# Convex Optimization 10-725, Lecture 17: Introduction to non-convex optimization

Yuanzhi Li

Assistant Professor, Carnegie Mellon University
Visiting Researcher, Microsoft

Today

# Last lecture

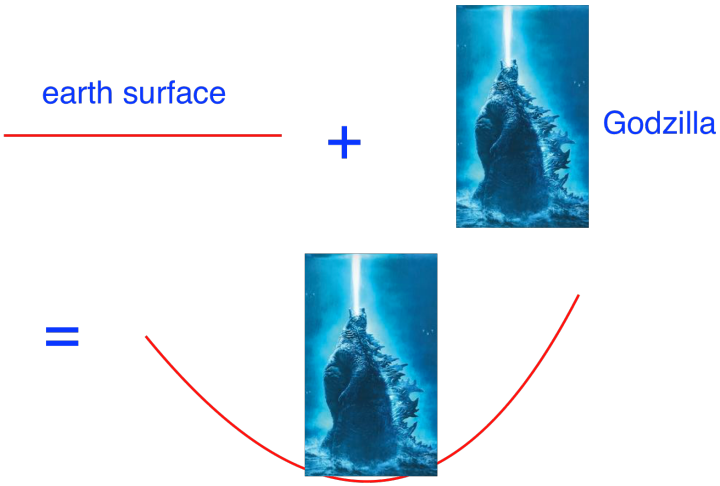
- We learnt that non-convexity is necessary for modern machine learning.
- Modern machine learning: Hierarchical Learning = Non-convex optimization.
- Classical machine learning: Shallow learning = convex optimization.

# This lecture

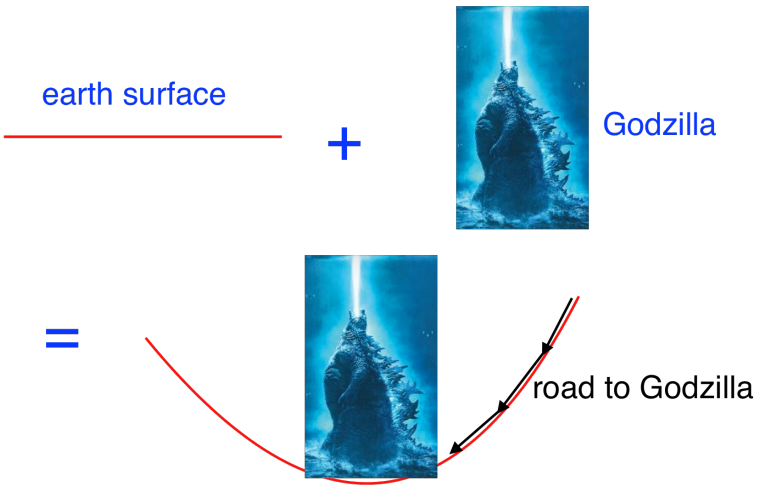
- We are going to learn some basic language of non-convex optimization: Critical points, local minimal, saddle points.
- We are going to learn algorithm Hessian Descent and noisy gradient descent for non-convex optimization.

# Convex optimization

- Convex optimization: Where is the Godzilla?

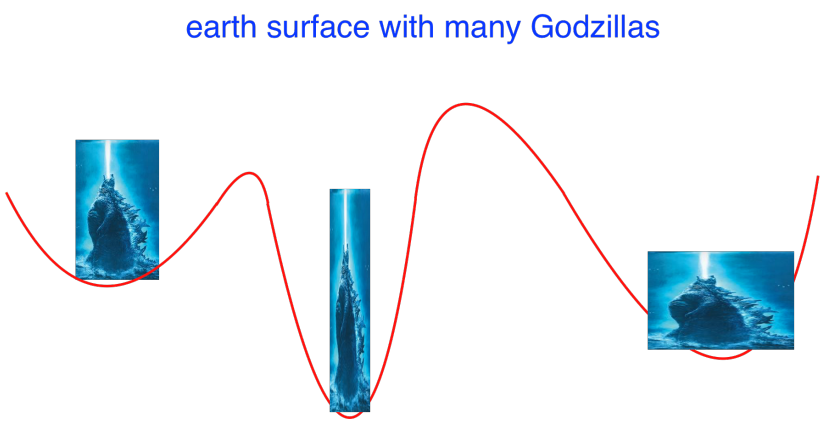



- Weight of the Godzilla: The Lipschitzness/smoothness.
- To find the Godzilla: follow the (negative) gradient direction.



