10-301/601: Introduction to Machine Learning Lecture 8 – Optimization for Machine Learning

Henry Chai

5/29/24

Front Matter

- Announcements:
  - HW3 released 5/23, due 6/4 at 11:59 PM
  - No recitation this week
- Recommended Readings:
  - None

Recall: Minimizing the Squared Error

$$\ell_{\mathcal{D}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{\theta}^T\boldsymbol{x}^{(n)} - y^{(n)}\right)^2 = \frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{x}^{(n)^T}\boldsymbol{\theta} - y^{(n)}\right)^2$$

$$= \frac{1}{N}\|X\boldsymbol{\theta} - \boldsymbol{y}\|_2^2 \quad \text{where } \|\boldsymbol{z}\|_2 = \sqrt{\sum_{d=1}^{D} z_d^2} = \sqrt{\boldsymbol{z}^T\boldsymbol{z}}$$

$$= \frac{1}{N}(X\boldsymbol{\theta} - \boldsymbol{y})^T(X\boldsymbol{\theta} - \boldsymbol{y})$$

$$= \frac{1}{N}(\boldsymbol{\theta}^T X^T X\boldsymbol{\theta} - 2\boldsymbol{\theta}^T X^T\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y})$$

$$\nabla_{\boldsymbol{\theta}}\ell_{\mathcal{D}}(\widehat{\boldsymbol{\theta}}) = \frac{1}{N}\left(2X^T X\widehat{\boldsymbol{\theta}} - 2X^T\boldsymbol{y}\right) = 0$$

$$\rightarrow X^T X\widehat{\boldsymbol{\theta}} = X^T\boldsymbol{y}$$

$$\rightarrow \widehat{\boldsymbol{\theta}} = (X^T X)^{-1}X^T\boldsymbol{y}$$

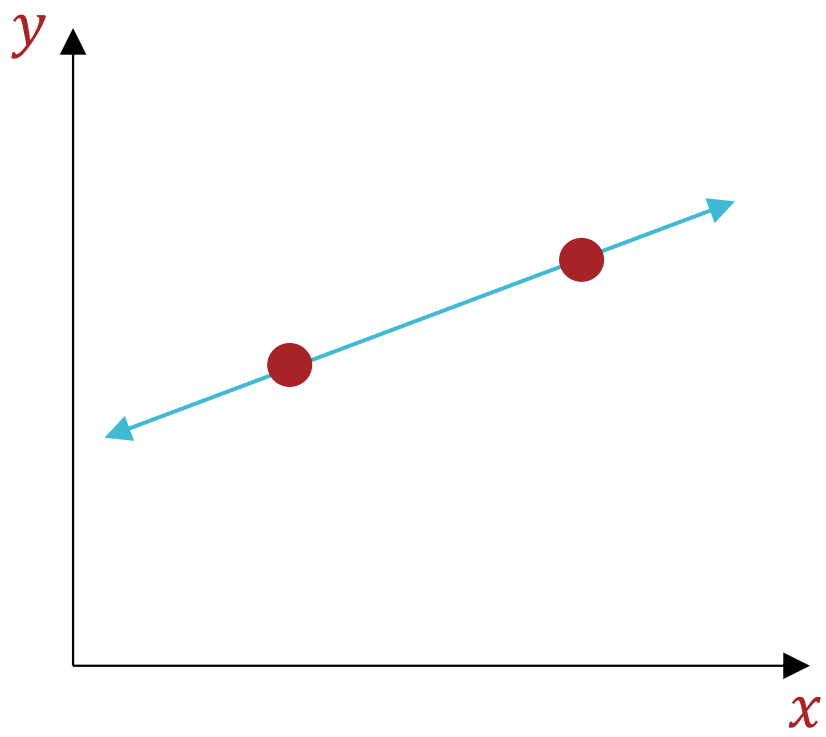
Recall: Closed Form Solution

$$\widehat{\boldsymbol{\theta}} = (X^T X)^{-1} X^T \boldsymbol{y}$$

1. Is $X^T X$ invertible?
   - When $N \gg D + 1$, $X^T X$ is (almost always) full rank and therefore, invertible!
   - If $X^T X$ is not invertible (occurs when one of the features is a linear combination of the others), what does that imply about our problem?
2. If so, how computationally expensive is inverting $X^T X$?
   - $X^T X \in \mathbb{R}^{D+1 \times D+1}$ so inverting $X^T X$ takes $O(D^3)$ time…
     - Computing $X^T X$ takes $O(ND^2)$ time
   - What alternative optimization method(s) can we use to minimize the mean squared error?

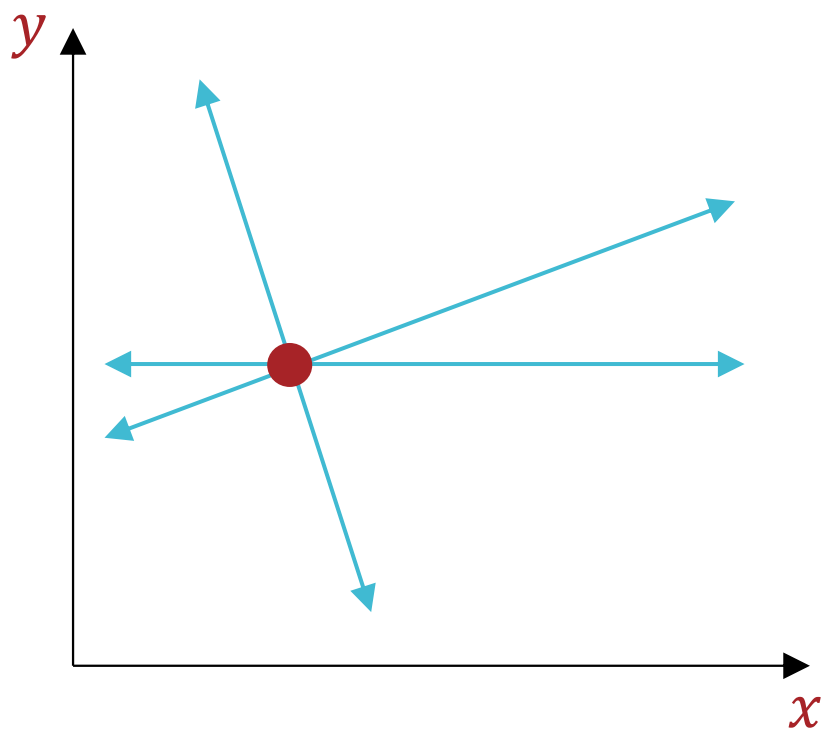
Linear Regression: Uniqueness

- Consider a 1D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of parameters $\boldsymbol{\theta}$) are there for the given dataset?



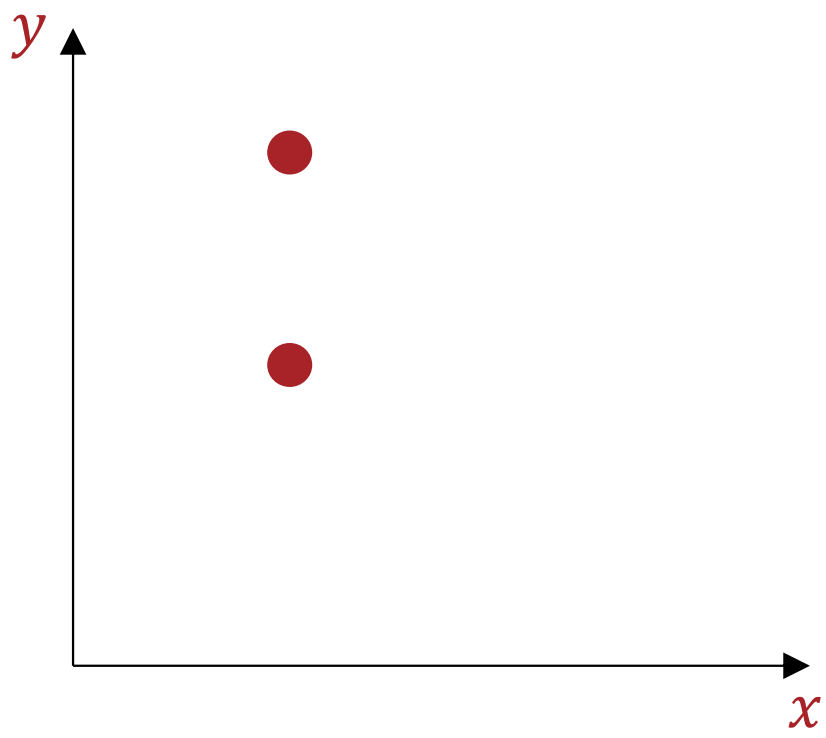
Linear Regression: Uniqueness

- Consider a 1D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of parameters $\boldsymbol{\theta}$) are there for the given dataset?

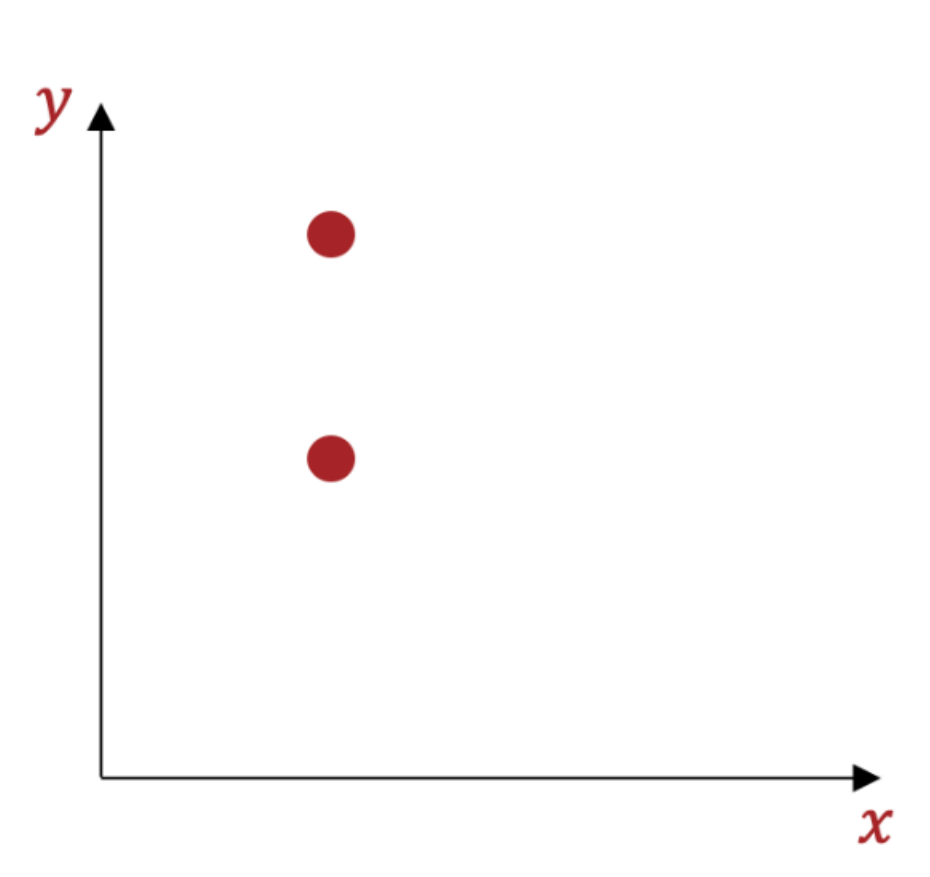


Linear Regression: Uniqueness

- Consider a 1D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of parameters $\boldsymbol{\theta}$) are there for the given dataset?



How many solutions optimal solutions are there for the given dataset?



