Introduction to non-convex optimization

Yuanzhi Li

Assistant Professor, Carnegie Mellon University

Random Date

A bit history of the speaker

- Current name: Yuanzhi Li.

A bit history of the speaker

- Current name: Yuanzhi Li.
- Previously used Names: Nan.

A bit history of the speaker

- Current name: Yuanzhi Li.
- Previously used Names: Nan.
- Current Age: 27.

A bit history of the speaker

- Current name: Yuanzhi Li.
- Previously used Names: Nan.
- Current Age: 27.
- Previous ages: from 0 to 26.

A bit history of the speaker

- Current name: Yuanzhi Li.
- Previously used Names: Nan.
- Current Age: 27.
- Previous ages: from 0 to 26.
- Current Position: First year Assistant Professor.

A bit history of the speaker

- Current name: Yuanzhi Li.
- Previously used Names: Nan.
- Current Age: 27.
- Previous ages: from 0 to 26.
- Current Position: First year Assistant Professor.
- Previous positions:

A bit history of the speaker

- Current name: Yuanzhi Li.
- Previously used Names: Nan.
- Current Age: 27.
- Previous ages: from 0 to 26.
- Current Position: First year Assistant Professor.
- Previous positions:
  - Postdoc (Stanford).

A bit history of the speaker

- Current name: Yuanzhi Li.
- Previously used Names: Nan.
- Current Age: 27.
- Previous ages: from 0 to 26.
- Current Position: First year Assistant Professor.
- Previous positions:
  - Postdoc (Stanford).
  - PhD (Princeton).

A bit history of the speaker

- Current name: Yuanzhi Li.
- Previously used Names: Nan.
- Current Age: 27.
- Previous ages: from 0 to 26.
- Current Position: First year Assistant Professor.
- Previous positions:
  - Postdoc (Stanford).
  - PhD (Princeton).
  - Bachelor(Tsinghua).

A bit history of the speaker

- Current name: Yuanzhi Li.
- Previously used Names: Nan.
- Current Age: 27.
- Previous ages: from 0 to 26.
- Current Position: First year Assistant Professor.
- Previous positions:
  - Postdoc (Stanford).
  - PhD (Princeton).
  - Bachelor(Tsinghua).
  - High school + middle school(The experimental school attached to Beijing Normal University).

A bit history of the speaker

- Current name: Yuanzhi Li.
- Previously used Names: Nan.
- Current Age: 27.
- Previous ages: from 0 to 26.
- Current Position: First year Assistant Professor.
- Previous positions:
  - Postdoc (Stanford).
  - PhD (Princeton).
  - Bachelor(Tsinghua).
  - High school + middle school(The experimental school attached to Beijing Normal University).
  - Elementary School(No.1 Fucheng Elementary School).

A bit history of the speaker

- Current name: Yuanzhi Li.
- Previously used Names: Nan.
- Current Age: 27.
- Previous ages: from 0 to 26.
- Current Position: First year Assistant Professor.
- Previous positions:
    - Postdoc (Stanford).
    - PhD (Princeton).
    - Bachelor(Tsinghua).
    - High school + middle school(The experimental school attached to Beijing Normal University).
    - Elementary School(No.1 Fucheng Elementary School).
    - Kindergarten(Shuguang Kindergarten).

Declare

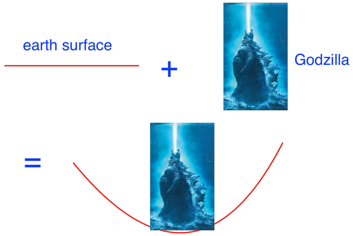
- This lecture is based on the paper “Neon2” by Zeyuan Allen-Zhu and myself (https://arxiv.org/abs/1711.06673) . Please do distribute.

Convex optimization

- Where is the Godzilla?

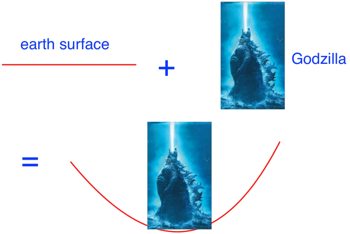
Convex optimization

- Where is the Godzilla?



Convex optimization

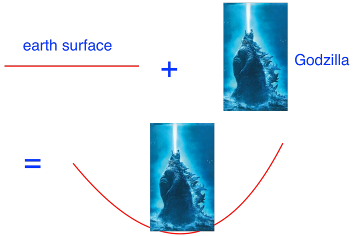
- Where is the Godzilla?



-
- Weight of the Godzilla: The smoothness / strong convexity.

Convex optimization

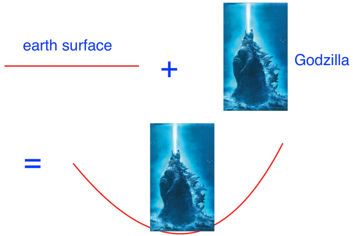
- Where is the Godzilla?



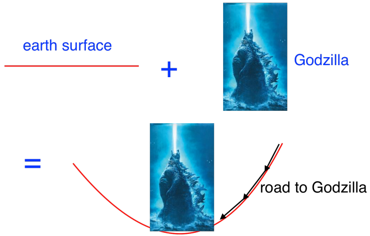
-
- Weight of the Godzilla: The smoothness / strong convexity.
- To find the Godzilla: follow the (negative) gradient direction.

Convex optimization

- Where is the Godzilla?



-
- Weight of the Godzilla: The smoothness / strong convexity.
- To find the Godzilla: follow the (negative) gradient direction.



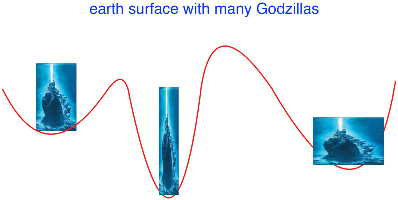
-

Non convex optimization

- Where are the Godzillas?

Non convex optimization

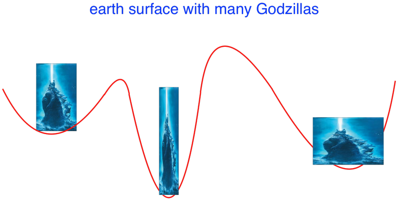
- Where are the Godzillas?



-

Non convex optimization

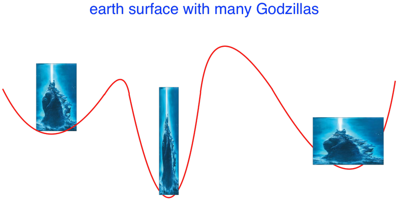
- Where are the Godzillas?



-
- Each Godzilla defines a local minima.

Non convex optimization

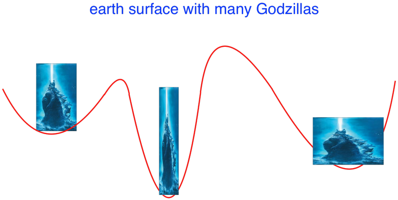
- Where are the Godzillas?



-
- Each Godzilla defines a local minima.
- The "heaviest" Godzilla: The global minima.

Non convex optimization

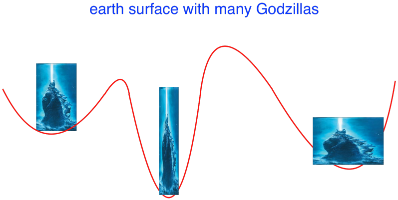
- Where are the Godzillas?



-
- Each Godzilla defines a local minima.
- The "heaviest" Godzilla: The global minima.
- 

Non convex optimization

- Where are the Godzillas?



-
- Each Godzilla defines a local minima.
- The "heaviest" Godzilla: The global minima.
- 

- Non-convex optimization: Can we find these Godzillas?

Non convex optimization

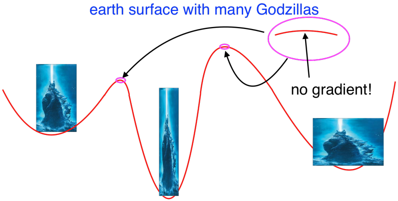
- Naive approach: Follow the (negative) gradient direction?

Non convex optimization

- Naive approach: Follow the (negative) gradient direction?
- Might not be able to find a single one!

Non convex optimization

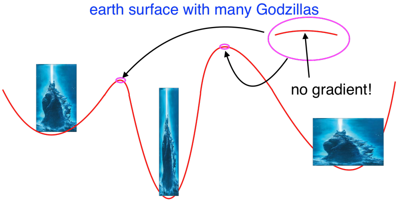
- Naive approach: Follow the (negative) gradient direction?
- Might not be able to find a single one!



-

Non convex optimization

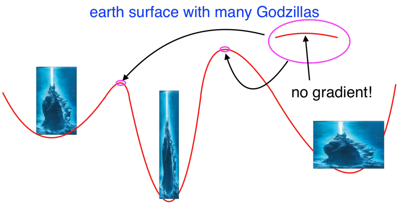
- Naive approach: Follow the (negative) gradient direction?
- Might not be able to find a single one!



-
- These are “saddle points”.

Non convex optimization

- Naive approach: Follow the (negative) gradient direction?
- Might not be able to find a single one!



-
- These are "saddle points".
- In fact, in high dimension, one can construct a function where gradient descent almost always stucks at a saddle point.

Non convex optimization: The goals

- Goal 1: Find at least one Godzilla, as fast as possible.

Non convex optimization: The goals

- Goal 1: Find at least one Godzilla, as fast as possible.
- Goal 2: Find the “heaviest” Godzilla.

Non convex optimization: The goals

- Goal 1: Find at least one Godzilla, as fast as possible.
- Goal 2: Find the "heaviest" Godzilla.
- Goal 1 can be done efficiently (the focus of this lecture).

Non convex optimization: The goals

- Goal 1: Find at least one Godzilla, as fast as possible.
- Goal 2: Find the "heaviest" Godzilla.
- Goal 1 can be done efficiently (the focus of this lecture).
- Goal 2 is in general hard, but possible in some settings (beyond this lecture, come to my course next semester if you want to know more).

Non convex optimization: Before going to the math

- Where do we use non-convex optimization? Why *** do we need to learn it?

Non convex optimization: Before going to the math

- Where do we use non-convex optimization? Why *** do we need to learn it?
- You didn't need to learn it – at least when it was ten years ago.

