535520: Optimization Algorithms

Lecture 4 — Gradient Descent

Ping-Chun Hsieh (謝秉均)

September 30, 2024

Announcement

- No class on 11/4 (Week 10)

- Let’s find a time to make up for the lecture last week
  - Originally, we expect to have 14 lectures (no lecture on 11/4 and in the final exam Week 16) + 1 poster presentation this semester
  - For the make-up lecture: Let’s still have a lecture in Week 16

Richard Sutton and “The Bitter Lesson”

The Bitter Lesson

Rich Sutton

March 13, 2019

The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. The ultimate reason for this is Moore's law, or rather its generalization of continued exponentially falling cost per unit of computation. Most AI research has been conducted as if the computation available to the agent were constant (in which case leveraging human knowledge would be one of the only ways to improve performance) but, over a slightly longer time than a typical research project, massively more computation inevitably becomes available. Seeking an improvement that makes a difference in the shorter term, researchers seek to leverage their human knowledge of the domain, but the only thing that matters in the long run is the leveraging of computation. These two need not run counter to each other, but in practice they tend to. Time spent on one is time not spent on the other. There are psychological commitments to investment in one approach or the other. And the human-knowledge approach tends to complicate methods in ways that make them less suited to taking advantage of general methods leveraging computation. There were many examples of AI researchers' belated learning of this bitter lesson, and it is instructive to review some of the most prominent.

In computer chess, the methods that defeated the world champion, Kasparov, in 1997, were based on massive, deep search. At the time, this was looked upon with dismay by the majority of computer-chess researchers who had pursued methods that leveraged human understanding of the special structure of chess. When a simpler, search-based approach with special hardware and software proved vastly more effective, these human-knowledge-based chess researchers were not good losers. They said that ``brute force" search may have won this time, but it was not a general strategy, and anyway it was not how people played chess. These researchers wanted methods based on human input to win and were disappointed when they did not.

A similar pattern of research progress was seen in computer Go, only delayed by a further 20 years. Enormous initial efforts went into avoiding search by taking advantage of human knowledge, or of the special features of the game, but all those efforts proved irrelevant, or worse, once search was applied effectively at scale. Also important was the use of learning by self play to learn a value function (as it was in many other games and even in chess, although learning did not play a big role in the 1997 program that first beat a world champion). Learning by self play, and learning in general, is like search in that it enables massive computation to be brought to bear. Search and learning are the two most important classes of techniques for utilizing massive amounts of computation in AI research. In computer Go, as in computer chess, researchers' initial effort was directed towards utilizing human understanding (so that less search was needed) and only much later was much greater success had by embracing search and learning.

In speech recognition, there was an early competition, sponsored by DARPA, in the 1970s. Entrants included a host of special methods that took advantage of human knowledge---knowledge of words, of phonemes, of the human vocal tract, etc. On the other side were newer methods that were more statistical in nature and did much more computation, based on hidden Markov models (HMMs). Again, the statistical methods won out over the human-knowledge-based methods. This led to a major change in all of natural language processing, gradually over decades, where statistics and computation came to dominate the field. The recent rise of deep learning in speech recognition is the most recent step in this consistent direction. Deep learning methods rely even less on human knowledge, and use even more computation, together with learning on huge training sets, to produce dramatically better speech recognition systems. As in the games, researchers always tried to make systems that worked the way the researchers thought their own minds worked---they tried to put that knowledge in their systems---but it proved ultimately counterproductive, and a colossal waste of researcher's time, when, through Moore's law, massive computation became available and a means was found to put it to good use.

In computer vision, there has been a similar pattern. Early methods conceived of vision as searching for edges, or generalized cylinders, or in terms of SIFT features. But today all this is discarded. Modern deep-learning neural networks use only the notions of convolution and certain kinds of invariances, and perform much better.

This is a big lesson. As a field, we still have not thoroughly learned it, as we are continuing to make the same kind of mistakes. To see this, and to effectively resist it, we have to understand the appeal of these mistakes. We have to learn the bitter lesson that building in how we think we think does not work in the long run. The bitter lesson is based on the historical observations that 1) AI researchers have often tried to build knowledge into their agents, 2) this always helps in the short term, and is personally satisfying to the researcher, but 3) in the long run it plateaus and even inhibits further progress, and 4) breakthrough progress eventually arrives by an opposing approach based on scaling computation by search and learning. The eventual success is tinged with bitterness, and often incompletely digested, because it is success over a favored, human-centric approach.

One thing that should be learned from the bitter lesson is the great power of general purpose methods, of methods that continue to scale with increased computation even as the available computation becomes very great. The two methods that seem to scale arbitrarily in this way are *search* and *learning*.

The second general point to be learned from the bitter lesson is that the actual contents of minds are tremendously, irredeemably complex; we should stop trying to find simple ways to think about the contents of minds, such as simple ways to think about space, objects, multiple agents, or symmetries. All these are part of the arbitrary, intrinsically-complex, outside world. They are not what should be built in, as their complexity is endless; instead we should build in only the meta-methods that can find and capture this arbitrary complexity. Essential to these methods is that they can find good approximations, but the search for them should be by our methods, not by us. We want AI agents that can discover like we can, not which contain what we have discovered. Building in our discoveries only makes it harder to see how the discovering process can be done.

Richard Sutton

“One thing that should be learned from the bitter lesson is the great power of general purpose methods, of methods that continue to scale with increased computation even as the available computation becomes very great.”

Gradient descent turns out to be one such method

http://www.incompleteideas.net/IncIdeas/BitterLesson.html

This Lecture

1. Gradient Descent for Convex Problems

2. Gradient Descent for Non-Convex Problems

- Reading Material:
  - Chapter 5 of Amir Beck's textbook "First-Order Methods in Optimization"
  - Chapters 2 & 5 of Dimitri Bertsekas's textbook "Nonlinear Programming"
  - Chapter 3 of Jorge Nocedel and Stephen Wright's textbook "Numerical Optimization"
  - Yuxin Chen's lecture note: https://yuxinchen2020.github.io/ele522_optimization/lectures/grad_descent_unconstrained.pdf

Convergence Rates of GD

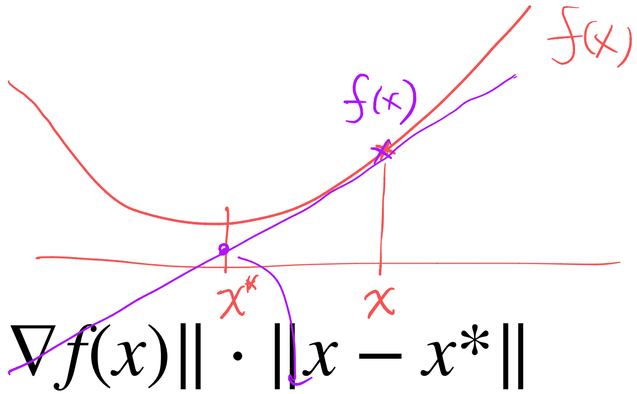
	Convergence Rate (under constant step sizes)
Quadratic problem	$\|x_t - x^*\|_2 = \left(\frac{\lambda_1(Q) - \lambda_n(Q)}{\lambda_1(Q) + \lambda_n(Q)}\right)^t \cdot \|x_0 - x^*\|_2$
Strongly convex and L-smooth	$\|x_t - x^*\| \leqslant \left(\frac{\kappa - 1}{\kappa + 1}\right)^t \cdot \|x_0 - x^*\|$
PL condition and L-smooth	$f\left(x_t\right) - f(x^*) \leqslant \left(1 - \frac{\mu}{L}\right)^t \cdot \left(f\left(x_0\right) - f(x^*)\right)$
Convex and L-smooth	$f(x_t) - f(x^*) \leqslant \frac{L}{2t} \cdot \|x_0 - x^*\|^2 = O(\frac{1}{t})$
Non-convex and L-smooth	$\min_{0 \leq k \leq T} \|\nabla f(x_k)\| \leq \sqrt{\frac{2L(f(x_0) - f(x^*))}{T}}, \quad \lim_{t \to \infty} \|\nabla f(x_t)\| = 0$

Intuition: Why can GD converge under convexity?

Cauchy-Schwarz:
$x, y \in \mathbb{R}^n$
$x^T y \leq \|x\|_1 \cdot \|y\|_2$

Ideally, if GD converges (to some point), then we expect $\|\nabla f(x_t)\| \to 0$ as $t \to \infty$

Recall from the definition of convexity:



$$f(x) - f(x^*) \leq \nabla f(x)^\top (x - x^*) \leq \|\nabla f(x)\| \cdot \|x - x^*\|$$

Convexity

$f(x) + \nabla f(x)^T (x^* - x)$

Suppose the domain is of finite radius, then $\|\nabla f(x)\| \to 0$ implies that $(f(x) - f(x^*)) \to 0$

"Astral space"

Quick Recap: GD for Quadratic Problems

Let's motivate the convergence of GD using a quadratic objective function

$$\min_{x\in\mathbb{R}^n} \quad f(x) := \frac{1}{2}(x-x^*)^\top Q(x-x^*), \; Q \text{ is pd} \quad (\lambda_1(Q) \geq \cdots \geq \lambda_n(Q) > 0)$$

GD update: $x_{k+1} = x_k - \eta_k \nabla f(x_k) =$

Theorem (Convergence Rate of GD for Quadratic Problems):

Under the step size $\eta_t \equiv \eta = 2/(\lambda_1(Q) + \lambda_n(Q))$, then we have

$$\|x_t - x^*\|_2 = \left(\frac{\lambda_1(Q) - \lambda_n(Q)}{\lambda_1(Q) + \lambda_n(Q)}\right)^t \cdot \|x_0 - x^*\|_2, \quad \forall t \in \mathbb{N}$$

Quick Recap: GD for Quadratic Problems

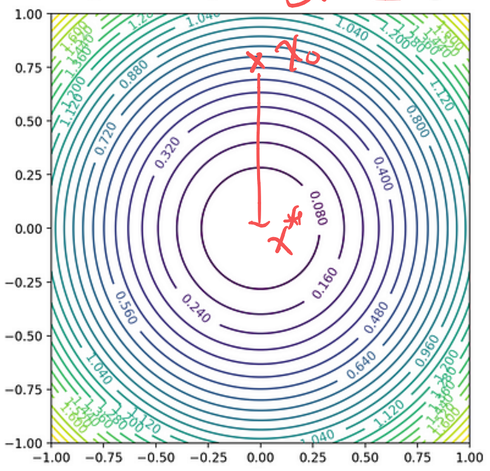
$$\|x_t - x^*\|_2 = \left(\frac{\lambda_1(Q) - \lambda_n(Q)}{\lambda_1(Q) + \lambda_n(Q)}\right)^t \cdot \|x_0 - x^*\|_2, \quad \forall t \in \mathbb{N}$$

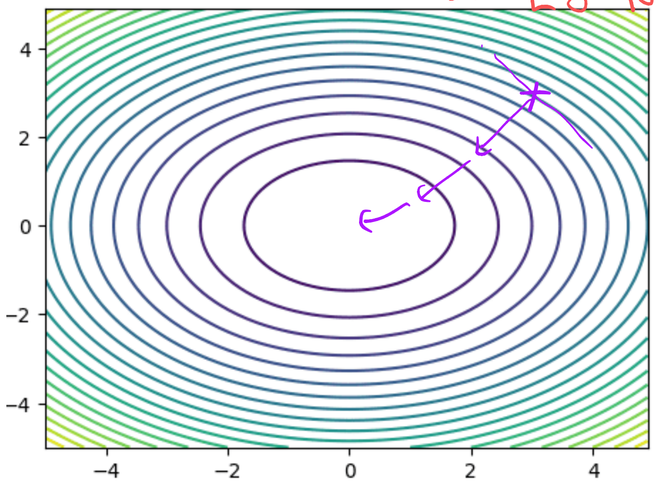
$$= \left(\frac{1 - C(Q)}{1 + C(Q)}\right)^t \cdot \|x_0 - x^*\|_2$$

