

Lecture 11

# Gradient Descent, Continued

DSC 40A

# Agenda

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

## Question 🤔

Take a moment to pause and reflect...

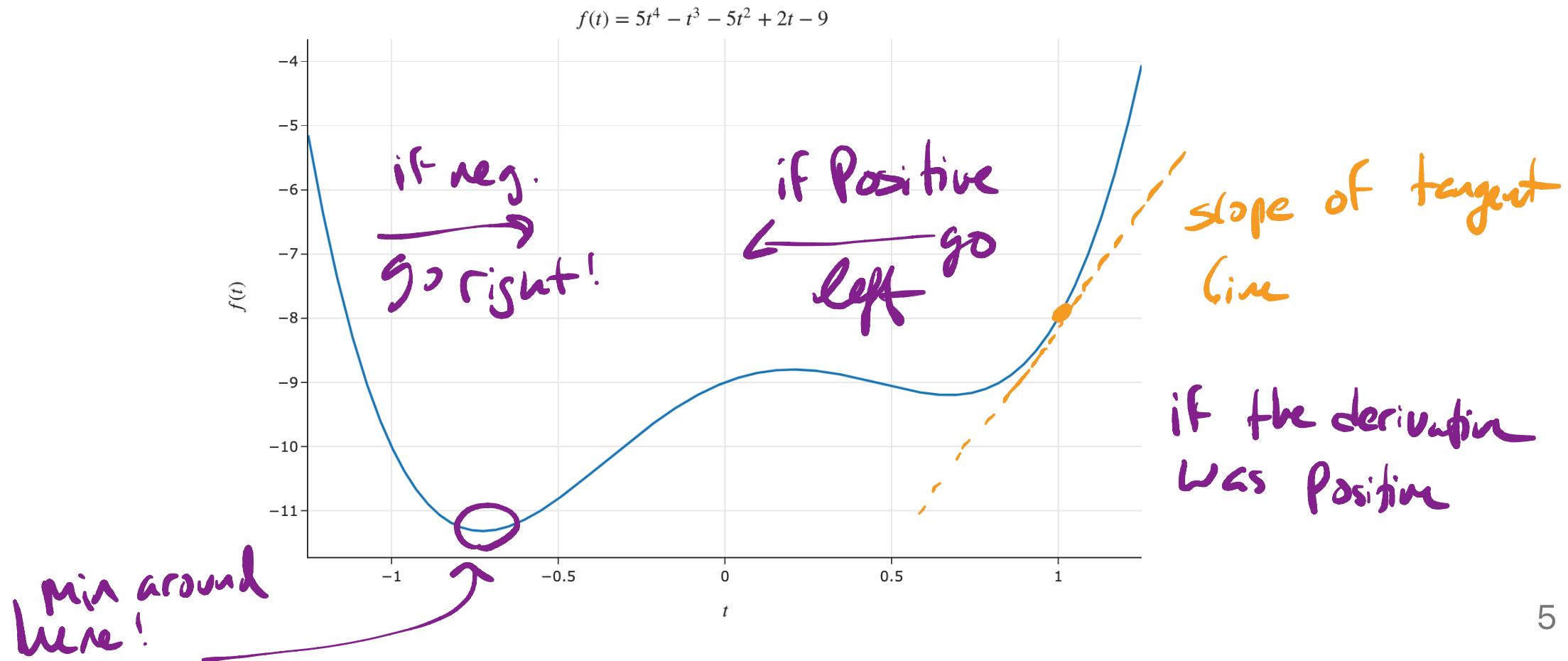
If you have any questions please post online to our forms/Q&A site.

Course staff will answer them ASAP!

# 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,  $\alpha$ . 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**:

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

learning Rate is how  
much movement

this negative here so we go opposite

- 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 (including ChatGPT)!

# 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,  $H(x) = h$ .
  - 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  $h_2$ .

$$\text{Data} = [-4, -2, 2, 4]$$

$$\alpha = \frac{1}{4} \quad h_0 = 4$$

$2h$

iteration 1  $h_0 = 4$

$$h_1 = h_0 - \alpha \frac{d R_{\text{sq}}}{dh}(h_0) \quad h_0 = 4$$

$$4 - \frac{1}{4} \cdot (2 \cdot 4) \boxed{= 2} = h_1$$

$$h_2 = h_1 - \alpha \cdot \frac{d R_{\text{sq}}}{dh}(h_1)$$

$$2 - \frac{1}{4} \cdot (2 \cdot 2) = 2 - 4 \frac{1}{4} \circled{= 1}$$

$$R_{\text{sq}}(h) = \frac{1}{n} \sum_{i=1}^n (y_i - h)^2$$

$$\frac{d R_{\text{sq}}}{dh} = \frac{1}{n} \sum_{i=1}^n 2(y_i - h)(-1)$$

$$= \frac{-2}{n} \sum_{i=1}^n (y_i - h)$$

$n = 4$

$$= \frac{-2}{4} \left( \sum_{i=1}^4 y_i - 4h \right) \xrightarrow{(4-2+2+4)} = 0$$

