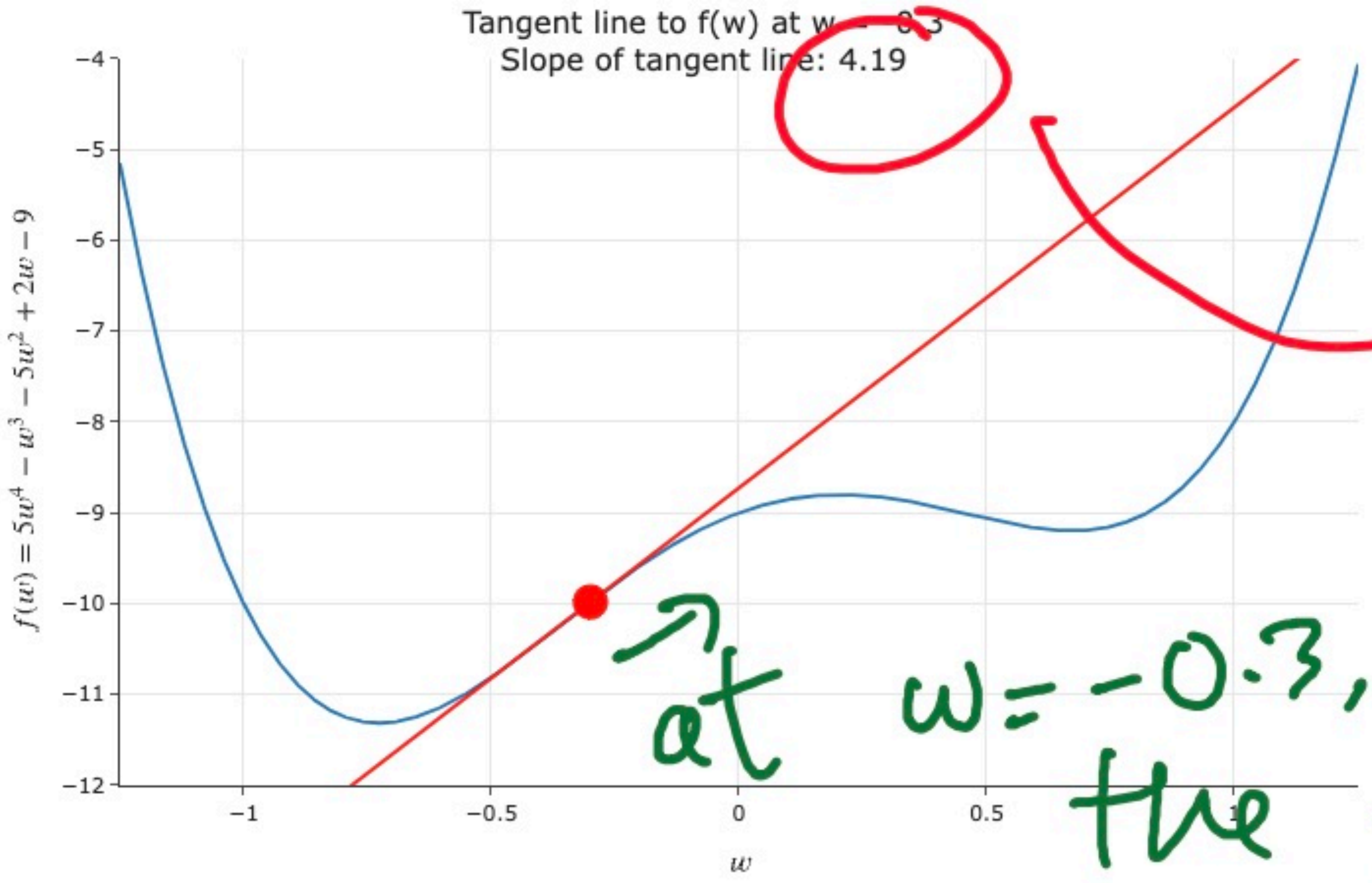
What does the derivative of a function tell us?

- **Goal**: Given a **differentiable** function $f(w)$, find the input w^* that minimizes $f(w)$.

- What does $\frac{d}{dw} f(w)$ mean?

In [3]:
```python
from ipywidgets import interact
interact(util.show_tangent, w0=(-1.5, 1.5));
```

w0 −0.30



derivative tells us the slope of tangent line!

at $w = -0.3$, the function is increasing!

