10-301/601: Introduction to Machine Learning Lecture 10 – Optimization for Machine Learning

Henry Chai

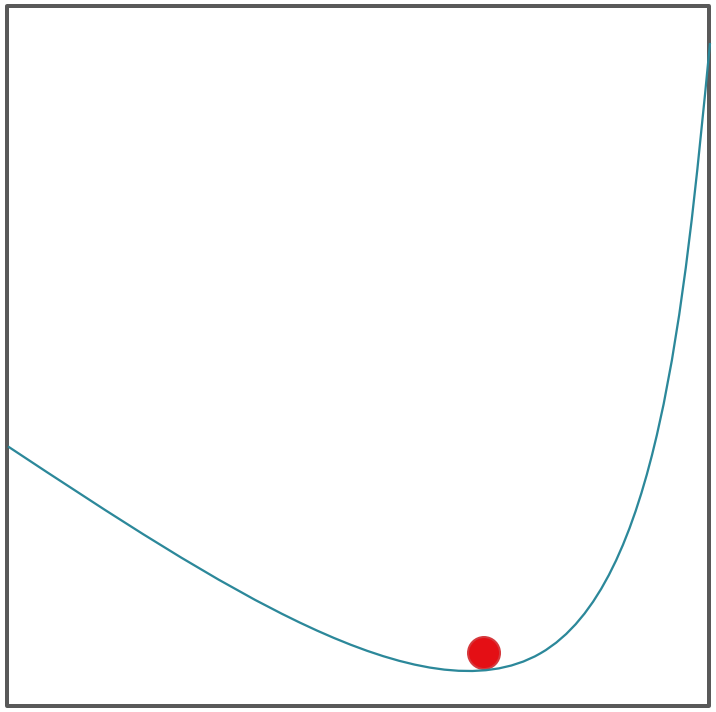
5/20/25

Front Matter

- Announcements:
  - HW2 released on 5/16, due 5/20 (today!) at 11:59 PM
  - HW3 to be released on 5/20 (today!), due 5/23 at 11:59 PM

Recall: Gradient Descent

- An iterative method for minimizing functions
- Requires the gradient to exist everywhere



- Good news: the negative conditional log-likelihood is *convex*!

Gradient Descent

- Input: $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}, \eta^{(0)}$

1. Initialize the parameters $\boldsymbol{\theta}^{(0)}$ and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Compute the gradient: $\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)$
   b. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta^{(0)} \nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)$
   c. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

Gradient Descent

- Input: $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}, \eta^{(0)}$

1. Initialize the parameters $\boldsymbol{\theta}^{(0)}$ and set $t = 0$
2. While $\left\|\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)\right\| > \epsilon$
   a. Compute the gradient:
      $\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)$
   b. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta^{(0)} \nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)$
   c. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

Gradient Descent

- Input: $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}, \eta^{(0)}$

1. Initialize the parameters $\boldsymbol{\theta}^{(0)}$ and set $t = 0$
2. While $t < T$
   a. Compute the gradient:
   $\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)$
   b. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta^{(0)} \nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)$
   c. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

Gradient Descent for Logistic Regression

- Input: $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}, \eta^{(0)}$

1. Initialize $\boldsymbol{\theta}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Compute the gradient:

$$\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right) = \sum_{n=1}^{N} \boldsymbol{x}^{(n)}\left(P\left(Y = 1 \middle| \boldsymbol{x}^{(n)}, \boldsymbol{\theta}^{(t)}\right) - y^{(n)}\right)$$

   b. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta^{(0)} \nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)$
   c. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

Why Gradient Descent for Logistic Regression?

- Input: $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}, \eta^{(0)}$

1. Initialize $\boldsymbol{\theta}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Compute the gradient:

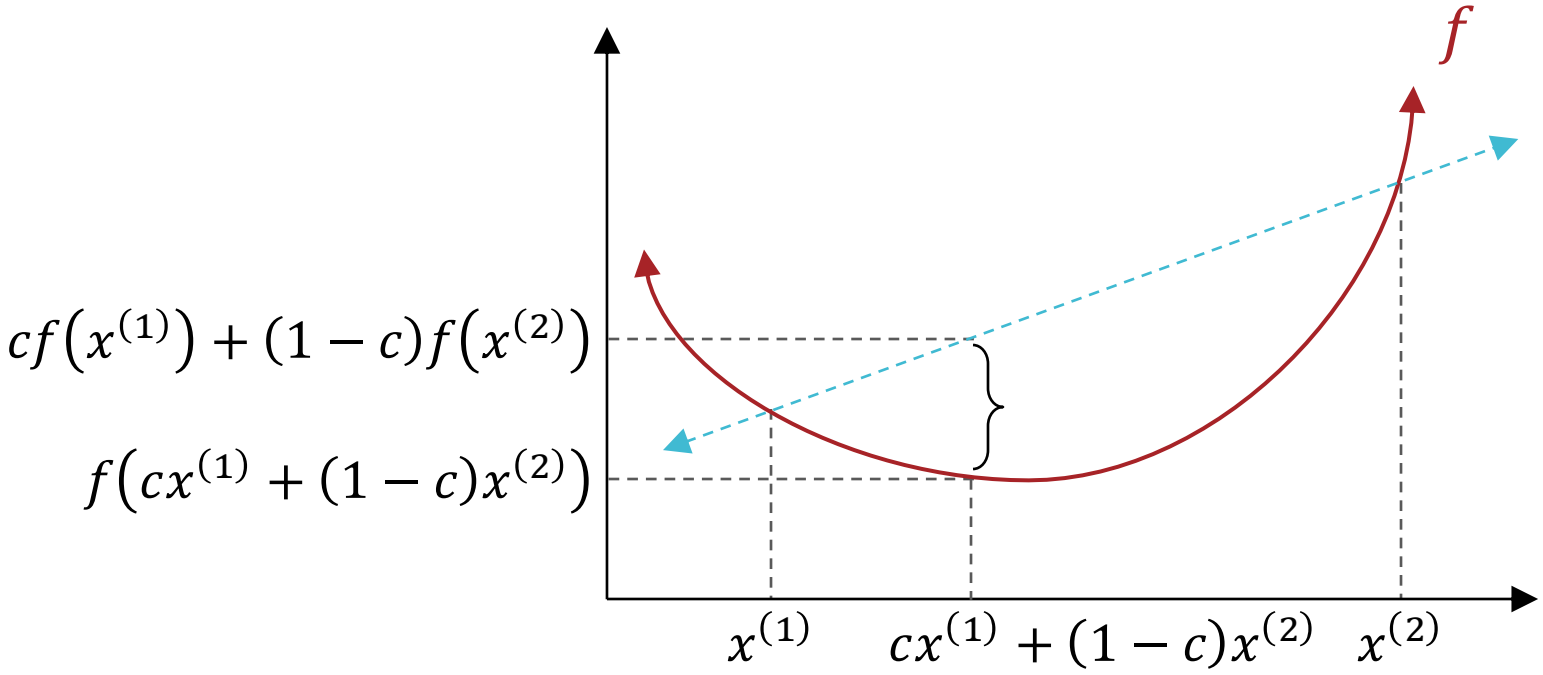
$$\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right) = \sum_{n=1}^{N} \boldsymbol{x}^{(n)}\left(P\left(Y = 1 \middle| \boldsymbol{x}^{(n)}, \boldsymbol{\theta}^{(t)}\right) - y^{(n)}\right)$$

