Lecture 11

Gradient Descent, Continued

DSC 40A, Summer 2024

Announcements

- The Midterm Exam is **tomorrow**!
- Some time for review in Discussion today, and Owen's OH 4-5p in HDSI 155.

Agenda

- Recap: Gradient descent.
- Convexity.
- More examples.
    - Huber loss.
    - Gradient descent with multiple variables.

Question 🤔

Answer at q.dsc40a.com

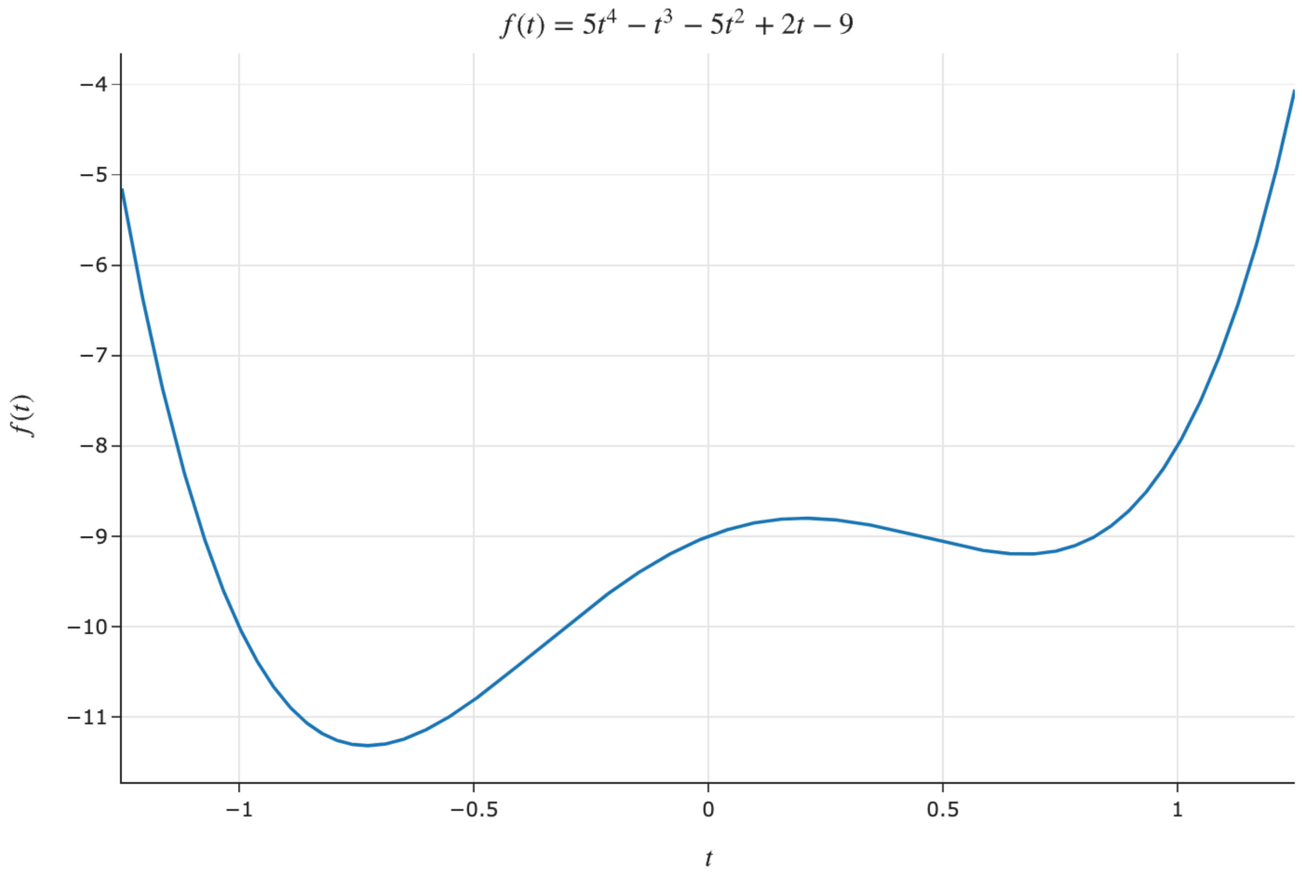
Remember, you can always ask questions at q.dsc40a.com!

If the direct link doesn't work, click the "🤔 Lecture Questions" link in the top right corner of dsc40a.com.

Overview: Gradient descent

What's the point?