$$= \frac{-2}{4} (0 - 4h) \circled{= 2h}$$

## Lingering questions

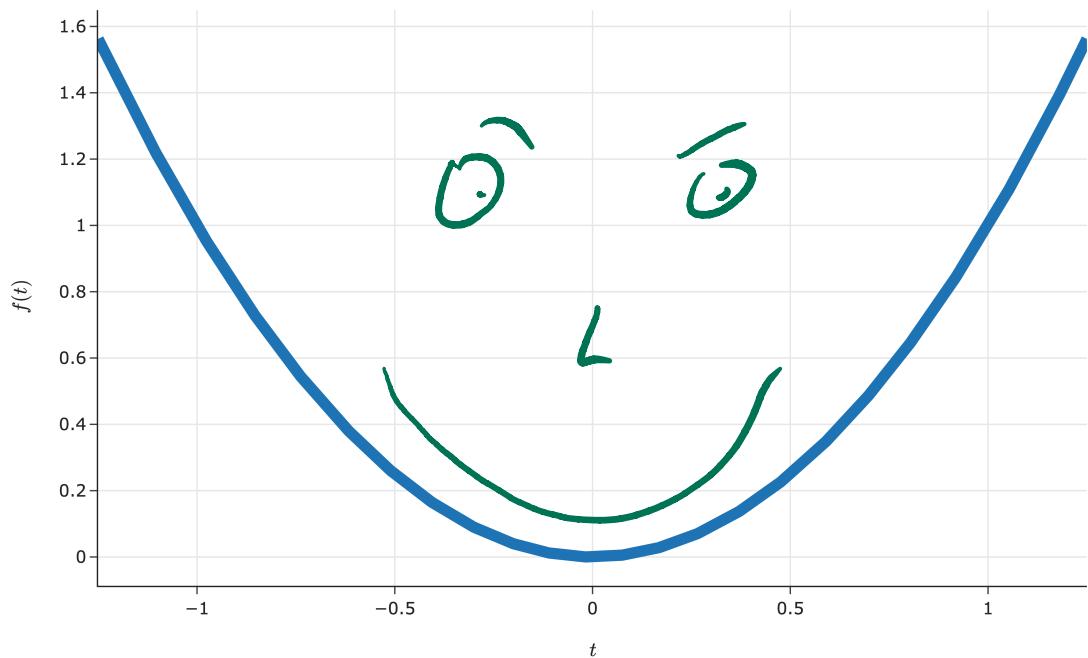
Now, we'll explore the following ideas:

- 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.:

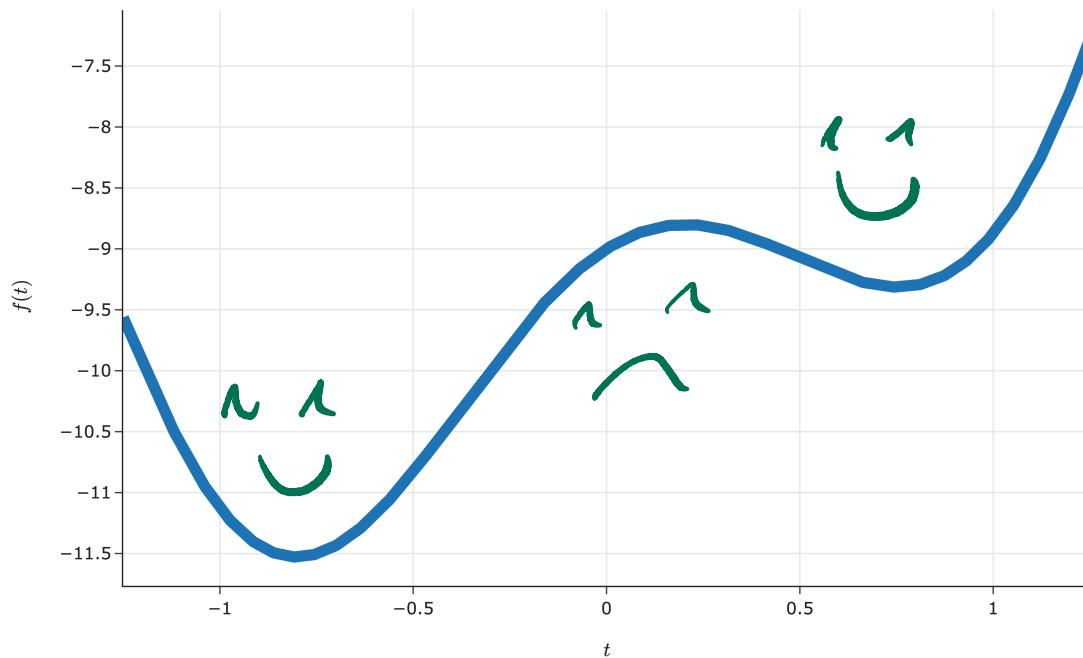
$$R_{\text{sq}}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^n (y_i - (w_0 + w_1 x_i))^2$$