   b. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta^{(0)} \nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)$
   c. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

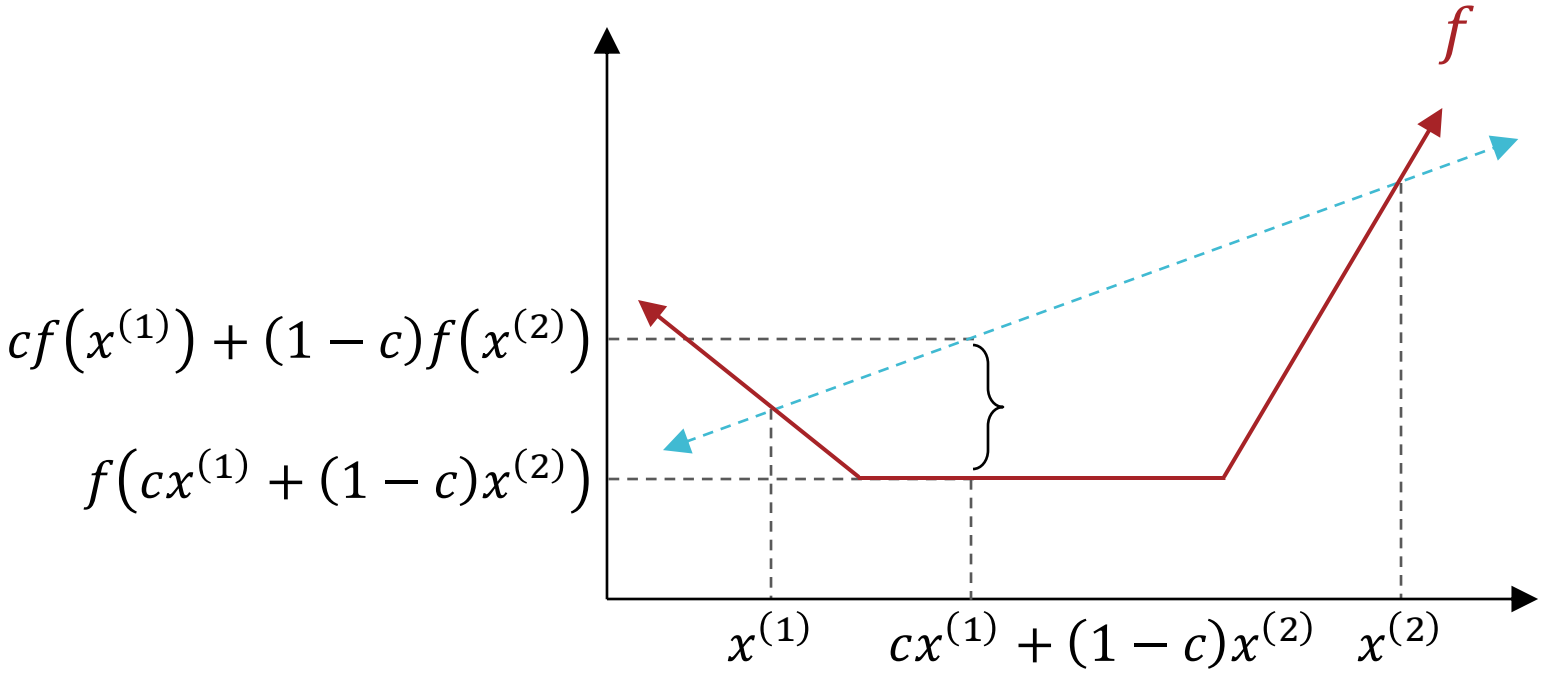
Convexity

- A function $f: \mathbb{R}^D \to \mathbb{R}$ is convex if
  $\forall\, \boldsymbol{x}^{(1)} \in \mathbb{R}^D, \boldsymbol{x}^{(2)} \in \mathbb{R}^D$ and $0 \leq c \leq 1$
  $$f\left(c\boldsymbol{x}^{(1)} + (1-c)\boldsymbol{x}^{(2)}\right) \leq cf\left(\boldsymbol{x}^{(1)}\right) + (1-c)f\left(\boldsymbol{x}^{(2)}\right)$$