# Non convex optimization

- Where are the Godzillas?

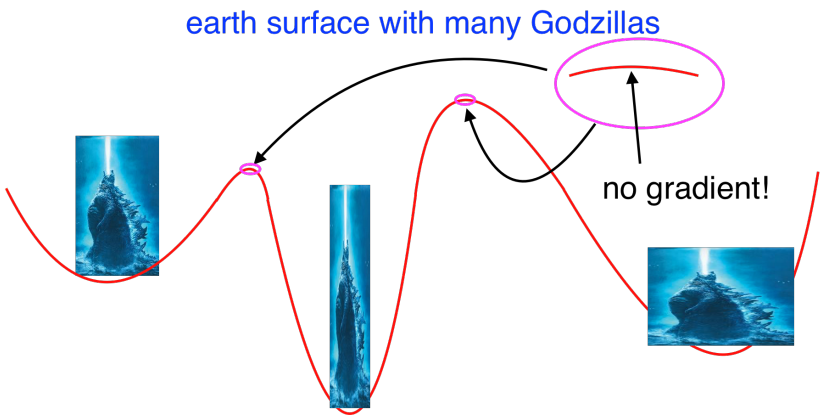



- 
- Each Godzilla defines a local minima.
- The "heaviest" Godzilla: The global minima.
- 

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

# Non convex optimization

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




- 
- There are "saddle points", whose gradients are also zero but not "locally minimal".
- In fact, in high dimension, one can construct a function where gradient descent almost always stuck at a saddle point.

# 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, will be covered later).
- In this lecture, we will see the non-convex optimization foundations: The basic definition, and the basic algorithm.

# Non convex optimization: The definition

- We start with the basic terminology for non-convex optimization: Local minima, global minima, saddle points.
- 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 $x$:

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

  Here, $\| * \|_2$ is the Euclidean norm, $a = b \pm c$ means $a \in [b-c, b+c]$.
- This holds for all functions, convex or not.

# Non convex optimization: The definition

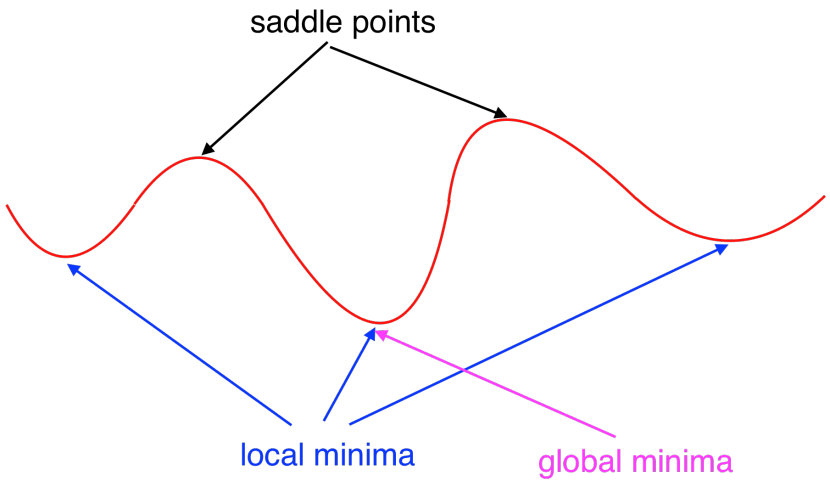
- Recall we defined the Lipschitzness of the hessian $\gamma$: For all $x, y \in \mathbb{R}^d$, $\|\nabla^2 f(x) - \nabla^2 f(y)\|_{sp} \leq \gamma \|x - y\|_2$. $\| * \|_{sp}$ is the spectral norm.
- This implies: For every $x, \tau \in \mathbb{R}^d$:

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

- For non-convex optimization: $\nabla^2 f(x)$ might not be positive semi-definite: This gives us local minimal and saddle points.