0 0%

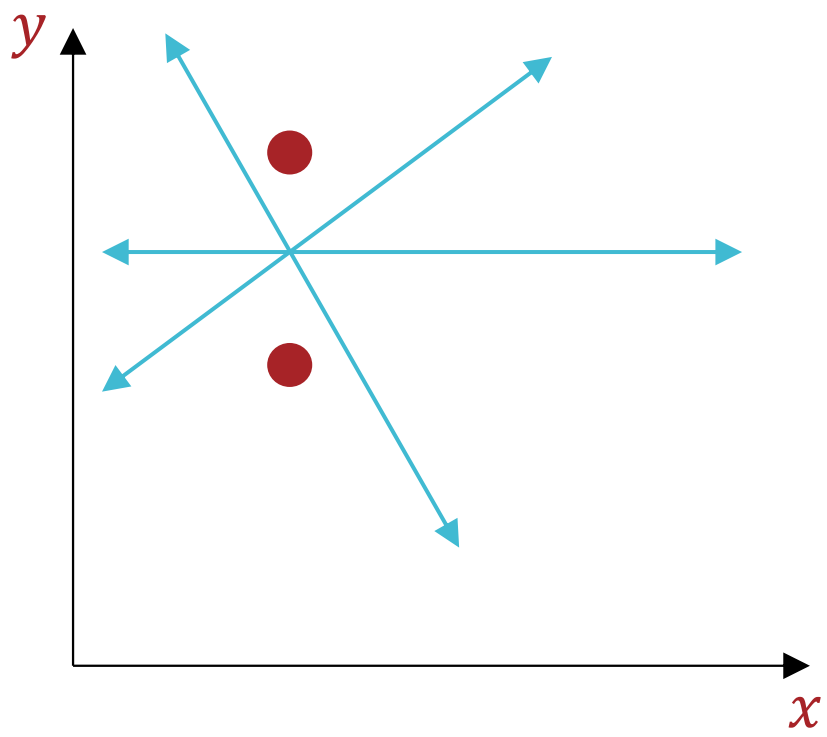
1 0%

2 0%

∞ 0%

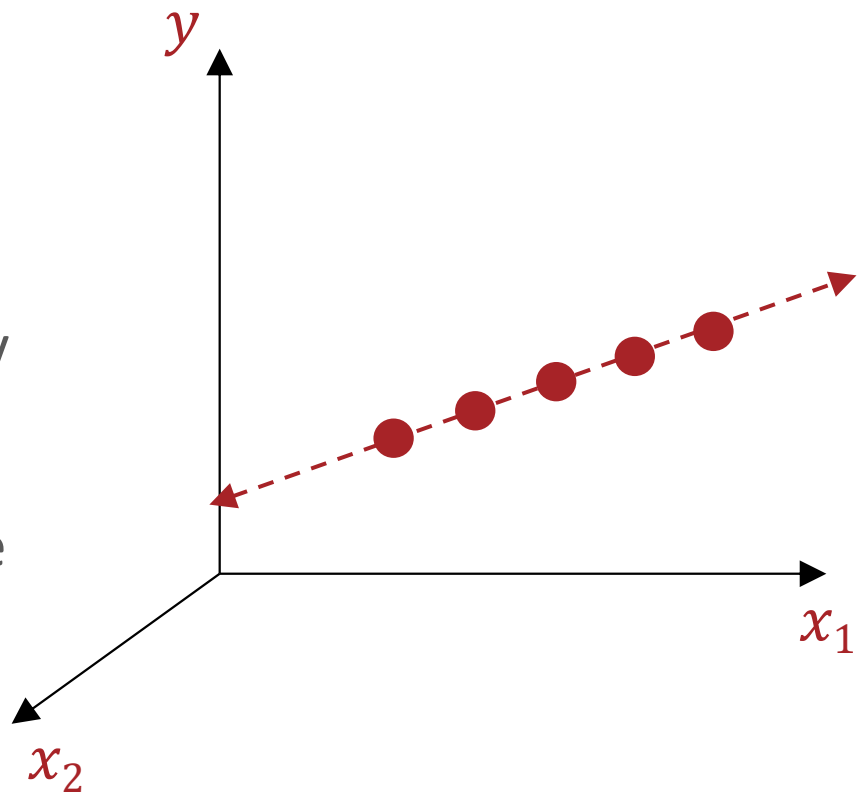
Linear Regression: Uniqueness

- Consider a 1D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of parameters $\boldsymbol{\theta}$) are there for the given dataset?



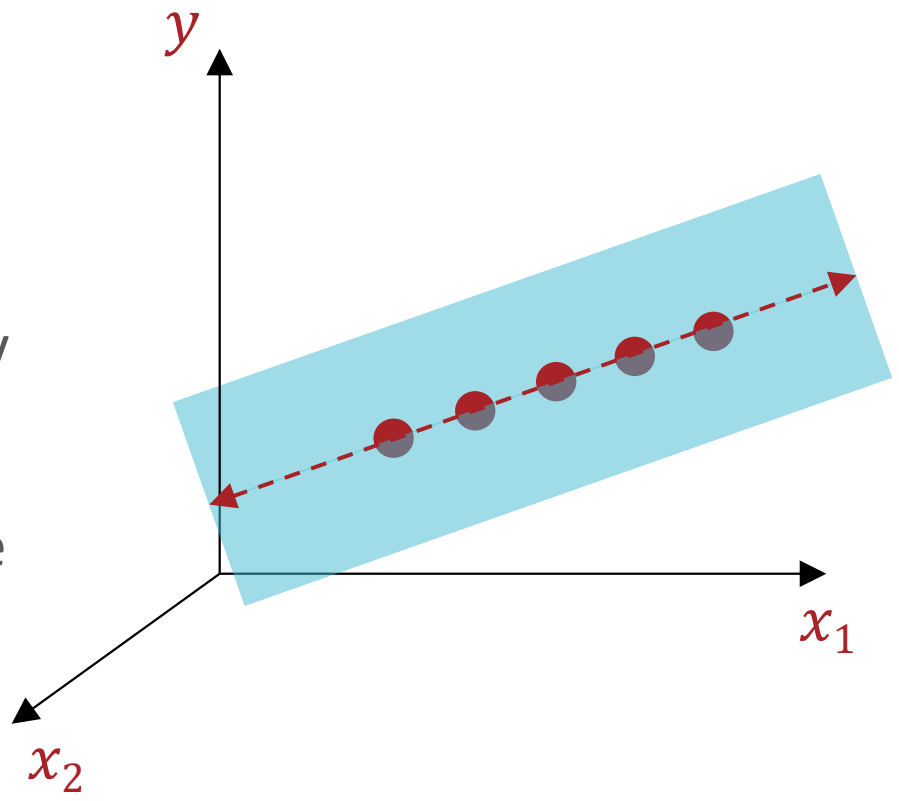
Linear Regression: Uniqueness

- Consider a 2D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of parameters $\boldsymbol{\theta}$) are there for the given dataset?



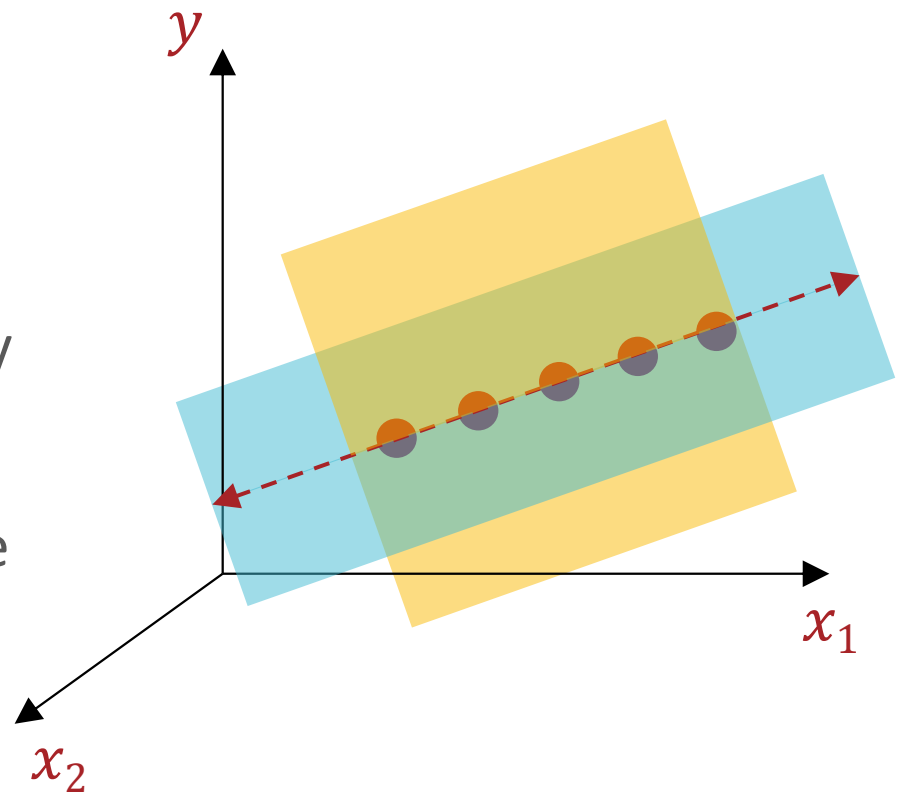
Linear Regression: Uniqueness

- Consider a 2D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of parameters $\boldsymbol{\theta}$) are there for the given dataset?



Linear Regression: Uniqueness

- Consider a 2D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of parameters $\boldsymbol{\theta}$) are there for the given dataset?



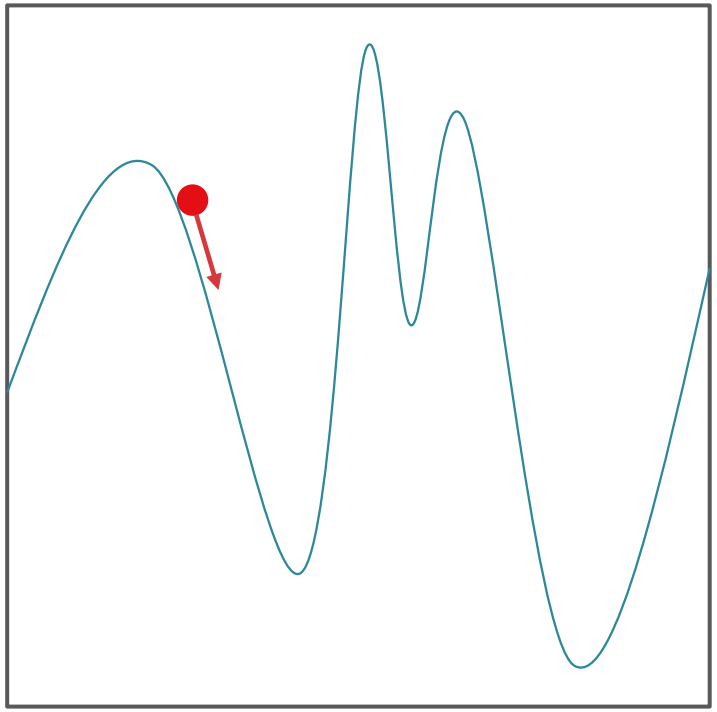
Closed Form Solution

$$\widehat{\boldsymbol{\theta}} = (X^T X)^{-1} X^T \boldsymbol{y}$$

1. Is $X^T X$ invertible?
   - When $N \gg D + 1$, $X^T X$ is (almost always) full rank and therefore, invertible!
   - If $X^T X$ is not invertible (occurs when one of the features is a linear combination of the others) then there are infinitely many solutions.
2. If so, how computationally expensive is inverting $X^T X$?
   - $X^T X \in \mathbb{R}^{D+1 \times D+1}$ so inverting $X^T X$ takes $O(D^3)$ time…
     - Computing $X^T X$ takes $O(ND^2)$ time
   - Can use gradient descent to (potentially) speed things up when N and D are large!