- **Goal**: Given a **differentiable** function $f(t)$, find the input t^* that minimizes $f(t)$.
- What does $\frac{d}{dt} f(t)$ mean?



Gradient descent

To minimize a **differentiable** function f:

- Pick a positive number, α. This number is called the **learning rate**, or **step size**.
- Pick an **initial guess**, t_0.
- Then, repeatedly update your guess using the **update rule**:

$$t_{i+1} = t_i - \alpha \frac{df}{dt}(t_i)$$

- Repeat this process until **convergence** – that is, when t doesn't change much.
- This procedure is called **gradient descent**.

What is gradient descent?

- Gradient descent is a numerical method for finding the input to a function f that minimizes the function.
- Why is it called **gradient** descent?
  - The gradient is the extension of the derivative to functions of multiple variables.
  - We will see how to use gradient descent with multivariate functions next class.
- What is a **numerical** method?
  - A numerical method is a technique for approximating the solution to a mathematical problem, often by using the computer.
- Gradient descent is **widely used** in machine learning, to train models from linear regression to neural networks and transformers (includng ChatGPT)!

See dsc40a.com/resources/lectures/lec10 for animated examples of gradient descent, and see this notebook for the associated code!

Gradient descent and empirical risk minimization

- While gradient descent can minimize other kinds of differentiable functions, its most common use case is in **minimizing empirical risk**.
- For example, consider:
    - The constant model, $H(x) = h$.
    - Squared loss.
    - The dataset $-4, -2, 2, 4$.
    - The initial guess $h_0 = 4$ and the learning rate $\alpha = \frac{1}{4}$.
- **Exercise**: Find h_1 and h_2.

Lingering questions

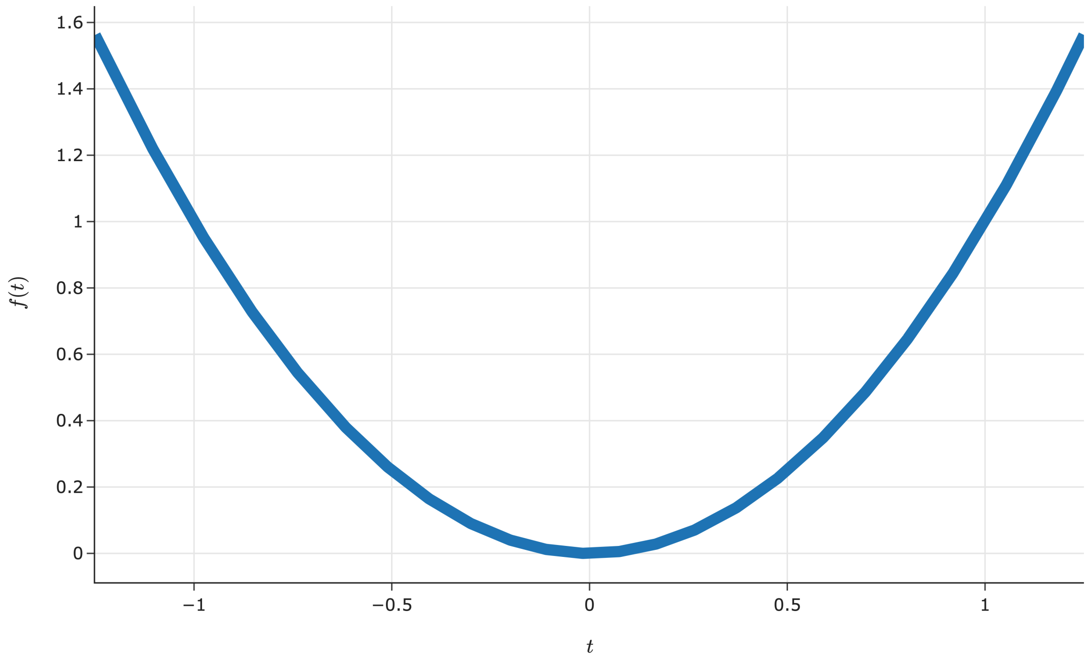
Now, we'll explore the following ideas:

- When is gradient descent *guaranteed* to converge to a global minimum?
  - What kinds of functions work well with gradient descent?
- How do I choose a step size?
- How do I use gradient descent to minimize functions of multiple variables, e.g.:

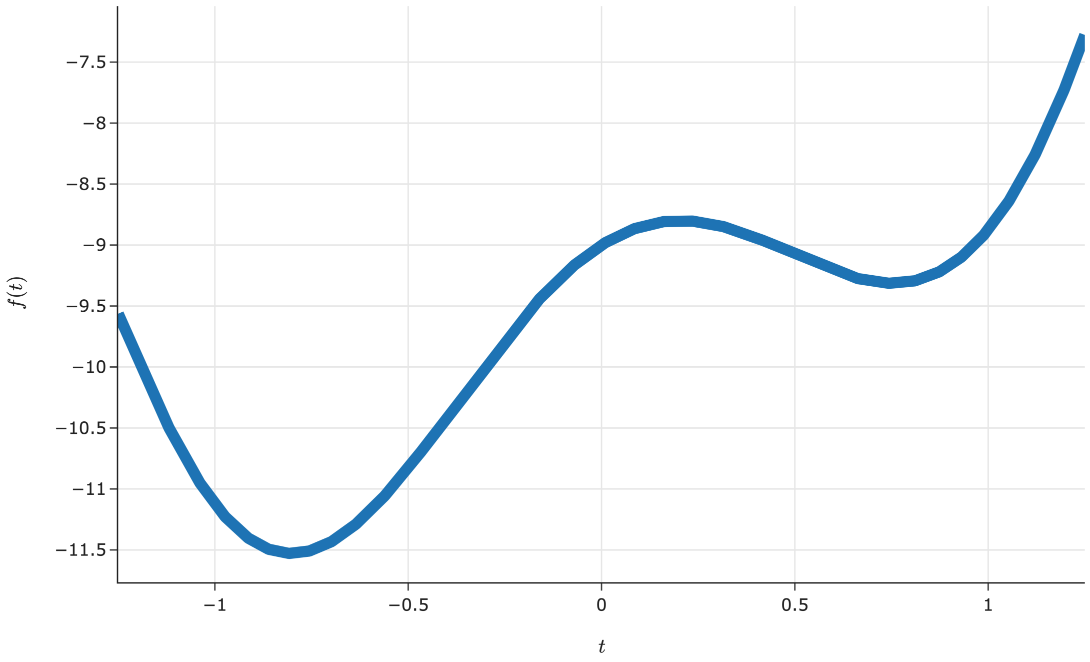
$$R_{\text{sq}}(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_0 + w_1 x_i))^2$$

When is gradient descent guaranteed to work?