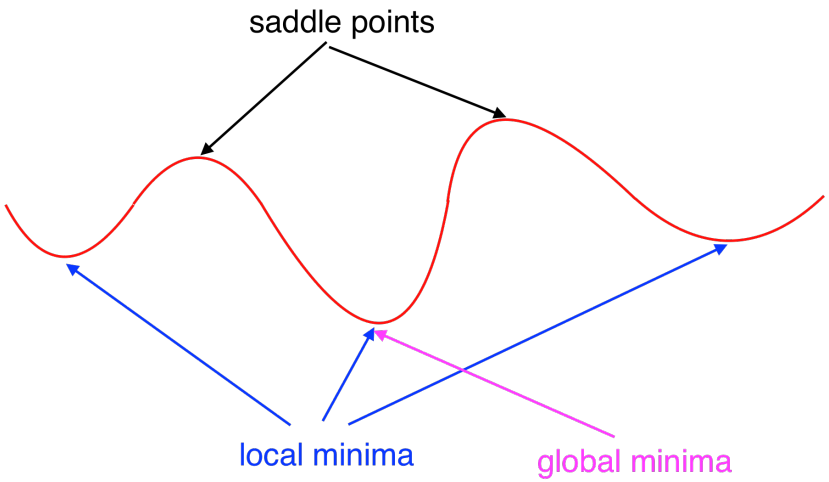
# Non convex optimization: The property

- 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?



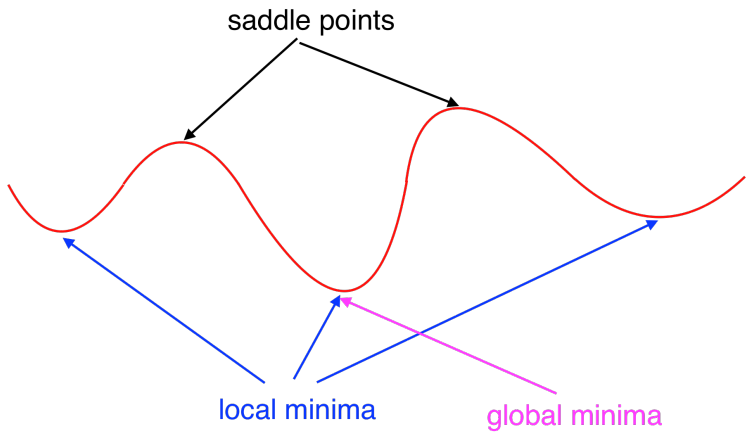

- There are global minima, local minima, saddle points.
- They are points whose gradients are zero, but have different properties.

# Non convex optimization: The property




- Non-convex landscape:
- A point $x$ is called a local minima (second-order local minima), if
- $\nabla f(x) = 0$ and $\nabla^2 f(x)$ is PSD (positive semi-definite, i.e. $\nabla^2 f(x) \succeq 0$).
- In particular, the global minima ($x^* = \text{argmin} f(x)$) is a special local minima.
- A point $x$ is called a saddle point, if
- $\nabla f(x) = 0$ and $\nabla^2 f(x)$ is not PSD – meaning there exists a $v \in \mathbb{R}^d$ such that $v^\top \nabla^2 f(x) v < 0$.