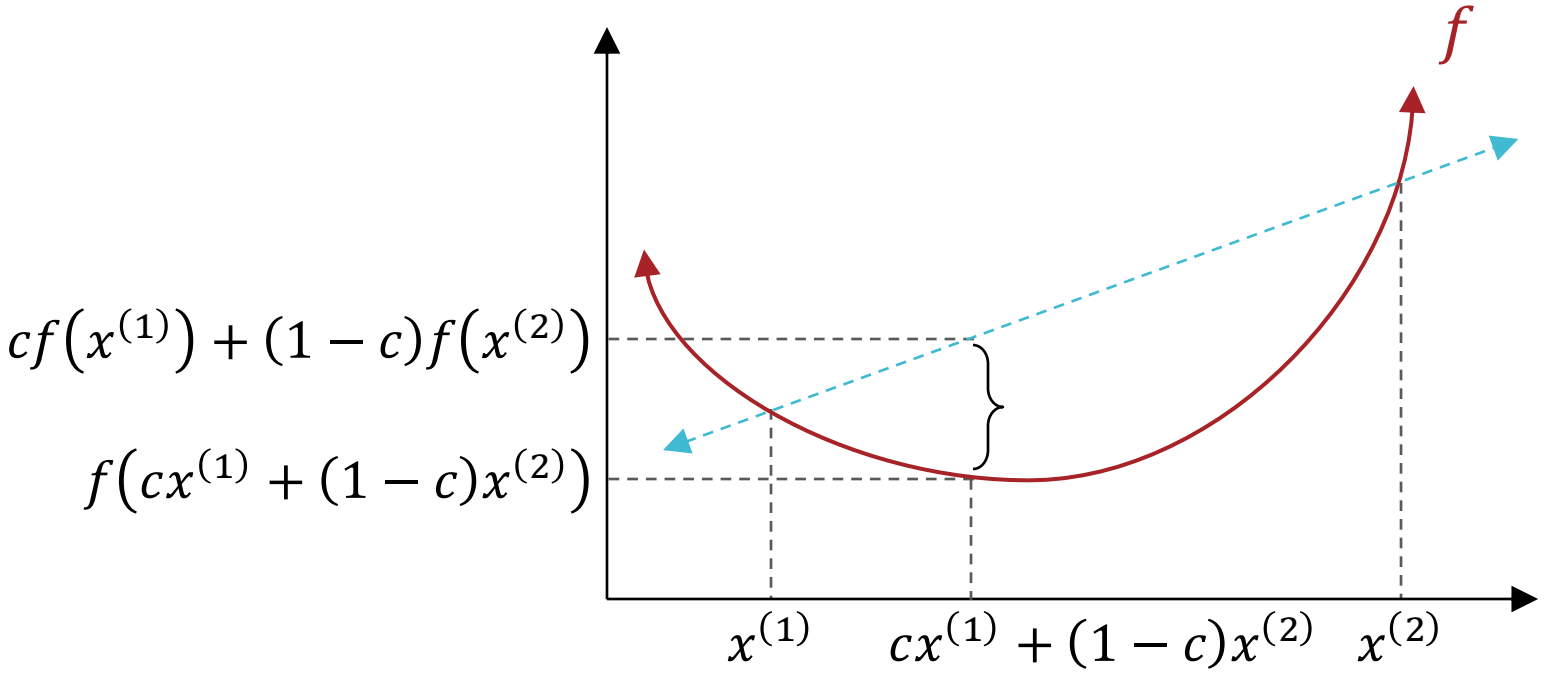
Convexity

- A function $f: \mathbb{R}^D \to \mathbb{R}$ is convex if
  $\forall\, \boldsymbol{x}^{(1)} \in \mathbb{R}^D, \boldsymbol{x}^{(2)} \in \mathbb{R}^D$ and $0 \leq c \leq 1$
  $$f\left(c\boldsymbol{x}^{(1)} + (1-c)\boldsymbol{x}^{(2)}\right) \leq cf\left(\boldsymbol{x}^{(1)}\right) + (1-c)f\left(\boldsymbol{x}^{(2)}\right)$$