Non convex optimization: Before going to the math

- Where do we use non-convex optimization? Why *** do we need to learn it?
- You didn't need to learn it – at least when it was ten years ago.
  - The problems solved in practice, especially in machine learning/statistics, are mostly convex.

Non convex optimization: Before going to the math

- Where do we use non-convex optimization? Why *** do we need to learn it?
- You didn't need to learn it – at least when it was ten years ago.
  - The problems solved in practice, especially in machine learning/statistics, are mostly convex.
    - Linear regression, logistic regression;

Non convex optimization: Before going to the math

- Where do we use non-convex optimization? Why *** do we need to learn it?
- You didn't need to learn it – at least when it was ten years ago.
  - The problems solved in practice, especially in machine learning/statistics, are mostly convex.
    - Linear regression, logistic regression;
    - Kernel methods;

Non convex optimization: Before going to the math

- Where do we use non-convex optimization? Why *** do we need to learn it?
- You didn't need to learn it – at least when it was ten years ago.
  - The problems solved in practice, especially in machine learning/statistics, are mostly convex.
    - Linear regression, logistic regression;
    - Kernel methods;
    - Linear programming, semi-definite programming, SOS (Sum Of Squares programming);

Non convex optimization: Before going to the math

- Where do we use non-convex optimization? Why *** do we need to learn it?
- You didn't need to learn it – at least when it was ten years ago.
  - The problems solved in practice, especially in machine learning/statistics, are mostly convex.
    - Linear regression, logistic regression;
    - Kernel methods;
    - Linear programming, semi-definite programming, SOS (Sum Of Squares programming);
  - But now, they are mostly non-convex, mainly for one reason:

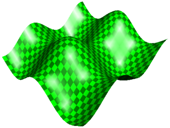
Non convex optimization: Before going to the math

- Where do we use non-convex optimization? Why *** do we need to learn it?
- You didn't need to learn it – at least when it was ten years ago.
  - The problems solved in practice, especially in machine learning/statistics, are mostly convex.
    - Linear regression, logistic regression;
    - Kernel methods;
    - Linear programming, semi-definite programming, SOS (Sum Of Squares programming);
  - But now, they are mostly non-convex, mainly for one reason:
  - Deep learning / Neural networks.

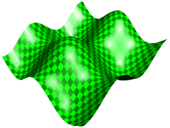
Non convex optimization: Before going to the math

- Where do we use non-convex optimization? Why *** do we need to learn it?
- You didn't need to learn it – at least when it was ten years ago.
  - The problems solved in practice, especially in machine learning/statistics, are mostly convex.
    - Linear regression, logistic regression;
    - Kernel methods;
    - Linear programming, semi-definite programming, SOS (Sum Of Squares programming);
  - But now, they are mostly non-convex, mainly for one reason:
  - Deep learning / Neural networks.
- Non-convex landscape:

Non convex optimization: Before going to the math

- Where do we use non-convex optimization? Why *** do we need to learn it?
- You didn't need to learn it – at least when it was ten years ago.
  - The problems solved in practice, especially in machine learning/statistics, are mostly convex.
    - Linear regression, logistic regression;
    - Kernel methods;
    - Linear programming, semi-definite programming, SOS (Sum Of Squares programming);
  - But now, they are mostly non-convex, mainly for one reason:
  - Deep learning / Neural networks.
- Non-convex landscape:
- 

Non convex optimization: Before going to the math

- Where do we use non-convex optimization? Why *** do we need to learn it?
- You didn't need to learn it – at least when it was ten years ago.
  - The problems solved in practice, especially in machine learning/statistics, are mostly convex.
    - Linear regression, logistic regression;
    - Kernel methods;
    - Linear programming, semi-definite programming, SOS (Sum Of Squares programming);
  - But now, they are mostly non-convex, mainly for one reason:
  - Deep learning / Neural networks.
- Non-convex landscape:
- 
- What can we say in this regime?

Non convex optimization: The definition

- We start with the definitions: smoothness, hessian Lipschitzness, local minima, saddle points etc.

Non convex optimization: The definition

- We start with the definitions: smoothness, hessian Lipschitzness, local minima, saddle points etc.
- Given a second-order differentiable function $f : \mathbb{R}^d \to \mathbb{R}$:

Non convex optimization: The definition

- We start with the definitions: smoothness, hessian Lipschitzness, local minima, saddle points etc.
- Given a second-order differentiable function $f : \mathbb{R}^d \to \mathbb{R}$:
- We can do a local taylor expansion of the function around any point x:

Non convex optimization: The definition

- We start with the definitions: smoothness, hessian Lipschitzness, local minima, saddle points etc.
- Given a second-order differentiable function $f : \mathbb{R}^d \to \mathbb{R}$:
- We can do a local taylor expansion of the function around any point x:
- $f(x+\tau) = f(x) + \langle \nabla f(x), \tau \rangle + \frac{1}{2}\tau^\top \nabla^2 f(x) \tau \pm O(\|\tau\|_2^3)$. $\| * \|_2$ is the Euclidean norm.

Non convex optimization: The definition

- We start with the definitions: smoothness, hessian Lipschitzness, local minima, saddle points etc.
- Given a second-order differentiable function $f : \mathbb{R}^d \to \mathbb{R}$:
- We can do a local taylor expansion of the function around any point x:
- $f(x+\tau) = f(x) + \langle \nabla f(x), \tau \rangle + \frac{1}{2}\tau^\top \nabla^2 f(x)\tau \pm O(\|\tau\|_2^3)$. $\| * \|_2$ is the Euclidean norm.
- Here, $a = b \pm c$ means $a \in [b-c, b+c]$.

Non convex optimization: The definition

- We start with the definitions: smoothness, hessian Lipschitzness, local minima, saddle points etc.
- Given a second-order differentiable function $f : \mathbb{R}^d \to \mathbb{R}$:
- We can do a local taylor expansion of the function around any point x:
- $f(x+\tau) = f(x) + \langle \nabla f(x), \tau \rangle + \frac{1}{2}\tau^\top \nabla^2 f(x) \tau \pm O(\|\tau\|_2^3)$. $\| * \|_2$ is the Euclidean norm.
- Here, $a = b \pm c$ means $a \in [b-c, b+c]$.
- Define: Lipschitzness: $L = \sup_{x \in \mathbb{R}^d} \|\nabla f(x)\|_2$.

Non convex optimization: The definition

- We start with the definitions: smoothness, hessian Lipschitzness, local minima, saddle points etc.
- Given a second-order differentiable function $f : \mathbb{R}^d \to \mathbb{R}$:
- We can do a local taylor expansion of the function around any point x:
- $f(x + \tau) = f(x) + \langle \nabla f(x), \tau \rangle + \frac{1}{2}\tau^\top \nabla^2 f(x)\tau \pm O(\|\tau\|_2^3)$. $\| * \|_2$ is the Euclidean norm.
- Here, $a = b \pm c$ means $a \in [b - c, b + c]$.
- Define: Lipschitzness: $L = \sup_{x \in \mathbb{R}^d} \|\nabla f(x)\|_2$.
- Lipschitzness implies: $|f(x) - f(y)| \leq L\|x - y\|_2$, for every $x, y \in \mathbb{R}$.

Non convex optimization: The definition

- Define: Smoothness $\beta = \sup_{x \in \mathbb{R}^d} \|\nabla^2 f(x)\|_{sp}$. $\| * \|_{sp}$ is the spectral norm.

Non convex optimization: The definition

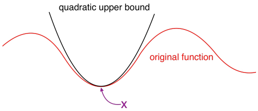
- Define: Smoothness $\beta = \sup_{x \in \mathbb{R}^d} \|\nabla^2 f(x)\|_{sp}$. $\| * \|_{sp}$ is the spectral norm.
- Smoothness implies:

Non convex optimization: The definition

- Define: Smoothness $\beta = \sup_{x \in \mathbb{R}^d} \|\nabla^2 f(x)\|_{sp}$. $\| * \|_{sp}$ is the spectral norm.
- Smoothness implies:
  - (Upper quadratic bound): For all $x, y \in \mathbb{R}^d$, $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\beta}{2}\|x - y\|_2^2$.

Non convex optimization: The definition

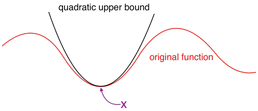
- Define: Smoothness $\beta = \sup_{x \in \mathbb{R}^d} \|\nabla^2 f(x)\|_{sp}$. $\| * \|_{sp}$ is the spectral norm.
- Smoothness implies:
  - (Upper quadratic bound): For all $x, y \in \mathbb{R}^d$, $f(y) \le f(x) + \langle \nabla f(x), y - x \rangle + \frac{\beta}{2}\|x - y\|_2^2$.



  -

Non convex optimization: The definition

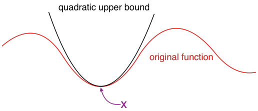
- Define: Smoothness $\beta = \sup_{x \in \mathbb{R}^d} \|\nabla^2 f(x)\|_{sp}$. $\| * \|_{sp}$ is the spectral norm.
- Smoothness implies:
  - (Upper quadratic bound): For all $x, y \in \mathbb{R}^d$, $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\beta}{2}\|x - y\|_2^2$.



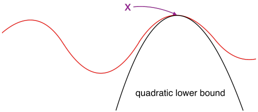
  -
  - (Lower quadratic bound): For all $x, y \in \mathbb{R}^d$, $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle - \frac{\beta}{2}\|x - y\|_2^2$.

Non convex optimization: The definition

- Define: Smoothness $\beta = \sup_{x \in \mathbb{R}^d} \|\nabla^2 f(x)\|_{sp}$. $\| * \|_{sp}$ is the spectral norm.
- Smoothness implies:
  - (Upper quadratic bound): For all $x, y \in \mathbb{R}^d$, $f(y) \le f(x) + \langle \nabla f(x), y - x \rangle + \frac{\beta}{2} \|x - y\|_2^2$.



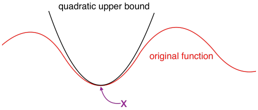
  -
  - (Lower quadratic bound): For all $x, y \in \mathbb{R}^d$, $f(y) \ge f(x) + \langle \nabla f(x), y - x \rangle - \frac{\beta}{2} \|x - y\|_2^2$.