# When is gradient descent guaranteed to work?

# Convex functions



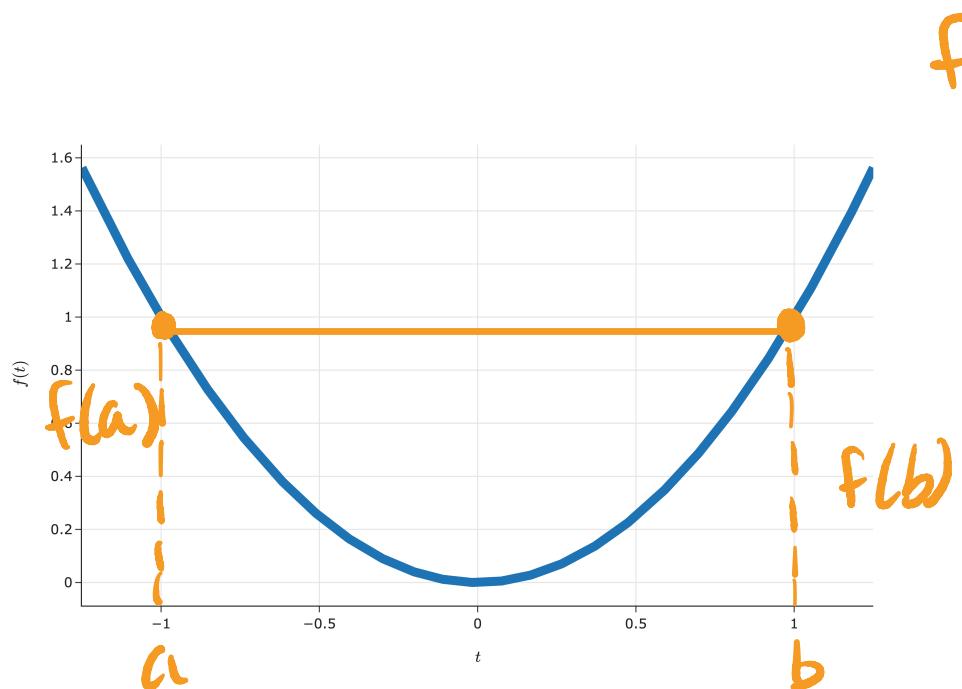
A convex function



A non-convex function

# Convexity

- A function  $f$  is **convex** if, for **every**  $a, b$  in the domain of  $f$ , the line segment between:  
 $(a, f(a))$  and  $(b, f(b))$   
does not go below the plot of  $f$ .



A convex function ✓



A non-convex function ✗

## Formal definition of convexity

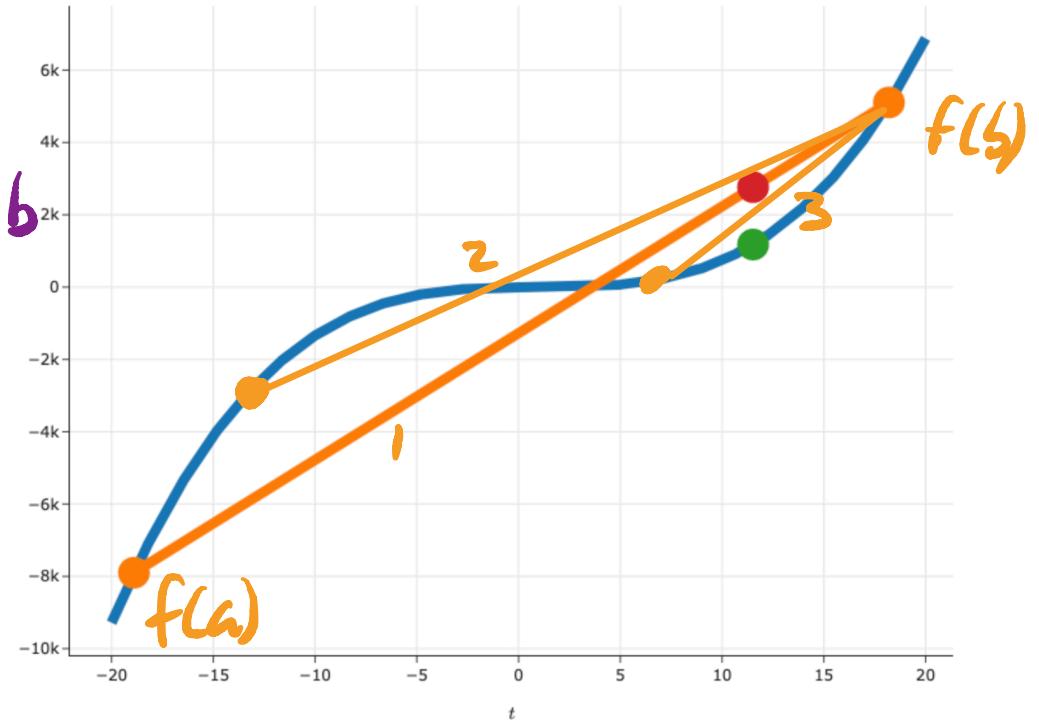
- A function  $f : \mathbb{R} \rightarrow \mathbb{R}$  is **convex** if, for every  $a, b$  in the domain of  $f$ , and for every  $t \in [0, 1]$ :