Convex functions



A **convex** function ✅



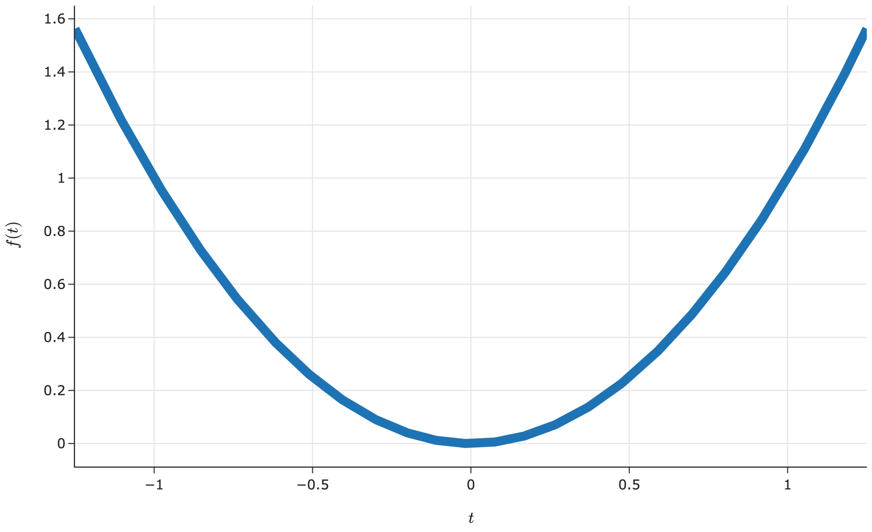
A **non-convex** function ❌

Convexity

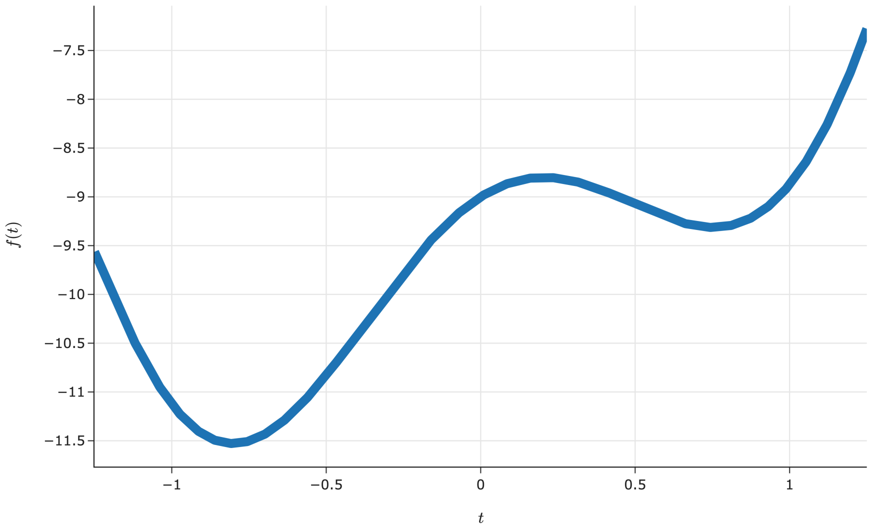
- A function f is **convex** if, for **every** a, b in the domain of f, the line segment between:

$$(a, f(a)) \text{ and } (b, f(b))$$

does not go below the plot of f.



A **convex** function ✅

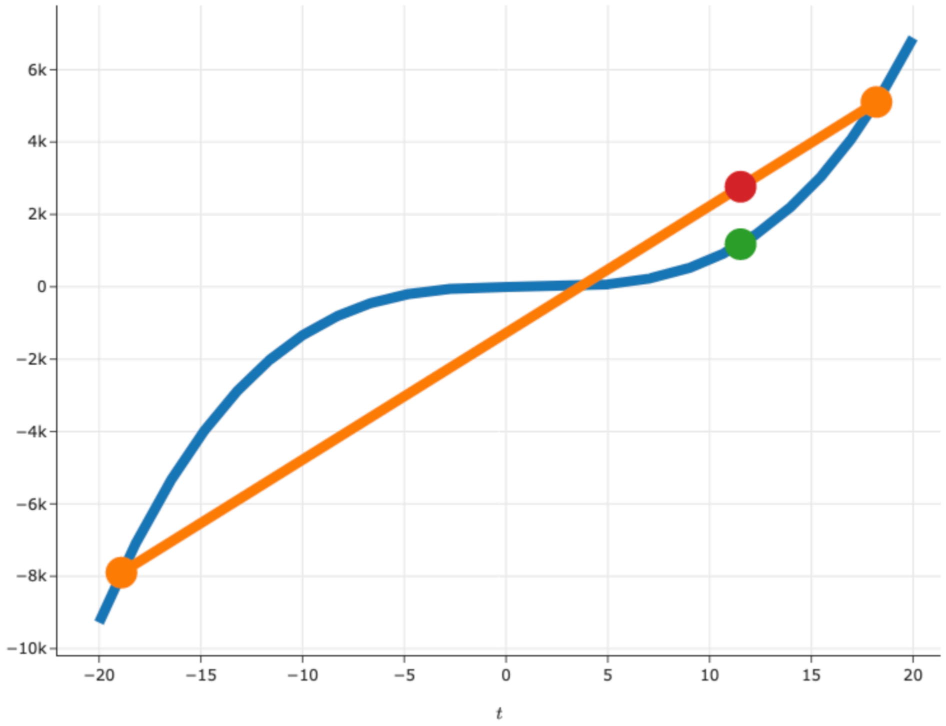


A **non-convex** function ❌

Formal definition of convexity

- A function $f : \mathbb{R} \to \mathbb{R}$ is **convex** if, for **every** a, b in the domain of f, and for every $t \in [0, 1]$:

$$\boxed{(1-t)f(a) + tf(b) \geq f((1-t)a + tb)}$$



- This is a formal way of restating the definition from the previous slide.

Question 🤔

Answer at q.dsc40a.com

Which of these functions are **not** convex?

- A. $f(x) = |x|$.
- B. $f(x) = e^x$.
- C. $f(x) = \sqrt{x - 1}$.
- D. $f(x) = (x - 3)^{24}$.
- E. More than one of the above are non-convex.

Second derivative test for convexity

- If $f(t)$ is a function of a single variable and is **twice** differentiable, then $f(t)$ is convex **if and only if**:

$$\frac{d^2 f}{dt^2}(t) \geq 0, \quad \forall\, t$$

- Example: $f(x) = x^4$ is convex.

Why does convexity matter?