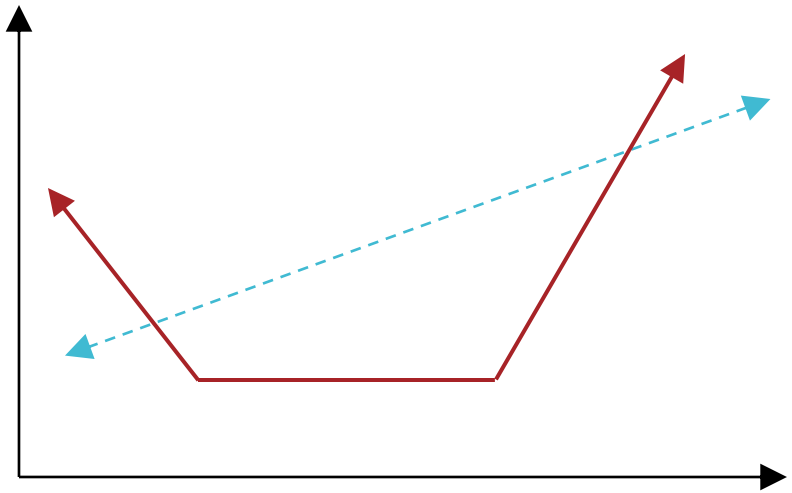


Convexity

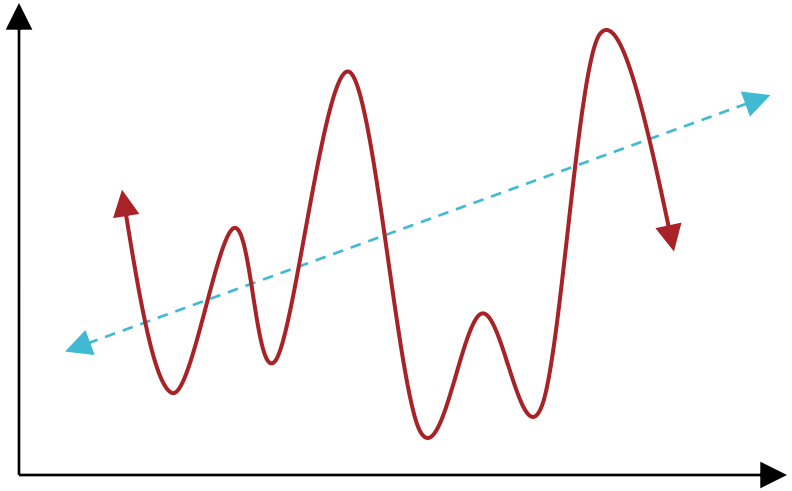
- A function $f: \mathbb{R}^D \to \mathbb{R}$ is *strictly* convex if
  $\forall\, \boldsymbol{x}^{(1)} \in \mathbb{R}^D, \boldsymbol{x}^{(2)} \in \mathbb{R}^D$ and $0 < c < 1$
  $$f\left(c\boldsymbol{x}^{(1)} + (1-c)\boldsymbol{x}^{(2)}\right) < cf\left(\boldsymbol{x}^{(1)}\right) + (1-c)f\left(\boldsymbol{x}^{(2)}\right)$$



Convexity

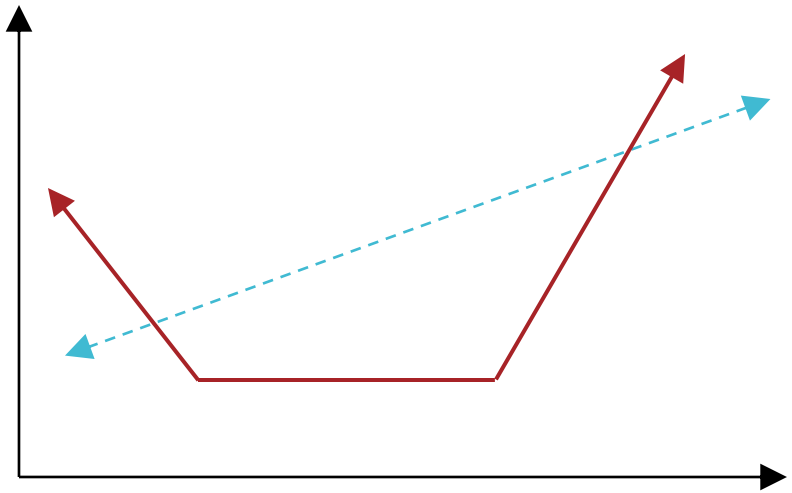


Convex functions



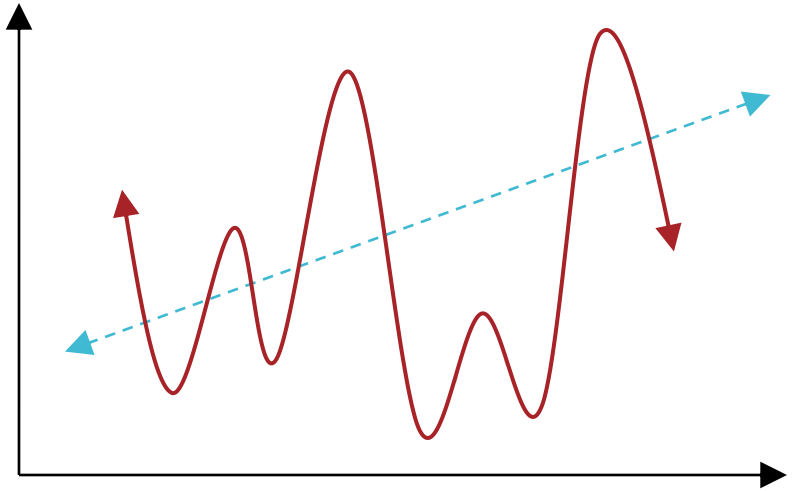
Non-convex functions

Convexity



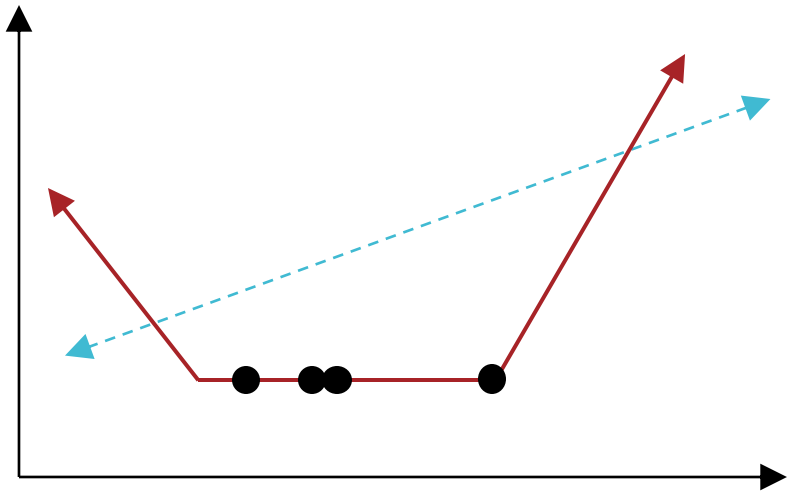
Given a function $f: \mathbb{R}^D \to \mathbb{R}$

- $\boldsymbol{x}^*$ is a global minimum iff $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) \; \forall \; \boldsymbol{x} \in \mathbb{R}^D$