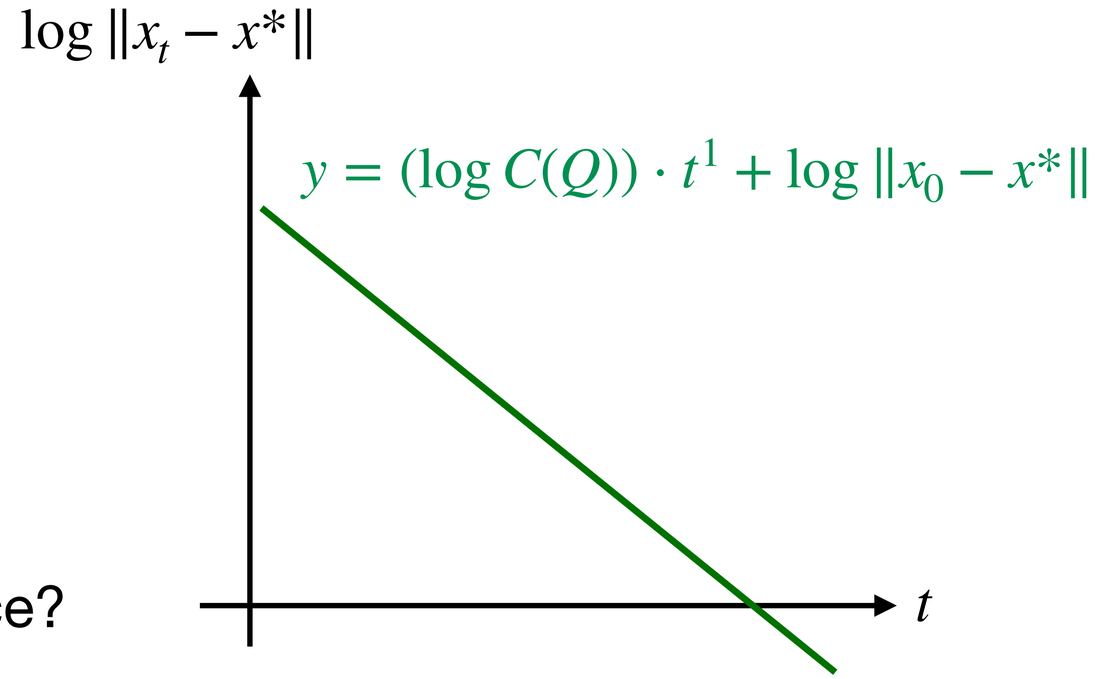
where $C(Q) := \frac{\lambda_n(Q)}{\lambda_1(Q)}$ is "condition number"

$C(Q) \in (0, 1]$

- Question: This is often called "geometric convergence" or "linear convergence"







How about "sub-linear" or "super-linear" convergence?

Terminology of Convergence Rates

- Let $e(x)$ denote the distance from optimality
  - Example: $e(x) = \|x - x^*\|$
  - Example: $e(x) = |f(x) - f(x^*)|$ (sub-optimality)
- Rate of convergence: The limit of the ratio of successive errors

$$\lim_{k \to \infty} \frac{e(x_{k+1})}{e(x_k)} = \beta$$

  - If $\beta = 1$: We call it a sub-linear rate of convergence
  - If $\beta \in (0,1)$: We call it a linear rate of convergence
  - If $\beta = 0$: We call it a super-linear rate of convergence

Proof: Convergence of GD for Quadratic Problems

Step 1: Consider the GD update

$$x_{t+1} - x^* = \left(x_t - \eta \cdot \nabla f(x^t)\right) - x^* = (I_n - \eta Q) \cdot (x_t - x^*)$$

This implies that

$$\|x_{t+1} - x^*\|_2 \le \|I_n - \eta \cdot Q\| \cdot \|x_t - x^*\|_2 \quad \text{..... (by definition of spetral norm)}$$

What norm?

Step 2: Moreover, we have

$$\|I_n - \eta Q\| = \max\left\{ |1 - \eta\lambda_1(Q)|, |1 - \eta\lambda_n(Q)| \right\}$$

$$\eta = \frac{2}{\lambda_1(Q)+\lambda_n(Q)} = 1 - \frac{2\lambda_n(Q)}{\lambda_1(Q) + \lambda_n(Q)}$$

$\frac{2\lambda_n}{\lambda_1+\lambda_n} \le 1$

Step 3: By repeating Step 1 recursively, we complete the proof.

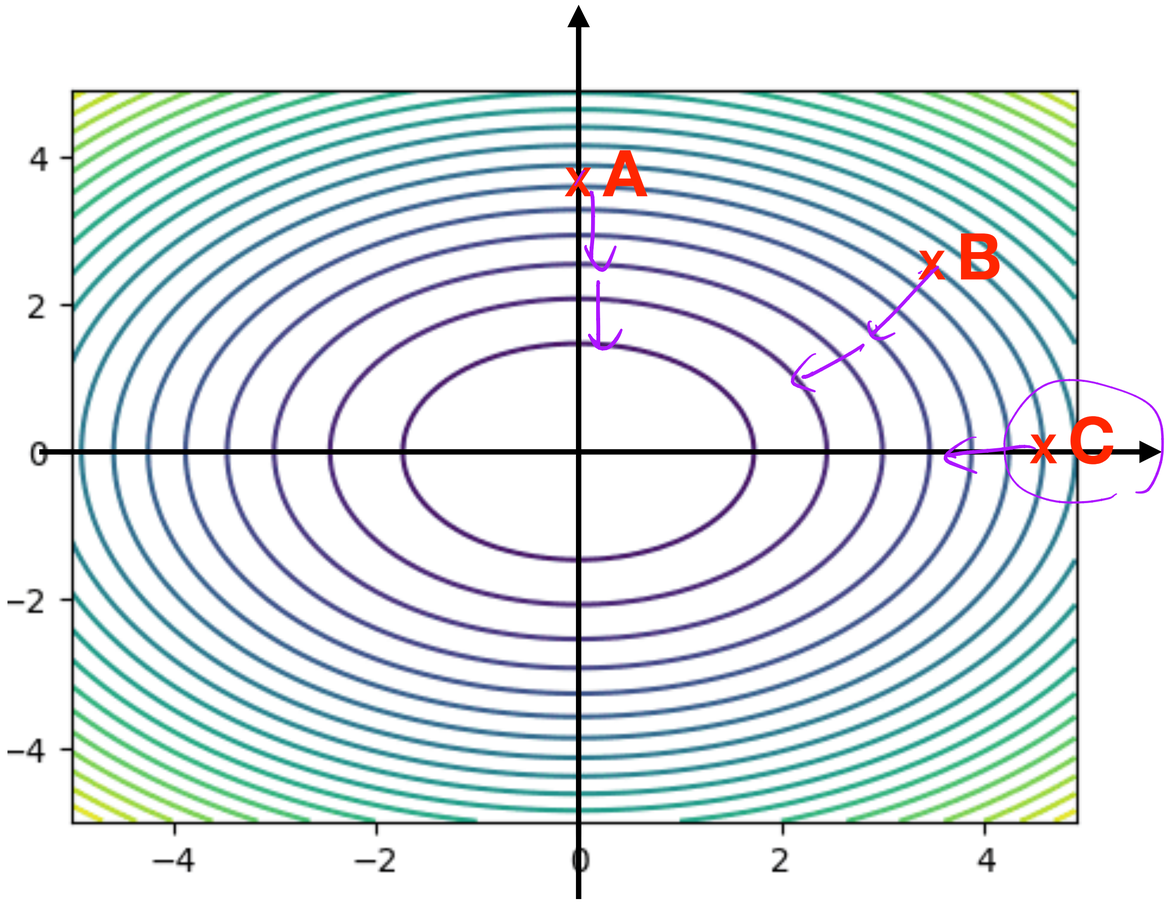
Spectral Norm:

Matrix $A \in \mathbb{R}^{n\times n}$

$\|A\| := \sup_{x:\|x\|_2=1} \|Ax\|$

Observations: GD with Constant Step Sizes

Consider $f(x) := \frac{1}{2} x^\top Q x$ with $Q = [[1,0],[0,10]]$



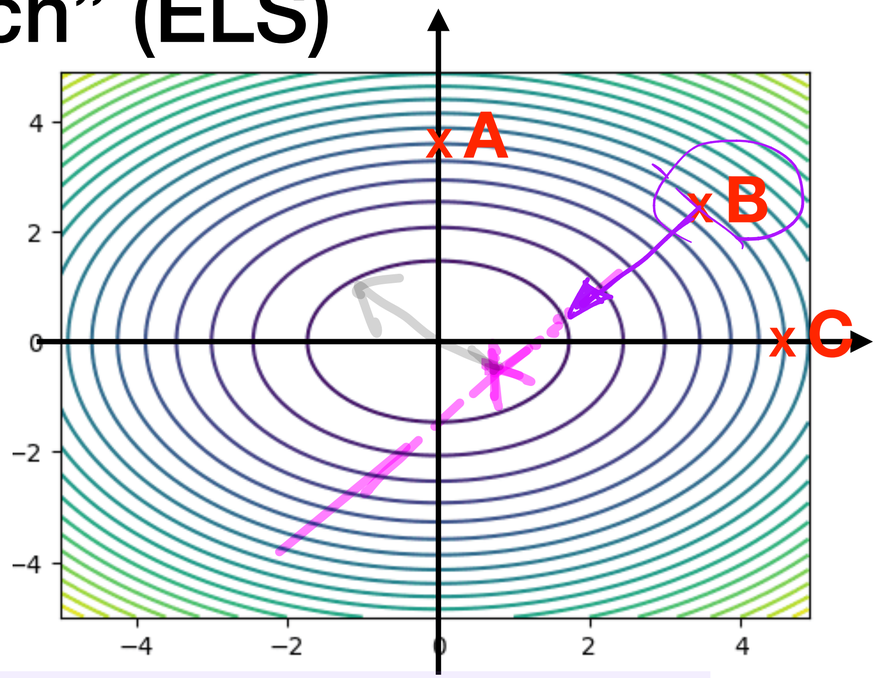
$$\eta = \frac{2}{\lambda_1(Q) + \lambda_n(Q)} = \frac{2}{11}$$

Can you find any special behavior of GD with constant step sizes?