Gradient descent

- To minimize a **differentiable** function f:

    1. Pick a positive number, α. This number is called the **learning rate**, or **step size**.
       Think of α as a hyperparameter of the minimization process.
    2. Pick an **initial guess**, $w^{(0)}$.
    3. Then, repeatedly update your guess using the **update rule**:

$$w^{(t+1)} = w^{(t)} - \alpha \frac{df}{dw}(w^{(t)})$$

walk opposite the direction of derivative

new guess of w^* = old guess of w^* − step size · derivative of f at old guess

- Repeat this process until **convergence** – that is, when the difference between $w^{(t)}$ and $w^{(t+1)}$ is small.

- This procedure is called **gradient descent**.

jupyter lec20-blank Last Checkpoint: a minute ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Python 3 (ipykernel)

Run

Code

```
w: -0.654, derivative at w: 1.6627
w: -0.6706, derivative at w: 1.3252
w: -0.6839, derivative at w: 1.0392
w: -0.6943, derivative at w: 0.8041
w: -0.7023, derivative at w: 0.6156
w: -0.7085, derivative at w: 0.4673
w: -0.7131, derivative at w: 0.3525
w: -0.7166, derivative at w: 0.2645
w: -0.7193, derivative at w: 0.1978
w: -0.7213, derivative at w: 0.1474
w: -0.7227, derivative at w: 0.1097
w: -0.7238, derivative at w: 0.0815
w: -0.7247, derivative at w: 0.0604
w: -0.7253, derivative at w: 0.0448
w: -0.7257, derivative at w: 0.0332
w: -0.726, derivative at w: 0.0246
w: -0.7263, derivative at w: 0.0182
w: -0.7265, derivative at w: 0.0134
w: -0.7266, derivative at w: 0.0099
w: -0.7267, derivative at w: 0.0074
w: -0.7268, derivative at w: 0.0054
w: -0.7268, derivative at w: 0.004
w: -0.7269, derivative at w: 0.003
w: -0.7269, derivative at w: 0.0022
w: -0.7269, derivative at w: 0.0016
w: -0.7269, derivative at w: 0.0012
w: -0.727, derivative at w: 0.0009
w: -0.727, derivative at w: 0.0007
w: -0.727, derivative at w: 0.0005
w: -0.727, derivative at w: 0.0004
```

think of gradient descent as an operation that finds where the derivative is 0!