*Line between  $f(a)$  &  $f(b)$*       *function between  $a, b$*

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

*line  $\geq$  function*

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

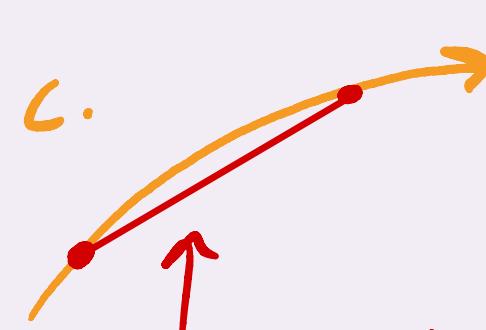
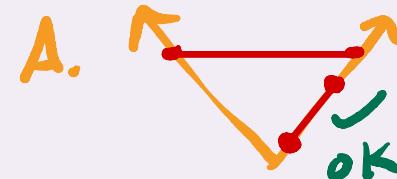


## Question 🤔

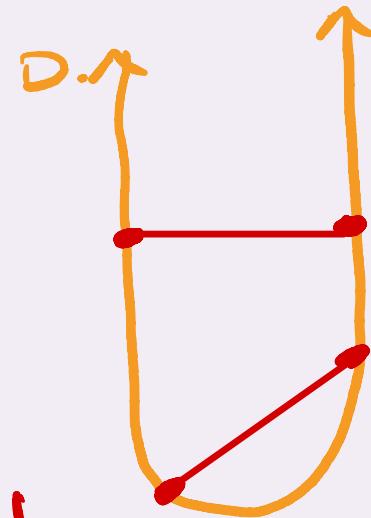
Pause the video and try to answer the question...

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.



Area under  
the function  
is no good!



## 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:

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

$\forall$  = for all

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

$$f'(x) = 4x^3$$

$$f''(x) = 12x^2 \geq 0 \quad \forall x$$

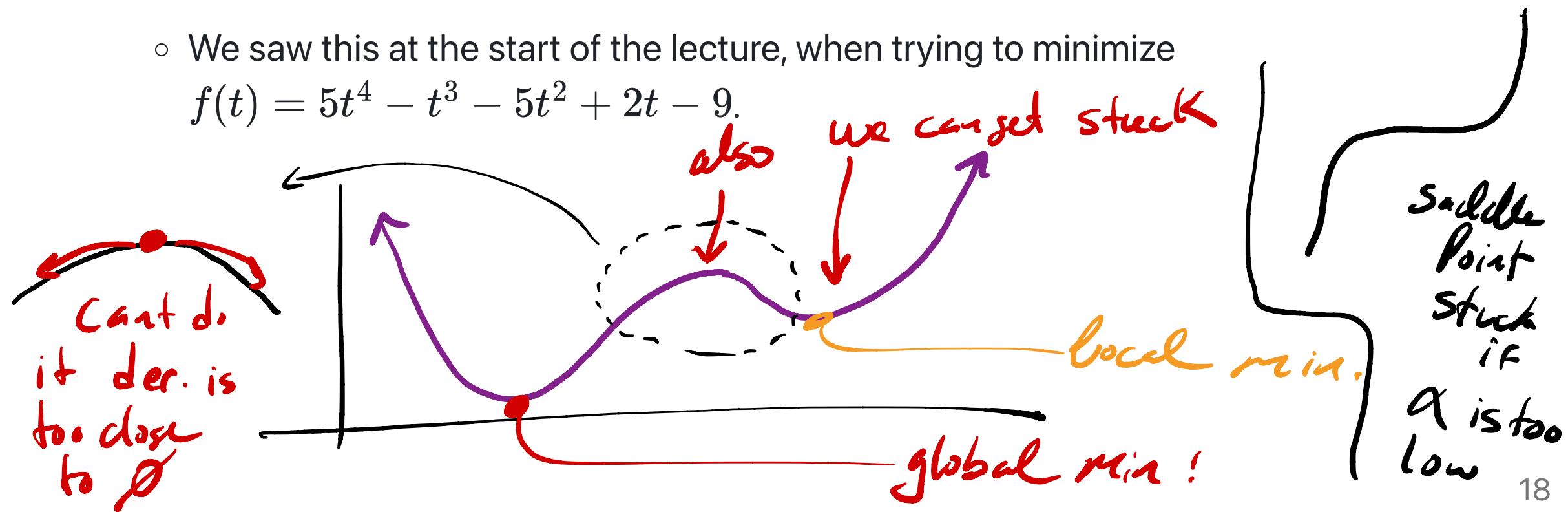
## Why does convexity matter?