# Non convex optimization: The property




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

**Warning**

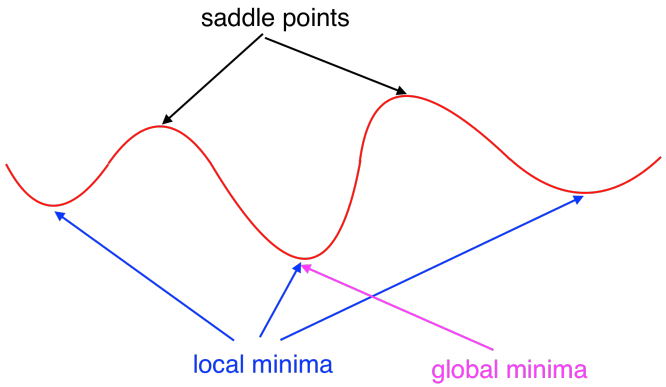
Be very clear: when people talk about local minima, they typically mean second oder local minima.

They DO NOT mean a point $x$ such that for every sufficiently small $\delta$, $f(x+\delta) \geq f(x)$

- $f(x) = x^3$ for $x = 0$ is a second order local minima.

# Non convex optimization: What do we want?

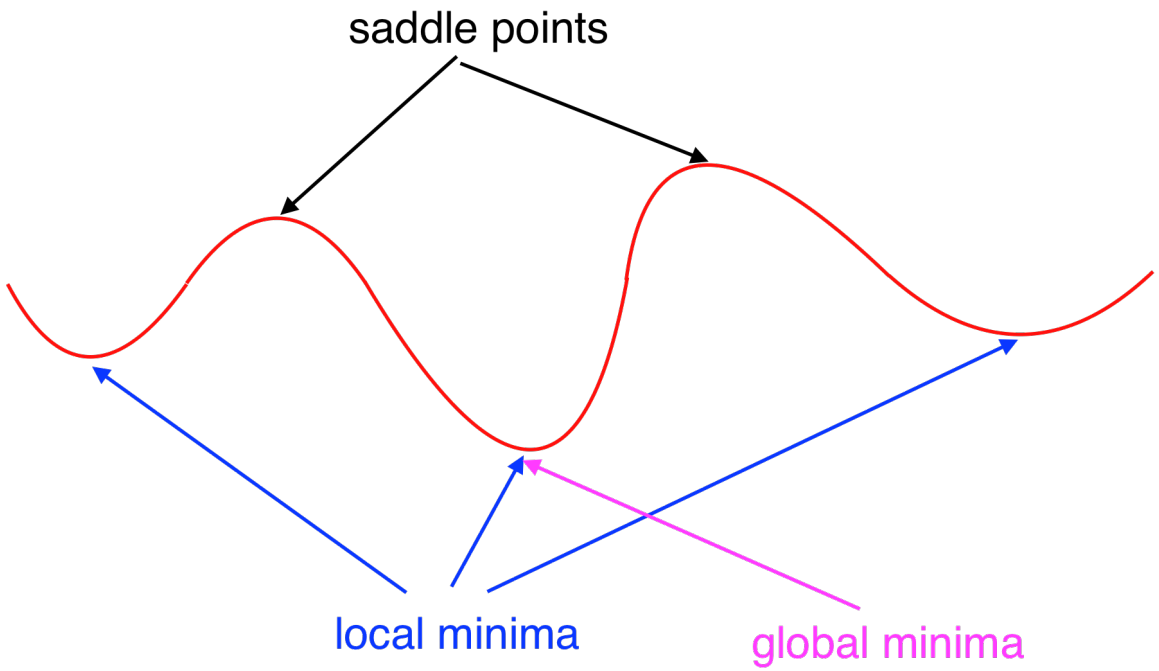
- What do we want when optimizing a non-convex function $f$?
- Recall: If we do gradient descent, we can find a point $x$ whose gradient $\nabla f(x) \approx 0$ efficiently, convex or not.
- But $\nabla f(x) = 0$ for a non-convex function does not mean that we are at the global minima.
- They can be (second-order) local minima or saddle points.




- This is because $\nabla f(x)$ might not be PSD for non-convex functions: Locally, when $\nabla f(x) = 0$:

$$f(x+\tau) \approx f(x) + \frac{1}{2}\tau^\top \nabla^2 f(x)\tau$$

# Non convex optimization: The first goal




- For non-convex optimization, the first goal is to at least find a (second order) local minima.
- Recall: Gradient Descent might stuck at a saddle points.
- How do we fix it?