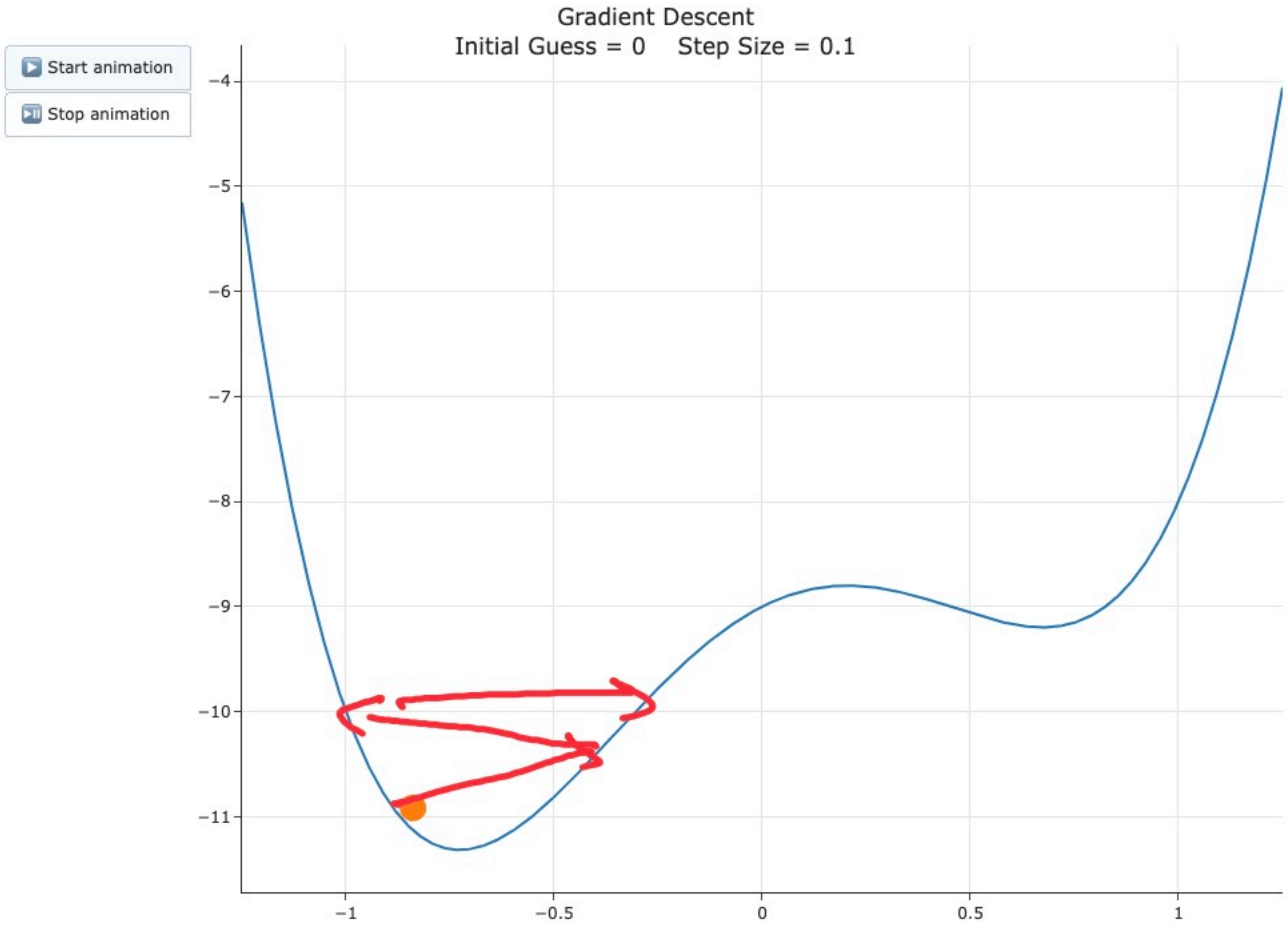
In [7]:
```
w
```

Out[7]: -0.7269745284952817

Visualizing $w^{(0)} = 0, \alpha = 0.1$

What if we use a different learning rate?

In [20]:
```
util.minimizing_animation(w0=0, alpha=0.1)
```



Lingering questions

- When is gradient descent *guaranteed* to converge to a global minimum? What kinds of functions work well with gradient descent?

- How do we choose a step size?

- How do we use gradient descent to minimize functions of multiple variables, e.g.:

"sub" gradient descent

$$R_{\text{ridge}}(\vec{w}) = \frac{1}{n}\|\vec{y} - X\vec{w}\|^2 + \lambda \sum_{j=1}^{d} w_j^2$$