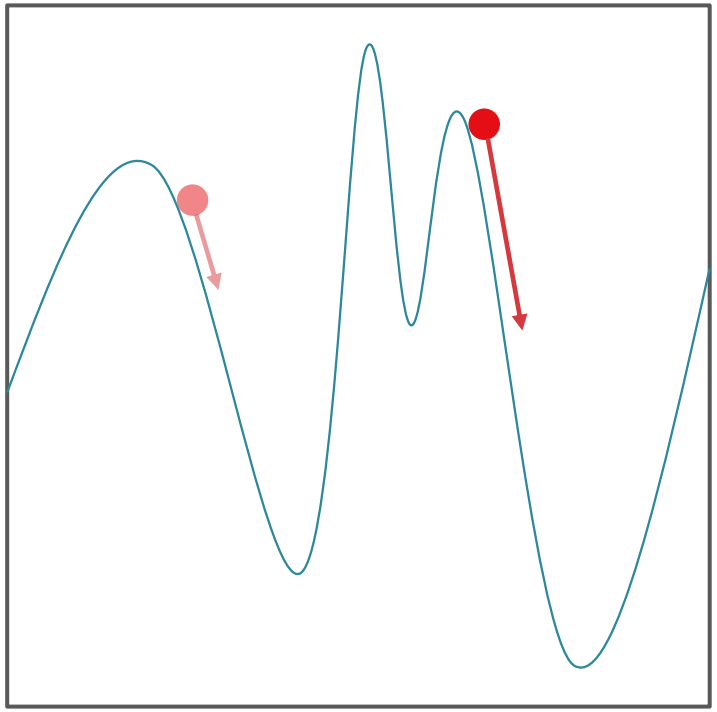
Gradient Descent: Intuition

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



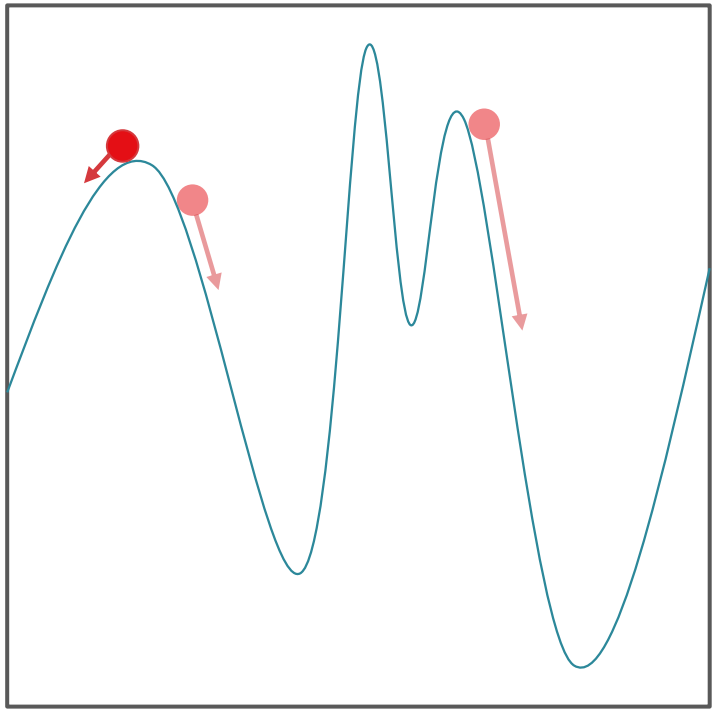
Gradient Descent: Intuition

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



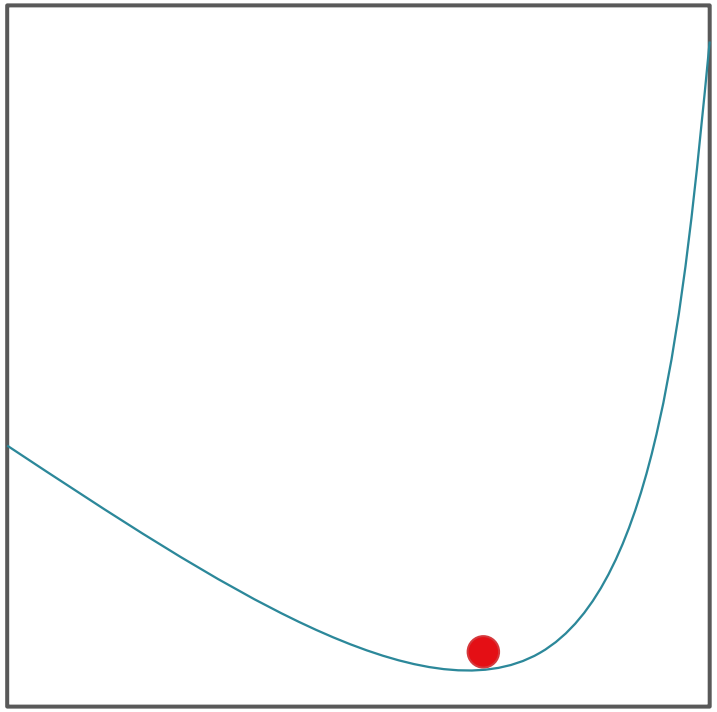
Gradient Descent: Intuition

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



Gradient Descent: Intuition

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



- Good news: the squared error is convex!

Minimizing the Squared Error

$$\ell_{\mathcal{D}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{\theta}^T\boldsymbol{x}^{(n)} - y^{(n)}\right)^2 = \frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{x}^{(n)^T}\boldsymbol{\theta} - y^{(n)}\right)^2$$

$$= \frac{1}{N}\|X\boldsymbol{\theta} - \boldsymbol{y}\|_2^2 \quad \text{where } \|\boldsymbol{z}\|_2 = \sqrt{\sum_{d=1}^{D} z_d^2} = \sqrt{\boldsymbol{z}^T\boldsymbol{z}}$$

$$= \frac{1}{N}(X\boldsymbol{\theta} - \boldsymbol{y})^T(X\boldsymbol{\theta} - \boldsymbol{y})$$

$$= \frac{1}{N}(\boldsymbol{\theta}^T X^T X\boldsymbol{\theta} - 2\boldsymbol{\theta}^T X^T\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y})$$

$$\nabla_{\boldsymbol{\theta}}\ell_{\mathcal{D}}(\boldsymbol{\theta}) = (2X^T X\boldsymbol{\theta} - 2X^T\boldsymbol{y})$$

$$H_{\boldsymbol{\theta}}\ell_{\mathcal{D}}(\boldsymbol{\theta}) = 2X^T X$$

$H_{\boldsymbol{\theta}}\ell_{\mathcal{D}}(\boldsymbol{\theta})$ is positive semi-definite

Gradient Descent: Step Direction

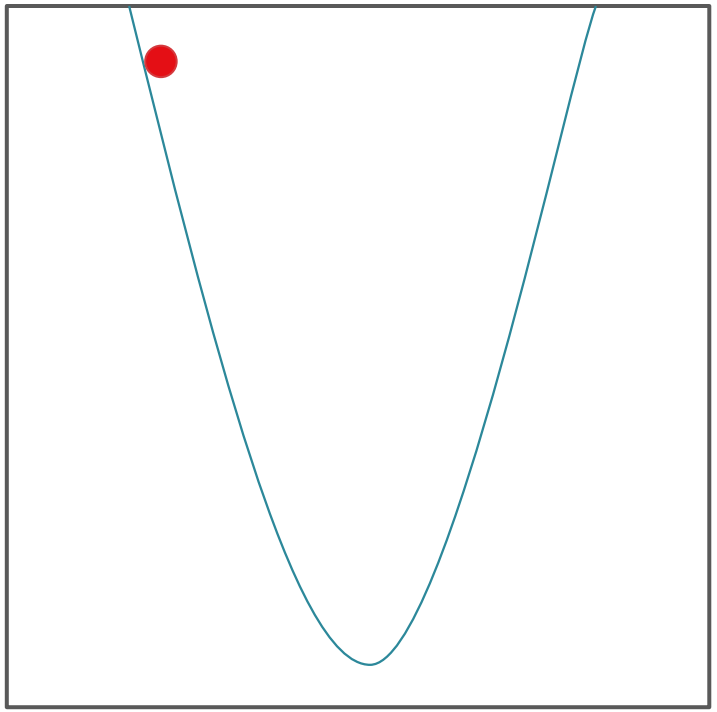
- Suppose the current parameter vector is $\boldsymbol{\theta}^{(t)}$
- Move some distance, η, in the "most downhill" direction, $\widehat{\boldsymbol{v}}$:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \eta\widehat{\boldsymbol{v}}$$

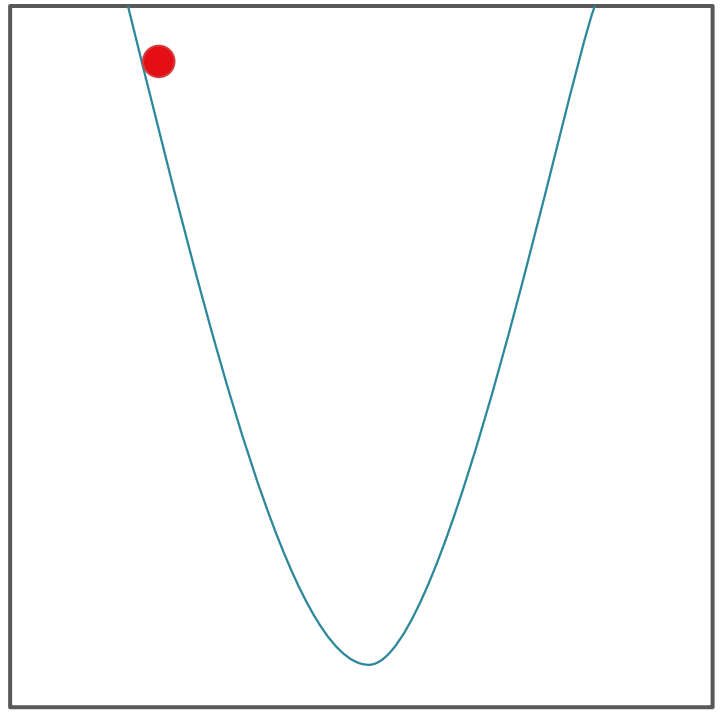
- The gradient points in the direction of steepest *increase* …
- … so $\widehat{\boldsymbol{v}}$ is a unit vector pointing in the opposite direction:

$$\widehat{\boldsymbol{v}}^{(t)} = -\frac{\nabla_{\boldsymbol{\theta}}\ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)}{\left\|\nabla_{\boldsymbol{\theta}}\ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)\right\|}$$

