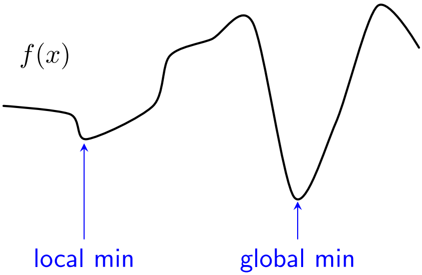
[1] If f is not differentiable, then *gradient* descent is not well defined, right?

Local vs. global minima

$x_\star$ is a *local minimum* if $f(x) \geq f(x_\star)$ within a small neighborhood.[2]

$x_\star$ is a *global minimum* if $f(x) \geq f(x_\star)$ for all $x \in \mathbb{R}^n$

In the worst case, finding the global minimum of an optimization problem is difficult. (The class of non-convex optimization problems is NP-hard.)



[2] if $\exists\, r > 0$ s.t. $\forall\, x$ s.t. $\|x - x_\star\| \leq r \Rightarrow f(x) \geq f(x_\star)$

What can we prove?

Without further assumptions, there is no hope of showing that GD finds the global minimum since GD can never "know" if it is stuck in a local minimum.

We cannot prove the function value converges to the global optimum. We instead prove $\nabla f(x_k) \to 0$. Roughly speaking, this is similar but weaker than proving that x_k converges to a local minimum.[3]

[3]Without further assumptions, we cannot show that x_k converges to a limit, and even x_k does converge to a limit, we cannot guarantee that that limit is not a saddle point or even a local maximum. Nevertheless, people commonly use the argument that x_k "usually" converges and that it is "unlikely" that the limit is a local maximum or a saddle point. More on this later.

$-\nabla f$ is steepest descent direction

From vector calculus, we know that ∇f is the steepest ascent direction, so $-\nabla f$ is the steepest descent direction. In other words,

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

is moving in the steepest descent direction, which is $-\nabla f(x_k)$ at the current position x_k, scaled by $\alpha_k > 0$.

Taylor expansion of f about x_k

$$f(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \mathcal{O}\big(\|x - x_k\|^2\big).$$

Plugging in x_{k+1}

$$f(x_{k+1}) = f(x_k) - \alpha_k \|\nabla f(x_k)\|^2 + \mathcal{O}(\alpha_k^2).$$

For small (cautious) α_k, GD step reduces function value.

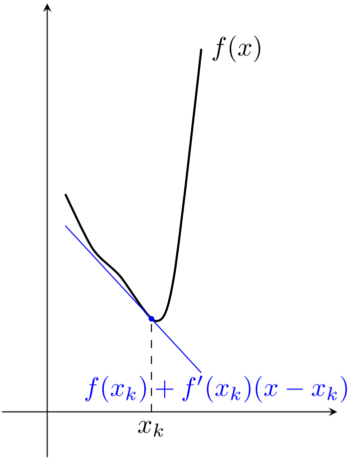
Is GD a "descent method"?

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

Without further assumptions, $-\nabla f(x_k)$ only provides directional information. How far should you go? How large should α_k be?

A step of GD need not result in descent, i.e., $f(x_{k+1}) > f(x_k)$ is possible.

Calculus only guarantees the accuracy of the Taylor expansion in an infinitesimal neighborhood.



Step size selection for GD

How do we choose the step size α_k and ensure convergence?

We consider 3 solutions:

- Make an assumption allowing us to choose α_k and ensures $f(x_k)$ will descend.
  - Estimate the L needed to choose α_k.
- Do a line search to ensure that $f(x_k)$ will descend.
- Drop the insistence that $f(x_k)$ must consistently go down.

Outline

Smooth non-convex GD

Smooth convex GD

GD for smooth non-convex functions

Consider the optimization problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x),$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is "L-smooth" (but not necessarily convex).

We consider GD with constant step size:

$$x_{k+1} = x_k - \alpha\nabla f(x_k).$$

(So $\alpha = \alpha_0 = \alpha_1 = \cdots$.)

We will show the following.

Theorem.

Assume $f\colon \mathbb{R}^n \to \mathbb{R}$ is L-smooth and $\inf f > -\infty$. Let $\alpha \in (0, 2/L)$. Then, the GD iterates satisfy $\nabla f(x_k) \to 0$.

L-smoothness

For $L > 0$, we say $f\colon \mathbb{R}^n \to \mathbb{R}$ is *L-smooth* if f is differentiable and

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \qquad \forall\, x, y \in \mathbb{R}^n.$$

I.e., $\nabla f\colon \mathbb{R}^n \to \mathbb{R}^n$ is L-Lipschitz continuous. We say f is *smooth* if it is L-smooth for some $L > 0$.