# Non convex optimization: The Hessian Descent

- We first introduce the most easily understandable non-convex optimization algorithm: The Hessian Descent:
- It is not used at all in practice, but it is still extremely important.
- It is the spirit behind all other algorithms to find a (second order) local minima.

**Warning**

Be very clear: when people talk about local minima, they typically mean second oder local minima.
They DO NOT mean a point $x$ such that for every sufficiently small $\delta$, $f(x+\delta) \geq f(x)$ – Such "local minima" CAN NOT BE FOUND EFFICIENTLY (there are many machine learning papers that switch definitions to cheat).

# Non convex optimization: The Hessian Descent

- Algorithm Hessian Descent:
- At every iteration $t$, do:
- Update (gradient descent): $y_{t+1} = x_t - \eta \nabla f(x_t)$.
- Update (hessian descent): Find a unit vector $v$ corresponding to the eigenvector of $\nabla^2 f(x_t)$ with the smallest (can be negative) eigenvalue, define

$$z_{t+1,1} = x_t - \eta v, \quad z_{t+1,2} = x_t + \eta v$$

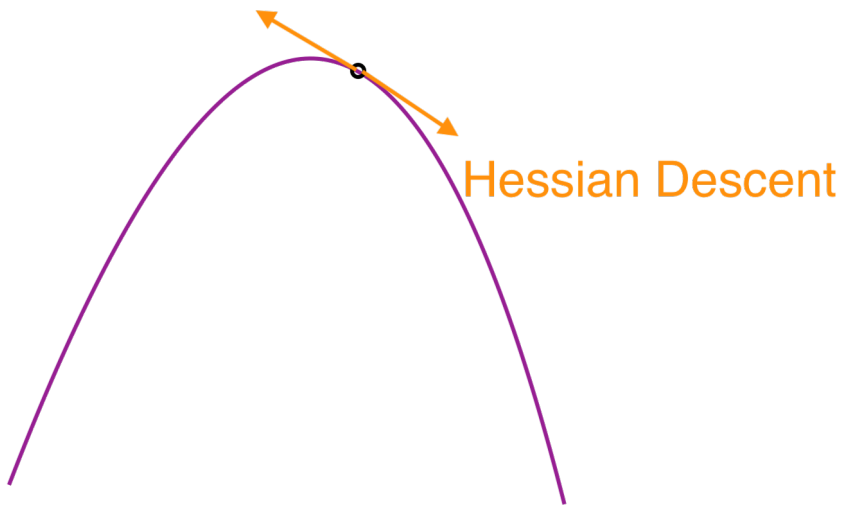
- Update

$$x_{t+1} = \text{argmin}\{f(y_{t+1}), f(z_{t+1,1}), f(z_{t+1,2})\}$$

# Non convex optimization: The Hessian Descent

- Hessian Descent Step: Find a unit vector $v$ corresponding to the eigenvector of $\nabla^2 f(x_t)$ with the minimal (can be negative) eigenvalue, define

$$z_{t+1,1} = x_t - \eta v, \quad z_{t+1,2} = x_t + \eta v$$



# Hessian descent: Convergence rate

- Recall: For a $L$-smooth function, convex or not, using $\eta \leq \frac{1}{L}$, we have:
- For the update $y_{t+1} = x_t - \eta \nabla f(x_t)$:

$$f(y_{t+1}) \leq f(x_t) - \frac{\eta}{2} \|\nabla f(x_t)\|_2^2$$

- In other words, Gradient Large $\Longrightarrow$ Large Objective Decrease.