- 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

$$f(t) = 5t^4 - t^3 - 5t^2 + 2t - 9.$$



$$\alpha = 0.09$$

gives good value

## 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  $\alpha$  is a constant.

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



- Remember:  $\alpha$  is the "step size", but the amount that our guess for  $t$  changes is  $\alpha \frac{df}{dt}(t_i)$ , not just  $\alpha$ .
- In future courses, you'll learn about "decaying" step sizes, where the value of  $\alpha$  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,

$$L_{\text{sq}}(y_i, H(x_i)) = (y_i - H(x_i))^2.$$

**Pros:** Differentiable  , Easy to Minimize 

**Cons:** Sensitive to outliers 

- Then, we learned about absolute loss,

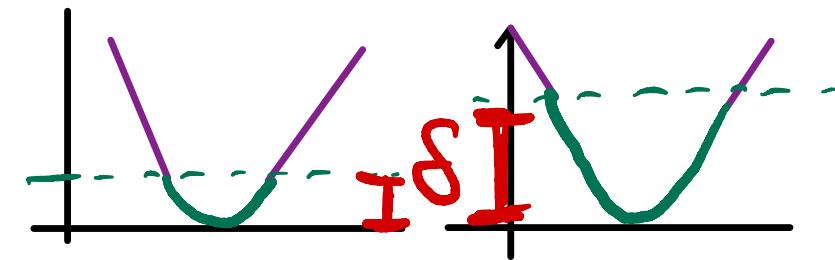
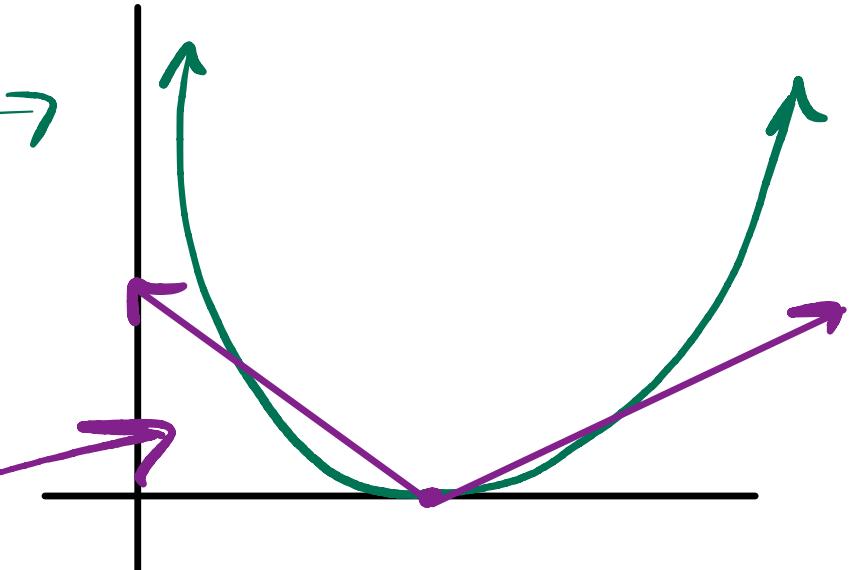
$$L_{\text{abs}}(y_i, H(x_i)) = |y_i - H(x_i)|.$$