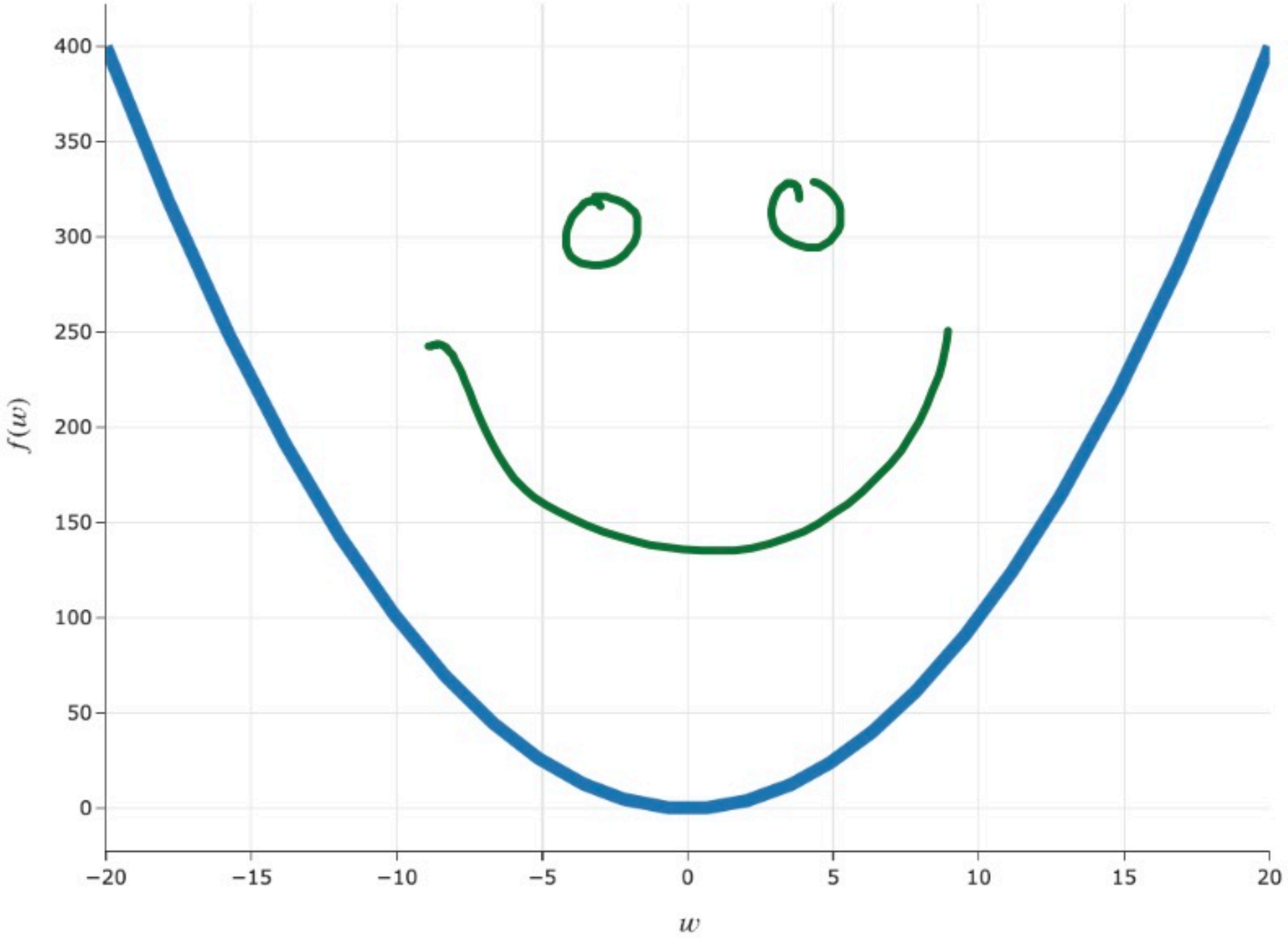
This is a function of $d + 1$ variables: $w_0, w_1, \ldots, w_d$.

abs value function is not differentiable everywhere