- Convex functions are (relatively) easy to minimize with gradient descent.
- **Theorem**: If $f(t)$ is convex and differentiable, then gradient descent converges to a **global minimum** of f, as long as the step size is small enough.
- **Why?**
  - Gradient descent converges when the derivative is 0.
  - For convex functions, the derivative is 0 only at one place – the global minimum.
  - In other words, if f is convex, gradient descent won't get "stuck" and terminate in places that aren't global minimums (local minimums, saddle points, etc.).

Nonconvex functions and gradient descent

- We say a function is **nonconvex** if it does not meet the criteria for convexity.
- Nonconvex functions are (relatively) difficult to minimize.
- Gradient descent **might** still work, but it's not guaranteed to find a global minimum.
    - We saw this at the start of the lecture, when trying to minimize $f(t) = 5t^4 - t^3 - 5t^2 + 2t - 9$.

Choosing a step size in practice

- In practice, choosing a step size involves a lot of trial-and-error.
- In this class, we've only touched on "constant" step sizes, i.e. where α is a constant.

$$t_{i+1} = t_i - \alpha \frac{df}{dt}(t_i)$$

- **Remember**: α is the "step size", but the amount that our guess for t changes is $\alpha \frac{df}{dt}(t_i)$, not just α.
- In future courses, you'll learn about "decaying" step sizes, where the value of α decreases as the number of iterations increases.
  - Intuition: take much bigger steps at the start, and smaller steps as you progress, as you're likely getting closer to the minimum.

More examples

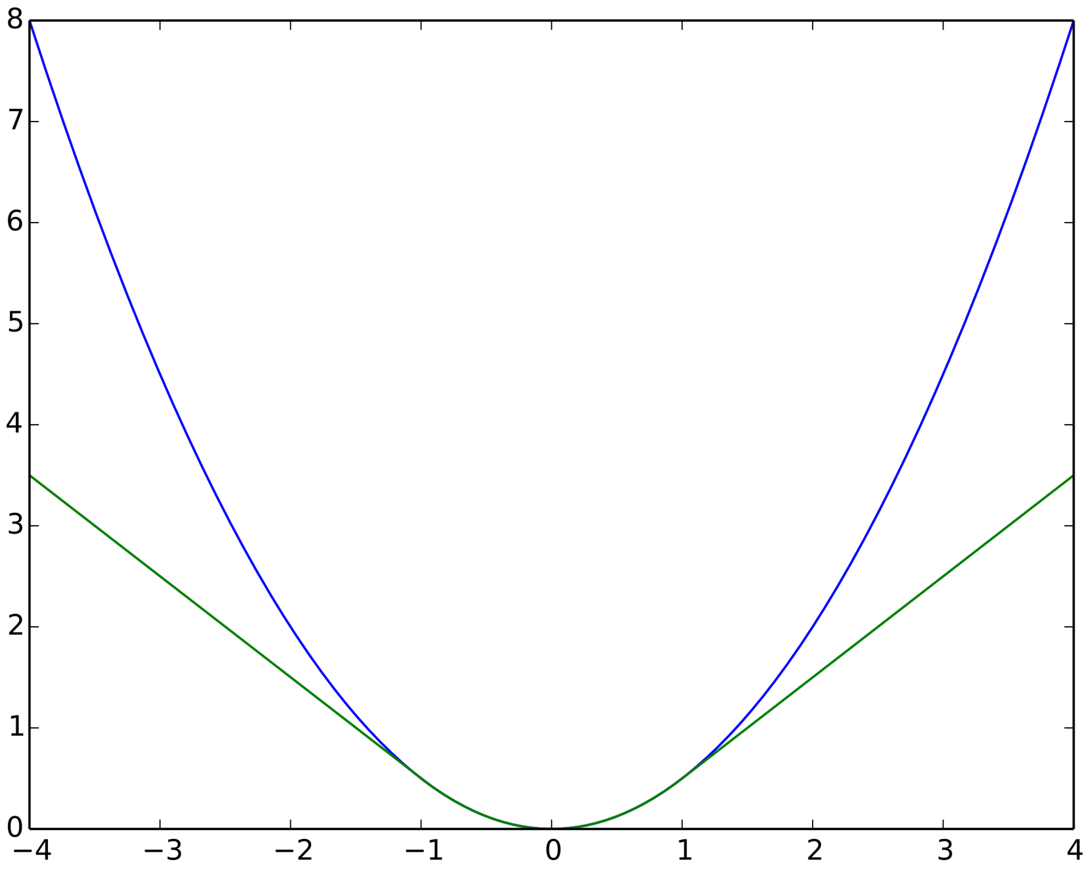
Example: Huber loss and the constant model

- First, we learned about squared loss, $L_{\text{sq}}(y_i, H(x_i)) = (y_i - H(x_i))^2$.