**Pros:** Robust to Outliers 

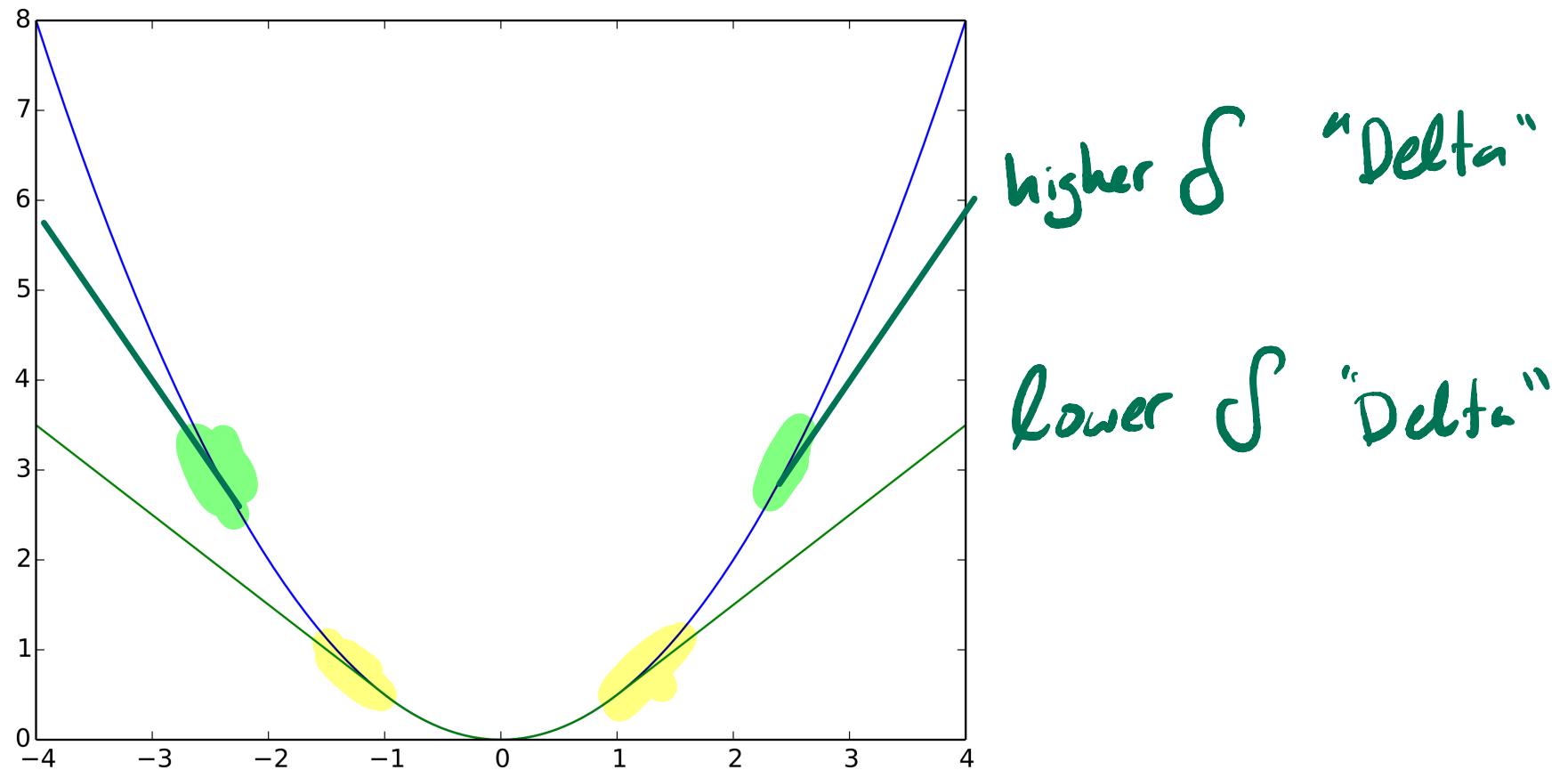
**Cons:** not Differentiable , harder to minimize  

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

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



$\boxed{\delta}$   
Delta



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!

Analytical Solution is impractical So we use a numerical Sol.

## 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  $\frac{\partial f}{\partial x_2}$ .

$$\frac{\partial f}{\partial x_1} = 2(x_1 - 2) + 2 = 2x_1 - 4 + 2 = \boxed{2x_1 - 2}$$
$$\frac{\partial f}{\partial x_2} = \boxed{-2(x_2 - 3)}$$

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

$\nabla$  nabula

## 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:

$$f(\vec{x}) = \vec{x} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \mathbb{R}^2 \rightarrow \mathbb{R}$$
$$f(\vec{x}) = (x_1 - 2)^2 + 2x_1 - (x_2 - 3)^2$$