Gradient Descent: Step Size

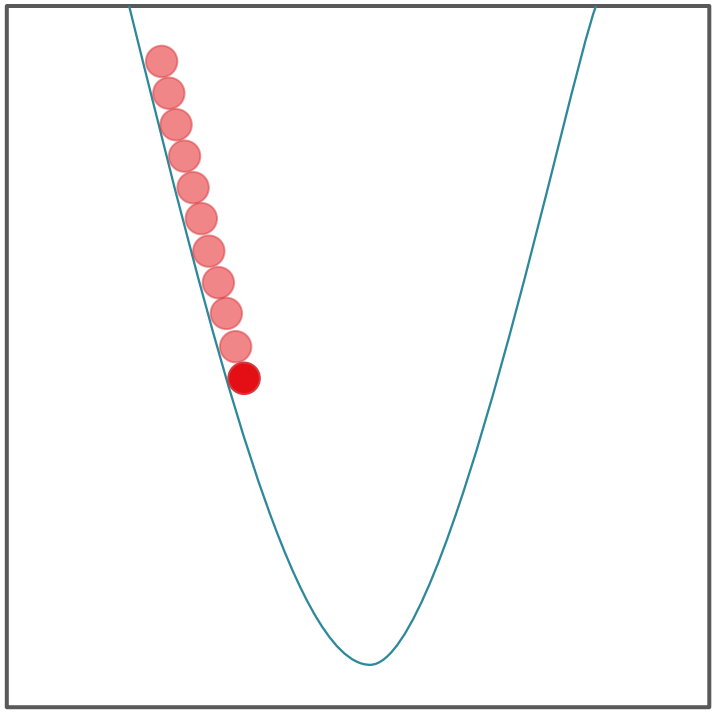


Small η

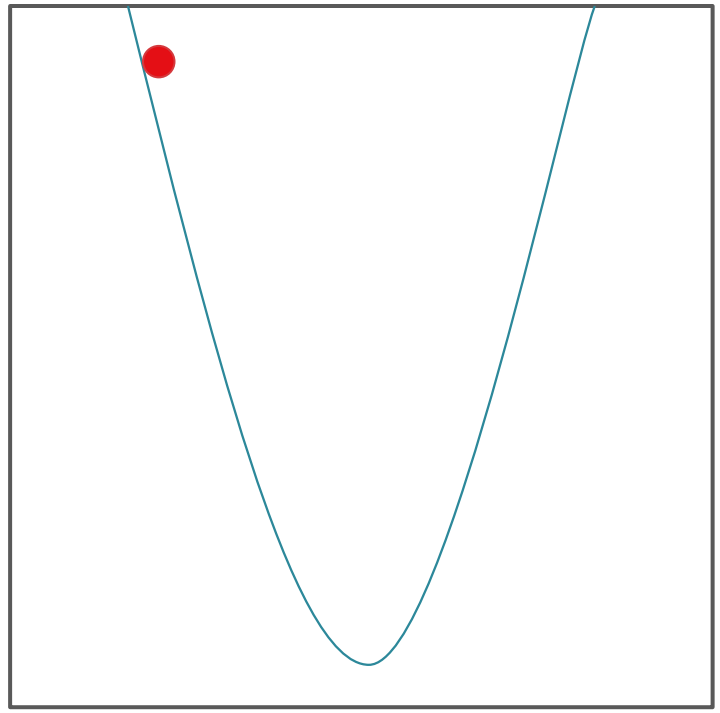


Large η

Gradient Descent: Step Size

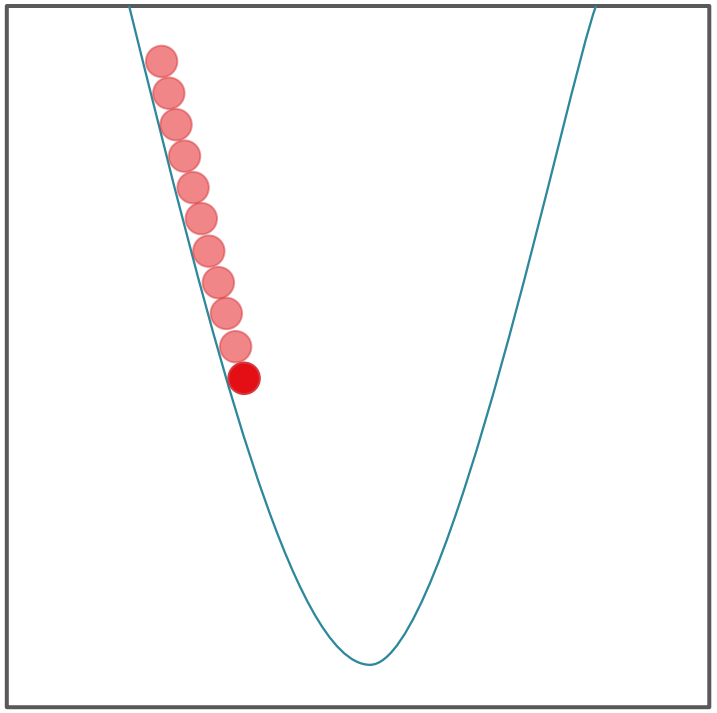


Small η

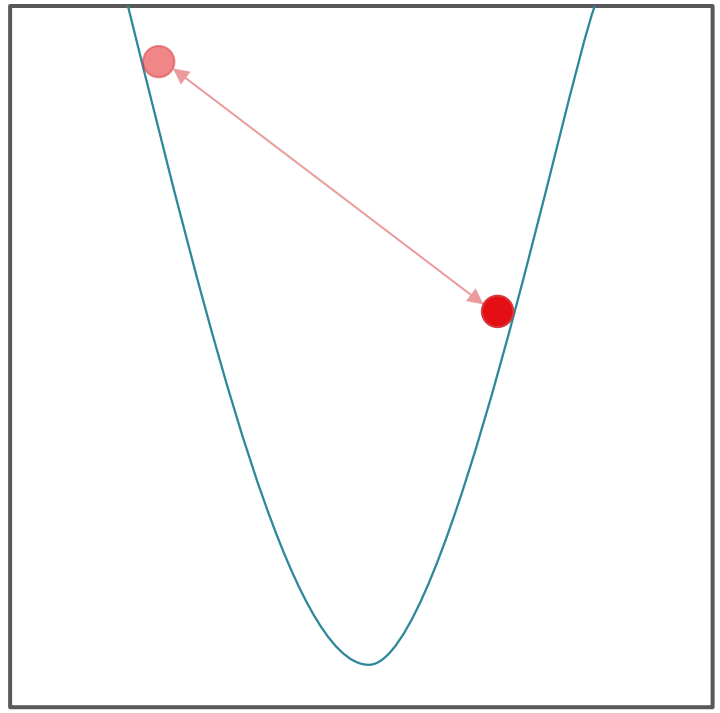


Large η

Gradient Descent: Step Size



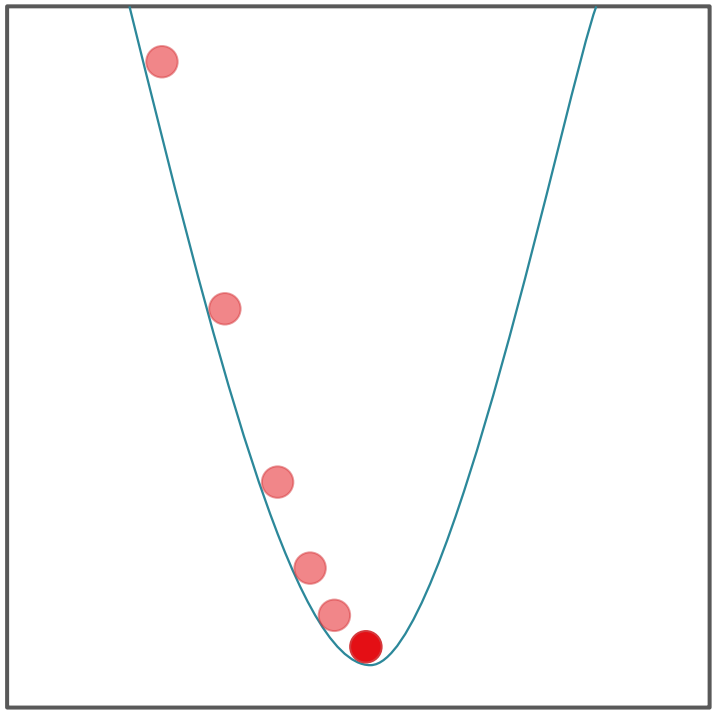
Small η



Large η

Gradient Descent: Step Size

- Use a variable $\eta^{(t)}$ instead of a fixed η!



- Set $\eta^{(t)} = \eta^{(0)} \left\| \nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}} \left(\boldsymbol{\theta}^{(t)} \right) \right\|$
- $\left\| \nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}} \left(\boldsymbol{\theta}^{(t)} \right) \right\|$ decreases as $\ell_{\mathcal{D}}$ approaches its minimum $\rightarrow \eta^{(t)}$ (hopefully) decreases over time

Gradient Descent

- $\widehat{\boldsymbol{v}}^{(t)} = -\frac{\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)}{\left\|\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)\right\|}$

- $\eta^{(t)} = \eta^{(0)}\left\|\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)\right\|$

- $$\begin{aligned}\boldsymbol{\theta}^{(t+1)} &= \boldsymbol{\theta}^{(t)} + \eta^{(t)}\widehat{\boldsymbol{v}}^{(t)} \\ &= \boldsymbol{\theta}^{(t)} + \left(\eta^{(0)}\left\|\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)\right\|\right)\left(-\frac{\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)}{\left\|\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)\right\|}\right) \\ &= \boldsymbol{\theta}^{(t)} - \eta^{(0)}\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)\end{aligned}$$

Gradient Descent

- 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:
      $\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}\left(\boldsymbol{\theta}^{(t)}\right)$
   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 $\boldsymbol{\theta}^{(0)}$ to all zeros 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

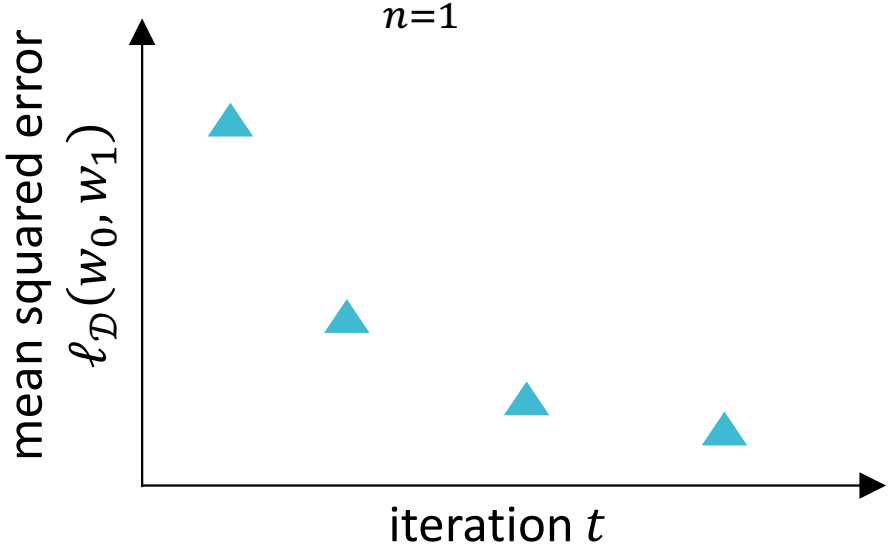
- Input: $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}, \eta^{(0)}$

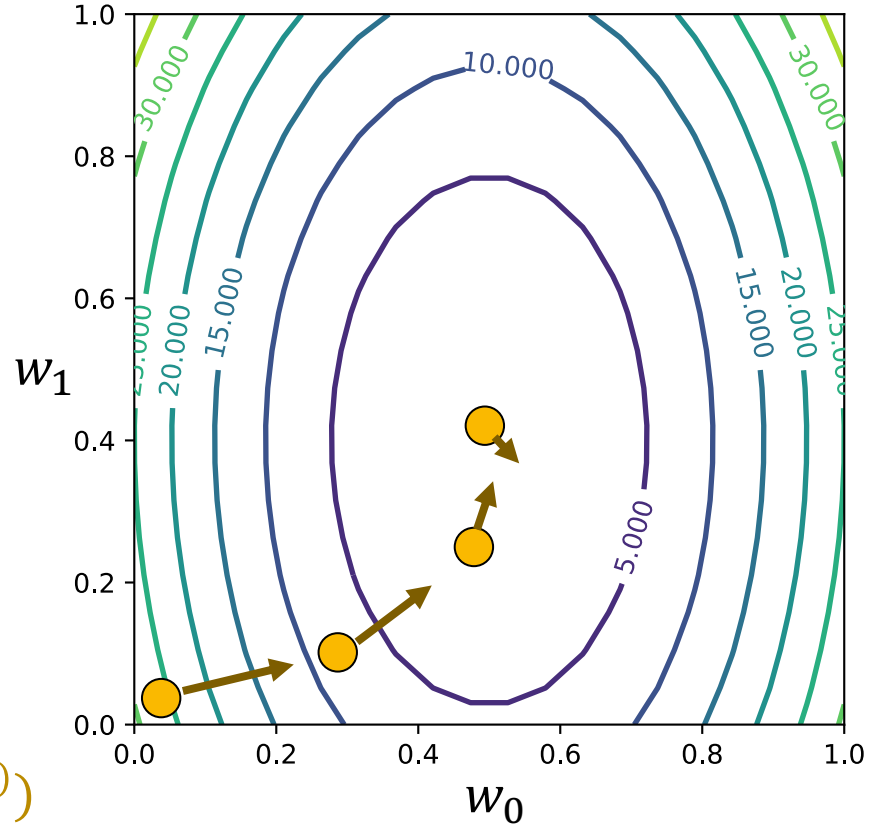
1. Initialize $\boldsymbol{\theta}^{(0)}$ to all zeros and set $t = 0$
2. While $t < T$
   a. Compute the gradient:
      $\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}} \left(\boldsymbol{\theta}^{(t)}\right)$
   b. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)}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

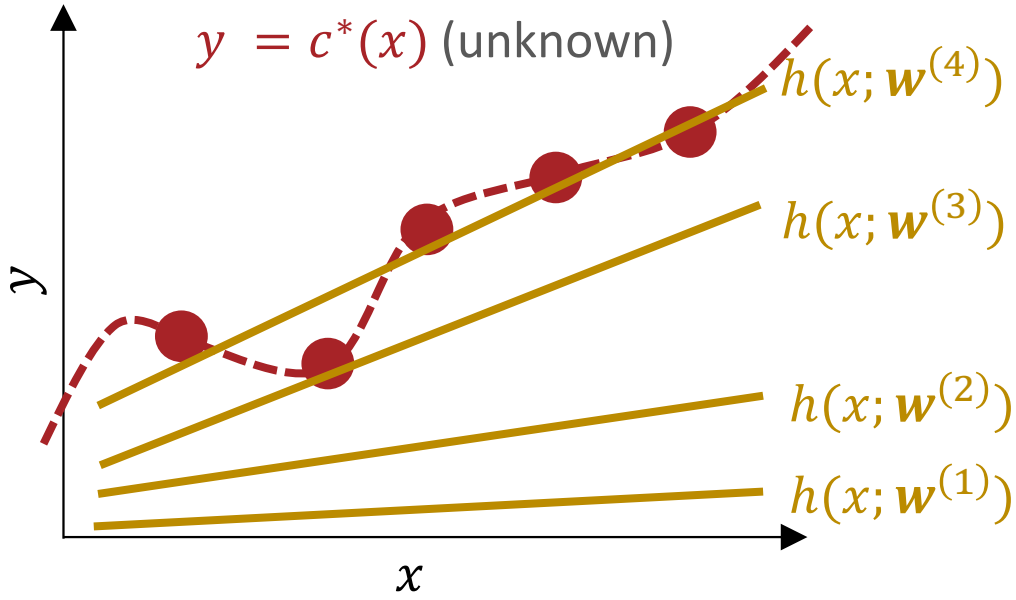
- Output: $\boldsymbol{\theta}^{(t)}$

Gradient Descent for Linear Regression

$$\ell_{\mathcal{D}}(w_0, w_1) = \frac{1}{N}\sum_{n=1}^{N}\left(w_1 x^{(n)} + w_0 - y^{(n)}\right)^2$$