Another Variant: GD with "Exact Line Search" (ELS)

To accelerate GD, we can choose the step sizes by

$$\eta_t = \arg\min_{\eta \geqslant 0} f\left(x_t - \eta \nabla f\left(x_t\right)\right)$$



Theorem (Convergence Rate of GD with ELS):

By applying GD with ELS to the quadratic problems, we have

$$f(x_t) - f(x^*) \leq \left(\frac{\lambda_1(Q) - \lambda_n(Q)}{\lambda_1(Q) + \lambda_n(Q)}\right)^{2t} \cdot \left(f(x_0) - f(x^*)\right), \quad \forall t \in \mathbb{N}$$

Remark: The convergence rate is actually the same of GD with constant η

Proof: Convergence of GD for Quadratic Problems (1/2)

Notation: $g_t \equiv \nabla f(x_t) = Q(x_t - x^*)$

Step 1: Under ELS, we have

$$\eta_t = \arg\min_{\eta \geqslant 0} f\left(x_t - \eta \nabla f\left(x_t\right)\right) = \frac{g_t^\top g_t}{g_t^\top Q g_t} \quad \text{(Why?)}$$

Step 2: Then, we can find $f(x_{t+1})$ as

$$\begin{aligned} f\left(x_{t+1}\right) &= \frac{1}{2}\left(x_t - \eta_i Q\left(x_t - x^*\right) - x^*\right)^\top Q\left(x_t - \eta_t Q\left(x_t - x^*\right) - x^*\right) \\ &= \frac{1}{2}\left(x_t - x^*\right)^\top Q\left(x_t - x^*\right) - \frac{\|g_t\|^4}{2 g_t^\top Q g_t} \\ &= \underbrace{\frac{1}{2}\left(x_t - x^*\right)^\top Q\left(x_t - x^*\right)}_{f(x_t)} \cdot \left(1 - \frac{\|g_t\|^4}{\left(g_t^\top Q g_t\right) \cdot \left(g_t^\top Q^{-1} \cdot Q \cdot Q^{-1} g\right)}\right) \end{aligned}$$

Proof: Convergence of GD for Quadratic Problems (2/2)

Step 3: To bound (A), we can use the "Kantorovich's inequality"

Lemma (Kantorovich's inequality):

Let Q be a symmetric and pd matrix. Then, for any $y \in \mathbb{R}\backslash\{0\}$,

$$\frac{\|y\|^4}{\left(y^\top Q y\right) \cdot \left(y^\top Q^{-1} y\right)} \geqslant \frac{4\lambda_1(Q) \cdot \lambda_n(Q)}{\left(\lambda_1(Q) + \lambda_n(Q)\right)^2}$$

As a result, we have

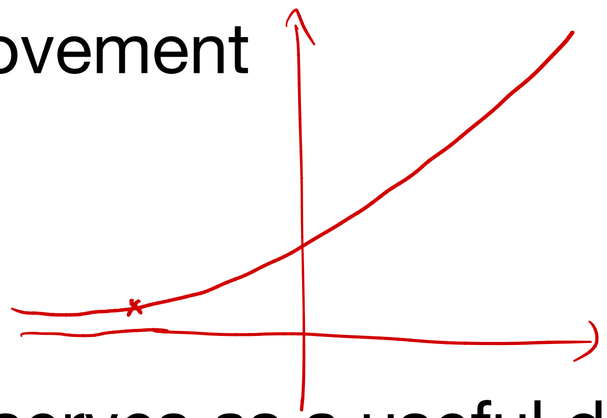
$$f(x_{t+1}) \leq \left(1 - \frac{4 \cdot \lambda_1(Q) \cdot \lambda_n(Q)}{\left(\lambda_1(Q) + \lambda_n(Q)\right)^2}\right) \cdot f\left(x_t\right) = \left(\frac{\lambda_1(Q) - \lambda_n(Q)}{\lambda_1(Q) + \lambda_n(Q)}\right)^2 f\left(x_t\right)$$

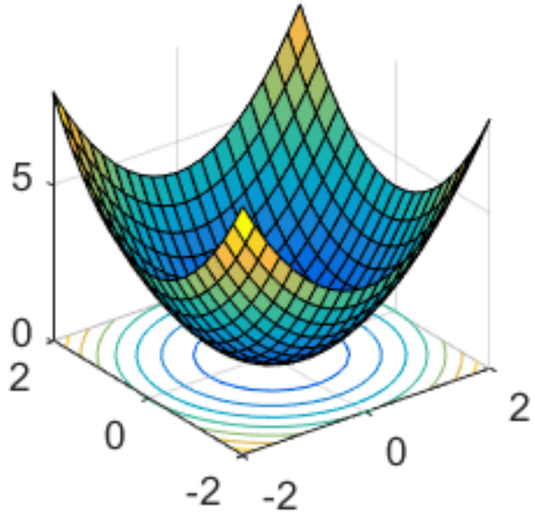
Let’s go beyond quadratic problems:

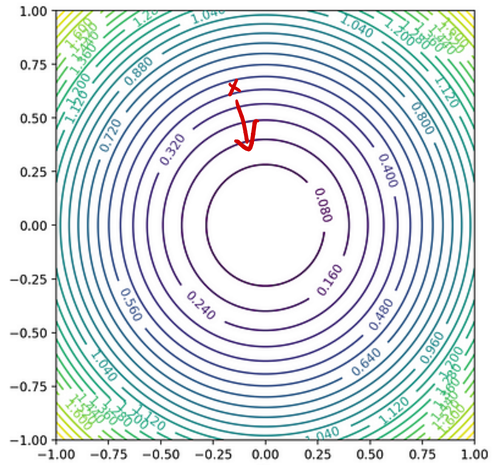
Strongly-convex and *smooth* problems

Why Strong Convexity and Smoothness?

Strong convexity: GD can always attain sufficient per-step improvement

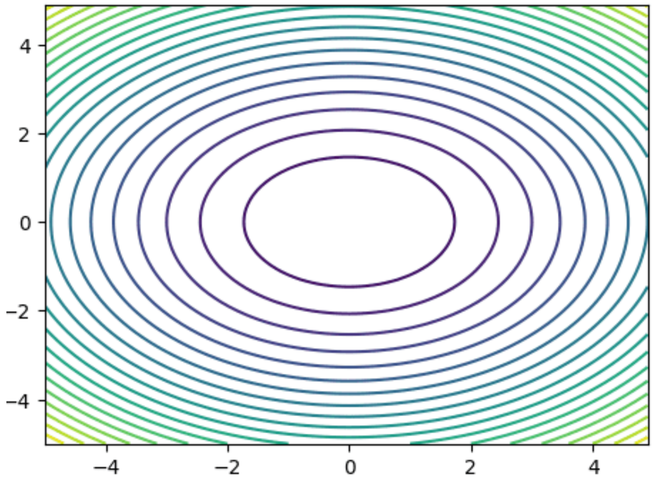


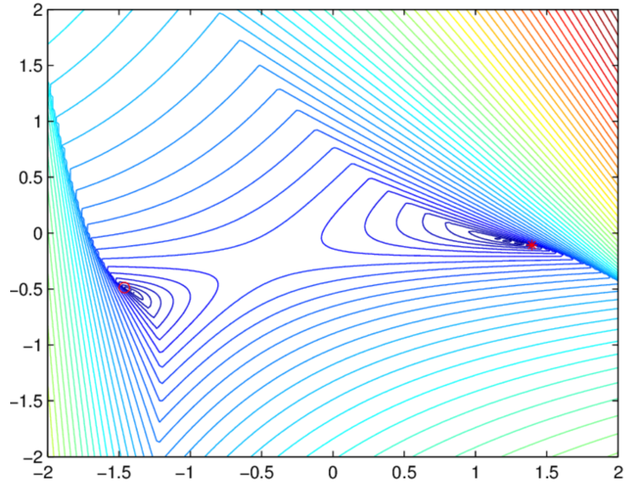




Smoothness: Gradient serves as a useful direction for improvement

GD $f(x_{t+1}) < f(x_t)$

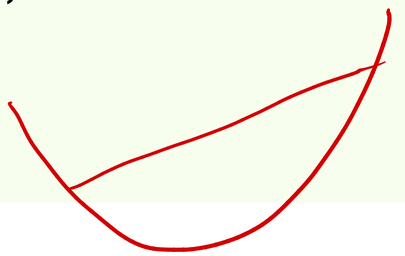




Strict Convexity vs Strong Convexity

Definition: A function $f : X \to \mathbb{R}$ is called strictly convex if its domain X is a convex set and for any $x, y \in X$ with $x \neq y$ and any $\alpha \in [0,1]$, we have

$$f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha) f(y)$$

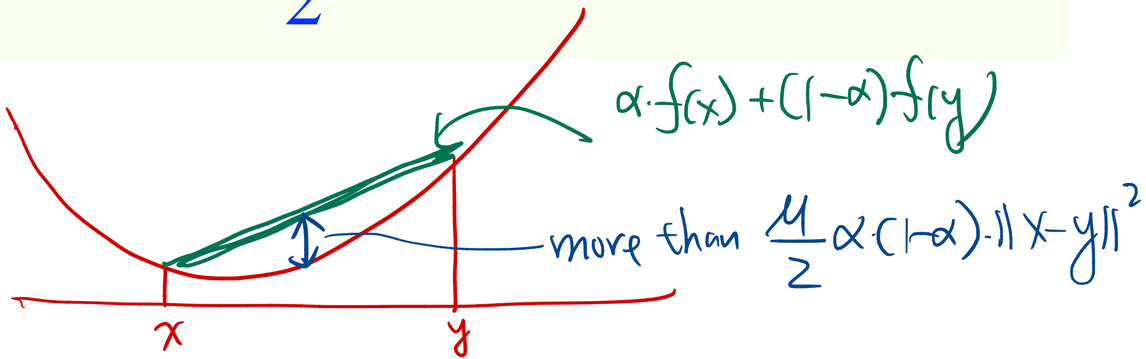


Intuition: "The segment lies strictly above the function"

Definition: A function $f : X \to \mathbb{R}$ is called μ-strongly convex if its domain X is a convex set and there exists some $\mu > 0$ such that for any $x, y \in X$

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y) - \frac{\mu}{2}\alpha(1-\alpha)\|x-y\|^2$$

Intuition: 1-dimensional case



An Alternative Definition of Strong Convexity

Theorem 1: Let $f : X \to \mathbb{R}$ be a *continuously differentiable* function. Then, the following are equivalent characterization of strong convexity.

(1) There exists some $\mu > 0$ such that for any $x, y \in X$,

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|y - x\|^2$$

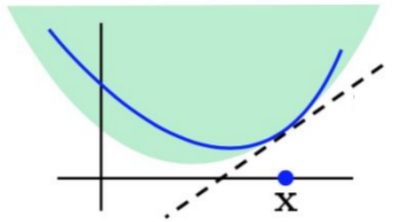
(2) There exists some $\mu > 0$ such that for any $x, y \in X$,

$$(\nabla f(x) - \nabla f(y))^\top (x - y) \geq \mu \|x - y\|^2$$

(3) Moreover, if f is twice continuously differentiable, then there exists some $\mu > 0$ such that for any $x \in X$, $\nabla^2 f(x) - \mu I \succ 0$

Intuition: Taylor expansion

(Proof: HW1 Problem)

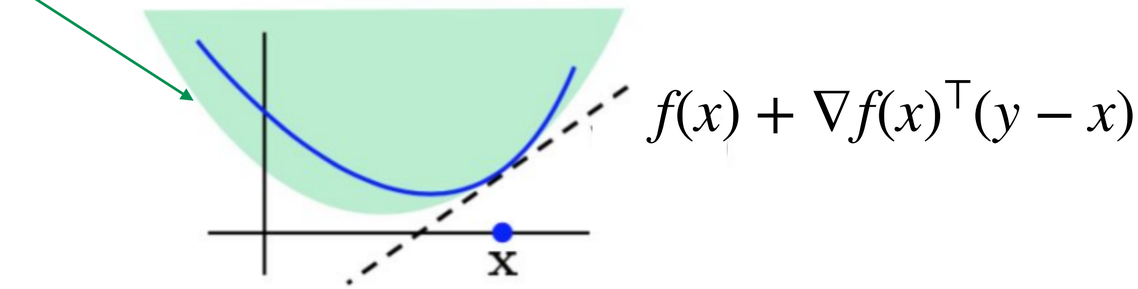


Connecting “Strict Convexity” and “Strong Convexity”

Theorem 2: Let $f : X \to \mathbb{R}$ be a *continuously differentiable* function with an open convex domain X. If f is strongly convex, then f is also strictly convex.