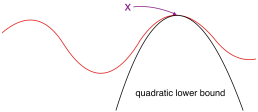
  -

Non convex optimization: The definition

- Define: Smoothness $\beta = \sup_{x \in \mathbb{R}^d} \|\nabla^2 f(x)\|_{sp}$. $\| * \|_{sp}$ is the spectral norm.
- Smoothness implies:
  - (Upper quadratic bound): For all $x, y \in \mathbb{R}^d$, $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\beta}{2} \|x - y\|_2^2$.



  -
  - (Lower quadratic bound): For all $x, y \in \mathbb{R}^d$, $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle - \frac{\beta}{2} \|x - y\|_2^2$.



  -
  - Note: For convex f, one shall have (lower linear bound): $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$.

Non convex optimization: The definition

- The Lipschitzness of the hessian γ: For all $x, y \in \mathbb{R}^d$, $\|\nabla^2 f(x) - \nabla^2 f(y)\|_{sp} \le \gamma \|x - y\|_2$.

Non convex optimization: The definition

- The Lipschitzness of the hessian γ: For all $x, y \in \mathbb{R}^d$, $\|\nabla^2 f(x) - \nabla^2 f(y)\|_{sp} \le \gamma \|x - y\|_2$.
- This implies (important): For every $x, \tau \in \mathbb{R}^d$:

$$f(x+\tau) = f(x) + \langle \nabla f(x), \tau \rangle + \frac{1}{2}\tau^\top \nabla^2 f(x) \tau \pm \gamma \|\tau\|_2^3$$

Non convex optimization: The definition

- The Lipschitzness of the hessian γ: For all $x, y \in \mathbb{R}^d$, $\|\nabla^2 f(x) - \nabla^2 f(y)\|_{sp} \le \gamma \|x - y\|_2$.
- This implies (important): For every $x, \tau \in \mathbb{R}^d$:

$$f(x+\tau) = f(x) + \langle \nabla f(x), \tau \rangle + \frac{1}{2}\tau^\top \nabla^2 f(x) \tau \pm \gamma \|\tau\|_2^3$$

- $\nabla^2 f(x)$ might not be positive semi-definite (PSD)! (Convex function $\iff$ $\nabla^2 f(x)$ is PSD for almost every x).

Non convex optimization: The property

- We proceed to define local minima, saddle points etc.

Non convex optimization: The property

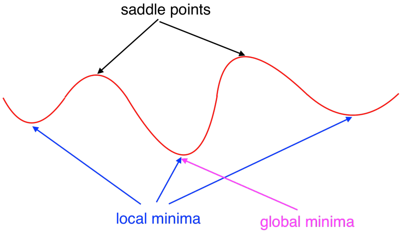
- We proceed to define local minima, saddle points etc.
- For convex function f: $\nabla f(x) = 0 \iff x$ is the global minima (e.g. $f(x) = \min_{y \in \mathbb{R}^d} f(y)$).

Non convex optimization: The property

- We proceed to define local minima, saddle points etc.
- For convex function f: $\nabla f(x) = 0 \iff x$ is the global minima (e.g. $f(x) = \min_{y \in \mathbb{R}^d} f(y)$).
- What about non-convex functions? $\nabla f(x) = 0$ implies?

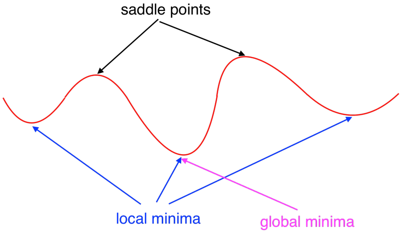
Non convex optimization: The property

- We proceed to define local minima, saddle points etc.
- For convex function f: $\nabla f(x) = 0 \iff x$ is the global minima (e.g. $f(x) = \min_{y \in \mathbb{R}^d} f(y)$).
- What about non-convex functions? $\nabla f(x) = 0$ implies?



-

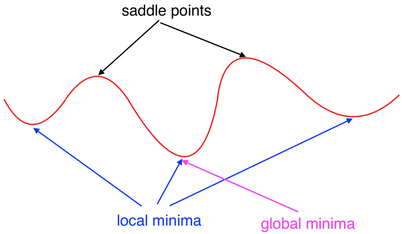
Non convex optimization: The property

- We proceed to define local minima, saddle points etc.
- For convex function f: $\nabla f(x) = 0 \iff x$ is the global minima (e.g. $f(x) = \min_{y \in \mathbb{R}^d} f(y)$).
- What about non-convex functions? $\nabla f(x) = 0$ implies?
- 

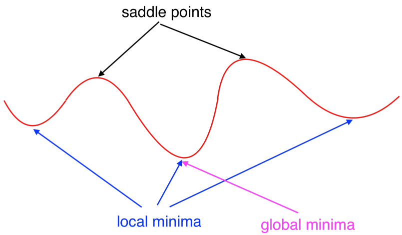
- Global minima, local minima, saddle points.

Non convex optimization: The property

- Non-convex landscape:

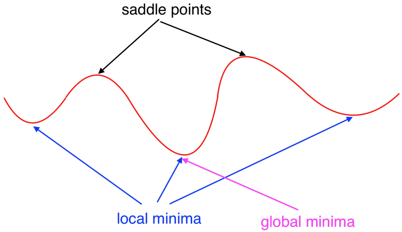


Non convex optimization: The property



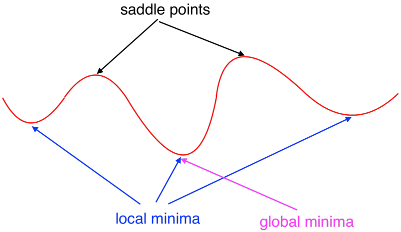
- Non-convex landscape:
- local minima (second-order local minima):

Non convex optimization: The property



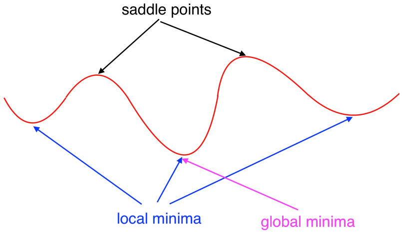
- Non-convex landscape:
- local minima (second-order local minima):
  - $\nabla f(x) = 0$ and $\nabla^2 f(x)$ is PSD (positive semi-definite, i.e. $\nabla^2 f(x) \succeq 0$).

Non convex optimization: The property



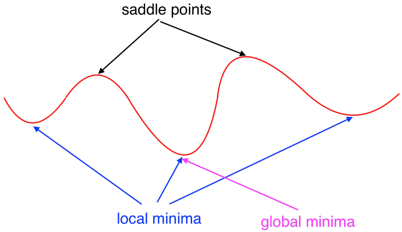
- Non-convex landscape:
- local minima (second-order local minima):
  - $\nabla f(x) = 0$ and $\nabla^2 f(x)$ is PSD (positive semi-definite, i.e. $\nabla^2 f(x) \succeq 0$).
- saddle point:

Non convex optimization: The property



- Non-convex landscape:
- local minima (second-order local minima):
  - $\nabla f(x) = 0$ and $\nabla^2 f(x)$ is PSD (positive semi-definite, i.e. $\nabla^2 f(x) \succeq 0$).
- saddle point:
  - $\nabla f(x) = 0$ and $\nabla^2 f(x)$ is not PSD.

Non convex optimization: The property



- Non-convex landscape:
- local minima (second-order local minima):
  - $\nabla f(x) = 0$ and $\nabla^2 f(x)$ is PSD (positive semi-definite, i.e. $\nabla^2 f(x) \succeq 0$).
- saddle point:
  - $\nabla f(x) = 0$ and $\nabla^2 f(x)$ is not PSD.
  - There exists a $v \in \mathbb{R}^d$ such that $v^\top \nabla^2 f(x) v < 0$.

Non convex optimization: The goal

- What do we want when optimizing a non-convex function f?

Non convex optimization: The goal

- What do we want when optimizing a non-convex function f?
- Finding the global minima is in general impossible (NP-hard) for non-convex functions.

Non convex optimization: The goal

- What do we want when optimizing a non-convex function f?
- Finding the global minima is in general impossible (NP-hard) for non-convex functions.
- Goal: Given a function $f : \mathbb{R}^d \to \mathbb{R}$, can we find a local minima efficiently?

Non convex optimization: The goal

- What do we want when optimizing a non-convex function f?
- Finding the global minima is in general impossible (NP-hard) for non-convex functions.
- Goal: Given a function $f : \mathbb{R}^d \to \mathbb{R}$, can we find a local minima efficiently?
- Given a function $f : \mathbb{R}^d \to \mathbb{R}$ that is β-smooth and γ- Lipschitz Hessian, for every $\varepsilon, \delta > 0$, find a point x such that:

- What do we want when optimizing a non-convex function f?
- Finding the global minima is in general impossible (NP-hard) for non-convex functions.
- Goal: Given a function $f : \mathbb{R}^d \to \mathbb{R}$, can we find a local minima efficiently?
- Given a function $f : \mathbb{R}^d \to \mathbb{R}$ that is β-smooth and γ- Lipschitz Hessian, for every $\varepsilon, \delta > 0$, find a point x such that:
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.

Non convex optimization: The goal

- What do we want when optimizing a non-convex function f?
- Finding the global minima is in general impossible (NP-hard) for non-convex functions.
- Goal: Given a function $f : \mathbb{R}^d \to \mathbb{R}$, can we find a local minima efficiently?
- Given a function $f : \mathbb{R}^d \to \mathbb{R}$ that is β-smooth and γ- Lipschitz Hessian, for every $\varepsilon, \delta > 0$, find a point x such that:
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.

Non convex optimization: The goal

- What do we want when optimizing a non-convex function f?
- Finding the global minima is in general impossible (NP-hard) for non-convex functions.
- Goal: Given a function $f : \mathbb{R}^d \to \mathbb{R}$, can we find a local minima efficiently?
- Given a function $f : \mathbb{R}^d \to \mathbb{R}$ that is β-smooth and γ- Lipschitz Hessian, for every $\varepsilon, \delta > 0$, find a point x such that:
  - $\|\nabla f(x)\|_2 \le \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- In time poly$(1/\varepsilon, 1/\delta, \gamma, \beta, d)$.