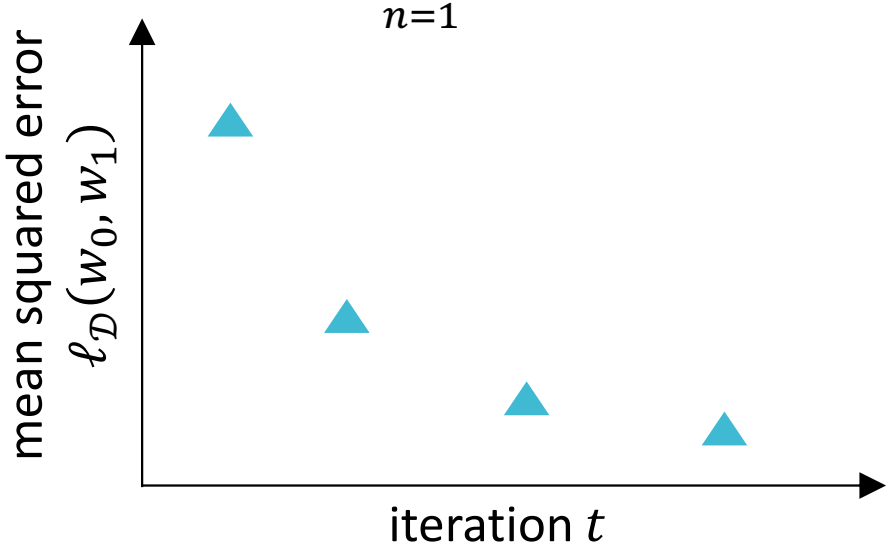


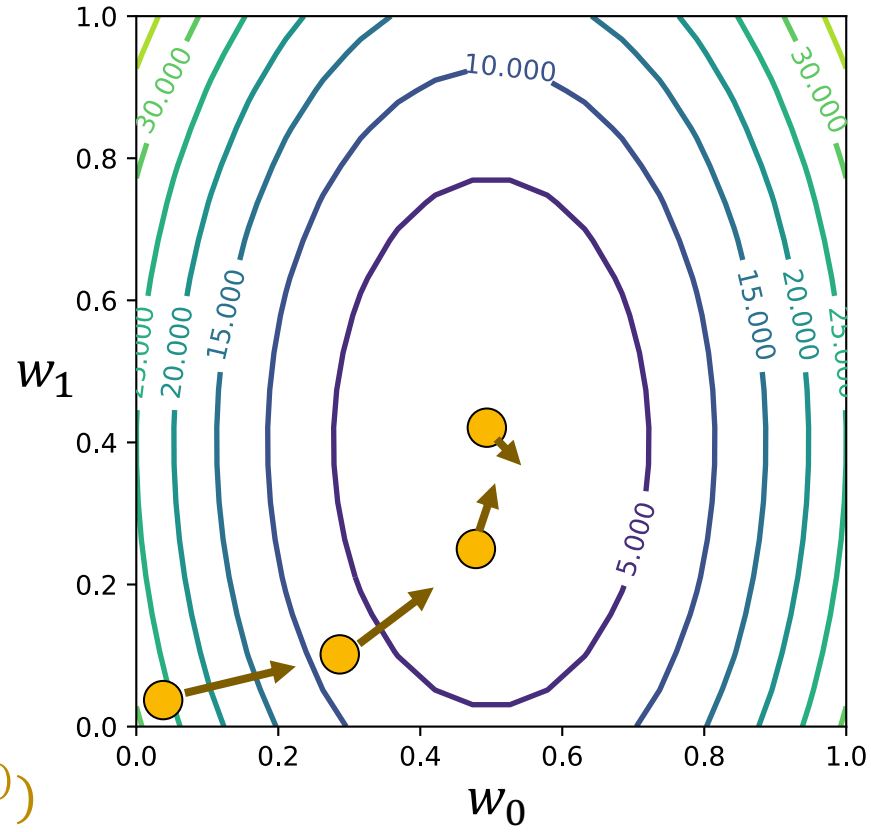


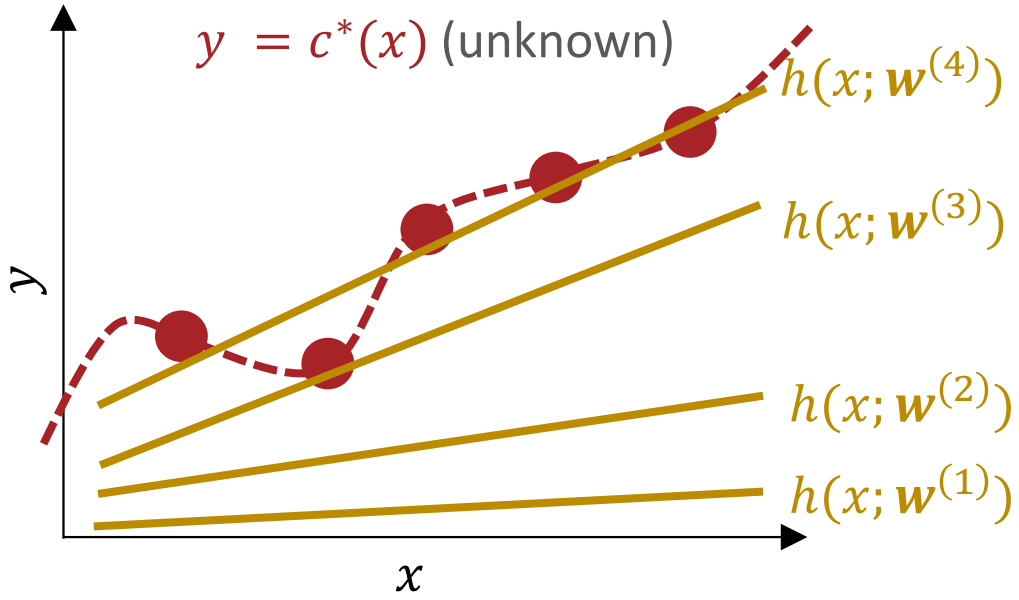
t	w_0	w_1	$\ell_{\mathcal{D}}(w_0, w_1)$

Why Gradient Descent for Linear Regression?

$$\ell_{\mathcal{D}}(w_0, w_1) = \frac{1}{N}\sum_{n=1}^{N}\left(w_1 x^{(n)} + w_0 - y^{(n)}\right)^2$$







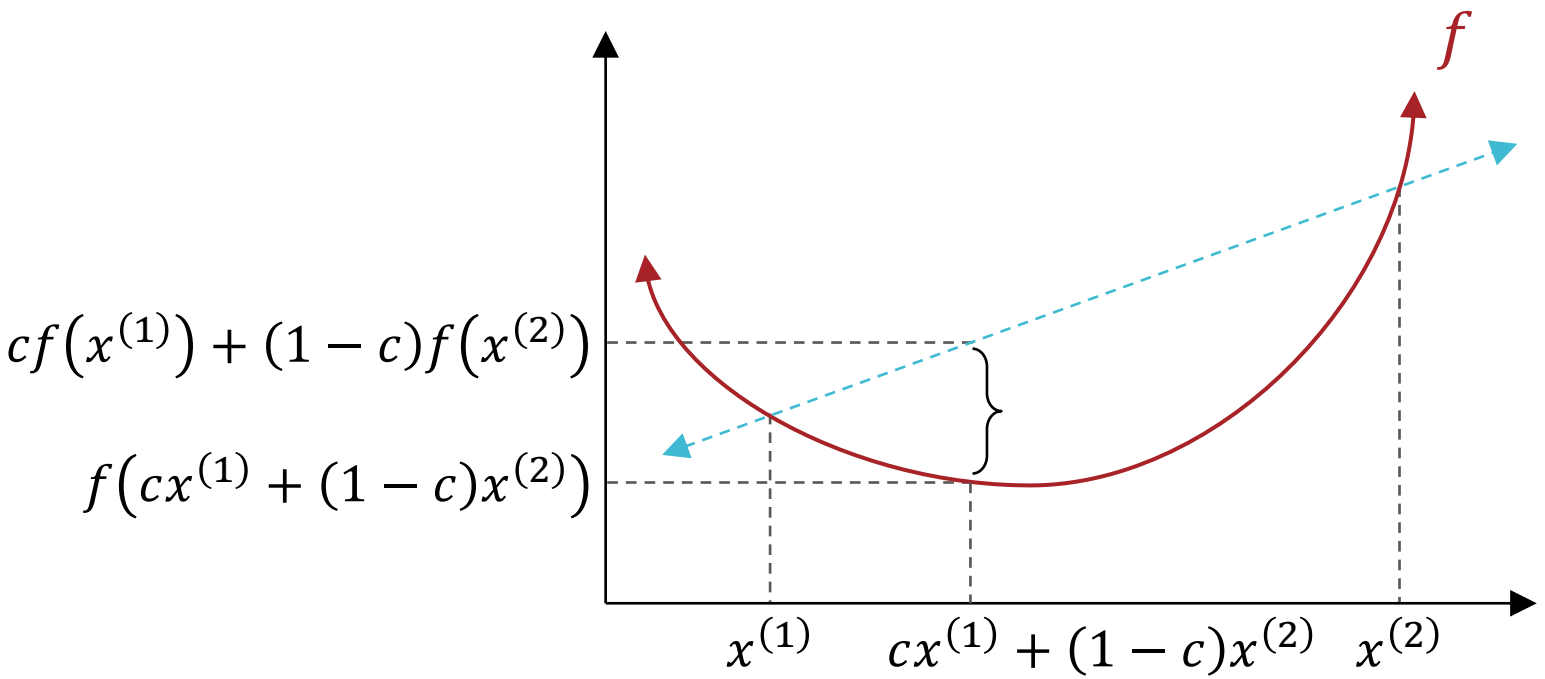
t	w_0	w_1	$\ell_{\mathcal{D}}(w_0, w_1)$
1	0.01	0.02	25.2
2	0.30	0.12	8.7
3	0.51	0.30	1.5
4	0.59	0.43	0.2

Convexity

- A function $f: \mathbb{R}^D \rightarrow \mathbb{R}$ is convex if

  $\forall\, \boldsymbol{x}^{(1)} \in \mathbb{R}^D, \boldsymbol{x}^{(2)} \in \mathbb{R}^D$ and $0 \leq c \leq 1$

$$f\left(c\boldsymbol{x}^{(1)} + (1-c)\boldsymbol{x}^{(2)}\right) \leq cf\left(\boldsymbol{x}^{(1)}\right) + (1-c)f\left(\boldsymbol{x}^{(2)}\right)$$