Intuition:

$$f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|y - x\|^2$$



$$f(x) + \nabla f(x)^\top (y - x)$$

Connecting “Strict Convexity” and “Strong Convexity”

Theorem 2: Let $f : X \to \mathbb{R}$ be a *continuously differentiable* function with an open convex domain X. If f is strongly convex, then f is also strictly convex.

Proof: Define $h(t) := f(x + t(y - x)), t \in \mathbb{R}$

Step 1: Consider $t, t' \in [0,1]$ such that $t < t'$

$$\underbrace{\big(\nabla f(x + t'(y - x)) - \nabla f(x + t(y - x))\big)^{\top}\big((t' - t)(y - x)\big)}_{= \left(\frac{dh(t')}{dt} - \frac{dh(t)}{dt}\right)(t' - t)} \geq \alpha(t' - t)^2\|y - x\|^2 > 0$$

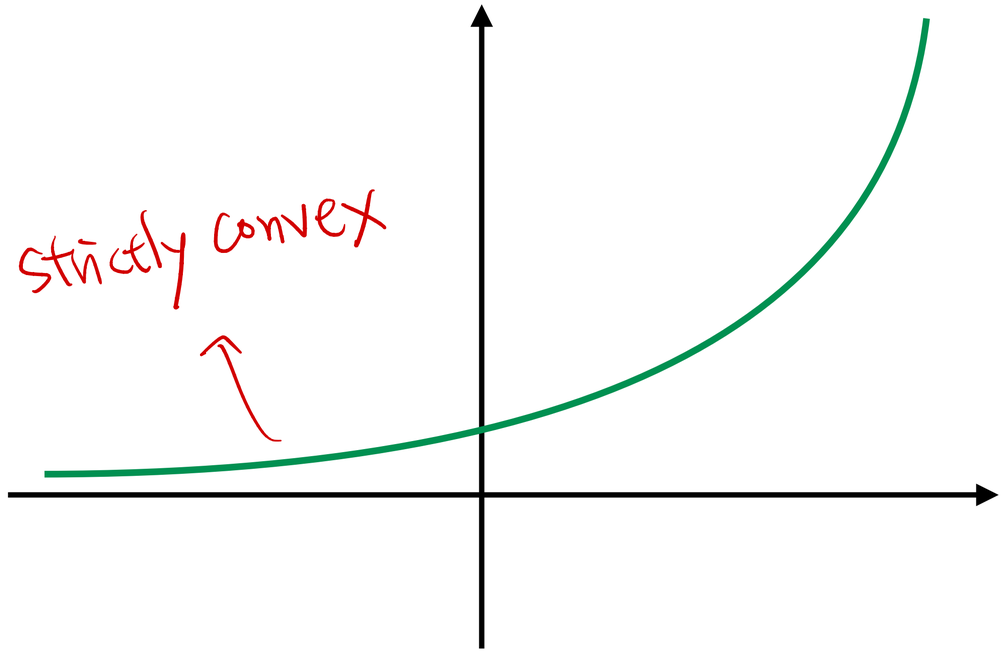
Step 2: By Step 1, we know $\frac{dh}{dt}$ is strictly increasing. As a result,

$$\frac{h(t) - h(0)}{t} = \frac{1}{t}\int_0^t \frac{dh(s)}{ds}ds < \frac{1}{1 - t}\int_t^1 \frac{dh(s)}{ds}ds = \frac{h(1) - h(t)}{1 - t} \qquad \text{(Why?)}$$

Step 3: Hence, we have $t \cdot h(1) + (1 - t)h(0) > h(t)$

“Strict Convexity” does NOT imply “Strong Convexity”

- A strictly convex function is NOT necessarily strongly convex
- Example: $f(x) = \exp(x)$ is not strongly convex on $\mathbb{R}$



Check $f''(x)$:

$f''(x) = \exp(x) \neq c$ for all $x \in \mathbb{R}$

$\Rightarrow$ $f(x)$ is "NOT" strongly convex.

Lipschitz Smoothness and L-Smoothness

Definition: $f : \mathbb{R}^n \to \mathbb{R}^m$ is called Lipschitz continuous if there exists $L < \infty$ such that for all $x, y \in \mathbb{R}^n$

$$\|f(x) - f(y)\| \leq L\|x - y\|$$

Definition: $f : \mathbb{R}^n \to \mathbb{R}$ is called L-smooth if it has *Lipschitz continuous gradients*, i.e., there exists $L < \infty$ such that for all $x, y \in \mathbb{R}^n$

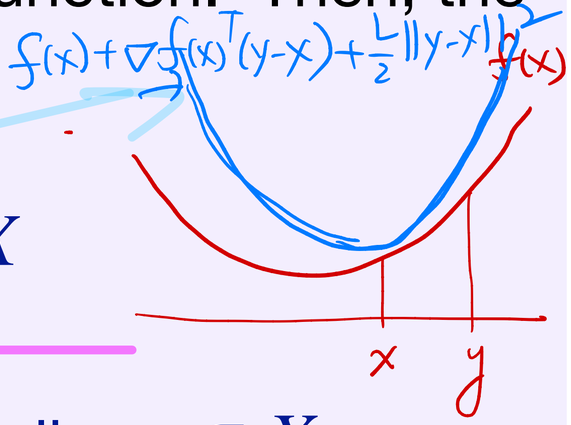
$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$$

Theorem 3: Let $f : X \to \mathbb{R}$ be *twice differentiable.* Then, f is L-smooth if and only if

$$\nabla^2 f(x) \leq LI$$

Equivalent Characterization of L-Smoothness for Convex Functions

Theorem 4: Let $f : X \to \mathbb{R}$ be a convex and differentiable function. Then, the following are equivalent characterization of L-smoothness:



(1) $f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|y - x\|^2$, for all $x, y \in X$

(2) $f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2L}\|\nabla f(y) - \nabla f(x)\|^2$, for all $x, y \in X$

(3) $(\nabla f(x) - \nabla f(y))^\top (y - x) \geq \frac{1}{L}\|\nabla f(y) - \nabla f(x)\|^2$, for all $x, y \in X$

1. $f(y) \leq f(x) + \nabla f(x)^T (y-x) + \frac{L}{2}\|y-x\|^2$ $\pm \nabla f(y) - \nabla f(y)$

+) 2. $f(x) \leq f(y) + \nabla f(y)^T (x-y) + \frac{L}{2}\|x-y\|^2$

$\leq f(y) +$

(For the details, please see Chapter 5.1.2 of Amir Beck's textbook)

In the next few slides, we focus on GD for

μ-strongly convex and *L-smooth* objective functions

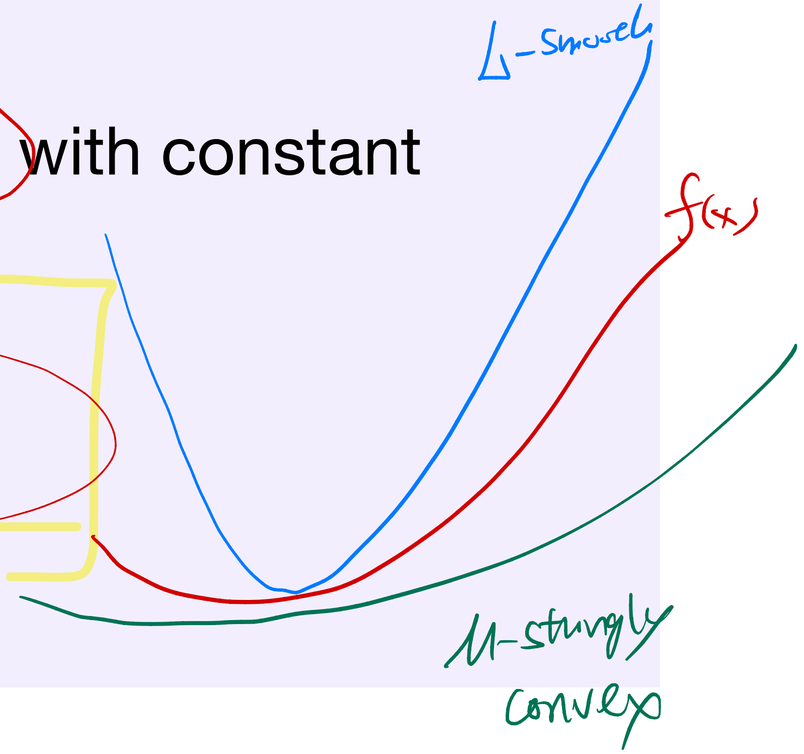
Convergence of GD for μ-Strongly Convex and Smooth Functions

Theorem (Convergence of GD under strong convexity and smoothness):

Let f be μ-strongly convex and L-smooth. Under GD with constant step sizes $\eta = 2/(\mu + L)$, we have

$$\|x_t - x^*\| \leqslant \left(\frac{\kappa - 1}{\kappa + 1}\right)^t \cdot \|x_0 - x^*\|$$

($\kappa := L/\mu$ is called the condition number)



Comparison:

- Step size: $\dfrac{2}{\mu + L}$ vs $\dfrac{2}{\lambda_1(Q) + \lambda_n(Q)}$
- Contraction: $\dfrac{\kappa - 1}{\kappa + 1}$ vs $\dfrac{1 - C(Q)}{1 + C(Q)}$

Proof: Convergence of GD for μ-Strongly Convex and Smooth Functions

Step 1: Let's rewrite

$$\nabla f(x_t) = \nabla f\left(x_t\right) - \nabla f\underbrace{(x^*)}_{x_t + 1 \cdot (x^* - x_t)} = \left(\int_0^1 \nabla^2 f\left(x_t + s \cdot \left(x^* - x_t\right)\right) \cdot ds \right) \left(x_t - x^*\right)$$

Step 2:

$$\|x_{t+1} - x^*\| = \|x_t - x^* - \eta \cdot \nabla f(x_t)\|$$

$$= \left\| \left(I - \eta \cdot \int_0^1 \nabla^2 f\left(x_t + s\left(x^* - x_t\right)\right) ds \right)(x_t - x^*) \right\|$$

$$= \underbrace{\sup_{0 \le s = 1} \left\| I - \eta \cdot \int_0^1 \nabla^2 f\left(x_t + s \cdot \left(x^* - x_t\right)\right) ds \right\|}_{\le |1 - \eta L|} \cdot \left\| x_t - x^* \right\|$$

Next Question: Do we still get “linear convergence” while relaxing the strong convexity condition?

Polyak-Łojasiewicz (PL) Condition in Non-Convex Optimization

Question: When can GD succeed under non-convex objective functions?