Non convex optimization: The first approach

- Recall the goal: find a point x such that

Non convex optimization: The first approach

- Recall the goal: find a point x such that
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.

Non convex optimization: The first approach

- Recall the goal: find a point x such that
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.

Non convex optimization: The first approach

- Recall the goal: find a point x such that
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- How do we do it?

Non convex optimization: The first approach

- Recall the goal: find a point x such that
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- How do we do it?
- Approach 1 (Algorithm Forklore):

Non convex optimization: The first approach

- Recall the goal: find a point x such that
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- How do we do it?
- Approach 1 (Algorithm Forklore):
- Do gradient descent, until we arrive at a point x' with $\|\nabla f(x')\|_2 \leq \varepsilon$.

Non convex optimization: The first approach

- Recall the goal: find a point x such that
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- How do we do it?
- Approach 1 (Algorithm Forklore):
- Do gradient descent, until we arrive at a point x' with $\|\nabla f(x')\|_2 \leq \varepsilon$.
  - Simple observation: For every β-smooth f,

$$f(x - \eta \nabla f(x)) \leq f(x) - \eta \|\nabla f(x)\|_2^2 + \eta^2 \beta^2 \|\nabla f(x)\|_2^2$$

Non convex optimization: The first approach

- Recall the goal: find a point x such that
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- How do we do it?
- Approach 1 (Algorithm Forklore):
- Do gradient descent, until we arrive at a point x' with $\|\nabla f(x')\|_2 \leq \varepsilon$.
  - Simple observation: For every β-smooth f,

$$f(x - \eta \nabla f(x)) \leq f(x) - \eta \|\nabla f(x)\|_2^2 + \eta^2 \beta^2 \|\nabla f(x)\|_2^2$$

  - Gradient large $\Longrightarrow$ decrease function value using gradient descent.

Non convex optimization: The first approach

- Recall the goal: find a point x such that
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- How do we do it?
- Approach 1 (Algorithm Forklore):
- Do gradient descent, until we arrive at a point x' with $\|\nabla f(x')\|_2 \leq \varepsilon$.
  - Simple observation: For every β-smooth f,

$$f(x - \eta \nabla f(x)) \leq f(x) - \eta \|\nabla f(x)\|_2^2 + \eta^2 \beta^2 \|\nabla f(x)\|_2^2$$

  - Gradient large $\implies$ decrease function value using gradient descent.
- Check if $\nabla^2 f(x') \succeq -\delta I$.

Non convex optimization: The first approach

- Recall the goal: find a point x such that
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- How do we do it?
- Approach 1 (Algorithm Forklore):
- Do gradient descent, until we arrive at a point x' with $\|\nabla f(x')\|_2 \leq \varepsilon$.
  - Simple observation: For every β-smooth f,

$$f(x - \eta \nabla f(x)) \leq f(x) - \eta \|\nabla f(x)\|_2^2 + \eta^2 \beta^2 \|\nabla f(x)\|_2^2$$

  - Gradient large $\implies$ decrease function value using gradient descent.
- Check if $\nabla^2 f(x') \succeq -\delta I$.
  - If not, find a unit vector v such that $v^\top \nabla^2 f(x') v \leq -\delta$. Can be done efficiently via eigenvectors solver.

Non convex optimization: The first approach

- Recall the goal: find a point x such that
  - $\|\nabla f(x)\|_2 \le \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- How do we do it?
- Approach 1 (Algorithm Forklore):
- Do gradient descent, until we arrive at a point x' with $\|\nabla f(x')\|_2 \le \varepsilon$.
  - Simple observation: For every β-smooth f,

$$f(x - \eta \nabla f(x)) \le f(x) - \eta \|\nabla f(x)\|_2^2 + \eta^2 \beta^2 \|\nabla f(x)\|_2^2$$

  - Gradient large $\Longrightarrow$ decrease function value using gradient descent.
- Check if $\nabla^2 f(x') \succeq -\delta I$.
  - If not, find a unit vector v such that $v^\top \nabla^2 f(x') v \le -\delta$. Can be done efficiently via eigenvectors solver.
  - Hessian descent: For a step size η, if $f(x' + \eta v) \le f(x' - \eta v)$, go to $x'' = x' + \eta v$. Otherwise go to $x'' = x' - \eta v$.

Non convex optimization: The first approach

- Recall the goal: find a point x such that
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \geq -\delta I$.
- How do we do it?
- Approach 1 (Algorithm Forklore):
- Do gradient descent, until we arrive at a point x' with $\|\nabla f(x')\|_2 \leq \varepsilon$.
  - Simple observation: For every β-smooth f,

$$f(x - \eta\nabla f(x)) \leq f(x) - \eta\|\nabla f(x)\|_2^2 + \eta^2\beta^2\|\nabla f(x)\|_2^2$$

  - Gradient large $\implies$ decrease function value using gradient descent.
- Check if $\nabla^2 f(x') \geq -\delta I$.
  - If not, find a unit vector v such that $v^\top \nabla^2 f(x') v \leq -\delta$. Can be done efficiently via eigenvectors solver.
  - Hessian descent: For a step size η, if $f(x' + \eta v) \leq f(x' - \eta v)$, go to $x'' = x' + \eta v$. Otherwise go to $x'' = x' - \eta v$.
- Repeat to gradient descent.

Non convex optimization: Hessian descent

- Recall: (important property): For every x', τ:

$$f(x' + \tau) = f(x') + \langle \nabla f(x'), \tau \rangle + \frac{1}{2}\tau^\top \nabla^2 f(x)\tau \pm \gamma \|\tau\|_2^3$$

Non convex optimization: Hessian descent

- Recall: (important property): For every x', τ:

$$f(x' + \tau) = f(x') + \langle \nabla f(x'), \tau \rangle + \frac{1}{2}\tau^\top \nabla^2 f(x)\tau \pm \gamma \|\tau\|_2^3$$

- Critical observation:

$$\begin{aligned}\frac{1}{2}\left(f(x' + \eta v) + f(x' - \eta v)\right) &\leq f(x') + \frac{\eta^2}{2} v^\top \nabla^2 f(x') v + \gamma \eta^3 \\ &\leq f(x') - \frac{\eta^2 \delta}{2} + \gamma \eta^3\end{aligned}$$

Non convex optimization: Hessian descent

- Recall: (important property): For every x', τ:

$$f(x' + \tau) = f(x') + \langle \nabla f(x'), \tau \rangle + \frac{1}{2}\tau^\top \nabla^2 f(x)\tau \pm \gamma \|\tau\|_2^3$$

- Critical observation:

$$\begin{aligned} \frac{1}{2}\Big(f(x' + \eta v) + f(x' - \eta v)\Big) &\leq f(x') + \frac{\eta^2}{2} v^\top \nabla^2 f(x') v + \gamma \eta^3 \\ &\leq f(x') - \frac{\eta^2 \delta}{2} + \gamma \eta^3 \end{aligned}$$

- Taking $\eta = \frac{\delta}{4\gamma}$, the function value is decreased by at least $\frac{\delta^3}{64\gamma^2}$: $f(x'') \leq f(x') - \frac{\delta^3}{64\gamma^2}$.

Non convex optimization: Hessian descent

- Recall: (important property): For every x', τ:

$$f(x' + \tau) = f(x') + \langle \nabla f(x'), \tau \rangle + \frac{1}{2}\tau^\top \nabla^2 f(x)\tau \pm \gamma \|\tau\|_2^3$$

- Critical observation:

$$\begin{aligned}\frac{1}{2}\left(f(x' + \eta v) + f(x' - \eta v)\right) &\leq f(x') + \frac{\eta^2}{2} v^\top \nabla^2 f(x') v + \gamma\eta^3 \\ &\leq f(x') - \frac{\eta^2 \delta}{2} + \gamma\eta^3\end{aligned}$$

- Taking $\eta = \frac{\delta}{4\gamma}$, the function value is decreased by at least $\frac{\delta^3}{64\gamma^2}$: $f(x'') \leq f(x') - \frac{\delta^3}{64\gamma^2}$.
- In other words, a hessian descent would decrease function value by $\Omega(\delta^3)$, when the negative eigenvalue of the hessian is $\leq -\delta$: The more non-convex, the hessian descent works better.

Non convex optimization: The first approach

- Recall the goal: find a point x such that:

Non convex optimization: The first approach

- Recall the goal: find a point x such that:
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.

Non convex optimization: The first approach

- Recall the goal: find a point x such that:
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.

Non convex optimization: The first approach

- Recall the goal: find a point x such that:
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- Suppose f is non-negative and the initial point x^{init} satisfies: $f(x^{init}) \leq 1$, then:

Non convex optimization: The first approach

- Recall the goal: find a point x such that:
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- Suppose f is non-negative and the initial point x^{init} satisfies: $f(x^{init}) \leq 1$, then:
- The first approach achieves the goal within: (ignoring poly(γ, β) factors)

Non convex optimization: The first approach

- Recall the goal: find a point x such that:
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- Suppose f is non-negative and the initial point x^{init} satisfies: $f(x^{init}) \leq 1$, then:
- The first approach achieves the goal within: (ignoring poly(γ, β) factors)
  - $O\left(\frac{1}{\varepsilon^2}\right)$ many gradient evaluations: gradient descent.

Non convex optimization: The first approach

- Recall the goal: find a point x such that:
  - $\|\nabla f(x)\|_2 \le \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- Suppose f is non-negative and the initial point x^{init} satisfies: $f(x^{init}) \le 1$, then:
- The first approach achieves the goal within: (ignoring poly(γ, β) factors)
  - $O\left(\frac{1}{\varepsilon^2}\right)$ many gradient evaluations: gradient descent.
  - $O\left(\frac{1}{\delta^3}\right)$ many eigenvectors solvers for the hessian matrix: hessian descent.

Non convex optimization: The second approach

- Recall: the first approach needs:

Non convex optimization: The second approach

- Recall: the first approach needs:
- $O\left(\frac{1}{\varepsilon^2}\right)$ many gradient evaluations: gradient descent.

Non convex optimization: The second approach

- Recall: the first approach needs:
- $O\left(\frac{1}{\varepsilon^2}\right)$ many gradient evaluations: gradient descent.
- $O\left(\frac{1}{\delta^3}\right)$ many eigenvectors solvers for the hessian matrix: hessian descent.