- Then, we learned about absolute loss, $L_{\text{abs}}(y_i, H(x_i)) = |y_i - H(x_i)|$.

- Let's look at a new loss function, **Huber loss**:

$$L_{\text{huber}}(y_i, H(x_i)) = \begin{cases} \frac{1}{2}(y_i - H(x_i))^2 & \text{if } |y_i - H(x_i)| \leq \delta \\ \delta \cdot \left(|y_i - H(x_i)| - \frac{1}{2}\delta\right) & \text{otherwise} \end{cases}$$



Squared loss in blue, **Huber** loss in green.

Note that both loss functions are convex!

Minimizing average Huber loss for the constant model

- For the constant model, $H(x) = h$:

$$L_{\text{huber}}(y_i, h) = \begin{cases} \frac{1}{2}(y_i - h)^2 & \text{if } |y_i - h| \leq \delta \\ \delta \cdot (|y_i - h| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$$

$$\implies \frac{\partial L}{\partial h}(h) = \begin{cases} -(y_i - h) & \text{if } |y_i - h| \leq \delta \\ -\delta \cdot \text{sign}(y_i - h) & \text{otherwise} \end{cases}$$

- So, the **derivative** of empirical risk is:

$$\frac{dR_{\text{huber}}}{dh}(h) = \frac{1}{n} \sum_{i=1}^{n} \begin{cases} -(y_i - h) & \text{if } |y_i - h| \leq \delta \\ -\delta \cdot \text{sign}(y_i - h) & \text{otherwise} \end{cases}$$

- It's **impossible** to set $\frac{dR_{\text{huber}}}{dh}(h) = 0$ and solve by hand: we need gradient descent!

Let's try this out in practice! Follow along in this notebook.

Minimizing functions of multiple variables

- Consider the function:

$$f(x_1, x_2) = (x_1 - 2)^2 + 2x_1 - (x_2 - 3)^2$$

- It has two **partial derivatives**: $\frac{\partial f}{\partial x_1}$ and $\frac{\partial f}{\partial x_2}$.

The gradient vector

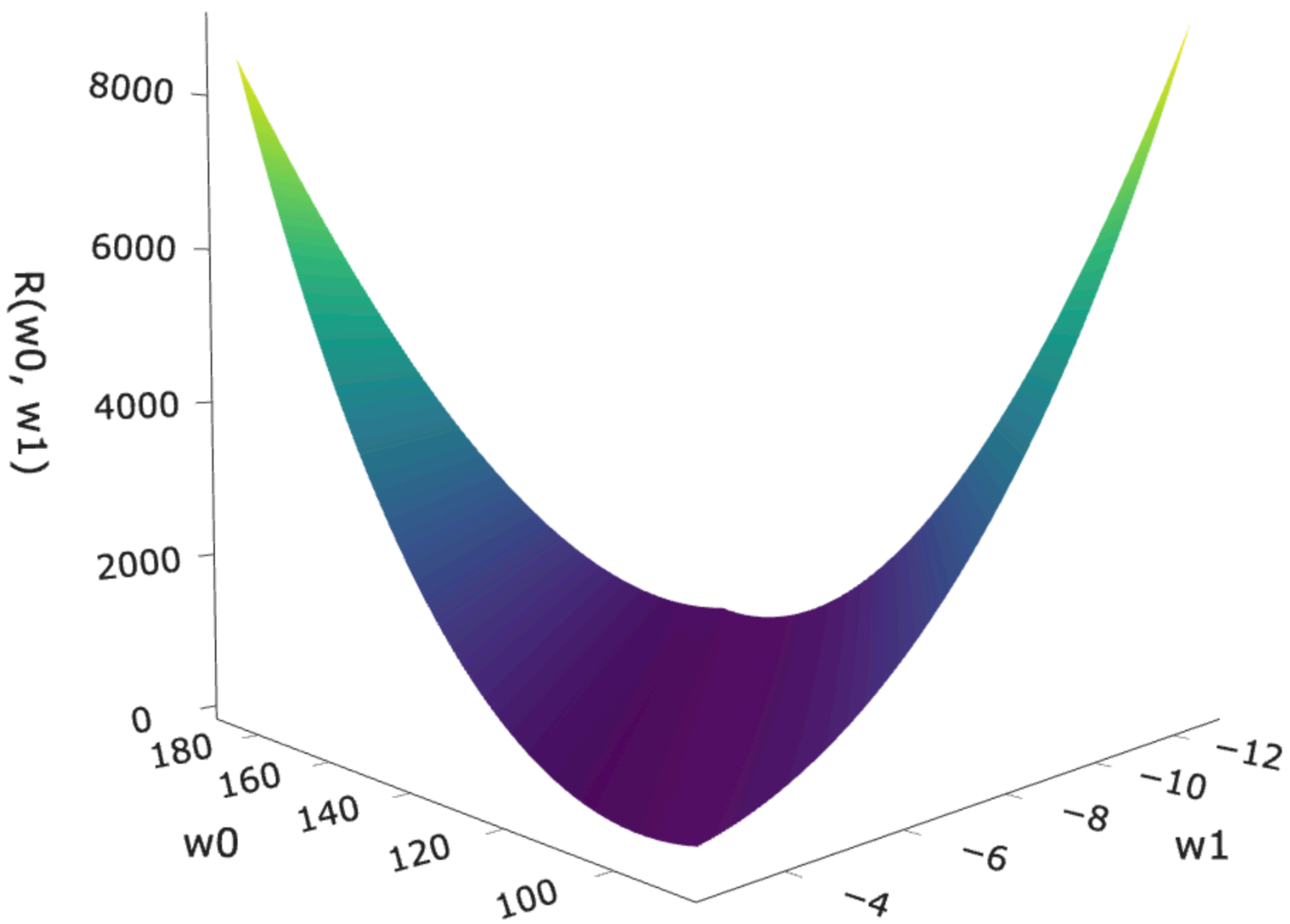
- If $f(\vec{x})$ is a function of multiple variables, then its **gradient**, $\nabla f(\vec{x})$, is a vector containing its partial derivatives.