Polyak-Łojasiewicz Condition

Gradient norm Sub-optimality gap

$$\|\nabla f(\theta)\|^2 \geq 2\mu \cdot (f(\theta) - f(\theta^*)) \quad \text{for some } \mu > 0$$

(aka "gradient dominance")

Boris Polyak

Stanisław Łojasiewicz

Interpretation:

- PL ensures that gradient grows fast as it moves away from the optimum
- PL ensures that every stationary point is a global optimum

Convergence of GD Under PL Condition

Theorem (Convergence of GD under PL and smoothness):

Let f satisfies PL condition and is L-smooth. Under GD with constant step sizes $\eta = 1/L$, we have

$$f(x_t) - f(x^*) \leqslant \left(1 - \frac{\mu}{L}\right)^t \cdot \big(f(x_0) - f(x^*)\big), \quad \forall t \in \mathbb{N}$$

Proof: $f(x_{t+1}) - f(x^*)$

Smoothness: $f(y) \leq f(x) + \nabla f(x)^T (y-x) + \frac{L}{2}\|y-x\|^2$

$y = x_{t+1}$ $x = x_t$ $y - x = -\frac{1}{L} \cdot \nabla f(x_t)$

$$\leqslant f(x_t) - f(x^*) - \frac{1}{2L}\|\nabla f(x_t)\|^2 \quad \ldots\ldots \text{(by Smoothness)}$$

$$\leqslant f(x_t) - f(x^*) - \frac{\mu}{L} \cdot (f(x_t) - f(x^*)) \quad \ldots\ldots \text{(by PL condition)}$$

$$= (1 - \frac{\mu}{L}) \cdot (f(x_t) - f(x^*)) \quad \ldots\ldots \text{(by rearranging the terms)}$$

Question: Any known problem that satisfies a PL-like condition?

Example 1: Overparametrized Linear Regression

Linear regression: Given N data samples $\{a_i \in \mathbb{R}^m, y_i \in \mathbb{R}\}$, find a linear model by minimizing

$$f(x) = \frac{1}{2}\sum_{i=1}^{n} (a_i^\top x - y_i)^2 = \|Ax - y\|^2$$

Hessian $m \times m$

Overparameterization: Model dimension m > sample size n (This regime occurs frequently in deep learning)

$AA^\top$ is psd but not pd.

Remark: This is a convex but not strongly convex problem (why?)

$$\nabla^2 f(x) = \sum_{i=1}^{n} a_i a_i^\top = AA^\top$$

$AA^\top$: $m \times m$

check: $x^\top (AA^\top) x \geq 0$

$A = [a_1, a_2, \cdots, a_n]$: $m \times n$

Notation:
$A = [a_1, \cdots, a_n]^\top \in \mathbb{R}^{n \times m}$ ~~$n \times m$~~ $m \times n$

Question: Does $f(x)$ satisfy the PL condition?

Example 1: Overparametrized Linear Regression

Let's show that $\|\nabla f(x)\|_2^2 \geq 2\lambda_{\min}\left(\boldsymbol{A}\boldsymbol{A}^\top\right)\left(f(x) - f(x^*)\right)$

Notation:

$\boldsymbol{A} = [a_1, \cdots, a_n]^\top \in \mathbb{R}^{n \times m}$

$y = [y_1, \cdots, y_n]^\top \in \mathbb{R}^{n \times 1}$

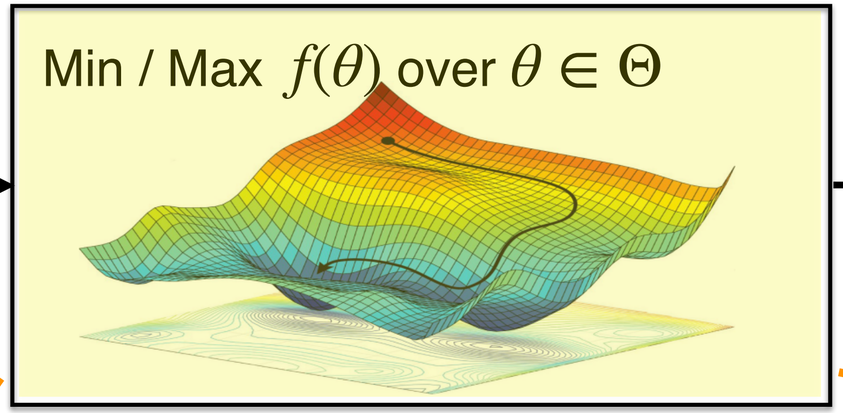
$$\nabla f(x) = \boldsymbol{A}^\top(\boldsymbol{A}x - y)$$

$$\|\nabla f(x)\|_2^2 = (\boldsymbol{A}x - y)^\top \boldsymbol{A}\boldsymbol{A}^\top(\boldsymbol{A}x - y) \quad \ldots\ldots \left(\|\nabla f(x)\|^2 = \nabla f(x)^\top \cdot \nabla f(x)\right)$$

$$\nabla f(x)^\top \nabla f(x) \geq \lambda_{\min}(\boldsymbol{A}\boldsymbol{A}^\top)\|\boldsymbol{A}x - y\|^2 \quad \ldots\ldots \left(\boldsymbol{A}\boldsymbol{A}^\top \text{ is psd.}\right)$$

$$= 2\lambda_{\min}(\boldsymbol{A}\boldsymbol{A}^\top) f(x) \quad \ldots\ldots \left(f(x) = \frac{1}{2}\cdot\|\boldsymbol{A}x - y\|^2\right)$$

Example 2: Policy Gradient in Reinforcement Learning



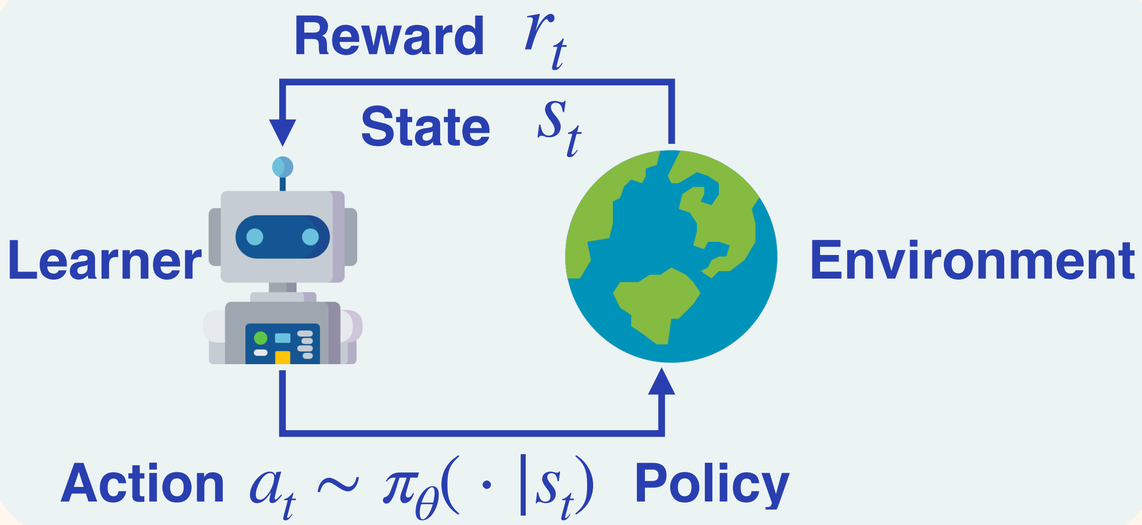
Next iterate θ_t

Information about f at θ_t

Policy Gradient Algorithm

μ : initial state distribution

γ : discount factor



RL objective function

$$f(\theta) \equiv V^{\pi_\theta}(\mu) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 \sim \mu; \pi_\theta\right]$$

Example 2: Non-Uniform PL in Reinforcement Learning

Non-Uniform PL-like Condition (Mei et al., ICML 2020)

Gradient norm — Non-uniformity — Sub-optimality gap

$$\left\| \frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta} \right\|_2 \geq \frac{\min_s \pi_\theta(a^*(s)|s)}{\sqrt{S} \cdot \left\| d_\rho^{\pi^*} / d_\mu^{\pi_\theta} \right\|_\infty} \cdot \left[V^*(\rho) - V^{\pi_\theta}(\rho) \right] .$$

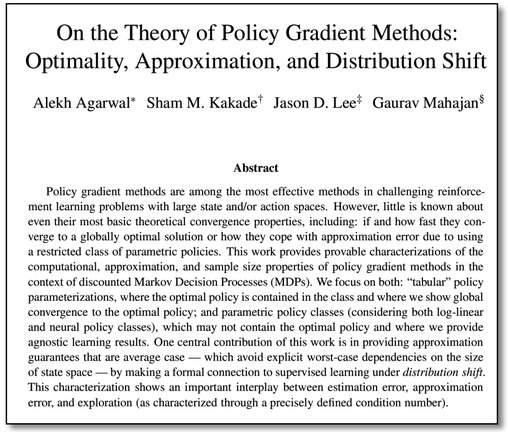
Nuance:

- Gradient could be extremely small if π_θ is far from an optimal one
- This "non-uniformity" results in complicated convergence analysis

Mei et al., "On the global convergence rates of softmax policy gradient methods," ICML 2020

Recent Breakthrough on Policy Gradient Theory in RL

(Agarwal et al., 2019)



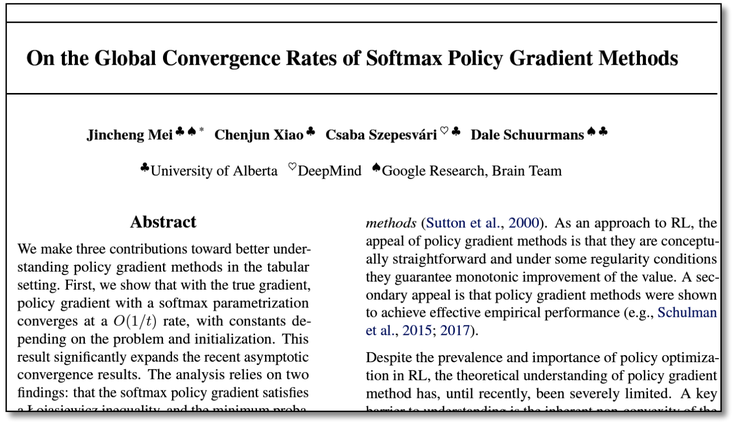
On the Theory of Policy Gradient Methods: Optimality, Approximation, and Distribution Shift

Alekh Agarwal* Sham M. Kakade† Jason D. Lee‡ Gaurav Mahajan§

Abstract

Policy gradient methods are among the most effective methods in challenging reinforcement learning problems with large state and/or action spaces. However, little is known about even their most basic theoretical convergence properties, including: if and how fast they converge to a globally optimal solution or how they cope with approximation error due to using a restricted class of parametric policies. This work provides provable characterizations of the computational, approximation, and sample size properties of policy gradient methods in the context of discounted Markov Decision Processes (MDPs). We focus on both: "tabular" policy parameterizations, where the optimal policy is contained in the class and where we show global convergence to the optimal policy; and parametric policy classes (considering both log-linear and neural policy classes), which may not contain the optimal policy and where we provide agnostic learning results. One central contribution of this work is in providing approximation guarantees that are average case — which avoid explicit worst-case dependencies on the size of state space — by making a formal connection to supervised learning under *distribution shift*. This characterization shows an important interplay between estimation error, approximation error, and exploration (as characterized through a precisely defined condition number).