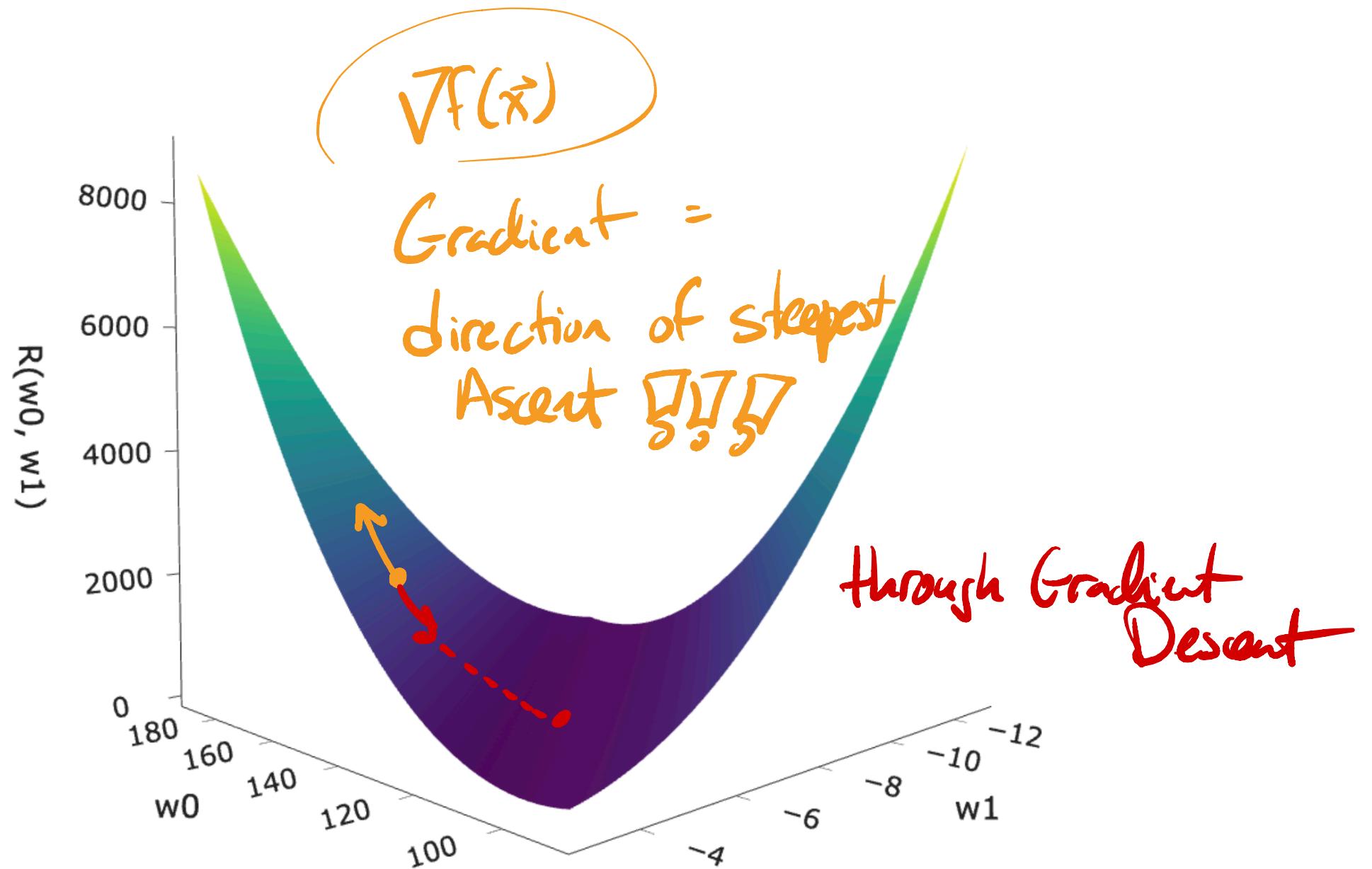
$$\nabla f(\vec{x}) = \begin{bmatrix} 2x_1 - 2 \\ 2x_2 + 6 \end{bmatrix}$$

- Example:

$$f(\vec{x}) = \vec{x}^T \vec{x}$$
$$\Rightarrow \nabla f(\vec{x}) = \begin{bmatrix} 2x_1 \\ 2x_2 \\ \vdots \\ 2x_n \end{bmatrix} = 2\vec{x}$$

Result in a scalar ?

$$x_1^2 + x_2^2 + \dots + x_n^2 \quad \frac{\partial F}{\partial x_i} = 2x_i$$



## 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  $\alpha$ , and update our guesses using:

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

*gradient ( $\vec{x}$ )*

## 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  $\vec{x}^{(2)}$ ?

$$\vec{x}^{(1)} = \vec{x}^{(0)} - \alpha \nabla f(\vec{x}^{(0)}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \frac{1}{3} \begin{bmatrix} 2 \cdot 0 - 2 \\ -(2 \cdot 0 - 6) \end{bmatrix} = -\frac{1}{3} \begin{bmatrix} -2 \\ 6 \end{bmatrix} = \begin{bmatrix} \frac{2}{3} \\ -2 \end{bmatrix}$$

$$\vec{x}^{(2)} = \begin{bmatrix} \frac{2}{3} \\ -2 \end{bmatrix} - \frac{1}{3} \begin{bmatrix} 2 \cdot \frac{2}{3} - 2 \\ -(2 \cdot (-2) - 6) \end{bmatrix}$$

next (2<sup>nd</sup>) iteration of  
G.D.