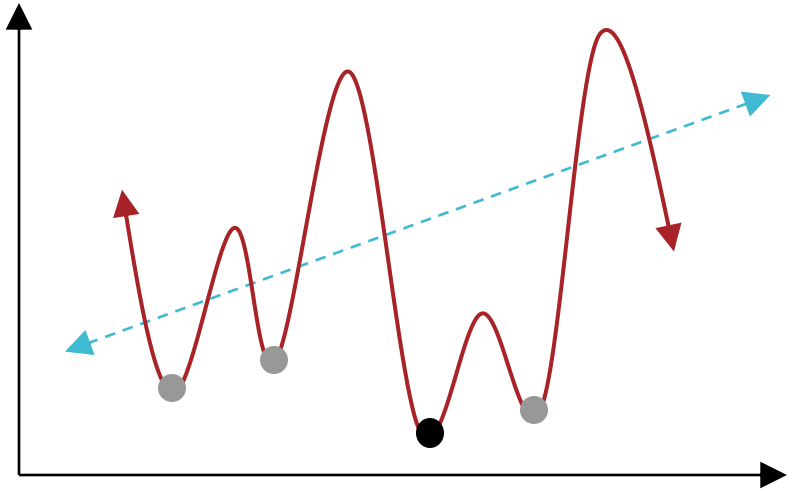
- $\boldsymbol{x}^*$ is a local minimum iff $\exists \; \epsilon$ s.t. $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) \; \forall \; \boldsymbol{x}$ s.t. $\|\boldsymbol{x} - \boldsymbol{x}^*\|_2 < \epsilon$

Convexity



Convex functions:

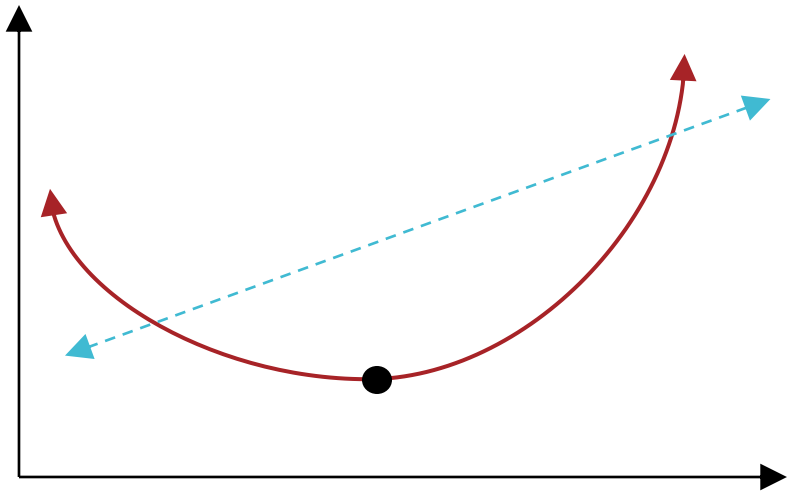
Each local minimum is a global minimum!



Non-convex functions:

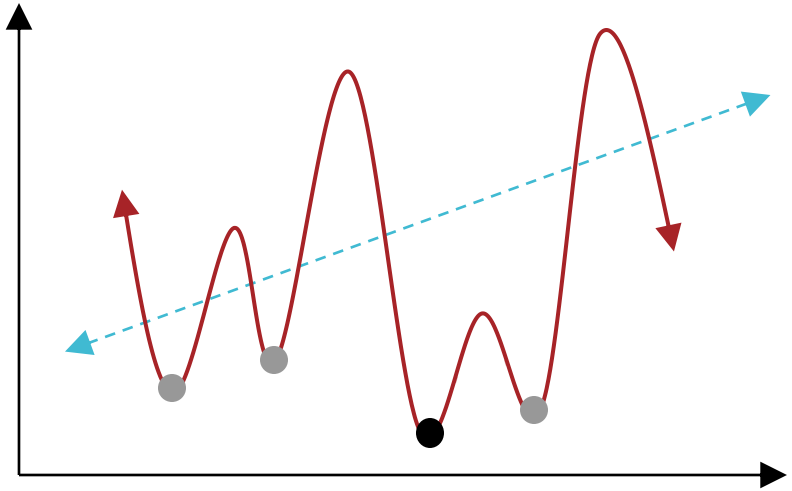
A local minimum may or may not be a global minimum…

Convexity



Strictly convex functions:

There exists a unique global minimum!

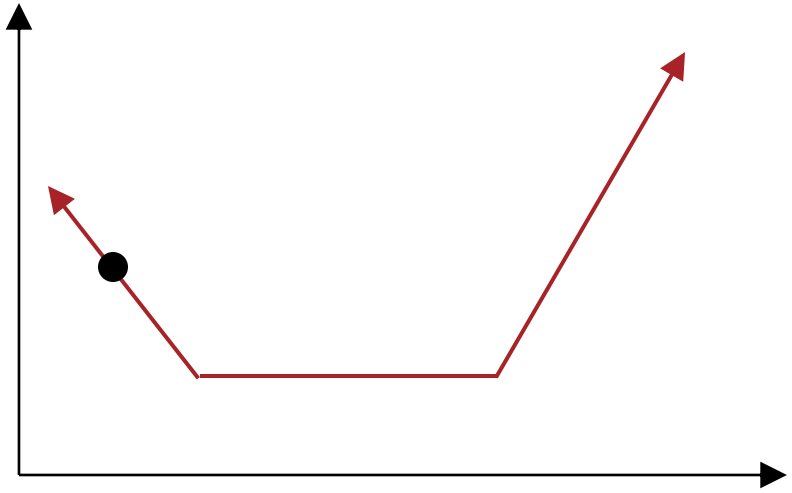


Non-convex functions:

A local minimum may or may not be a global minimum…

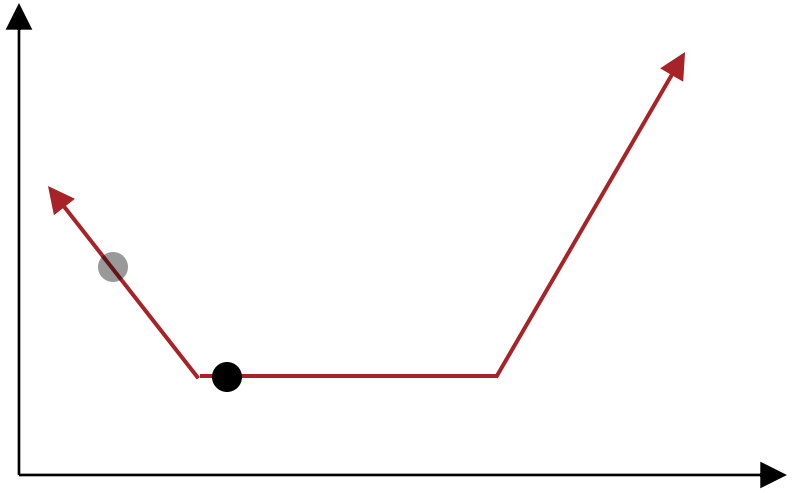
Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Works great if the objective function is convex!



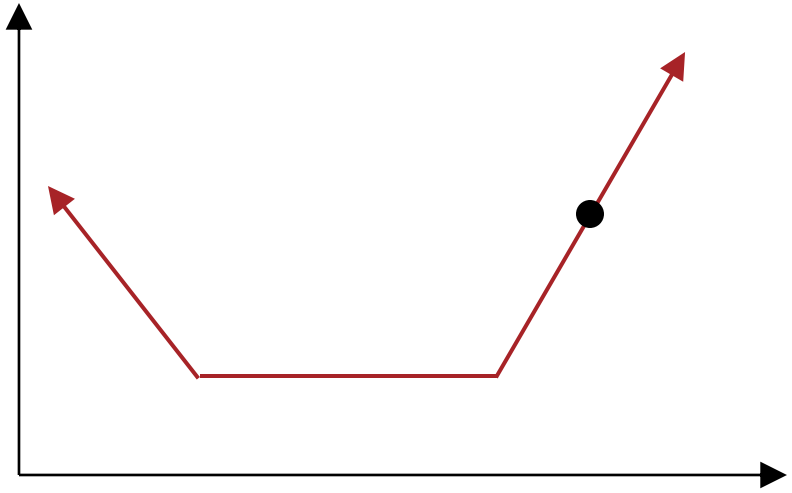
Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Works great if the objective function is convex!



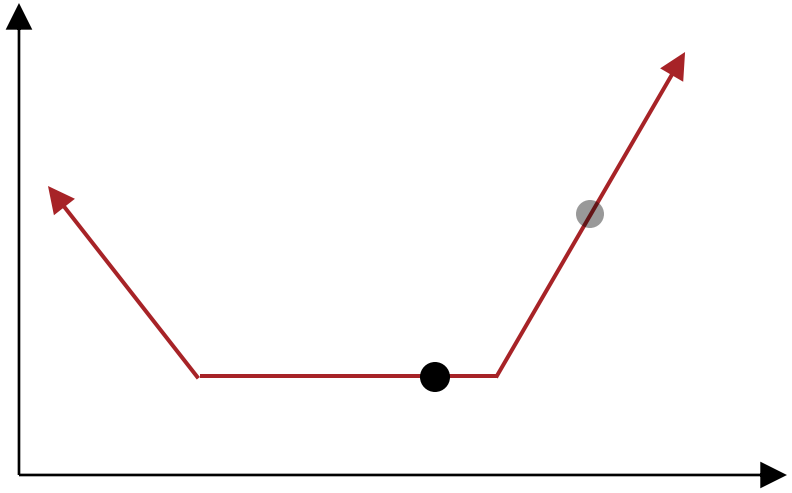
Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Works great if the objective function is convex!



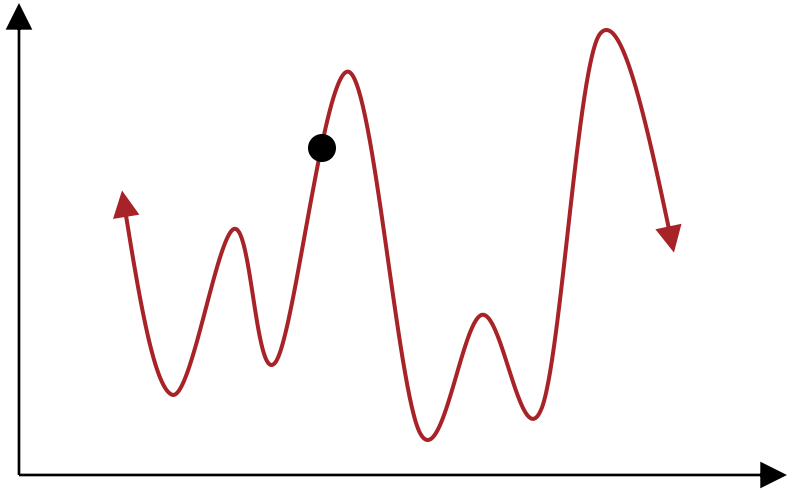
Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Works great if the objective function is convex!