# Hessian descent: Convergence rate

- For Hessian Descent Step: Find a unit vector $v$ corresponding to the eigenvector of $\nabla f(x_t)$ with the smallest (can be negative) eigenvalue, define

$$z_{t+1,1} = x_t - \eta v, \quad z_{t+1,2} = x_t + \eta v$$

- Recall: When $f$ is $\gamma$-Hessian Lipschitz: For every $x, \tau$:

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

- Therefore, we have

$$\frac{1}{2}\left(f(x+\tau) + f(x-\tau)\right) \le f(x) + \frac{1}{2}\tau^\top \nabla^2 f(x)\tau + \gamma\|\tau\|_2^3$$

# Hessian descent: Convergence rate

- For Hessian Descent Step: Find a unit vector $v$ corresponding to the eigenvector of $\nabla f(x_t)$ with the minimal (can be negative) eigenvalue, define

$$z_{t+1,1} = x_t - \eta v, \quad z_{t+1,2} = x_t + \eta v$$

- We also have:

$$\frac{1}{2}\left(f(x+\tau) + f(x-\tau)\right) \le f(x) + \frac{1}{2}\tau^\top \nabla^2 f(x)\tau + \gamma\|\tau\|_2^3$$

- Apply here with $\tau = \eta v$, we have:

$$\frac{1}{2}\left(f(z_{t+1,1}) + f(z_{t+1,2})\right) \le f(x_t) + \frac{\eta^2}{2} v^\top \nabla^2 f(x_t) v + \gamma\eta^3\|v\|_2^3$$

# Hessian descent: Convergence rate

- For Hessian Descent Step: Find a unit vector $v$ corresponding to the eigenvector of $\nabla f(x_t)$ with the minimal (can be negative) eigenvalue, define

$$z_{t+1,1} = x_t - \eta v, \quad z_{t+1,2} = x_t + \eta v$$

- Now we have:

$$\frac{1}{2}\left(f(z_{t+1,1}) + f(z_{t+1,2})\right) \le f(x_t) + \frac{\eta^2}{2} v^\top \nabla^2 f(x_t) v + \gamma \eta^3 \|v\|_2^3$$

- Suppose $v^\top \nabla^2 f(x_t) v = -\delta$ for $\delta > 0$, for every $\eta \le \frac{\delta}{4\gamma}$, we have

$$\frac{1}{2}\left(f(z_{t+1,1}) + f(z_{t+1,2})\right) \le f(x_t) - \frac{\eta^2}{4}\delta$$

- In other words, Hessian Negative (locally highly non-convex) $\Longrightarrow$ Large Objective Decrease.

# Hessian descent: Convergence rate

- Now we have: Suppose $v^\top \nabla^2 f(x_t) v = -\delta$ for $\delta > 0$, for every $\eta \le \frac{\delta}{4\gamma}$, we have

$$\frac{1}{2}\left(f(z_{t+1,1}) + f(z_{t+1,2})\right) \le f(x_t) - \frac{\eta^2}{4}\delta$$

- In other words, Hessian Negative $\Longrightarrow$ Large Objective Decrease.
- On the other hand, recall for the update $y_{t+1} = x_t - \eta \nabla f(x_t)$:

$$f(y_{t+1}) \le f(x_t) - \frac{\eta}{2}\|\nabla f(x_t)\|_2^2$$

- In other words, Gradient Large $\Longrightarrow$ Large Objective Decrease.
- Together, algorithm Hessian Descent can find a point whose gradient is close to zero, and whose Hessian is close to PSD efficiently.

# Beyond Hessian descent

- Recall we mentioned: Hessian Descent is not used at all in practice.
- This is due to a simple reason: (noisy) Gradient Descent can already find a point whose gradient is close to zero, and whose Hessian is close to PSD efficiently.
- In other words, (noisy) Gradient Descent can find an approximate (second-order) local minima efficiently.

# Noisy Gradient Descent

- Algorithm (noisy) Gradient Descent: At every iteration, update:

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

- For every $T_0$ iterations, further update $x_{t+1} \leftarrow x_{t+1} + \xi_{t+1}$, where $\xi_{t+1} \sim \mathcal{N}(0, \sigma^2 I)$ is a small Gaussian Noise.
- Gradient Descent might stuck at a saddle points, but adding a little bit noise can already help gradient descent algorithm efficiently escape saddle points.
- In practice, since we are using stochastic gradient descent for most of the problems, adding such noise is usually unnecessary.