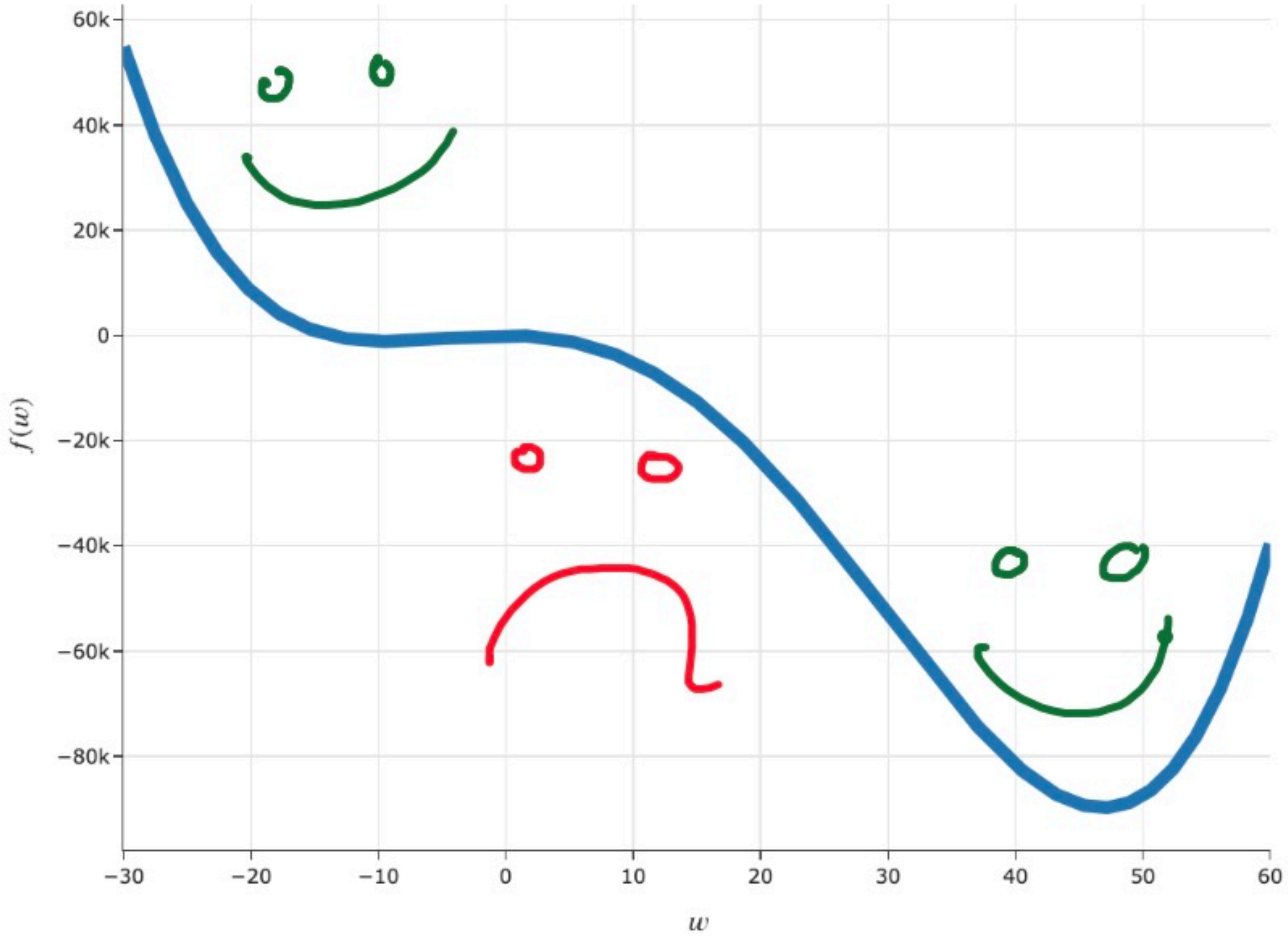
- **Question**: Why **can't** we use gradient descent to find $\vec{w}^*_{\text{LASSO}}$?