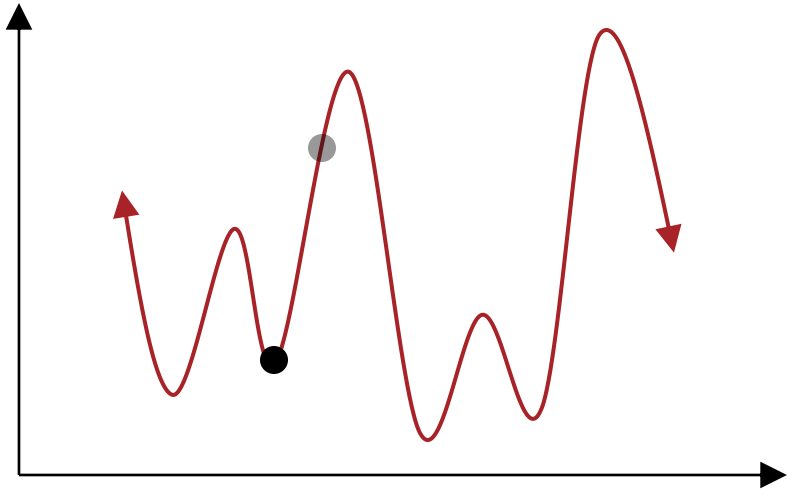
Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Not ideal if the objective function is non-convex…



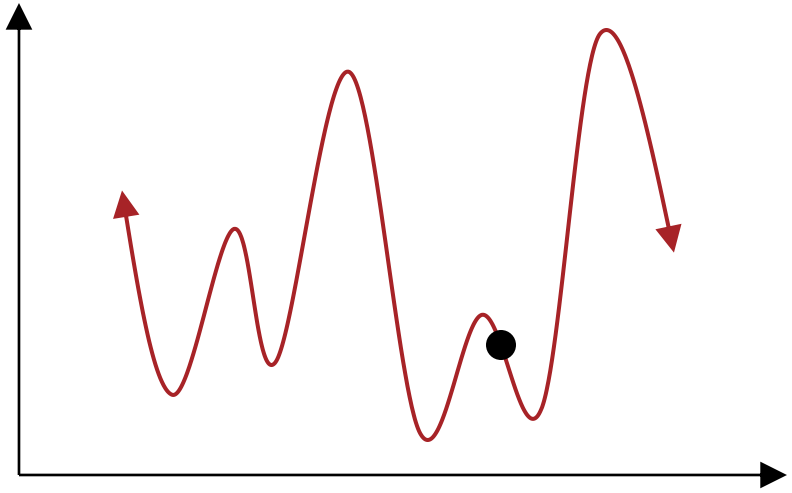
Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Not ideal if the objective function is non-convex…



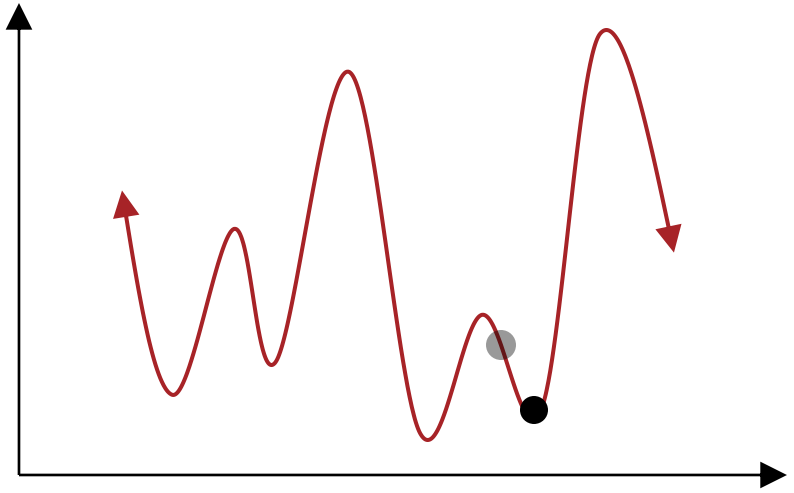
Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Not ideal if the objective function is non-convex…



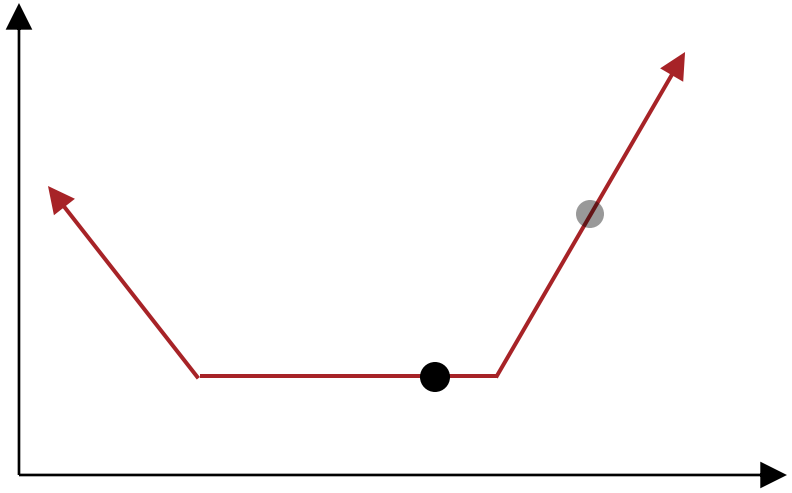
Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Not ideal if the objective function is non-convex…



Gradient Descent for Logistic Regression

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - The negative conditional log-likelihood is convex! But not *strictly* convex…



Gradient Descent for Logistic Regression

- Input: $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}, \eta^{(0)}$

1. Initialize $\boldsymbol{\theta}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Compute the gradient:

$$\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right) = \sum_{n=1}^{N} \boldsymbol{x}^{(n)}\left(P\left(Y = 1 \middle| \boldsymbol{x}^{(n)}, \boldsymbol{\theta}^{(t)}\right) - y^{(n)}\right)$$

   b. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta^{(0)} \nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)$
   c. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

Stochastic Gradient Descent for Logistic Regression

- Input: $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}, \eta_{SGD}^{(0)}$

1. Initialize $\boldsymbol{\theta}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Randomly sample a data point from $\mathcal{D}$, $\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)$
   b. Compute the *pointwise* gradient:
   $$\nabla_{\boldsymbol{\theta}} \ell^{(n)}\left(\boldsymbol{\theta}^{(t)}\right) = \boldsymbol{x}^{(n)}\left(P\left(Y = 1 \middle| \boldsymbol{x}^{(n)}, \boldsymbol{\theta}^{(t)}\right) - y^{(n)}\right)$$
   c. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta_{SGD}^{(0)} \nabla_{\boldsymbol{\theta}} \ell^{(n)}\left(\boldsymbol{\theta}^{(t)}\right)$
   d. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