(Mei et al., 2020)



On the Global Convergence Rates of Softmax Policy Gradient Methods

Jincheng Mei ♣♠* **Chenjun Xiao** ♣ **Csaba Szepesvári** ♡♣ **Dale Schuurmans** ♠♣

♣University of Alberta ♡DeepMind ♠Google Research, Brain Team

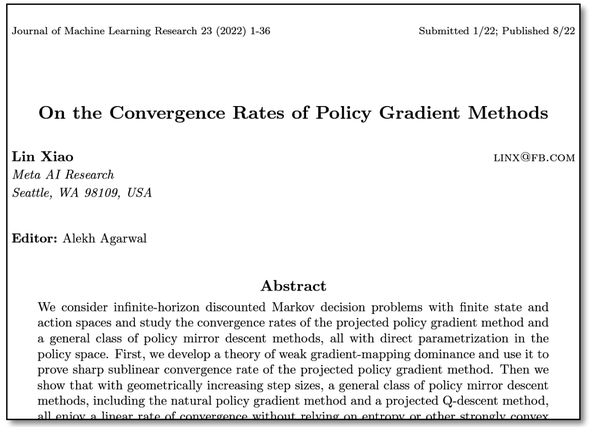
Abstract

We make three contributions toward better understanding policy gradient methods in the tabular setting. First, we show that with the true gradient, policy gradient with a softmax parametrization converges at a $O(1/t)$ rate, with constants depending on the problem and initialization. This result significantly expands the recent asymptotic convergence results. The analysis relies on two findings: that the softmax policy gradient satisfies

methods (Sutton et al., 2000). As an approach to RL, the appeal of policy gradient methods is that they are conceptually straightforward and under some regularity conditions they guarantee monotonic improvement of the value. A secondary appeal is that policy gradient methods were shown to achieve effective empirical performance (e.g., Schulman et al., 2015; 2017).

Despite the prevalence and importance of policy optimization in RL, the theoretical understanding of policy gradient method has, until recently, been severely limited. A key

(Xiao, 2022)



Journal of Machine Learning Research 23 (2022) 1-36 Submitted 1/22; Published 8/22

On the Convergence Rates of Policy Gradient Methods

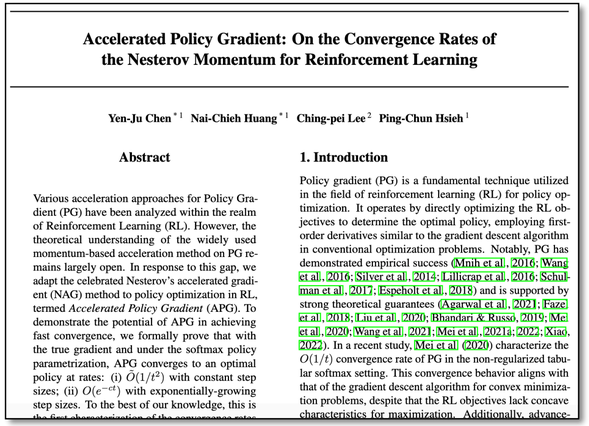
Lin Xiao LINX@FB.COM
Meta AI Research
Seattle, WA 98109, USA

Editor: Alekh Agarwal

Abstract

We consider infinite-horizon discounted Markov decision problems with finite state and action spaces and study the convergence rates of the projected policy gradient method and a general class of policy mirror descent methods, all with direct parametrization in the policy space. First, we develop a theory of weak gradient-mapping dominance and use it to prove sharp sublinear convergence rate of the projected policy gradient method. Then we show that with geometrically increasing step sizes, a general class of policy mirror descent methods, including the natural policy gradient method and a projected Q-descent method, all enjoy a linear rate of convergence without relying on entropy or other strongly convex

(Chen et al., 2024)



Accelerated Policy Gradient: On the Convergence Rates of the Nesterov Momentum for Reinforcement Learning

Yen-Ju Chen [*1] **Nai-Chieh Huang** [*1] **Ching-pei Lee** [2] **Ping-Chun Hsieh** [1]

Abstract

Various acceleration approaches for Policy Gradient (PG) have been analyzed within the realm of Reinforcement Learning (RL). However, the theoretical understanding of the widely used momentum-based acceleration method on PG remains largely open. In response to this gap, we adapt the celebrated Nesterov's accelerated gradient (NAG) method to policy optimization in RL, termed *Accelerated Policy Gradient* (APG). To demonstrate the potential of APG in achieving fast convergence, we formally prove that with the true gradient and under the softmax policy parametrization, APG converges to an optimal policy at rates: (i) $\tilde{O}(1/t^2)$ with constant step sizes; (ii) $O(e^{-ct})$ with exponentially-growing step sizes. To the best of our knowledge, this is

1. Introduction

Policy gradient (PG) is a fundamental technique utilized in the field of reinforcement learning (RL) for policy optimization. It operates by directly optimizing the RL objectives to determine the optimal policy, employing first-order derivatives similar to the gradient descent algorithm in conventional optimization problems. Notably, PG has demonstrated empirical success (Mnih et al., 2016; Wang et al., 2016; Silver et al., 2014; Lillicrap et al., 2016; Schulman et al., 2017; Espeholt et al., 2018) and is supported by strong theoretical guarantees (Agarwal et al., 2021; Fazel et al., 2018; Liu et al., 2020; Bhandari & Russo, 2019; Mei et al., 2020; Wang et al., 2021; Mei et al., 2021a; 2022; Xiao, 2022). In a recent study, Mei et al. (2020) characterize the $O(1/t)$ convergence rate of PG in the non-regularized tabular softmax setting. This convergence behavior aligns with that of the gradient descent algorithm for convex minimization problems, despite that the RL objectives lack concave characteristics for maximization. Additionally, advance-

COLT 2019

Asymptotic convergence to optimum under PG

ICML 2020

The first convergence rate of $O(1/t)$ under PG

JMLR 2022

Similar rates for a large class of PG

ICML 2024

Our recent result: $\tilde{O}(1/t^2)$ under Accelerated PG

GD for *convex* and L-*smooth* objective functions

Convergence of GD for Convex and Smooth Functions

Theorem (Convergence of GD under convexity and smoothness):

Let f be convex and L-smooth. Under GD with constant step sizes $\eta = 1/L$, we have

$$f(x_T) - f(x^*) \leqslant \frac{L}{2T} \cdot \|x_0 - x^*\|^2, \quad \forall T \in \mathbb{N}$$

$= O(\frac{1}{T})$

How is this convergence rate compared to the strongly convex case?

A Useful Tool: "Descent Lemma"

Descent Lemma:

Let f be an L-smooth function (and not necessarily convex). Then, under GD with step size $\eta \leq 1/L$, we have

$$f(x_{t+1}) \leq f(x_t) - \frac{\eta}{2}\|\nabla f(x_t)\|^2$$

$x_{t+1} = x_t - \eta \cdot \nabla f(x_t)$

Proof:

$$f(x_{t+1}) \leq f(x_t) + \nabla f(x_t)^\top (x_{t+1} - x_t) + \frac{L}{2}\|x_{t+1} - x_t\|^2 \quad \cdots\cdots (\text{ by smoothness })$$

$$= f(x_t) + \nabla f(x_t)^\top \big(-\eta \nabla f(x_t)\big) + \frac{L}{2}\big\|\eta \nabla f(x_t)\big\|^2 \quad \cdots\cdots (\text{ by GD })$$

$$= f(x_t) - \frac{\eta}{2}\|\nabla f(x_t)\|^2 \quad \cdots\cdots (\text{ by the step size } \eta \leq \tfrac{1}{L})$$

$\left(-\eta + \frac{L}{2}\eta^2\right) \cdot \|\nabla f(x_t)\|^2$

Proof: Convergence of GD for Convex and Smooth Functions

Step 1: Let's quantify the distance from x^*

$$\|x_t - x^*\|^2 = \left\|\left(x_{t-1} - \eta\,\nabla f(x_{t-1})\right) - x^*)\right\|^2$$

$$= \|x_{t-1} - x^*\|^2 - 2\eta\,\nabla f(x_{t-1})^\top (x_{t-1} - x^*) + \eta^2 \|\nabla f(x_{t-1})\|^2$$

By reorganizing the terms, we have

$$\nabla f(x_{t-1})^\top (x_{t-1} - x^*) = \frac{1}{2\eta}\left(\|x_{t-1} - x^*\|_2^2 - \|x_t - x^*\|_2^2\right) + \frac{\eta}{2}\|\nabla f(x_{t-1})\|_2^2$$

Step 2:

$$f(x_{t-1}) - f(x^*) \le \nabla f(x_{t-1})^\top (x_{t-1} - x^*) = \frac{1}{2\eta}\left(\|x_{t-1} - x^*\|_2^2 - \|x_t - x^*\|_2^2\right) + \frac{\eta}{2}\|\nabla f(x_{t-1})\|_2^2$$

(Why?)

$$\le \frac{1}{2\eta}\left(\|x_{t-1} - x^*\|_2^2 - \|x_t - x^*\|_2^2\right) + (f(x_{t-1}) - f(x_t))$$

(Why?)

This implies

$$(f(x_t) - f(x^*)) \le \frac{1}{2\eta}\left(\|x_{t-1} - x^*\|_2^2 - \|x_t - x^*\|_2^2\right)$$

Proof: Convergence of GD for Convex and Smooth Functions (Cont.)

Step 3: By taking the summation over $t = 1, \cdots, T$

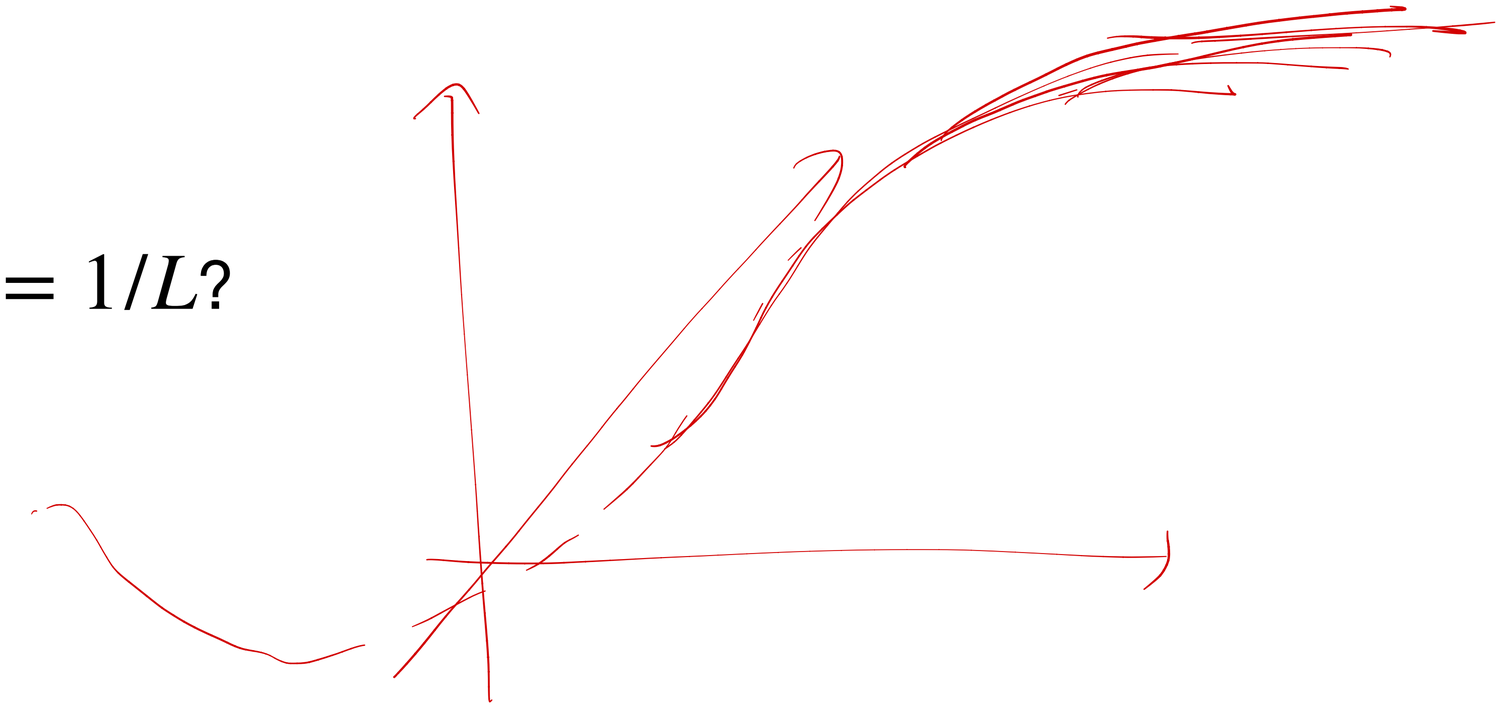
$$\sum_{t=1}^{T} (f(x_t) - f(x^*)) \leq \frac{1}{2\eta} \|x_0 - x^*\|_2^2$$