# Noisy Gradient Descent: Intuition

- By the update rule:

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

- We know that

$$\nabla f(x_{t+1}) = \nabla f(x_t) - \eta \nabla^2 f(x_t) \nabla f(x_t) + R_t$$

- Where $\|R_t\|_2 = O(\eta^2)$.
- At every iteration, we are updating

$$\nabla f(x_{t+1}) \approx (I - \eta \nabla^2 f(x_t)) \nabla f(x_t)$$

# Noisy Gradient Descent: Intuition

- At every iteration, we are updating

$$\nabla f(x_{t+1}) \approx (I - \eta \nabla^2 f(x_t)) \nabla f(x_t)$$

- Now, suppose $\nabla f(x_t)$ is sufficiently small for all $t = T_1, T_1 + 1, \cdots, T_1 + T$ iterations for a fairly large $T$ (otherwise, we can decrease the objective value simply using gradient descent),
- By the update rule:

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

  We know that $x_t$ is close to $x_{T_1}$ for all $t \in [T_1, T_1 + T]$.
- Since Hessian is Lipschitz, $\nabla^2 f(x_t)$ is close to $\nabla^2 f(x_{T_1})$ as well. This implies that

$$\nabla f(x_{t+1}) \approx (I - \eta \nabla^2 f(x_{T_1})) \nabla f(x_t)$$

# Noisy Gradient Descent: Intuition

- Now we are updating:

$$\nabla f(x_{t+1}) \approx (I - \eta \nabla^2 f(x_{T_1})) \nabla f(x_t)$$

- This is the so-called power method, it will converge to the eigenvector of $(I - \eta \nabla^2 f(x_{T_1}))$ with the largest eigenvalue, as long as the starting point $\nabla f(x_{T_1})$ is not very adversarial, that is why we need to add a bit random noise.
- In particular, when $\nabla^2 f(x_{T_1})$ is not PSD, then $\|(I - \eta \nabla^2 f(x_{T_1}))\|_2 > 1$, which means that gradient will become large again after some iterations, and gradient descent will decrease objective again.
- The algorithm will stop only if (1). Gradient is small (2). Hessian is approximately PSD.

# Noisy Gradient Descent: More find grind analysis

- Suppose $\|\nabla f(x_t)\|_2 \le \rho$ is for all $t = T_1, T_1+1, \cdots, T_1+T_0-1$, where $T_1 = kT_0+1$ for some integer $k$:
- By the update rule:

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

- We know that $\|x_t - x_{T_1}\|_2 \le \eta\rho T_0$ for all $t \in [T_1, T_1+T_0-1]$.
- This implies that

$$\left\|\nabla f(x_{t+1}) - (I - \eta\nabla^2 f(x_{T_1}))\nabla f(x_t)\right\|_2 \le O(\gamma\eta\rho T_0)$$

- Suppose there is a unit eigenvector $v$ such that $v^\top \nabla^2 f(x_{T_1}) v = -\delta$ for some $\delta > 0$, then we can obtain:

$$|\langle v, \nabla f(x_{t+1})\rangle| \ge (1+\eta\delta)|\langle v, \nabla f(x_t)\rangle| - \gamma\eta\rho T_0$$

- For sufficiently small $\rho$, $|\langle v, \nabla f(x_{t+1})\rangle|$ is increasing at a geometry rate.

# Noisy Gradient Descent: Summary

- The rate of convergence for noisy gradient descent is $\mathrm{poly}(1/\varepsilon, 1/\delta)$ to find a point whose gradient is smaller than $\varepsilon$ and whose hessian is $\geq -\delta I$.
- The rate is not very important, just the spirit is important: gradient descent with a little bit noise can already converge to an approximate (second order) local minima efficiently.