## Example: Gradient descent for simple linear regression

- To find optimal model parameters for the model  $H(x) = w_0 + w_1x$  and squared loss, we minimized empirical risk:

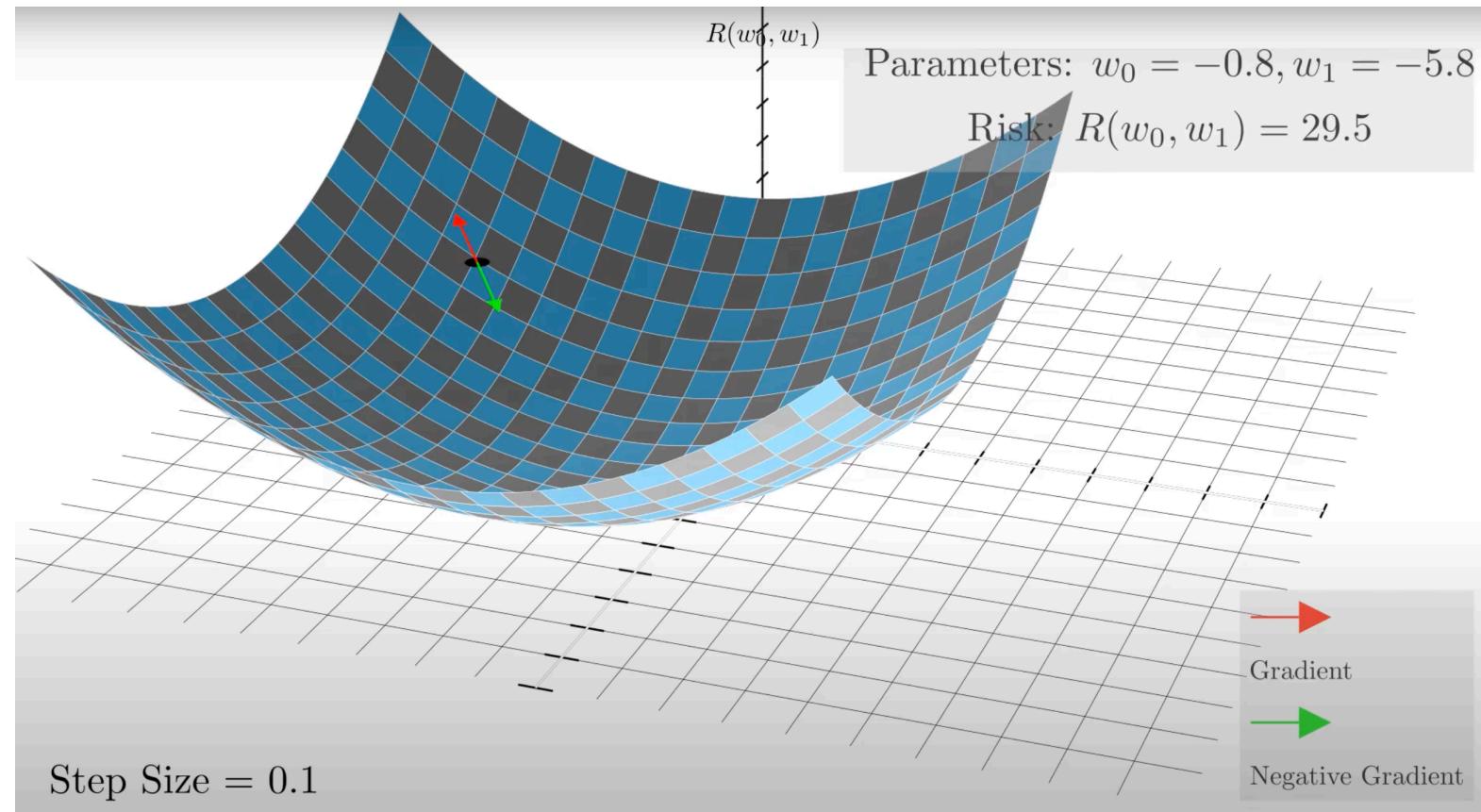
$$R_{\text{sq}}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^n (y_i - (w_0 + w_1 x_i))^2$$

- 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



Thank you  
Jack!

Let's watch [this animation](#) that Jack made.

## What's next?

- We'll start talking about probability.
- :)