Non convex optimization: The second approach

- Recall: the first approach needs:
- $O\left(\frac{1}{\varepsilon^2}\right)$ many gradient evaluations: gradient descent.
- $O\left(\frac{1}{\delta^3}\right)$ many eigenvectors solvers for the hessian matrix: hessian descent.
- Can we do it faster?

Non convex optimization: The second approach

- Recall: the first approach needs:
- $O\left(\frac{1}{\varepsilon^2}\right)$ many gradient evaluations: gradient descent.
- $O\left(\frac{1}{\delta^3}\right)$ many eigenvectors solvers for the hessian matrix: hessian descent.
- Can we do it faster?
- Yes, we can reduce the number of gradient evaluations and completely get rid of eigenvectors solvers.

Non convex optimization: The second approach

- Recall: the first approach needs:
- $O\left(\frac{1}{\varepsilon^2}\right)$ many gradient evaluations: gradient descent.
- $O\left(\frac{1}{\delta^3}\right)$ many eigenvectors solvers for the hessian matrix: hessian descent.
- Can we do it faster?
- Yes, we can reduce the number of gradient evaluations and completely get rid of eigenvectors solvers.
- Algorithm Neon2.

Non convex optimization: The second approach

- Recall: the first approach needs:
- $O\left(\frac{1}{\varepsilon^2}\right)$ many gradient evaluations: gradient descent.
- $O\left(\frac{1}{\delta^3}\right)$ many eigenvectors solvers for the hessian matrix: hessian descent.
- Can we do it faster?
- Yes, we can reduce the number of gradient evaluations and completely get rid of eigenvectors solvers.
- Algorithm Neon2.
- Approach:

Non convex optimization: The second approach

- Recall: the first approach needs:
- $O\left(\frac{1}{\varepsilon^2}\right)$ many gradient evaluations: gradient descent.
- $O\left(\frac{1}{\delta^3}\right)$ many eigenvectors solvers for the hessian matrix: hessian descent.
- Can we do it faster?
- Yes, we can reduce the number of gradient evaluations and completely get rid of eigenvectors solvers.
- Algorithm Neon2.
- Approach:
  - Reducing the number of gradient evaluations at the cost of increasing the number of hessian eigenvectors solvers.

Non convex optimization: The second approach

- Recall: the first approach needs:
- $O\left(\frac{1}{\varepsilon^2}\right)$ many gradient evaluations: gradient descent.
- $O\left(\frac{1}{\delta^3}\right)$ many eigenvectors solvers for the hessian matrix: hessian descent.
- Can we do it faster?
- Yes, we can reduce the number of gradient evaluations and completely get rid of eigenvectors solvers.
- Algorithm Neon2.
- Approach:
  - Reducing the number of gradient evaluations at the cost of increasing the number of hessian eigenvectors solvers.
  - Then reducing the number of hessian eigenvectors solvers at the cost of increasing the number of gradient evaluations.

Non convex optimization: The second approach

- Recall: the first approach needs:
- $O\left(\frac{1}{\varepsilon^2}\right)$ many gradient evaluations: gradient descent.
- $O\left(\frac{1}{\delta^3}\right)$ many eigenvectors solvers for the hessian matrix: hessian descent.
- Can we do it faster?
- Yes, we can reduce the number of gradient evaluations and completely get rid of eigenvectors solvers.
- Algorithm Neon2.
- Approach:
  - Reducing the number of gradient evaluations at the cost of increasing the number of hessian eigenvectors solvers.
  - Then reducing the number of hessian eigenvectors solvers at the cost of increasing the number of gradient evaluations.
  - Sounds fishy? Loopy argument? We shall see.

Non convex optimization: The second approach

- Basic idea: What if f is a convex function? Can we reduce the number of gradient evaluations to find a point x with $\|\nabla f(x)\|_2 \leq \varepsilon$?

Non convex optimization: The second approach

- Basic idea: What if f is a convex function? Can we reduce the number of gradient evaluations to find a point x with $\|\nabla f(x)\|_2 \leq \varepsilon$?
- Yes, we can reduce the number of gradient evaluations.

Non convex optimization: The second approach

- Basic idea: What if f is a convex function? Can we reduce the number of gradient evaluations to find a point x with $\|\nabla f(x)\|_2 \leq \varepsilon$?
- Yes, we can reduce the number of gradient evaluations.
- Tool: Accelerated gradient descent (AGD). [Nesterov 1983]

Non convex optimization: The second approach

- Basic idea: What if f is a convex function? Can we reduce the number of gradient evaluations to find a point x with $\|\nabla f(x)\|_2 \leq \varepsilon$?
- Yes, we can reduce the number of gradient evaluations.
- Tool: Accelerated gradient descent (AGD). [Nesterov 1983]
- Recall: AGD finds an x with $f(x) \leq \min_{y \in \mathbb{R}^d} f(y) + \varepsilon^2$:

Non convex optimization: The second approach

- Basic idea: What if f is a convex function? Can we reduce the number of gradient evaluations to find a point x with $\|\nabla f(x)\|_2 \le \varepsilon$?
- Yes, we can reduce the number of gradient evaluations.
- Tool: Accelerated gradient descent (AGD). [Nesterov 1983]
- Recall: AGD finds an x with $f(x) \le \min_{y \in \mathbb{R}^d} f(y) + \varepsilon^2$:
  - In $O\left(\frac{1}{\varepsilon}\right)$ many gradient evaluations for any smooth, convex function f.

Non convex optimization: The second approach

- Basic idea: What if f is a convex function? Can we reduce the number of gradient evaluations to find a point x with $\|\nabla f(x)\|_2 \leq \varepsilon$?
- Yes, we can reduce the number of gradient evaluations.
- Tool: Accelerated gradient descent (AGD). [Nesterov 1983]
- Recall: AGD finds an x with $f(x) \leq \min_{y \in \mathbb{R}^d} f(y) + \varepsilon^2$:
  - In $O\left(\frac{1}{\varepsilon}\right)$ many gradient evaluations for any smooth, convex function f.
  - In $O\left(\frac{1}{\sqrt{\alpha}} \log \frac{1}{\varepsilon}\right)$ many gradient evaluations if f is α-strongly convex.

Non convex optimization: The second approach

- Basic idea: What if f is a convex function? Can we reduce the number of gradient evaluations to find a point x with $\|\nabla f(x)\|_2 \leq \varepsilon$?
- Yes, we can reduce the number of gradient evaluations.
- Tool: Accelerated gradient descent (AGD). [Nesterov 1983]
- Recall: AGD finds an x with $f(x) \leq \min_{y \in \mathbb{R}^d} f(y) + \varepsilon^2$:
  - In $O\left(\frac{1}{\varepsilon}\right)$ many gradient evaluations for any smooth, convex function f.
  - In $O\left(\frac{1}{\sqrt{\alpha}} \log \frac{1}{\varepsilon}\right)$ many gradient evaluations if f is α-strongly convex.
- By the 1-smoothness of f, $f(x) \leq \min_{y \in \mathbb{R}^d} f(y) + \varepsilon^2$ implies that $\|\nabla f(x)\|_2 \leq \varepsilon$.

Non convex optimization: The second approach

- Basic idea: What if f is a convex function? Can we reduce the number of gradient evaluations to find a point x with $\|\nabla f(x)\|_2 \leq \varepsilon$?
- Yes, we can reduce the number of gradient evaluations.
- Tool: Accelerated gradient descent (AGD). [Nesterov 1983]
- Recall: AGD finds an x with $f(x) \leq \min_{y \in \mathbb{R}^d} f(y) + \varepsilon^2$:
  - In $O\left(\frac{1}{\varepsilon}\right)$ many gradient evaluations for any smooth, convex function f.
  - In $O\left(\frac{1}{\sqrt{\alpha}} \log \frac{1}{\varepsilon}\right)$ many gradient evaluations if f is α-strongly convex.
- By the 1-smoothness of f, $f(x) \leq \min_{y \in \mathbb{R}^d} f(y) + \varepsilon^2$ implies that $\|\nabla f(x)\|_2 \leq \varepsilon$.
- For smooth, convex function f: AGD can find a point x with $\|\nabla f(x)\|_2 \leq \varepsilon$ in $O\left(\frac{1}{\varepsilon}\right)$ iterations.

Non convex optimization: The second approach

- Basic idea: What if f is a convex function? Can we reduce the number of gradient evaluations to find a point x with $\|\nabla f(x)\|_2 \leq \varepsilon$?
- Yes, we can reduce the number of gradient evaluations.
- Tool: Accelerated gradient descent (AGD). [Nesterov 1983]
- Recall: AGD finds an x with $f(x) \leq \min_{y \in \mathbb{R}^d} f(y) + \varepsilon^2$:
  - In $O\left(\frac{1}{\varepsilon}\right)$ many gradient evaluations for any smooth, convex function f.
  - In $O\left(\frac{1}{\sqrt{\alpha}} \log \frac{1}{\varepsilon}\right)$ many gradient evaluations if f is α-strongly convex.
- By the 1-smoothness of f, $f(x) \leq \min_{y \in \mathbb{R}^d} f(y) + \varepsilon^2$ implies that $\|\nabla f(x)\|_2 \leq \varepsilon$.
- For smooth, convex function f: AGD can find a point x with $\|\nabla f(x)\|_2 \leq \varepsilon$ in $O\left(\frac{1}{\varepsilon}\right)$ iterations.
- Recall: gradient descent needs $O\left(\frac{1}{\varepsilon^2}\right)$ iterations.

Non convex optimization: The second approach

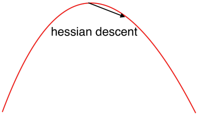
- For non-convex function, we can do:

Non convex optimization: The second approach

- For non-convex function, we can do:
  - (Truly non-convex): If $\nabla^2 f(x)$ has a very negative eigenvalue, then we do a hessian descent.

Non convex optimization: The second approach

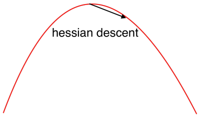
- For non-convex function, we can do:
  - (Truly non-convex): If $\nabla^2 f(x)$ has a very negative eigenvalue, then we do a hessian descent.