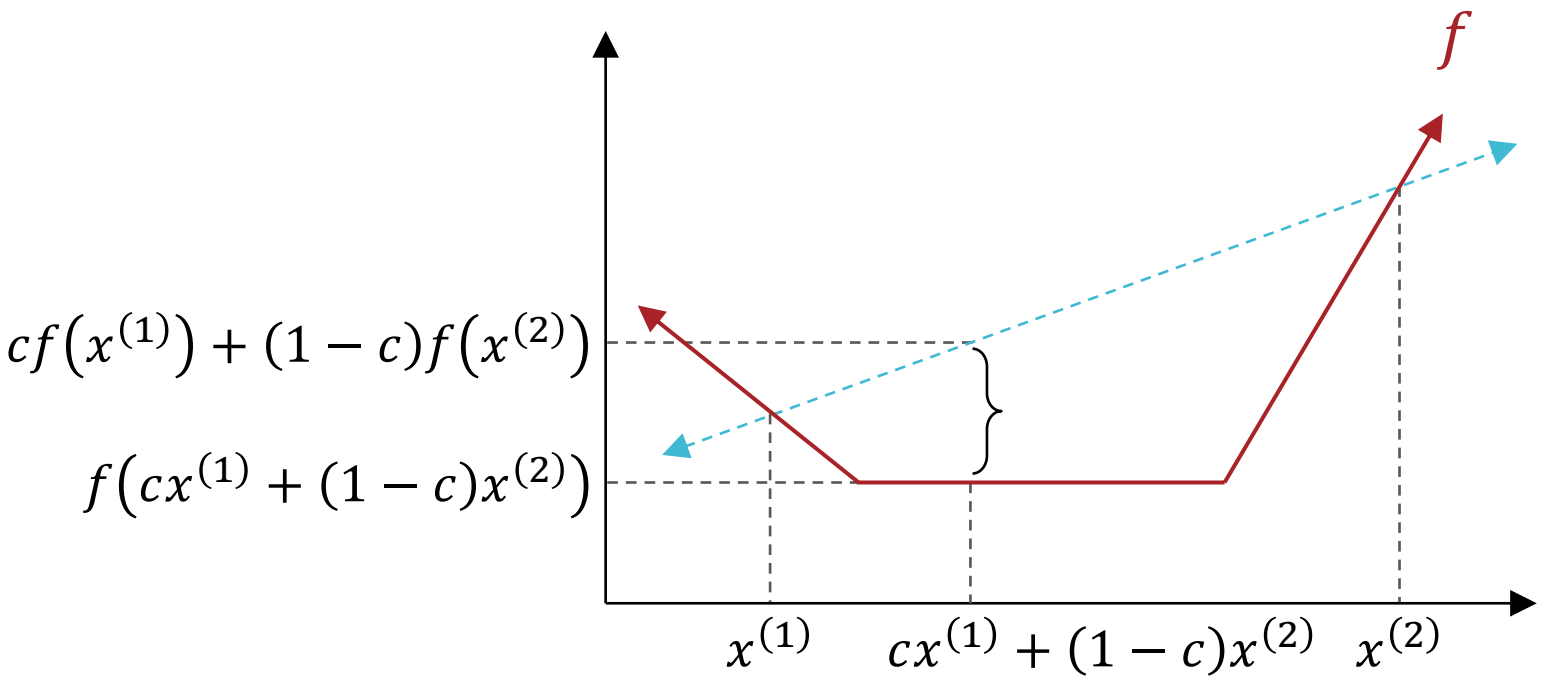


Convexity

- A function $f: \mathbb{R}^D \rightarrow \mathbb{R}$ is convex if

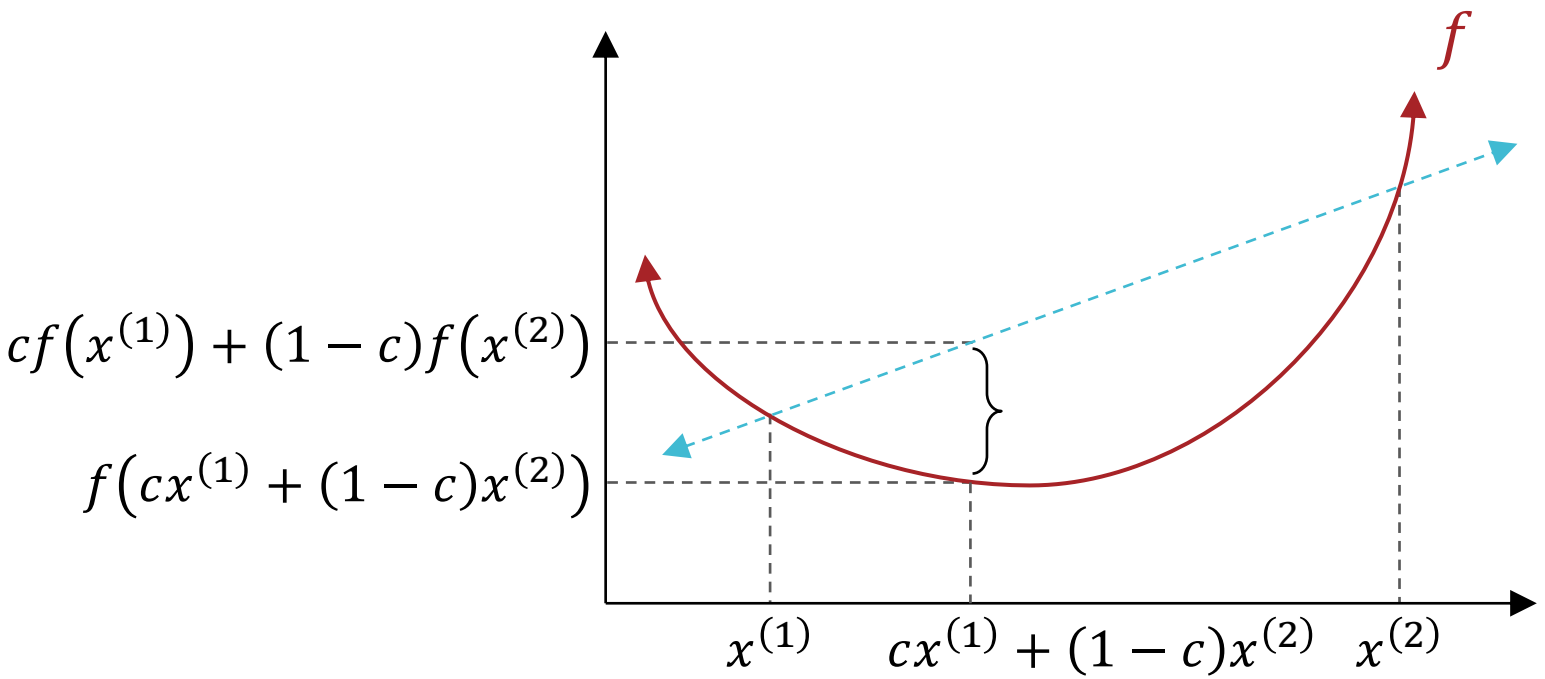
  $\forall\, \boldsymbol{x}^{(1)} \in \mathbb{R}^D, \boldsymbol{x}^{(2)} \in \mathbb{R}^D$ and $0 \leq c \leq 1$

$$f\left(c\boldsymbol{x}^{(1)} + (1-c)\boldsymbol{x}^{(2)}\right) \leq cf\left(\boldsymbol{x}^{(1)}\right) + (1-c)f\left(\boldsymbol{x}^{(2)}\right)$$

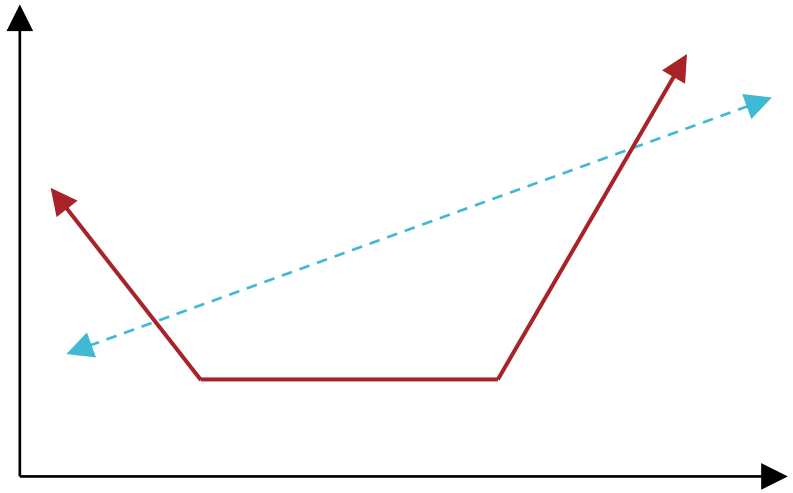


Convexity

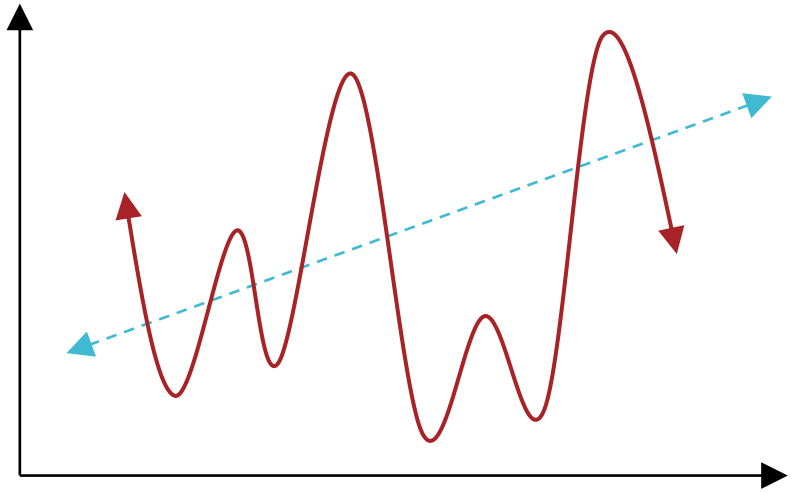
- A function $f: \mathbb{R}^D \rightarrow \mathbb{R}$ is *strictly* convex if
  $\forall\, \boldsymbol{x}^{(1)} \in \mathbb{R}^D, \boldsymbol{x}^{(2)} \in \mathbb{R}^D$ and $0 < c < 1$
  $$f\left(c\boldsymbol{x}^{(1)} + (1-c)\boldsymbol{x}^{(2)}\right) < cf\left(\boldsymbol{x}^{(1)}\right) + (1-c)f\left(\boldsymbol{x}^{(2)}\right)$$



Convexity

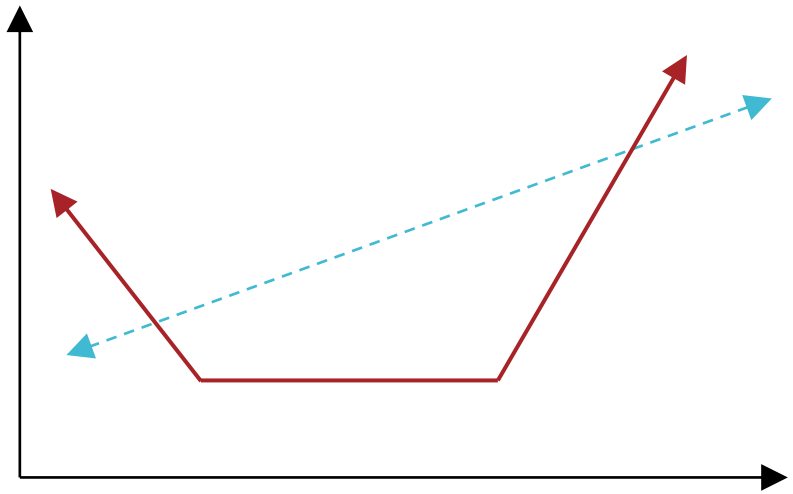


Convex functions



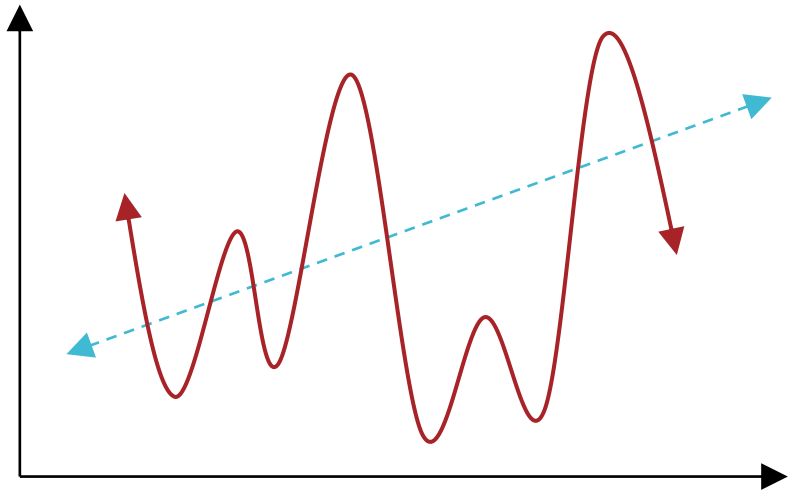
Non-convex functions

Convexity



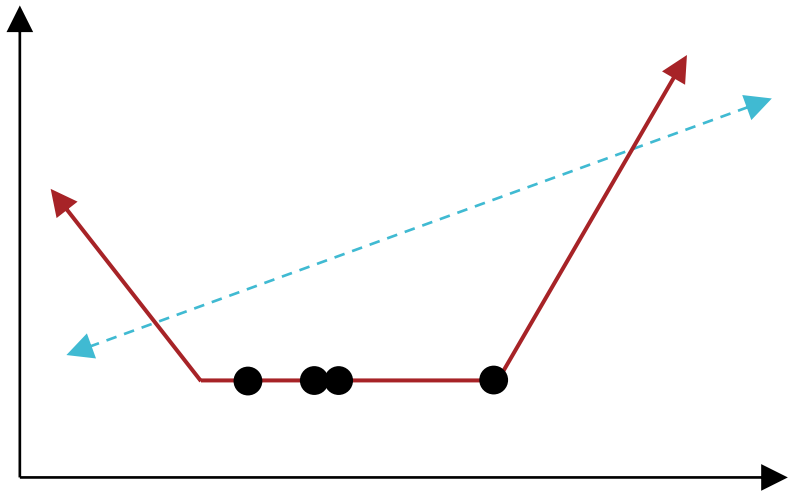
Given a function $f: \mathbb{R}^D \rightarrow \mathbb{R}$