$$R_{\text{LASSO}}(\vec{w}) = \frac{1}{n}\|\vec{y} - X\vec{w}\|^2 + \lambda \sum_{j=1}^{d} |w_d|$$

What makes a function convex?

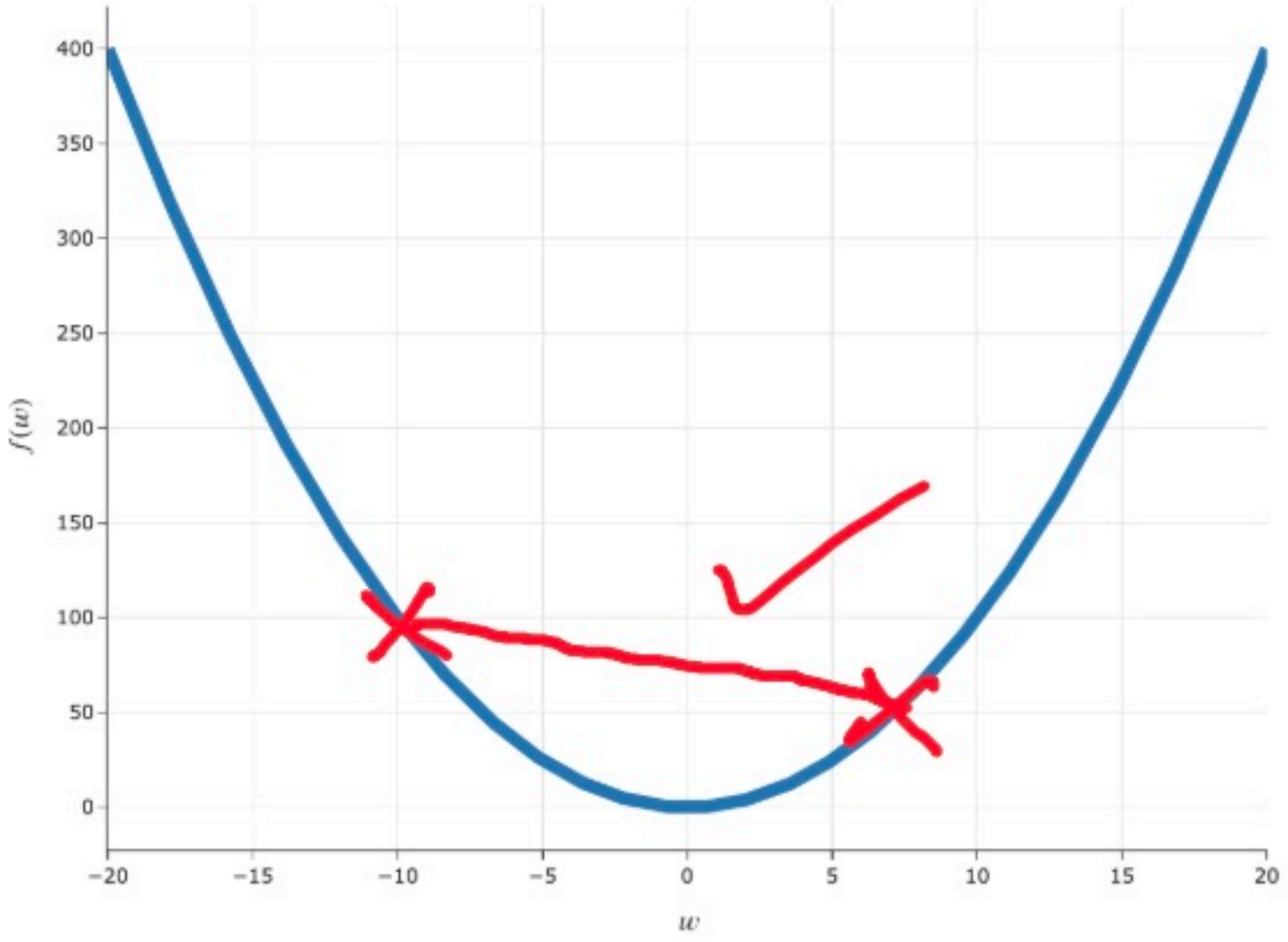


A **convex** function ✅.