Since GD is a descent algorithm, we have $f(x_0) \geq f(x_1) \geq \cdots \geq f(x_T)$

$$f(x_T) - f(x^*) \leq \frac{1}{T} \sum_{t=1}^{T} (f(x_t) - f(x^*)) \leq \frac{1}{2T\eta} \|x_0 - x^*\|_2^2 = \frac{L}{2T} \|x_0 - x^*\|_2^2$$

Remarks

- Why selecting the step size $\eta = 1/L$?



- What's the hidden assumption about x^* when we state the result

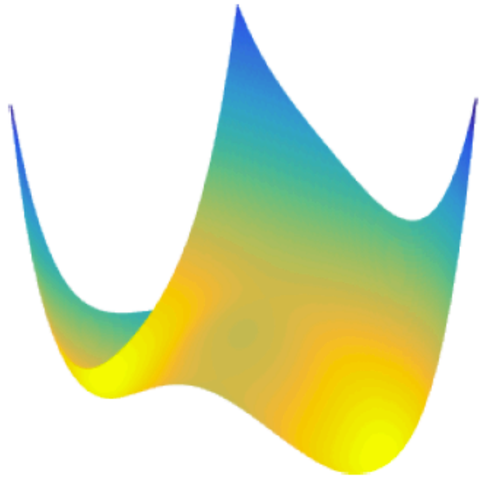
$$f(x_T) - f(x^*) \leqslant \frac{L}{2T} \cdot \|x_0 - x^*\|^2 \text{ ?}$$

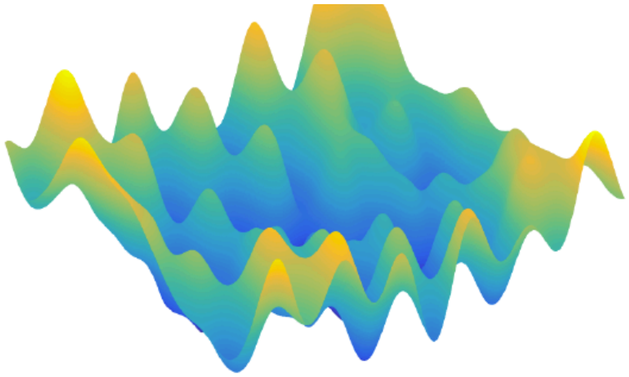
Assume $\|x^*\|_2 < \infty$

$= O\left(\frac{1}{T}\right)$

GD for non-convex and smooth objective functions

GD for General Non-Convex Problems?





- Many machine learning problems are non-convex
  - Mixture models
  - Learning deep neural networks
  - Meta learning (e.g., MAML)

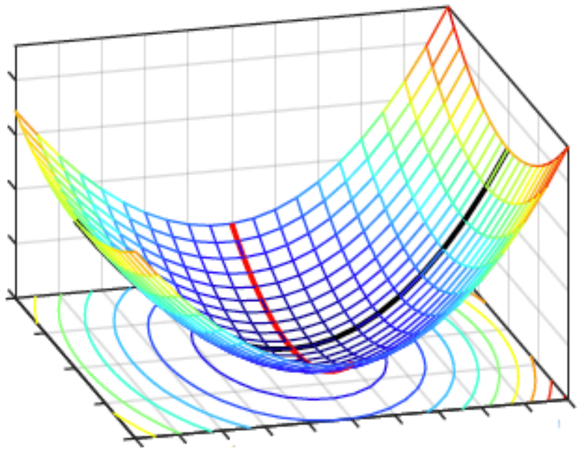
- Challenge
  - Bumps and local minima everywhere
  - No algorithm can solve non-convex problems efficiently in all cases

(Slide Credit: Yuxin Chen)

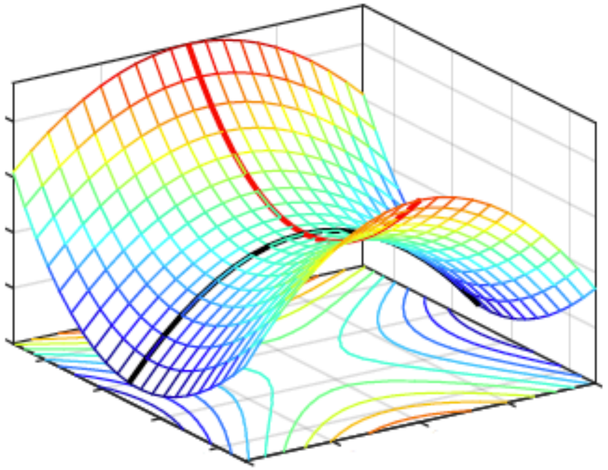
Typical Convergence Guarantees for Non-Convex Problems

- No efficient global convergence to global minima in general
- But, we may still hope for convergence to (nearly-)stationary points (i.e. $\|\nabla f(x)\| \leq \epsilon$)

There are at least 2 types of stationary points



global and local minimum



saddle point

Escaping Saddle Points under GD for Non-Convex Problems

- GD cannot always escape saddle points
- Example: if x^0 happens to be a saddle point, then GD is trapped (as $\nabla f(x^0) = 0$)

- Existing results: Under mild conditions (strict saddle property), randomly initialized GD converges to local minimum with probability 1

  Lee et al., Gradient Descent Only Converges to Minimizers (COLT 2016)

  Pascanu et al., On the saddle point problem for non-convex optimization (NIPS 2014)

- As a result, we are happy with finding a (nearly-)stationary points with

$$\|\nabla f(x)\| \leq \epsilon$$

Convergence of GD for Non-Convex and Smooth Functions

Theorem (Convergence of GD under only smoothness):

Let f be L-smooth. Under GD with constant step sizes $\eta = 1/L$, we have

(1) $\|\nabla f(x_t)\| \to 0, \quad \text{as } t \to \infty$ (asymptotic convergence)

(2) $\min_{0 \le k \le T} \|\nabla f(x_k)\| \le \sqrt{\frac{2L(f(x_0) - f(x^*))}{T}}$ (convergence rate)

"min-iterate" or "best-iterate" convergence $\not\Rightarrow$ "last-iterate" convergence

Implication: GD reaches a nearly-stationary point after sufficiently many iterations

Question: Does (1) imply (2)? And how about vice versa?

Proof: Convergence of GD for Non-Convex and Smooth Functions

Step 1: By the descent lemma under GD with $\eta = 1/L$, we have

$$f(x_{t+1}) \leq f(x_t) - \frac{1}{2L}\|\nabla f(x_t)\|^2$$

$$\frac{1}{2L}\|\nabla f(x_t)\|^2 \leq f(x_t) - f(x_{t+1})$$

Step 2: By taking the telescoping sum of the above,

$$\underbrace{\lim_{T\to\infty} \frac{1}{2L}\sum_{k=0}^{T-1}\|\nabla f(x_t)\|^2}_{LHS} \leq \lim_{T\to\infty} f(x_0) - f(x_T)$$

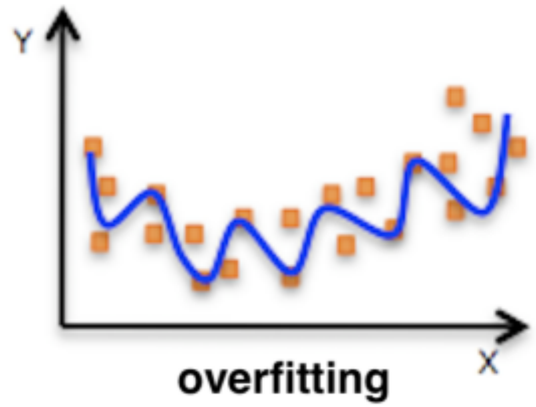
Step3: As LHS is non-negative and upper bounded, we know $\lim_{t\to\infty}\|\nabla f(x_t)\| = 0$

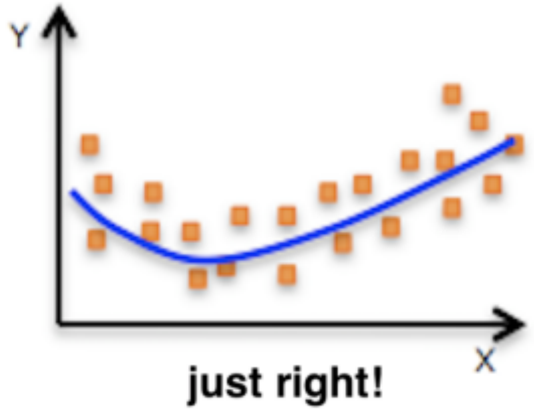
Moreover, $$\min_{0\leq k\leq T}\|\nabla f(x_k)\|^2 \leq \frac{1}{T}\sum_{k=0}^{T-1}\|\nabla f(x_t)\|^2 \leq \frac{2L(f(x_0) - f(x^*))}{T}$$

GD for Fully-Connected Neural Nets

Gradient Descent vs Neural Networks

- Back in 2010-2016, deep neural nets has shown significant empirical success in supervised learning
- Since 2016, a lot of research interests are focused on the question: "Why can GD converge for neural nets?"
- Answer: "overparameterization" (model size $\gg$ training samples)





(Figure Source: Mahdi Soltanolkotabi)

Example: Overparametrized Non-Linear Least-Squares

Non-linear least-squares regression: Given n data samples $\{x_i \in \mathbb{R}^p, y_i \in \mathbb{R}\}$, find a nonlinear model $f(\theta)$ by minimizing

$$L(\theta) = \frac{1}{2}\sum_{i=1}^{n}(f(x_i;\theta) - y_i)^2 = \frac{1}{2}\|\boldsymbol{f}(\theta) - \boldsymbol{y}\|^2$$

where $\boldsymbol{y} := [y_1, \cdots, y_n]^\top, f(\theta) := [f(x_1, \theta), \cdots, f(x_n;\theta)]^\top$

Overparameterization: Model dimension p > sample size n

Run GD on this nonlinear least-squares problem: $\theta_{t+1} = \theta_t - \eta_t \nabla L(\theta_t)$