Interpretation 1: ∇f does not change too rapidly. This makes the first-order Taylor expansion reliable beyond an infinitesimal neighborhood. (Further quantified on next slide.)

If f twice-continuously differentiable, then L-smoothness is equivalent to

$$-L \leq \lambda_{\min}(\nabla^2 f(x)) \leq \lambda_{\max}(\nabla^2 f(x)) \leq L, \qquad \forall\, x \in \mathbb{R}^n.$$

Interpretation 2: The curvature f, quantified by $\nabla^2 f$, has lower and upper bounds $\pm L$.

The name "smoothness", as used in optimization, is somewhat confusing because in other areas of mathematics, "smoothness" often refers to infinite differentiability.

Smoothness ⇒ first-order Taylor has small remainder

For GD to work with a fixed non-adaptive step size, we need assurance that the first-order Taylor expansion is a good approximation within a sufficiently large neighborhood. L-smoothness provides this assurance.

Lemma.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be L-smooth. Then

$$\left| f(x+\delta) - \big(f(x) + \langle \nabla f(x), \delta \rangle\big) \right| \le \frac{L}{2}\|\delta\|^2, \qquad \forall\, x, \delta \in \mathbb{R}^n.$$

Note

$$R_1(\delta; x) = f(x+\delta) - \big(f(x) + \langle \nabla f(x), \delta \rangle\big)$$

is the remainder between f and its first-order Taylor expansion about x. This lemma provides a quantitative bound $|R_1(\delta; x)| \le \mathcal{O}(\|\delta\|^2)$.

L-smoothness lower and upper bounds

The claimed inequality

$$\left| f(x+\delta) - \left(f(x) + \langle \nabla f(x), \delta \rangle\right) \right| \leq \frac{L}{2}\|\delta\|^2$$

is equivalent to

$$f(x) + \langle \nabla f(x), \delta \rangle - \frac{L}{2}\|\delta\|^2 \overset{(*)}{\leq} f(x+\delta) \overset{(\#)}{\leq} f(x) + \langle \nabla f(x), \delta \rangle + \frac{L}{2}\|\delta\|^2.$$

We will only prove the upper bound inequality $\overset{(\#)}{\leq}$. The lower bound inequality $\overset{(*)}{\leq}$ follows from the same reasoning with some sign changes. (Also, we only use $\overset{(\#)}{\leq}$.)

Proof of the upper bound $\overset{(\#)}{\leq}$. Define $g : \mathbb{R} \to \mathbb{R}$ by

$$g(t) = f(x + t\,\delta).$$

Then g is differentiable, and its derivative is

$$g'(t) = \langle \nabla f(x + t\,\delta), \delta \rangle.$$

Next, observe that g' is $(L\|\delta\|^2)$-Lipschitz continuous. Indeed,

$$\begin{aligned} |g'(t_1) - g'(t_0)| &= \big|\langle \nabla f(x + t_1\,\delta) - \nabla f(x + t_0\,\delta), \delta \rangle\big| \\ &\leq \big\|\nabla f(x + t_1\,\delta) - \nabla f(x + t_0\,\delta)\big\| \|\delta\| \leq L\|\delta\|^2 |t_1 - t_0|. \end{aligned}$$

Finally, we conclude that

$$\begin{aligned} f(x + \delta) = g(1) = g(0) + \int_0^1 g'(t)\,dt \\ \leq f(x) \ + \ \int_0^1 \big(g'(0) + L\|\delta\|^2 t\big)\,dt \\ = f(x) + \langle \nabla f(x), \delta \rangle + \frac{L}{2}\|\delta\|^2. \end{aligned}$$

□

Summability lemma

Lemma.

Let $V_0, V_1, \ldots \in \mathbb{R}$ and $S_0, S_1, \ldots \in \mathbb{R}$ be nonnegative sequences satisfying

$$V_{k+1} \leq V_k - S_k$$

for $k = 0, 1, \ldots$. Then $S_k \to 0$.

Key idea. S_k measures progress (decrease) made in iteration k. Since $V_k \geq 0$, V_k cannot decrease forever, so the progress (magnitude of S_k) must diminish to 0.

Proof. Sum the inequality from $i = 0$ to k

$$V_{k+1} + \sum_{i=0}^{k} S_i \leq V_0.$$

Let $k \to \infty$

$$\sum_{i=0}^{\infty} S_i \leq V_0 - \lim_{k\to\infty} V_k \leq V_0$$

Since $\sum_{i=0}^{\infty} S_i < \infty$, we conclude $S_i \to 0$. $\square$

Convergence proof for smooth non-convex functions

Theorem.