  -

Non convex optimization: The second approach

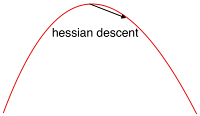
- For non-convex function, we can do:
  - (Truly non-convex): If $\nabla^2 f(x)$ has a very negative eigenvalue, then we do a hessian descent.



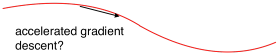
  -
  - (Approximately convex): $\nabla^2 f(x)$ only contains small negative eigenvalues, can we still do accelerated gradient descent?

Non convex optimization: The second approach

- For non-convex function, we can do:
  - (Truly non-convex): If $\nabla^2 f(x)$ has a very negative eigenvalue, then we do a hessian descent.



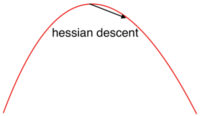
  -
  - (Approximately convex): $\nabla^2 f(x)$ only contains small negative eigenvalues, can we still do accelerated gradient descent?



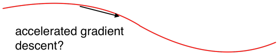
  -

Non convex optimization: The second approach

- For non-convex function, we can do:
  - (Truly non-convex): If $\nabla^2 f(x)$ has a very negative eigenvalue, then we do a hessian descent.



  -
  - (Approximately convex): $\nabla^2 f(x)$ only contains small negative eigenvalues, can we still do accelerated gradient descent?



  -
  - Yes, we can do it.

Non convex optimization: The second approach

- General plan: Each iteration, we first find the eigenvector of $\nabla^2 f(x)$ with the most negative eigenvalue.

Non convex optimization: The second approach

- General plan: Each iteration, we first find the eigenvector of $\nabla^2 f(x)$ with the most negative eigenvalue.
  - If the eigenvalue is too negative: do hessian descent.

Non convex optimization: The second approach

- General plan: Each iteration, we first find the eigenvector of $\nabla^2 f(x)$ with the most negative eigenvalue.
  - If the eigenvalue is too negative: do hessian descent.
  - Otherwise, do accelerated gradient descent.

Non convex optimization: The second approach

- General plan: Each iteration, we first find the eigenvector of $\nabla^2 f(x)$ with the most negative eigenvalue.
  - If the eigenvalue is too negative: do hessian descent.
  - Otherwise, do accelerated gradient descent.
- In this way, we can reduce the number of gradient evaluations at the cost of increasing the number of hessian eigenvectors solvers.

Non convex optimization: The second approach

- Let us now do the calculation for the exact numbers:

Non convex optimization: The second approach

- Let us now do the calculation for the exact numbers:
- For simplicity, I will assume $\beta = \gamma = 1$.

Non convex optimization: The second approach

- Let us now do the calculation for the exact numbers:
- For simplicity, I will assume $\beta = \gamma = 1$.
- Taking $\delta_1 = \frac{1}{100}\varepsilon^{0.5}$ (the "threshold" of large v.s. small for the negative eigenvalue), then:

Non convex optimization: The second approach

- Let us now do the calculation for the exact numbers:
- For simplicity, I will assume $\beta = \gamma = 1$.
- Taking $\delta_1 = \frac{1}{100}\varepsilon^{0.5}$ (the "threshold" of large v.s. small for the negative eigenvalue), then:
- If $\nabla^2 f(x_0) \geq -\delta_1 I$, we do accelerated gradient descent.

Non convex optimization: The second approach

- Let us now do the calculation for the exact numbers:
- For simplicity, I will assume $\beta = \gamma = 1$.
- Taking $\delta_1 = \frac{1}{100}\varepsilon^{0.5}$ (the "threshold" of large v.s. small for the negative eigenvalue), then:
- If $\nabla^2 f(x_0) \succeq -\delta_1 I$, we do accelerated gradient descent.
- Otherwise, we do hessian descent, which (recall!) will decrease function value by $\Omega(\delta_1^3) = \Omega(\varepsilon^{1.5})$.

Non convex optimization: The second approach

- Let us now do the calculation for the exact numbers:
- For simplicity, I will assume $\beta = \gamma = 1$.
- Taking $\delta_1 = \frac{1}{100}\varepsilon^{0.5}$ (the "threshold" of large v.s. small for the negative eigenvalue), then:
- If $\nabla^2 f(x_0) \succeq -\delta_1 I$, we do accelerated gradient descent.
- Otherwise, we do hessian descent, which (recall!) will decrease function value by $\Omega(\delta_1^3) = \Omega(\varepsilon^{1.5})$.
- So, we can do at most $O\left(\frac{1}{\varepsilon^{1.5}}\right)$ many iterations of the hessian descent.

Non convex optimization: The second approach

- The magic step for AGD when $\nabla^2 f(x_0) \succeq -\delta_1 I$:

Non convex optimization: The second approach

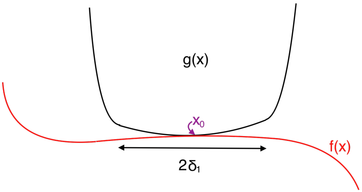
- The magic step for AGD when $\nabla^2 f(x_0) \geq -\delta_1 I$:
- Define function $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$, where

Non convex optimization: The second approach

- The magic step for AGD when $\nabla^2 f(x_0) \succeq -\delta_1 I$:
- Define function $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$, where
- $h_{x_0,\delta_1}(x) = 4 \times 1_{\|x - x_0\|_2 \geq \delta_1}(\|x - x_0\|_2 - \delta_1)^2$.

Non convex optimization: The second approach

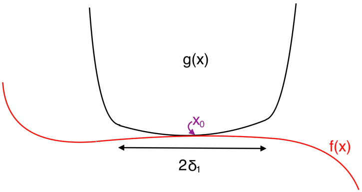
- The magic step for AGD when $\nabla^2 f(x_0) \succeq -\delta_1 I$:
- Define function $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$, where
- $h_{x_0,\delta_1}(x) = 4 \times 1_{\|x-x_0\|_2 \geq \delta_1}(\|x - x_0\|_2 - \delta_1)^2$.



-

Non convex optimization: The second approach

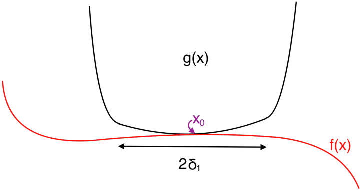
- The magic step for AGD when $\nabla^2 f(x_0) \succeq -\delta_1 I$:
- Define function $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$, where
- $h_{x_0,\delta_1}(x) = 4 \times 1_{\|x-x_0\|_2 \geq \delta_1}(\|x - x_0\|_2 - \delta_1)^2$.



-
- Critical observation: $g(x)$ is δ_1 strongly convex.

Non convex optimization: The second approach

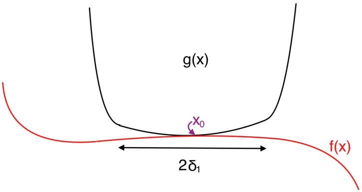
- The magic step for AGD when $\nabla^2 f(x_0) \succeq -\delta_1 I$:
- Define function $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$, where
- $h_{x_0,\delta_1}(x) = 4 \times 1_{\|x - x_0\|_2 \geq \delta_1} (\|x - x_0\|_2 - \delta_1)^2$.



-
- Critical observation: $g(x)$ is δ_1 strongly convex.
    - When $\|x - x_0\|_2 \leq \delta_1$: Using the $\nabla^2 f(x) \succeq -2\delta_1 I$ and the strong convexity of $4\delta_1 \|x - x_0\|_2^2$.

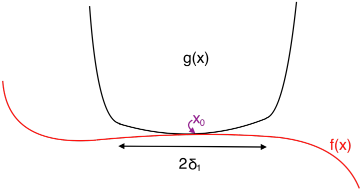
Non convex optimization: The second approach

- The magic step for AGD when $\nabla^2 f(x_0) \succeq -\delta_1 I$:
- Define function $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$, where
- $h_{x_0,\delta_1}(x) = 4 \times 1_{\|x-x_0\|_2 \geq \delta_1} (\|x - x_0\|_2 - \delta_1)^2$.

- 

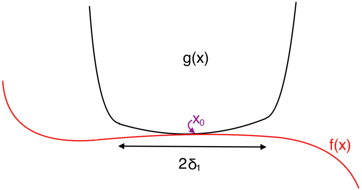
- Critical observation: $g(x)$ is δ_1 strongly convex.
  - When $\|x - x_0\|_2 \leq \delta_1$: Using the $\nabla^2 f(x) \succeq -2\delta_1 I$ and the strong convexity of $4\delta_1 \|x - x_0\|_2^2$.
  - When $\|x - x_0\|_2 \geq \delta_1$: Using the strong convexity of $h_{x_0,\delta_1}(x)$.

Non convex optimization: The second approach



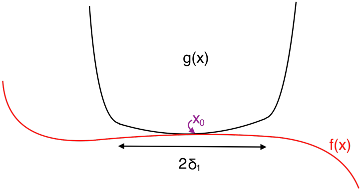
•

Non convex optimization: The second approach



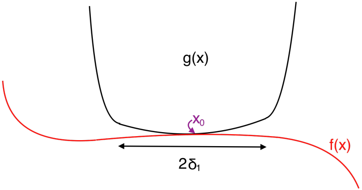
-
- $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$.

Non convex optimization: The second approach

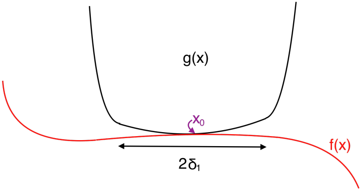


-
- $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$.
- Now, use accelerated gradient descent on the δ_1-strongly convex function g, we can find a point x_1 with

Non convex optimization: The second approach

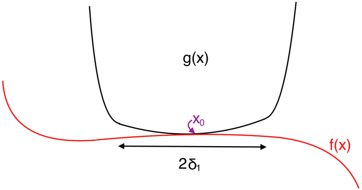


-
- $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$.
- Now, use accelerated gradient descent on the δ_1-strongly convex function g, we can find a point x_1 with
  - $g(x_1) \leq g(x_0) + \varepsilon^2$;



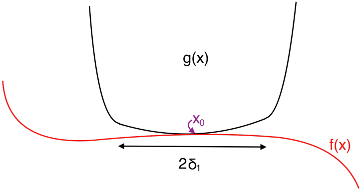
- $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0, \delta_1}(x)$.
- Now, use accelerated gradient descent on the δ_1-strongly convex function g, we can find a point x_1 with
  - $g(x_1) \le g(x_0) + \varepsilon^2$;
  - $\|\nabla g(x_1)\|_2 \le \varepsilon^2$.