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



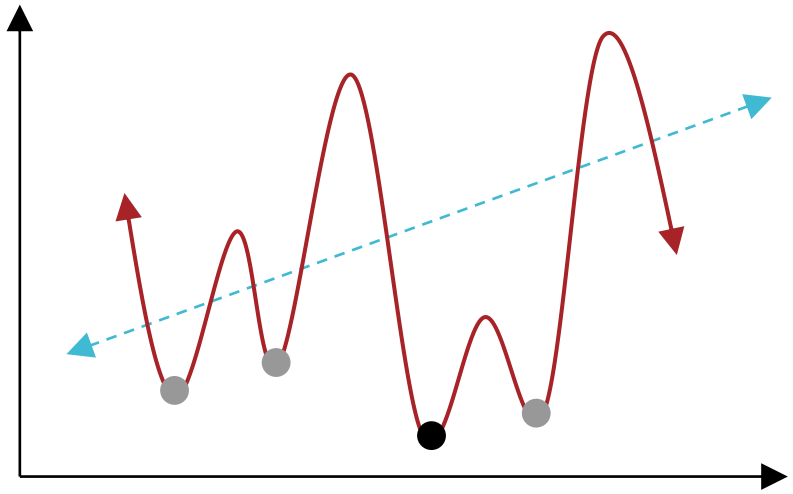
- $\boldsymbol{x}^*$ is a local minimum iff $\exists \; \epsilon$ s.t. $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) \; \forall \; \boldsymbol{x}$ s.t. $\|\boldsymbol{x} - \boldsymbol{x}^*\|_2 < \epsilon$

Convexity



Convex functions:

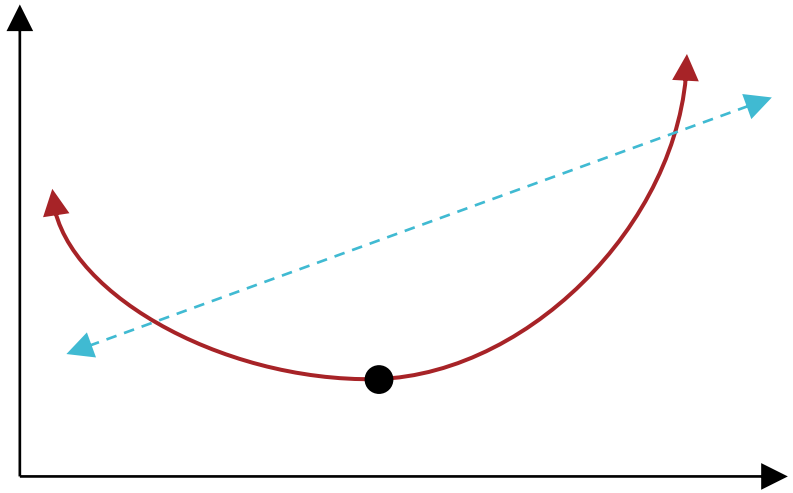
Each local minimum is a global minimum!



Non-convex functions:

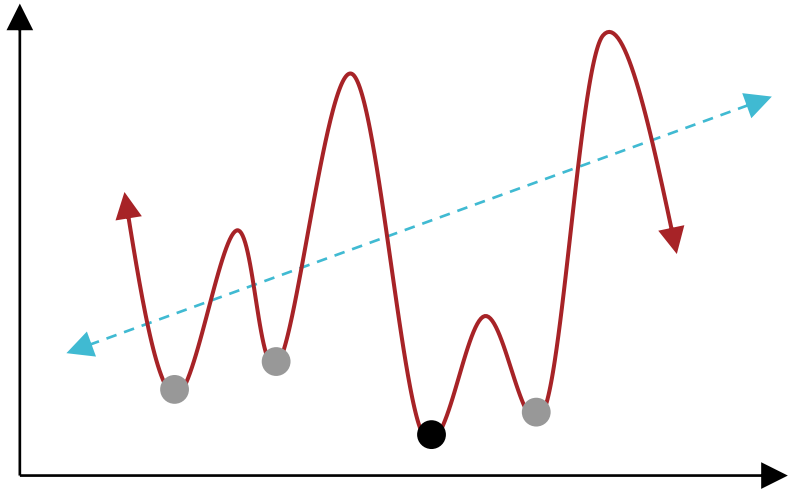
A local minimum may or may not be a global minimum…

Convexity



Strictly convex functions:

There exists a unique global minimum!

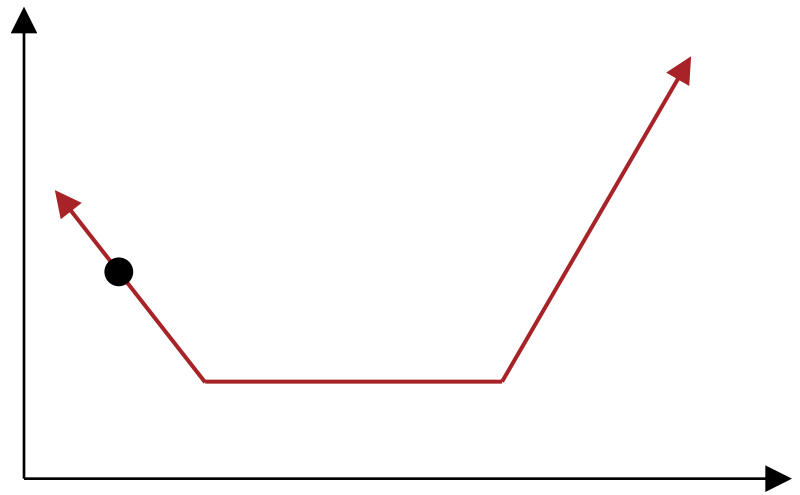


Non-convex functions:

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

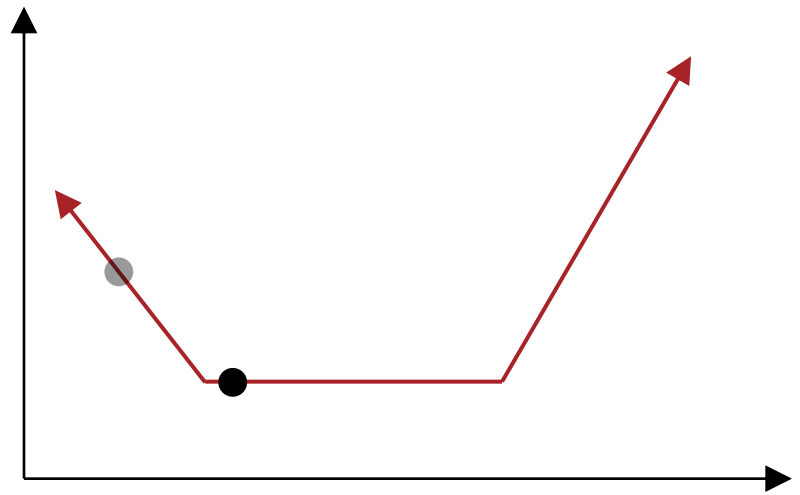
Gradient Descent & Convexity

- 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!



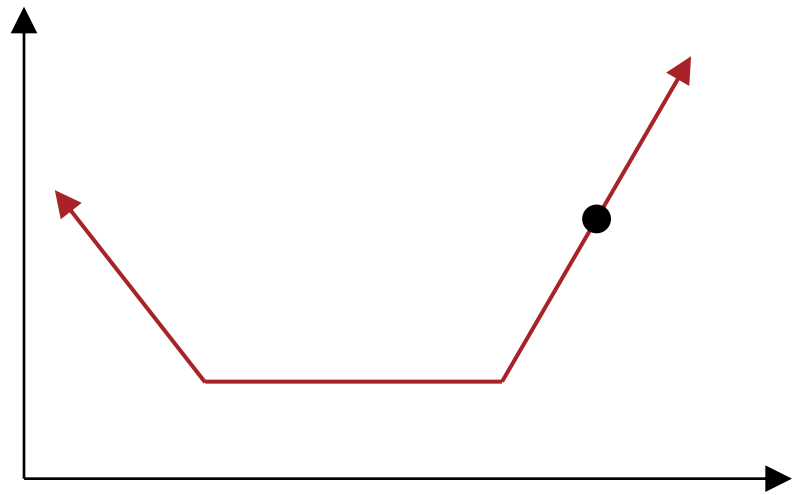
Gradient Descent & Convexity

- 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!



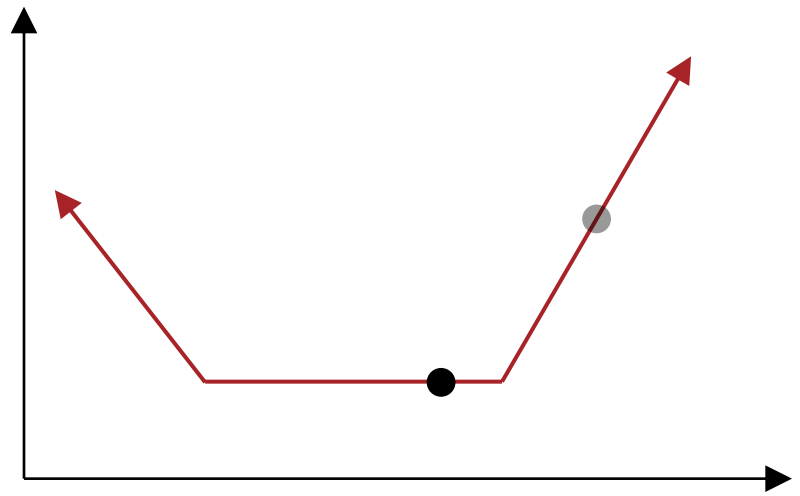
Gradient Descent & Convexity

- 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!



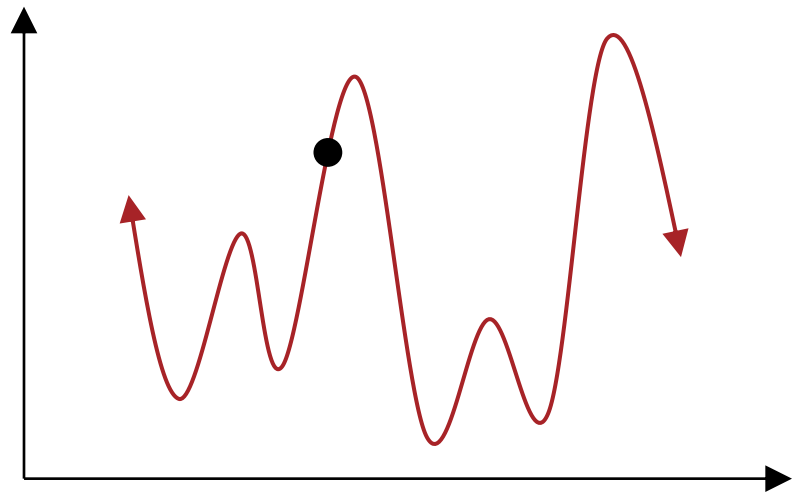
Gradient Descent & Convexity

- 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!