- Example:

$$f(\vec{x}) = (x_1 - 2)^2 + 2x_1 - (x_2 - 3)^2$$
$$\nabla f(\vec{x}) = \begin{bmatrix} 2x_1 - 2 \\ 2x_2 - 6 \end{bmatrix}$$

- Example:

$$f(\vec{x}) = \vec{x}^T \vec{x}$$
$$\implies \nabla f(\vec{x}) =$$



R(w0, w1)
8000
6000
4000
2000
0
180
160
140
120
100
w0
−12
−10
−8
−6
−4
w1

Gradient descent for functions of multiple variables

- Example:

$$f(x_1, x_2) = (x_1 - 2)^2 + 2x_1 - (x_2 - 3)^2$$

$$\nabla f(\vec{x}) = \begin{bmatrix} 2x_1 - 2 \\ 2x_2 - 6 \end{bmatrix}$$

- The minimizer of f is a vector, $\vec{x}^* = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix}$.
- We start with an initial guess, $\vec{x}^{(0)}$, and step size α, and update our guesses using:

$$\vec{x}^{(i+1)} = \vec{x}^{(i)} - \alpha \nabla f(\vec{x}^{(i)})$$

Exercise

$$f(x_1, x_2) = (x_1 - 2)^2 + 2x_1 - (x_2 - 3)^2$$

$$\nabla f(\vec{x}) = \begin{bmatrix} 2x_1 - 2 \\ 2x_2 - 6 \end{bmatrix}$$

$$\vec{x}^{(i+1)} = \vec{x}^{(i)} - \alpha \nabla f(\vec{x}^{(i)})$$

Given an initial guess of $\vec{x}^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and a step size of $\alpha = \frac{1}{3}$, perform **two** iterations of gradient descent. What is $\vec{x}^{(2)}$?