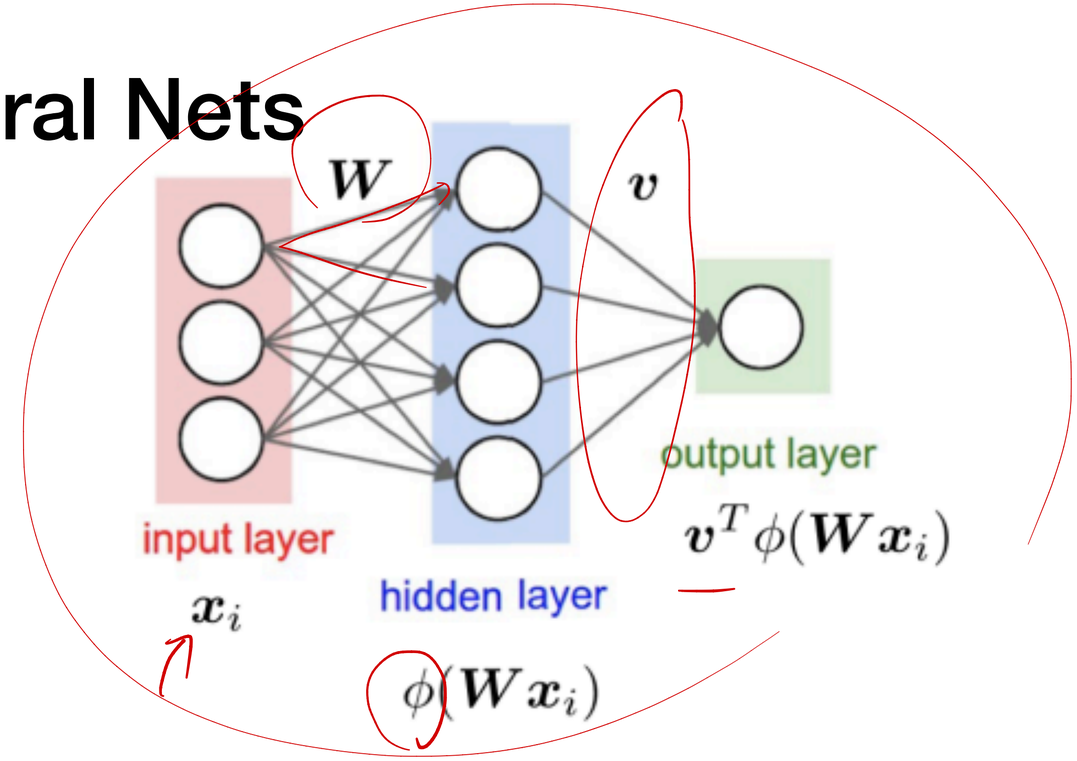
Gradient and Jacobian: $\nabla L(\theta) = J(\theta)^\top(\boldsymbol{f}(\theta) - \boldsymbol{y})$ (Compared to linear case?)

where $J(\theta) = \dfrac{\partial \boldsymbol{f}(\theta)}{\partial \theta} \in \mathbb{R}^{n\times p}$ is called the Jacobian $= [\nabla f(x_1;\theta) \cdots \nabla f(x_n;\theta)]$

Samet Oymak and Mahdi Soltanolkotabi, “Overparameterized Nonlinear Learning: Gradient Descent Takes the Shortest Path?,” ICML 2019

Example: One-Hidden Layer Neural Nets

- Training data:
  $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)$
- Loss:
  $\mathcal{L}(\boldsymbol{v}, \boldsymbol{W}) := \sum_{i=1}^{n} \left(\boldsymbol{v}^T \phi(\boldsymbol{W}\boldsymbol{x}_i) - y_i\right)^2$
- Algorithm: gradient descent
  with random Gaussian initialization



Theorem (Oymak and Soltanolkotabi 2019)

As long as

$$\#\text{parameters} \gtrsim (\#\text{of training data})^2$$

Then, with high probability

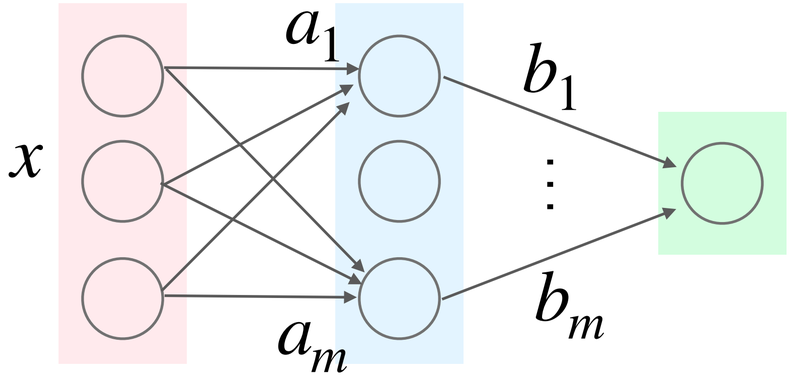
- *Zero training error:* $\mathcal{L}(\boldsymbol{v}_\tau, \boldsymbol{W}_\tau) \leq (1-\rho)^\tau \mathcal{L}(\boldsymbol{v}_0, \boldsymbol{W}_0)$
- *Iterates remain close to initialization*

Samet Oymak and Mahdi Soltanolkotabi, "Towards moderate overparameterization: global convergence guarantees for training shallow neural networks," 2019

(Slide Credit: Mahdi Soltanolkotabi)

An Alternative Explanation: Neural Tangent Kernel

An empirical observation about NNs:



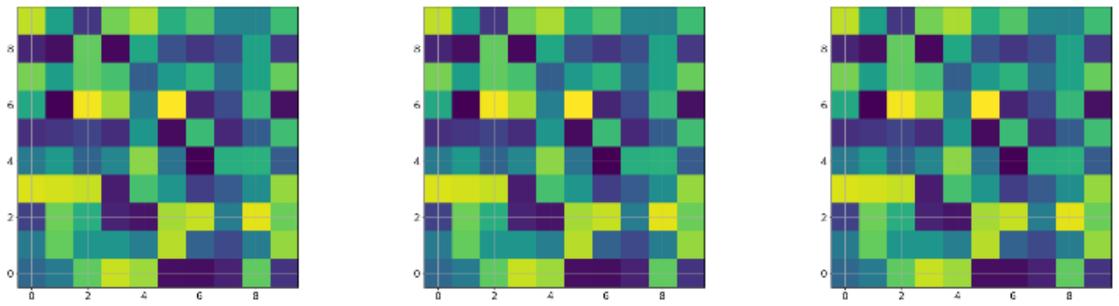
$$f_m(x;\Theta) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} b_i \cdot \sigma(a_i^\top x)$$

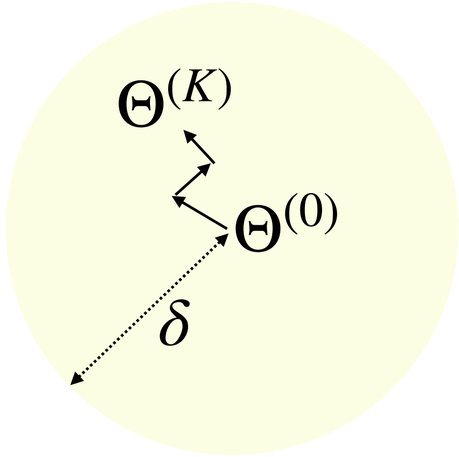
(σ: ReLU activation)

($\Theta \equiv \{[a_i]_{i=1}^m, [b_i]_{i=1}^m\}$

(Random Initialization)

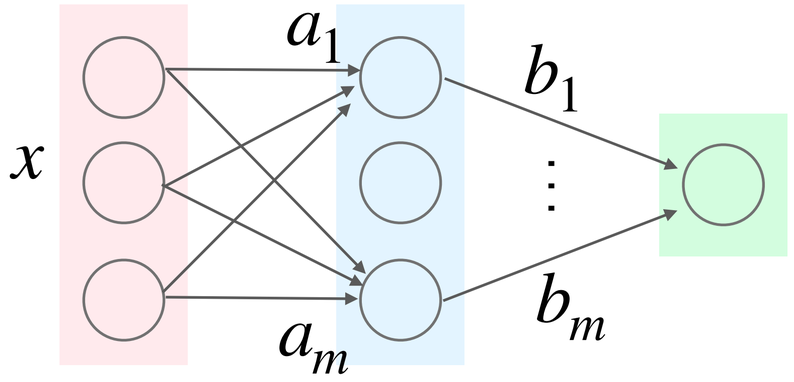
$\Theta^{(0)}$ —GD→ $\Theta^{(1)}$ —GD→ $\Theta^{(2)}$





NN parameters “almost” static (under large m)

A Primer on Neural Tangent Kernel (NTK)



$$f_m(x;\Theta) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} b_i \cdot \sigma(a_i^\top x)$$

(σ: ReLU activation)

($\Theta \equiv \{[a_i]_{i=1}^m, [b_i]_{i=1}^m\}$

First-order Taylor expansion:

$$f_m(x;\Theta) = f_m(x;\Theta^{(0)}) + \nabla_\Theta f_m(x;\Theta^{(0)})^\top (\Theta - \Theta^{(0)}) + O(\|\Theta - \Theta^{(0)}\|^2)$$

Viewed as a feature map in kernel methods

(NTK)

Neural tangent kernel function [Jacot et al., 2018]**:**

$$\mathbf{H}_m(x,x') := \langle \nabla_\Theta f_m(x;\Theta^{(0)}), \nabla_\Theta f_m(x';\Theta^{(0)}) \rangle$$

$$\downarrow m \to \infty$$

$$\mathbf{H}(x,x') = \mathbb{E}_{a,b}[b^2 \sigma'(a^\top x)\sigma'(a^\top x')\langle x, x'\rangle] + \mathbb{E}_a[\sigma(a^\top x)\sigma(a^\top x')]$$

Minimize squared loss

$(\Theta-\Theta^0)^\top H^{-1}(\Theta-\Theta^0)$

Jacot, Gabriel, and Hongler, "Neural Tangent Kernel: Convergence and Generalization in Neural Networks," NeurIPS 2018

References: GD in Overparameterization Regime

(ICLR 2019)

GRADIENT DESCENT PROVABLY OPTIMIZES OVER-PARAMETERIZED NEURAL NETWORKS

Simon S. Du*
Machine Learning Department
Carnegie Mellon University
ssdu@cs.cmu.edu

Xiyu Zhai*
Department of EECS
Massachusetts Institute of Technology
xiyuzhai@mit.edu

Barnabás Poczós
Machine Learning Department
Carnegie Mellon University
bapozos@cs.cmu.edu

Aarti Singh
Machine Learning Department
Carnegie Mellon University
aartisingh@cmu.edu

(NeurIPS 2018)

Neural Tangent Kernel: Convergence and Generalization in Neural Networks

Arthur Jacot
École Polytechnique Fédérale de Lausanne
arthur.jacot@netopera.net

Franck Gabriel
Imperial College London and École Polytechnique Fédérale de Lausanne
franckrgabriel@gmail.com

Clément Hongler
École Polytechnique Fédérale de Lausanne
clement.hongler@gmail.com

(ICML 2019)

Overparameterized Nonlinear Learning: Gradient Descent Takes the Shortest Path?

Samet Oymak [1] **Mahdi Soltanolkotabi** [2]