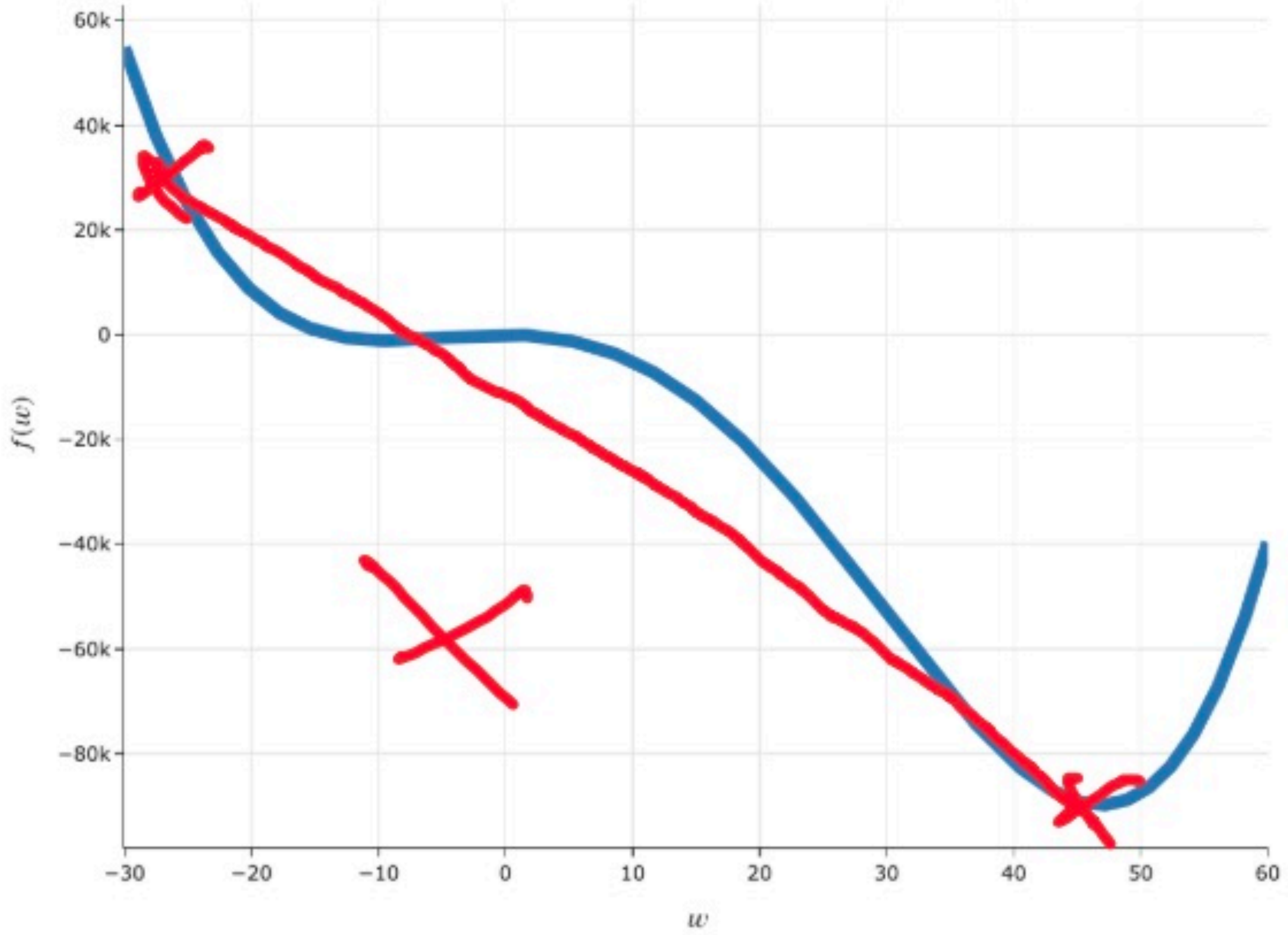


A **non-convex** function ❌.

Intuitive definition of convexity



A **convex** function ✅.