Example: Gradient descent for simple linear regression

- To find optimal model parameters for the model $H(x) = w_0 + w_1 x$ and squared loss, we minimized empirical risk:

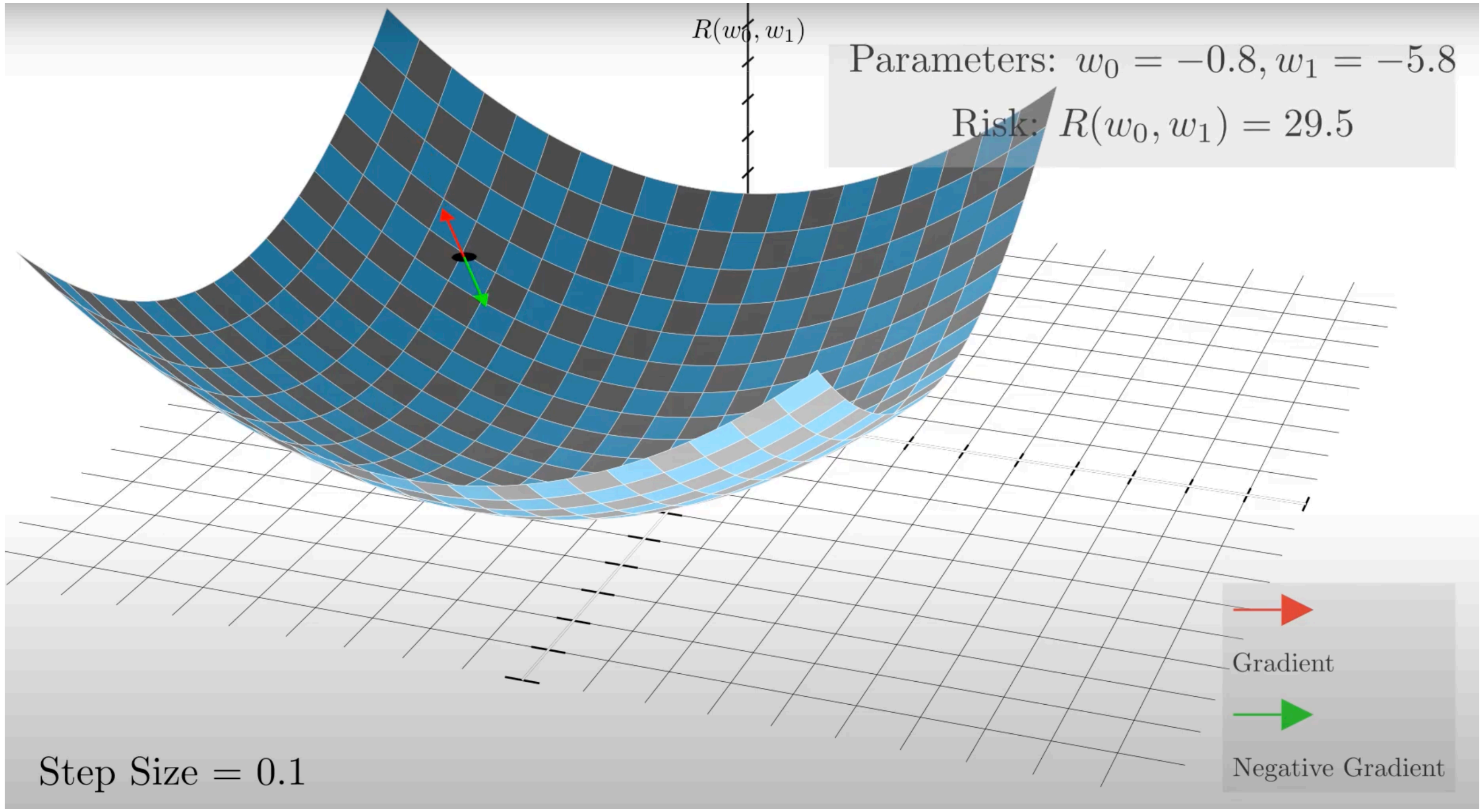
$$R_{\text{sq}}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - (w_0 + w_1 x_i))^2$$

- This is a function of multiple variables, and is differentiable, so it has a gradient!

$$\nabla R(\vec{w}) = \begin{bmatrix} -\frac{2}{n} \sum_{i=1}^{n} (y_i - (w_0 + w_1 x_i)) \\ -\frac{2}{n} \sum_{i=1}^{n} (y_i - (w_0 + w_1 x_i)) x_i \end{bmatrix}$$

- **Key idea**: To find w_0^* and w_1^*, we *could* use gradient descent!

Gradient descent for simple linear regression, visualized



Let's watch 🎥 this animation that Jack made.

What's next?

- The Midterm Exam is tomorrow, in this room!
- In Homework 5, you'll see a few questions involving today's material:
    - A question about convexity.
    - A question about implementing gradient descent to find optimal parameters for a model that is **not linear in its parameters**.
- On Monday, we'll start talking about probability.
    - Homework 5 will have a probability problem taken from a past DSC 10 exam, to help you refresh.