Non convex optimization: The second approach



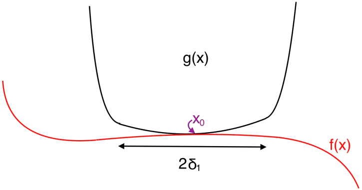
-
- $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$.
- Now, use accelerated gradient descent on the δ_1-strongly convex function g, we can find a point x_1 with
  - $g(x_1) \le g(x_0) + \varepsilon^2$;
  - $\|\nabla g(x_1)\|_2 \le \varepsilon^2$.
- in $O\left(\frac{1}{\sqrt{\delta_1}} \log \frac{1}{\varepsilon}\right)$ gradient evaluations.

Non convex optimization: The second approach



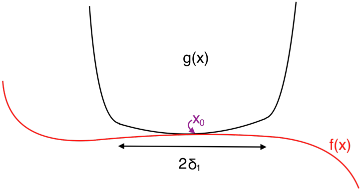
-

Non convex optimization: The second approach



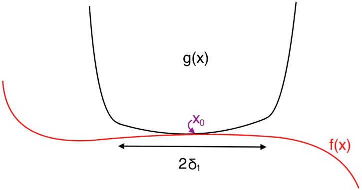
-
- Recall: $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$

Non convex optimization: The second approach



-
- Recall: $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$
    - $g(x_1) \le g(x_0) + \varepsilon^2$;

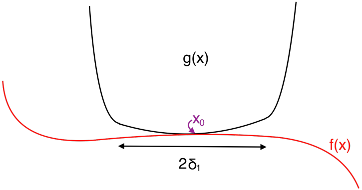
Non convex optimization: The second approach



-

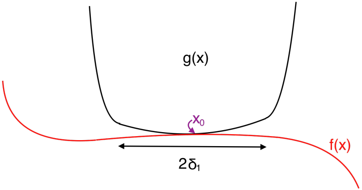
- Recall: $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$
  - $g(x_1) \leq g(x_0) + \varepsilon^2$;
  - $\|\nabla g(x_1)\|_2 \leq \varepsilon^2$.

Non convex optimization: The second approach



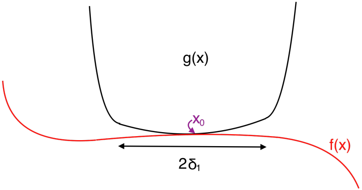
-
- Recall: $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$
    - $g(x_1) \leq g(x_0) + \varepsilon^2$;
    - $\|\nabla g(x_1)\|_2 \leq \varepsilon^2$.
- Two cases for x_1:

Non convex optimization: The second approach



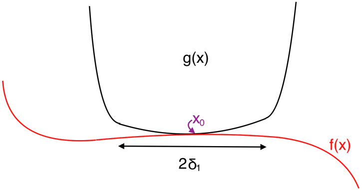
-
- Recall: $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$
  - $g(x_1) \le g(x_0) + \varepsilon^2$;
  - $\|\nabla g(x_1)\|_2 \le \varepsilon^2$.
- Two cases for x_1:
  - $\|x_1 - x_0\|_2 \ge \delta_1$, then

Non convex optimization: The second approach



- Recall: $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$
    - $g(x_1) \leq g(x_0) + \varepsilon^2$;
    - $\|\nabla g(x_1)\|_2 \leq \varepsilon^2$.
- Two cases for x_1:
    - $\|x_1 - x_0\|_2 \geq \delta_1$, then
        - $f(x_1) \leq g(x_1) - 4\delta_1^3 \leq f(x_0) - \Omega(\varepsilon^{1.5})$: Decrease function value.

Non convex optimization: The second approach



- Recall: $g(x) = f(x) + 4\delta_1 \|x - x_0\|_2^2 + h_{x_0,\delta_1}(x)$
  - $g(x_1) \leq g(x_0) + \varepsilon^2$;
  - $\|\nabla g(x_1)\|_2 \leq \varepsilon^2$.
- Two cases for x_1:
  - $\|x_1 - x_0\|_2 \geq \delta_1$, then
    - $f(x_1) \leq g(x_1) - 4\delta_1^3 \leq f(x_0) - \Omega(\varepsilon^{1.5})$: Decrease function value.
  - $\|x_1 - x_0\|_2 \leq \delta_1$, then $\|\nabla f(x_1)\|_2 \leq \|\nabla g(x_1)\|_2 + 8\delta_1^2 \leq \varepsilon$: Gradient is small.

Non convex optimization: The second approach

- What did we do?

Non convex optimization: The second approach

- What did we do?
- We use 1 hessian eigenvector solver and $\tilde{O}\left(\frac{1}{\sqrt{\delta_1}}\right) = \tilde{O}\left(\frac{1}{\varepsilon^{0.25}}\right)$ gradient evaluations, we obtain at least one of the following:

Non convex optimization: The second approach

- What did we do?
- We use 1 hessian eigenvector solver and $\tilde{O}\left(\frac{1}{\sqrt{\delta_1}}\right) = \tilde{O}\left(\frac{1}{\varepsilon^{0.25}}\right)$ gradient evaluations, we obtain at least one of the following:
  - Decrease the function value by at least $\Omega(\varepsilon^{1.5})$ (Can happen for at most $O\left(\frac{1}{\varepsilon^{1.5}}\right)$ times).

Non convex optimization: The second approach

- What did we do?
- We use 1 hessian eigenvector solver and $\tilde{O}\left(\frac{1}{\sqrt{\delta_1}}\right) = \tilde{O}\left(\frac{1}{\varepsilon^{0.25}}\right)$ gradient evaluations, we obtain at least one of the following:
  - Decrease the function value by at least $\Omega(\varepsilon^{1.5})$ (Can happen for at most $O\left(\frac{1}{\varepsilon^{1.5}}\right)$ times).
  - Find a point x with $\|\nabla f(x)\|_2 \le \varepsilon$.

Non convex optimization: The second approach

- What did we do?
- We use 1 hessian eigenvector solver and $\tilde{O}\left(\frac{1}{\sqrt{\delta_1}}\right) = \tilde{O}\left(\frac{1}{\varepsilon^{0.25}}\right)$ gradient evaluations, we obtain at least one of the following:
  - Decrease the function value by at least $\Omega(\varepsilon^{1.5})$ (Can happen for at most $O\left(\frac{1}{\varepsilon^{1.5}}\right)$ times).
  - Find a point x with $\|\nabla f(x)\|_2 \leq \varepsilon$.
- In total, we can find a point x with $\|\nabla f(x)\|_2 \leq \varepsilon$ in $\tilde{O}\left(\frac{1}{\varepsilon^{1.75}}\right)$ gradient evaluations and $O\left(\frac{1}{\varepsilon^{1.5}}\right)$ calls of hessian eigenvectors solver.

Non convex optimization: The second approach

- What did we do?
- We use 1 hessian eigenvector solver and $\tilde{O}\left(\frac{1}{\sqrt{\delta_1}}\right) = \tilde{O}\left(\frac{1}{\varepsilon^{0.25}}\right)$ gradient evaluations, we obtain at least one of the following:
  - Decrease the function value by at least $\Omega(\varepsilon^{1.5})$ (Can happen for at most $O\left(\frac{1}{\varepsilon^{1.5}}\right)$ times).
  - Find a point x with $\|\nabla f(x)\|_2 \leq \varepsilon$.
- In total, we can find a point x with $\|\nabla f(x)\|_2 \leq \varepsilon$ in $\tilde{O}\left(\frac{1}{\varepsilon^{1.75}}\right)$ gradient evaluations and $O\left(\frac{1}{\varepsilon^{1.5}}\right)$ calls of hessian eigenvectors solver.
- Recall: Gradient descent needs $\tilde{O}\left(\frac{1}{\varepsilon^{2}}\right)$ gradient evaluations.

Non convex optimization: The second approach

- The last piece: Reducing the call to hessian eigenvectors solver to gradient evaluations: Recall we need in total $O\left(\frac{1}{\varepsilon^{1.5}}\right)$ calls of hessian eigenvectors solver.

Non convex optimization: The second approach

- The last piece: Reducing the call to hessian eigenvectors solver to gradient evaluations: Recall we need in total $O\left(\frac{1}{\varepsilon^{1.5}}\right)$ calls of hessian eigenvectors solver.
- Goal of each hessian eigenvectors solver: Suppose there is a unit vector v with $v^\top \nabla^2 f(x) v \leq -\delta_1$, we need to find a unit vector w with

$$w^\top \nabla^2 f(x) w \leq -0.9\delta_1$$

Non convex optimization: The second approach

- The last piece: Reducing the call to hessian eigenvectors solver to gradient evaluations: Recall we need in total $O\left(\frac{1}{\varepsilon^{1.5}}\right)$ calls of hessian eigenvectors solver.
- Goal of each hessian eigenvectors solver: Suppose there is a unit vector v with $v^\top \nabla^2 f(x) v \leq -\delta_1$, we need to find a unit vector w with

$$w^\top \nabla^2 f(x) w \leq -0.9\delta_1$$

- Can we do it within $O\left(\frac{1}{\sqrt{\delta_1}}\right) = O\left(\frac{1}{\varepsilon^{0.25}}\right)$ gradient evaluations?

Non convex optimization: Last piece of Neon2:

- How to find eigenvectors of a matrix M?

Non convex optimization: Last piece of Neon2:

- How to find eigenvectors of a matrix M?
- Power method: z_0 is a random unit vector, update $z_{t+1} = \frac{Mz_t}{\|Mz_t\|_2}$.

Non convex optimization: Last piece of Neon2:

- How to find eigenvectors of a matrix M?
- Power method: z_0 is a random unit vector, update $z_{t+1} = \frac{Mz_t}{\|Mz_t\|_2}$.
- Power method finds unit vector w with $w^\top \nabla^2 f(x) w \le -0.9\delta_1$ in $O\left(\frac{1}{\delta_1}\right)$ iterations of computing M times a vector.