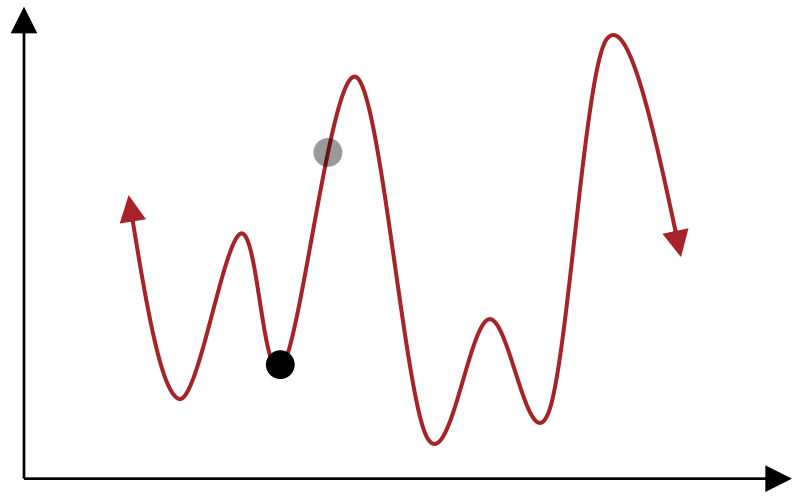
Gradient Descent & Convexity

- 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…



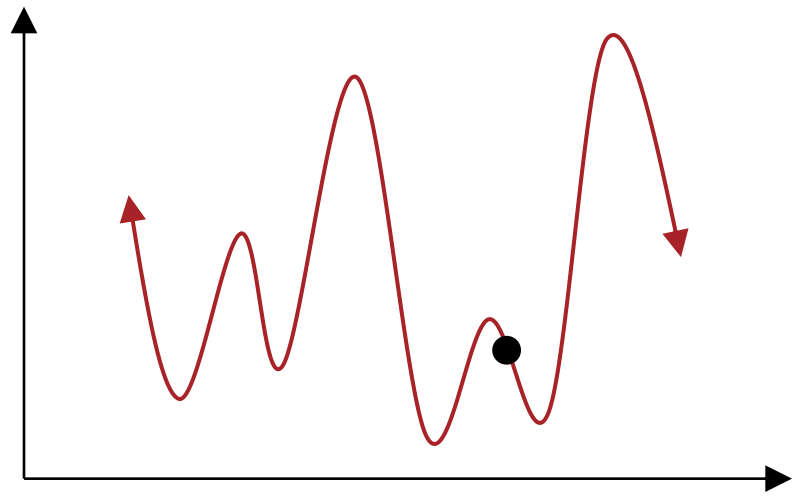
Gradient Descent & Convexity

- 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…



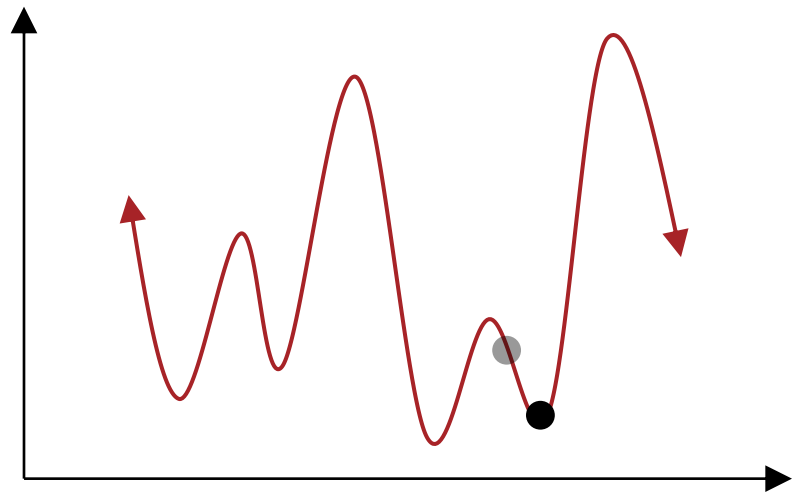
Gradient Descent & Convexity

- 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…



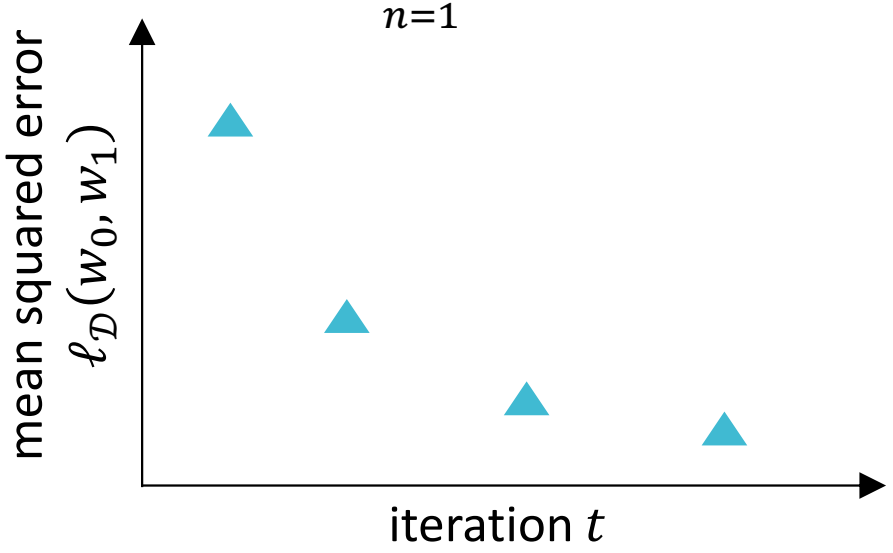
Gradient Descent & Convexity

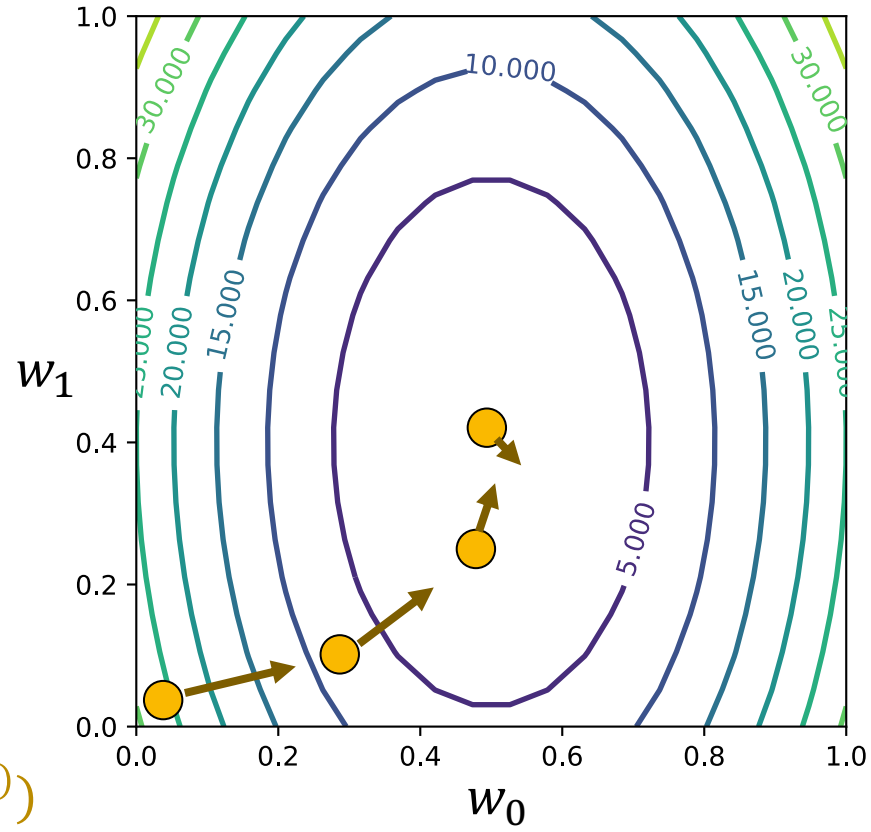
- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Not ideal if the objective function is non-convex…

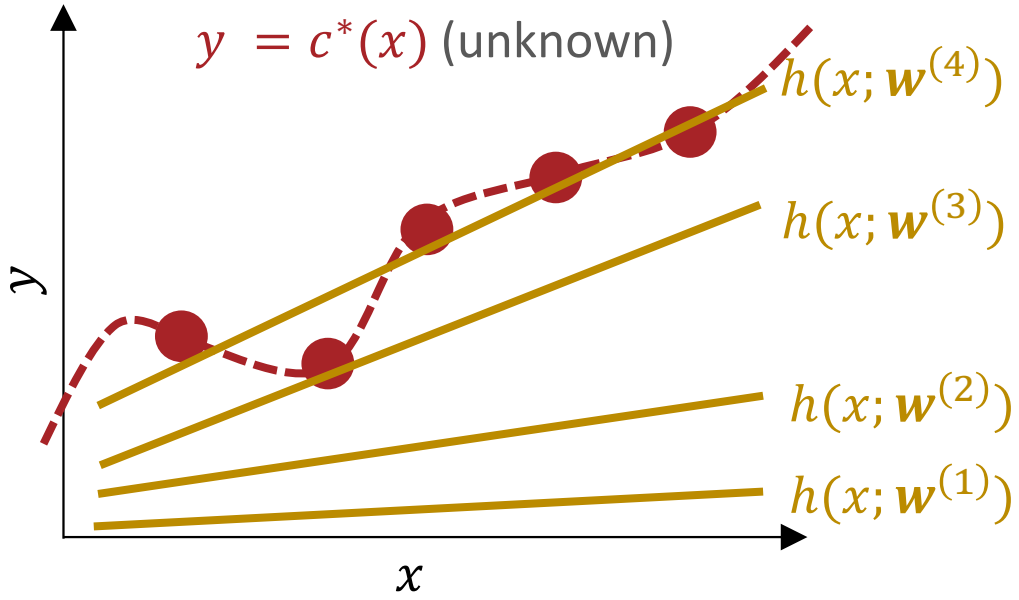


The mean squared error is convex (but not always strictly convex)

$$\ell_{\mathcal{D}}(w_0, w_1) = \frac{1}{N}\sum_{n=1}^{N}\left(w_1 x^{(n)} + w_0 - y^{(n)}\right)^2$$



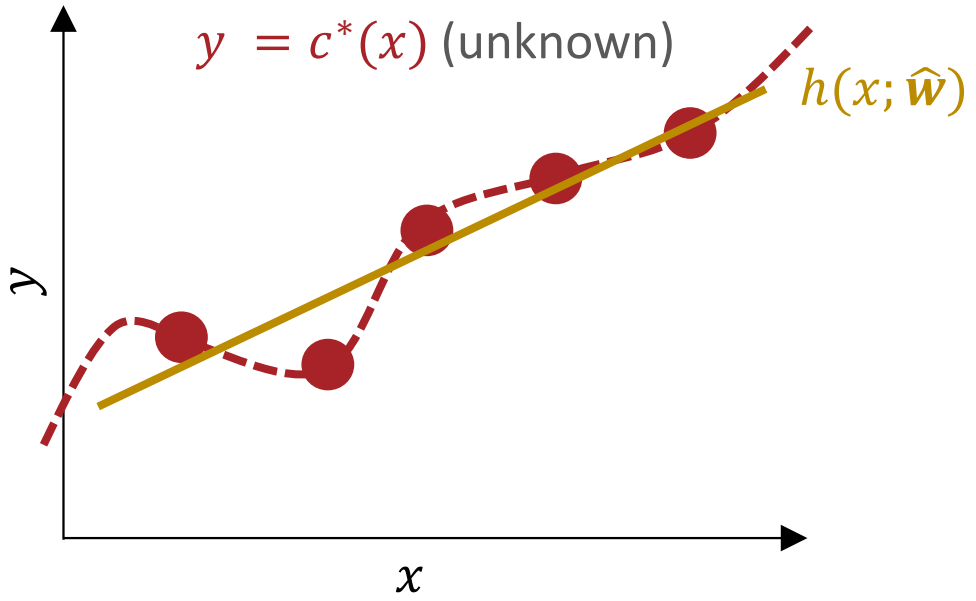


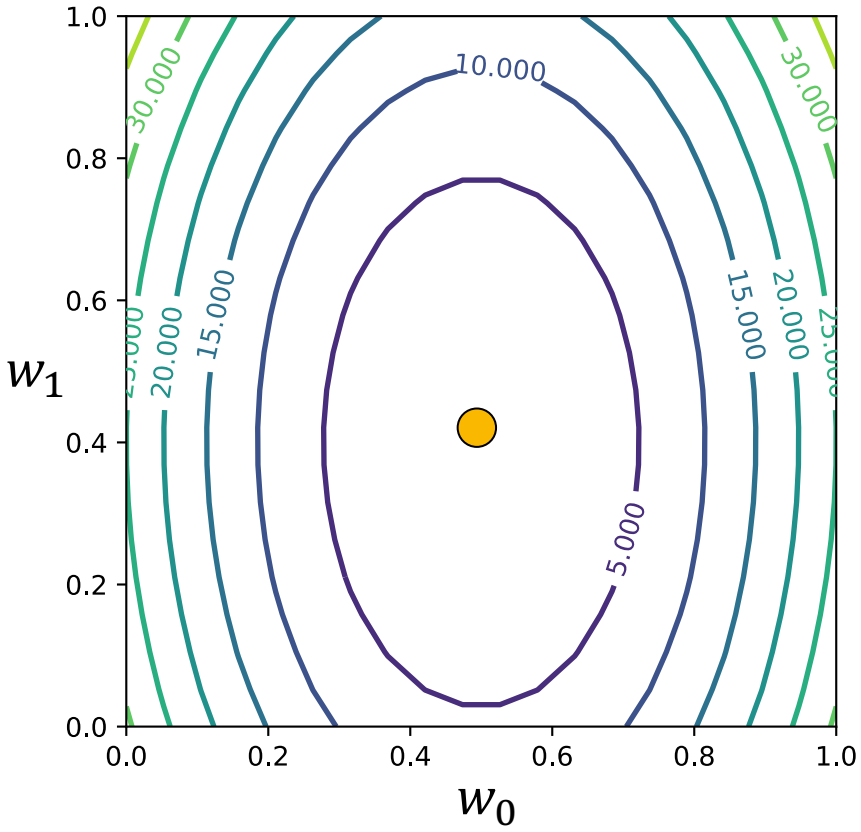


t	w_0	w_1	$\ell_{\mathcal{D}}(w_0, w_1)$
1	0.01	0.02	25.2
2	0.30	0.12	8.7
3	0.51	0.30	1.5
4	0.59	0.43	0.2

Closed Form Optimization

$$\widehat{\boldsymbol{\theta}} = (X^T X)^{-1} X^T \boldsymbol{y}$$





t	w_0	w_1	$\ell_{\mathcal{D}}(w_0, w_1)$
1	0.59	0.43	0.2

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