Stochastic Gradient Descent

- If the data point is sampled uniformly at random, then the expected value of the pointwise gradient is proportional to the full gradient:

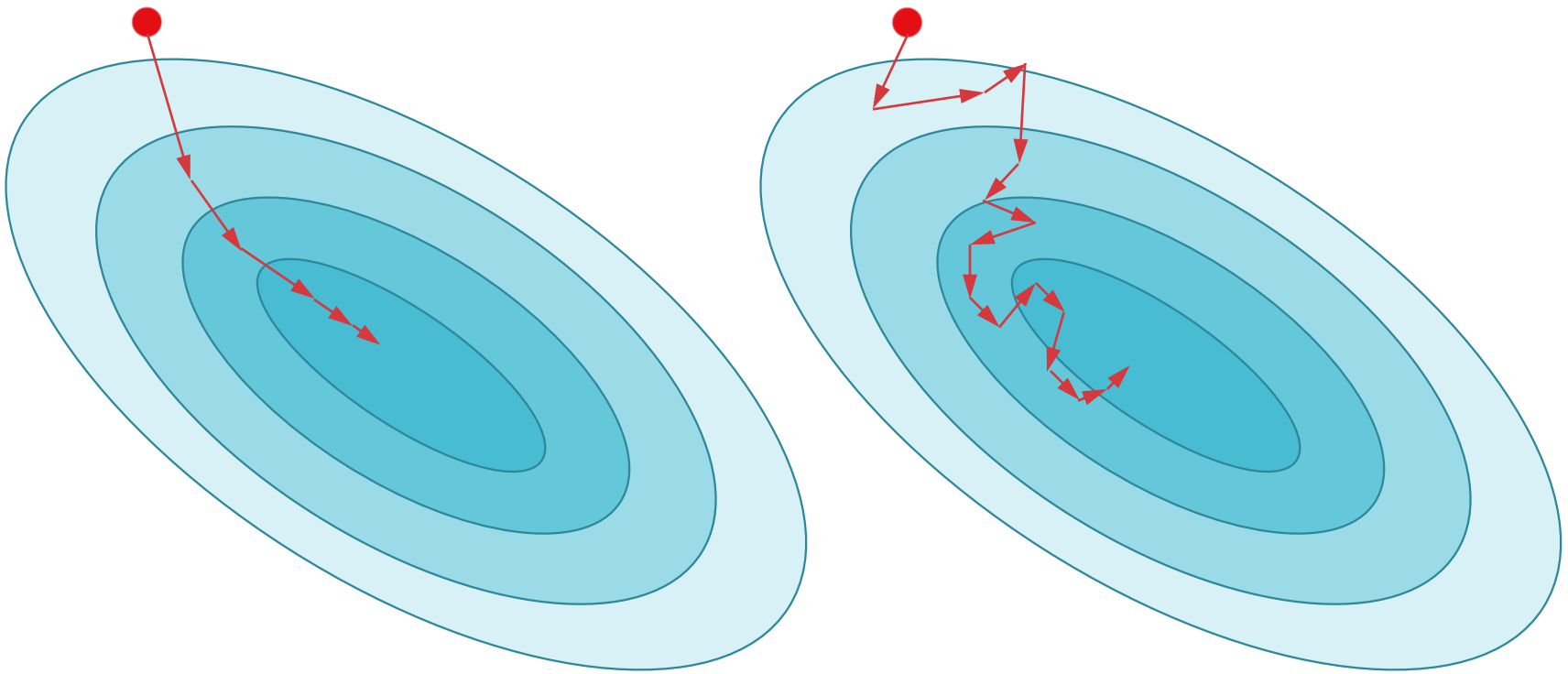
$$E\left[\nabla_{\boldsymbol{\theta}} \ell_{\boldsymbol{x}^{(n)}, y^{(n)}}\left(\boldsymbol{\theta}^{(t)}\right)\right] = \frac{1}{N}\sum_{n=1}^{N} \nabla_{\boldsymbol{\theta}} \ell^{(n)}\left(\boldsymbol{\theta}^{(t)}\right)$$

$$= \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}^{(n)}\left(P\left(Y = 1 \middle| \boldsymbol{x}^{(n)}, \boldsymbol{\theta}^{(t)}\right) - y^{(n)}\right)$$

$$= \frac{1}{N}\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)$$

- In practice, the data set is randomly shuffled then looped through so that each data point is used equally often

Stochastic Gradient Descent vs. Gradient Descent



Gradient Descent

Stochastic Gradient Descent

Mini-batch Stochastic Gradient Descent

- Input: $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}, \eta_{MB}^{(0)}, B$

1. Initialize $\boldsymbol{\theta}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Randomly sample B data points from $\mathcal{D}$:

$$\mathcal{D}_{batch}\left\{\left(\boldsymbol{x}^{(b)}, y^{(b)}\right)\right\}_{b=1}^{B}$$

   b. Compute the gradient w.r.t. the sampled *batch*:

$$\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}_{batch}}\left(\boldsymbol{\theta}^{(t)}\right) = \sum_{b=1}^{B} \boldsymbol{x}^{(b)}\left(P\left(Y = 1 \middle| \boldsymbol{x}^{(b)}, \boldsymbol{\theta}^{(t)}\right) - y^{(b)}\right)$$

   c. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta_{MB}^{(0)} \nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}_{batch}}\left(\boldsymbol{\theta}^{(t)}\right)$
   d. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

Key Takeaways

- Convexity vs. non-convexity
  - Strong vs. weak convexity
  - Implications for local, global and unique optima
- Gradient descent
  - Effect of step size
  - Termination criteria
- Stochastic gradient descent vs. gradient descent