Assume $f\colon \mathbb{R}^n \to \mathbb{R}$ is L-smooth and $\inf f > -\infty$. Let $\alpha \in (0, 2/L)$. Then, the GD iterates satisfy $\nabla f(x_k) \to 0$.

Proof. Use the Lipschitz gradient lemma with $x = x_k$ and $\delta = -\alpha \nabla f(x_k)$ to obtain

$$f(x_{k+1}) \leq f(x_k) - \alpha\Big(1 - \tfrac{\alpha L}{2}\Big)\|\nabla f(x_k)\|^2,$$

and

$$\overbrace{\Big(f(x_{k+1}) - \inf_x f(x)\Big)}^{\overset{\text{def}}{=} V_{k+1}} \leq \overbrace{\Big(f(x_k) \ - \ \inf_x f(x)\Big)}^{\overset{\text{def}}{=} V_k} - \overbrace{\underbrace{\alpha\Big(1 - \tfrac{\alpha L}{2}\Big)}_{\substack{>0 \\ \text{for } \alpha \in (0, 2/L)}} \|\nabla f(x_k)\|^2}^{\overset{\text{def}}{=} S_k}.$$

By the summability lemma, we have $\|\nabla f(x_k)\|^2 \to 0$ and thus $\nabla f(x_k) \to 0$. $\square$

GD experiments and curvature

GD with line search

Consider

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x),$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is differentiable but not necessarily smooth.

GD with *exact line search*

$$\begin{aligned} g_k &= \nabla f(x_k) \\ \alpha_k &\in \underset{\alpha \in \mathbb{R}}{\operatorname{argmin}}\, f(x_k - \alpha g_k) \\ x_{k+1} &= x_k - \alpha_k \nabla f(x_k) \end{aligned}$$

performs a one-dimensional search in the direction of the gradient.

Theorem.

Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be differentiable. Then GD with exact line search satisfies

$$f(x_k) \searrow f_\infty \in [-\infty, \infty).$$

Proof. By construction, we have $f(x_{k+1}) \leq f(x_k)$. A non-increasing sequence of real numbers converges to a value in $[-\infty, \infty)$. □

GD with inexact line search

Computing the exact line search is often expensive and unnecessary.
GD with *inexact line search*

$$
\begin{aligned}
g_k &= \nabla f(x_k) \\
\alpha_k &= \text{InexLineSearch}(f, x_k, g_k) \\
x_{k+1} &= x_k - \alpha_k \nabla f(x_k)
\end{aligned}
$$

InexLineSearch(f, x, g) :

$\alpha \leftarrow \beta$ // some initial constant > 0

if $g == 0$: return α

while $f(x - \alpha g) \geq f(x)$

  $\alpha \leftarrow \alpha/2$

return α

This inexact line search is also called a *backtracking line search*.

Theorem.

If f is differentiable, the line search terminates in finite steps.

Proof. Since f is differentiable,

$$f(x - \alpha g) = f(x) - \alpha \|g\|^2 + o(\alpha)$$

and there is a threshold $A > 0$ such that $f(x - \alpha g) < f(x)$ for $\alpha \in (0, A)$. The halving process of α eventually results in $f(x - \alpha g) < f(x)$ (by coincidence) or enters the interval $\alpha \in (0, A)$. □

GD with inexact line search

The starting step size $\beta > 0$ is a parameter to be tuned.

With large β, we have to perform the backtracking loop many times, but we have the opportunity to take a long step.

With small β, the backtracking loop may terminate more quickly, but we won't take steps larger than β.

One can modify the algorithm to adaptively decrease or increase β based on the history of backtracking.

How to choose the starting point x_0

Most (if not all) optimization algorithms require a starting point x_0. It is optimal to choose x_0 to be close (or equal to) $x_\star$, but, of course, we don't know where $x_\star$ is.

If one has an estimate of $x_\star$ based on problem structure, should utilize it.

In convex optimization problems, we often have convergence to the global minimum regardless of x_0, so it is okay to choose $x_0 = 0$.

For non-convex optimization problems, the general prescription is to start with $x_0 =$ random noise.

In some non-convex optimization problems (such as training deep neural networks), one must not use $x_0 = 0$, and a well-tuned random initialization is crucial.

Outline

Smooth non-convex GD

Smooth convex GD

Convex optimization

The problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x),$$

is a *convex optimization* problem if $f : \mathbb{R}^n \to \mathbb{R}$ is convex, i.e., if

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta) f(y), \qquad \forall\, x, y \in \mathbb{R}^n,\ \theta \in [0,1].$$

Finding the global minimum of a convex problem is tractable.

> *"In fact, the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity."*
> *— R. Tyrrell Rockafellar, in SIAM Review, 1993*

(In other areas of mathematics, linear things tend to be easier, while nonlinear things tend to be significantly harder, but not in optimization.)