Non convex optimization: Last piece of Neon2:

- How to find eigenvectors of a matrix M?
- Power method: z_0 is a random unit vector, update $z_{t+1} = \frac{Mz_t}{\|Mz_t\|_2}$.
- Power method finds unit vector w with $w^\top \nabla^2 f(x) w \le -0.9\delta_1$ in $O\left(\frac{1}{\delta_1}\right)$ iterations of computing M times a vector.
- Acceleration: Lanzos method/ Chevbyshev polynomial methods.

Non convex optimization: Last piece of Neon2:

- How to find eigenvectors of a matrix M?
- Power method: z_0 is a random unit vector, update $z_{t+1} = \frac{Mz_t}{\|Mz_t\|_2}$.
- Power method finds unit vector w with $w^\top \nabla^2 f(x) w \leq -0.9\delta_1$ in $O\left(\frac{1}{\delta_1}\right)$ iterations of computing M times a vector.
- Acceleration: Lanzos method/ Chevbyshev polynomial methods.
- Finds unit vector w with $w^\top \nabla^2 f(x) w \leq -0.9\delta_1$ in $O\left(\frac{1}{\sqrt{\delta_1}}\right)$ iterations of computing M times a vector.

Non convex optimization: Last piece of Neon2:

- How to find eigenvectors of a matrix M?
- Power method: z_0 is a random unit vector, update $z_{t+1} = \frac{Mz_t}{\|Mz_t\|_2}$.
- Power method finds unit vector w with $w^\top \nabla^2 f(x) w \leq -0.9\delta_1$ in $O\left(\frac{1}{\delta_1}\right)$ iterations of computing M times a vector.
- Acceleration: Lanzos method/ Chevbyshev polynomial methods.
- Finds unit vector w with $w^\top \nabla^2 f(x) w \leq -0.9\delta_1$ in $O\left(\frac{1}{\sqrt{\delta_1}}\right)$ iterations of computing M times a vector.
- Computing M times a vector? How to compute $\nabla^2 f(x) z$ for a vector z in our case? Easy:

Non convex optimization: Last piece of Neon2:

- How to find eigenvectors of a matrix M?
- Power method: z_0 is a random unit vector, update $z_{t+1} = \frac{Mz_t}{\|Mz_t\|_2}$.
- Power method finds unit vector w with $w^\top \nabla^2 f(x) w \leq -0.9\delta_1$ in $O\left(\frac{1}{\delta_1}\right)$ iterations of computing M times a vector.
- Acceleration: Lanzos method/ Chevbyshev polynomial methods.
- Finds unit vector w with $w^\top \nabla^2 f(x) w \leq -0.9\delta_1$ in $O\left(\frac{1}{\sqrt{\delta_1}}\right)$ iterations of computing M times a vector.
- Computing M times a vector? How to compute $\nabla^2 f(x) z$ for a vector z in our case? Easy:
- $\nabla^2 f(x) z = \lim_{\eta \to 0} \frac{\nabla f(x+\eta z) - \nabla f(x)}{\eta}$: Only two gradient evaluations for a sufficiently small η.

Non convex optimization: Last piece of Neon2:

- How to find eigenvectors of a matrix M?
- Power method: z_0 is a random unit vector, update $z_{t+1} = \frac{Mz_t}{\|Mz_t\|_2}$.
- Power method finds unit vector w with $w^\top \nabla^2 f(x) w \leq -0.9\delta_1$ in $O\left(\frac{1}{\delta_1}\right)$ iterations of computing M times a vector.
- Acceleration: Lanzos method/ Chevbyshev polynomial methods.
- Finds unit vector w with $w^\top \nabla^2 f(x) w \leq -0.9\delta_1$ in $O\left(\frac{1}{\sqrt{\delta_1}}\right)$ iterations of computing M times a vector.
- Computing M times a vector? How to compute $\nabla^2 f(x) z$ for a vector z in our case? Easy:
- $\nabla^2 f(x) z = \lim_{\eta \to 0} \frac{\nabla f(x+\eta z) - \nabla f(x)}{\eta}$: Only two gradient evaluations for a sufficiently small η.
- Critical Lemma of Neon2: η only needs to be $\frac{1}{\text{poly}(1/\delta_1)}$ small, and the approximation error won't mess up the eigenvectors solver.

Non convex optimization: The second approach

- Critical Lemma of Neon2: η only needs to be $\frac{1}{\text{poly}(1/\delta_1)}$ small, and the approximation error won't mess up the eigenvectors solver.

Non convex optimization: The second approach

- Critical Lemma of Neon2: η only needs to be $\frac{1}{\text{poly}(1/\delta_1)}$ small, and the approximation error won't mess up the eigenvectors solver.
- In general, when you have some errors in the internal computation of an optimization algorithm, would it mess up the entire algorithm?

Non convex optimization: The second approach

- Critical Lemma of Neon2: η only needs to be $\frac{1}{\text{poly}(1/\delta_1)}$ small, and the approximation error won't mess up the eigenvectors solver.
- In general, when you have some errors in the internal computation of an optimization algorithm, would it mess up the entire algorithm?
- This is the "stability analysis" of optimization algorithms, you can not learn it in any course (it is very hard).

Non convex optimization: The second approach

- Critical Lemma of Neon2: η only needs to be $\frac{1}{\text{poly}(1/\delta_1)}$ small, and the approximation error won't mess up the eigenvectors solver.
- In general, when you have some errors in the internal computation of an optimization algorithm, would it mess up the entire algorithm?
- This is the "stability analysis" of optimization algorithms, you can not learn it in any course (it is very hard).
- But you should know the answer: In general, the errors won't mess up the optimization algorithms (at least for gradient descent, mirror descent and accelerated gradient descent via linear coupling).

Non convex optimization: The second approach

- Find a point x such that:

Non convex optimization: The second approach

- Find a point x such that:
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.

Non convex optimization: The second approach

- Find a point x such that:
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.

Non convex optimization: The second approach

- Find a point x such that:
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- Suppose f is non-negative and the initial point x^{init} satisfies: $f(x^{init}) \leq 1$, then

Non convex optimization: The second approach

- Find a point x such that:
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- Suppose f is non-negative and the initial point x^{init} satisfies: $f(x^{init}) \leq 1$, then
- The second approach (Neon2) achieves the goal within: (ignoring poly(γ, β) factors)

Non convex optimization: The second approach

- Find a point x such that:
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- Suppose f is non-negative and the initial point x^{init} satisfies: $f(x^{init}) \leq 1$, then
- The second approach (Neon2) achieves the goal within: (ignoring poly(γ, β) factors)
  - $\tilde{O}\left(\frac{1}{\varepsilon^{1.75}}\right) + O\left(\frac{1}{\delta^{3.5}}\right)$ many gradient evaluations of f.

Non convex optimization: The second approach

- Find a point x such that:
  - $\|\nabla f(x)\|_2 \leq \varepsilon$.
  - $\nabla^2 f(x) \succeq -\delta I$.
- Suppose f is non-negative and the initial point x^{init} satisfies: $f(x^{init}) \leq 1$, then
- The second approach (Neon2) achieves the goal within: (ignoring poly(γ, β) factors)
  - $\tilde{O}\left(\frac{1}{\varepsilon^{1.75}}\right) + O\left(\frac{1}{\delta^{3.5}}\right)$ many gradient evaluations of f.
- This is essentially the only theorem you need to know for general non-convex optimization problems.

Non convex optimization: Non-general problems?

- However, unlike convex optimization, non-convex optimization is rarely given as a general problem: $\min f(x)$.

Non convex optimization: Non-general problems?

- However, unlike convex optimization, non-convex optimization is rarely given as a general problem: $\min f(x)$.
- According to Sanjeev Arora: Optimization algorithm is not the correct language for non-convex optimization.

Non convex optimization: Non-general problems?

- However, unlike convex optimization, non-convex optimization is rarely given as a general problem: $\min f(x)$.
- According to Sanjeev Arora: Optimization algorithm is not the correct language for non-convex optimization.
- We should use the special structure properties of f (for example f is a given by a neural network) to optimize it faster, instead of purely relying on optimization algorithms.

Non convex optimization: Non-general problems?

- However, unlike convex optimization, non-convex optimization is rarely given as a general problem: $\min f(x)$.
- According to Sanjeev Arora: Optimization algorithm is not the correct language for non-convex optimization.
- We should use the special structure properties of f (for example f is a given by a neural network) to optimize it faster, instead of purely relying on optimization algorithms.
- You have learnt Neon2, the only optimization algorithm you need to know for general non-convex optimization, which is:

Non convex optimization: Non-general problems?

- However, unlike convex optimization, non-convex optimization is rarely given as a general problem: $\min f(x)$.
- According to Sanjeev Arora: Optimization algorithm is not the correct language for non-convex optimization.
- We should use the special structure properties of f (for example f is a given by a neural network) to optimize it faster, instead of purely relying on optimization algorithms.
- You have learnt Neon2, the only optimization algorithm you need to know for general non-convex optimization, which is:
- 1% of non-convex optimization :)

Non convex optimization: Non-general problems?

- However, unlike convex optimization, non-convex optimization is rarely given as a general problem: $\min f(x)$.
- According to Sanjeev Arora: Optimization algorithm is not the correct language for non-convex optimization.
- We should use the special structure properties of f (for example f is a given by a neural network) to optimize it faster, instead of purely relying on optimization algorithms.
- You have learnt Neon2, the only optimization algorithm you need to know for general non-convex optimization, which is:
- 1% of non-convex optimization :)
- The rest 99% relies on understanding the structure of f, and we can say much more than just finding a local minima.

Non convex optimization: Non-general problems?

- However, unlike convex optimization, non-convex optimization is rarely given as a general problem: $\min f(x)$.
- According to Sanjeev Arora: Optimization algorithm is not the correct language for non-convex optimization.
- We should use the special structure properties of f (for example f is a given by a neural network) to optimize it faster, instead of purely relying on optimization algorithms.
- You have learnt Neon2, the only optimization algorithm you need to know for general non-convex optimization, which is:
- 1% of non-convex optimization :)
- The rest 99% relies on understanding the structure of f, and we can say much more than just finding a local minima.
- One example (further reading): Optimizing non-convex, non-smooth ReLU neural networks via SGD to global minima: A Convergence Theorem of Deep Learning via Over-parameterization.