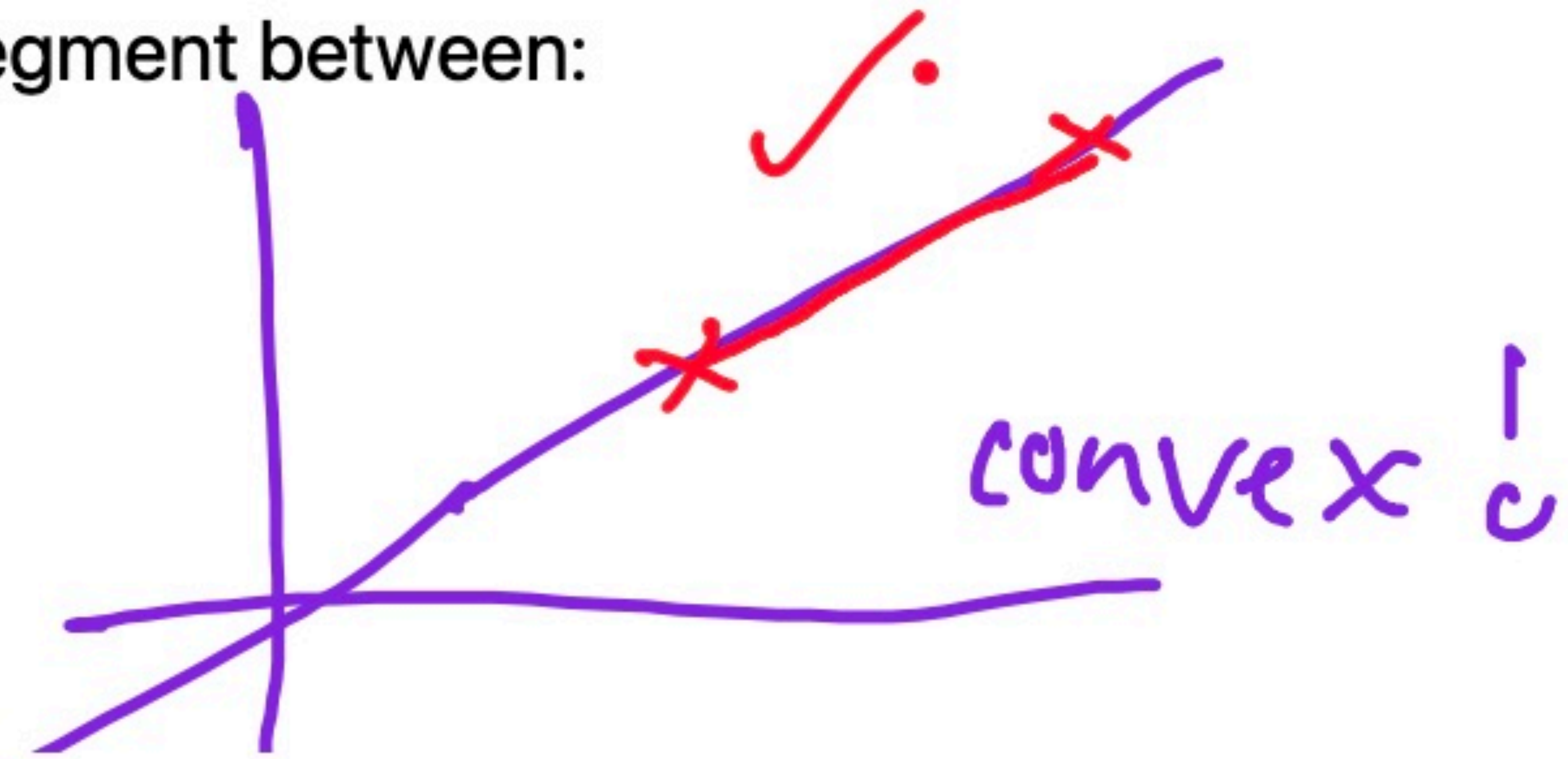


A **non-convex** function ❌.

- 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.



doesn't apply to every function!

not differentiable

Second derivative test for convexity

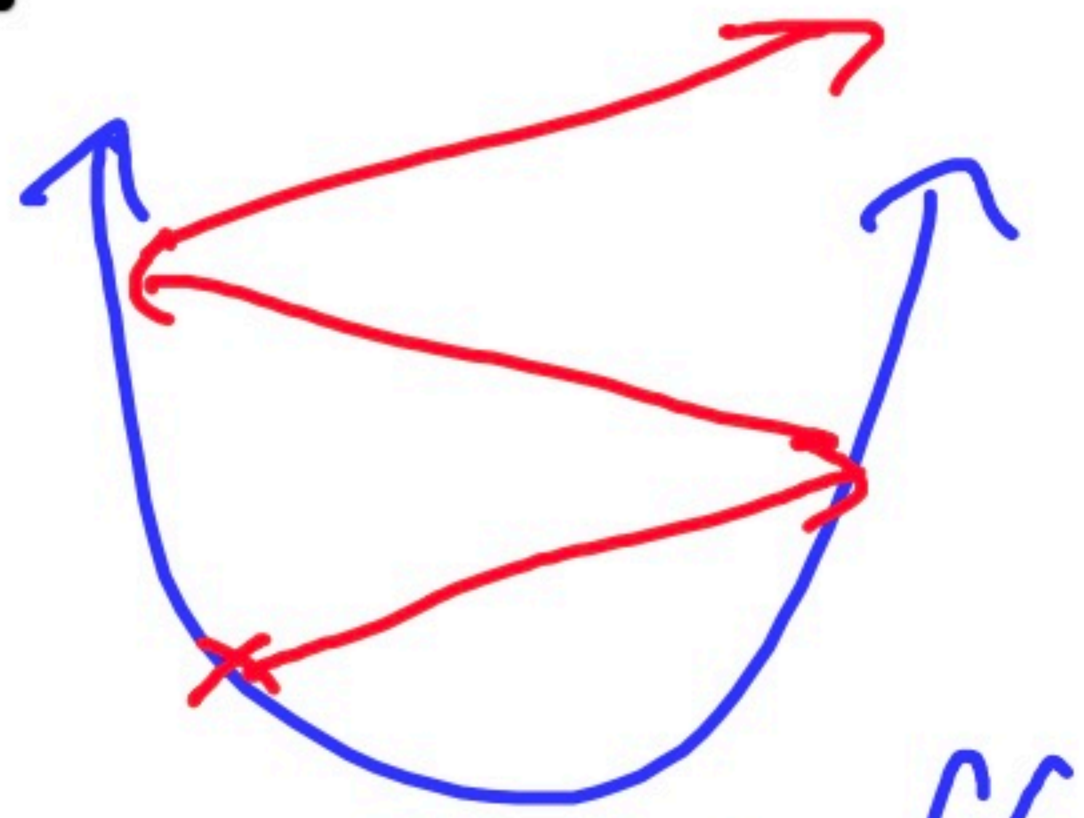
- If $f(w)$ is a function of a single variable and is **twice** differentiable, then $f(w)$ is convex **if and only if**:

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

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

$$\frac{df}{dw} = 4w^3 \rightarrow \frac{d^2 f}{dw^2} = 12w^2 \geq 0$$

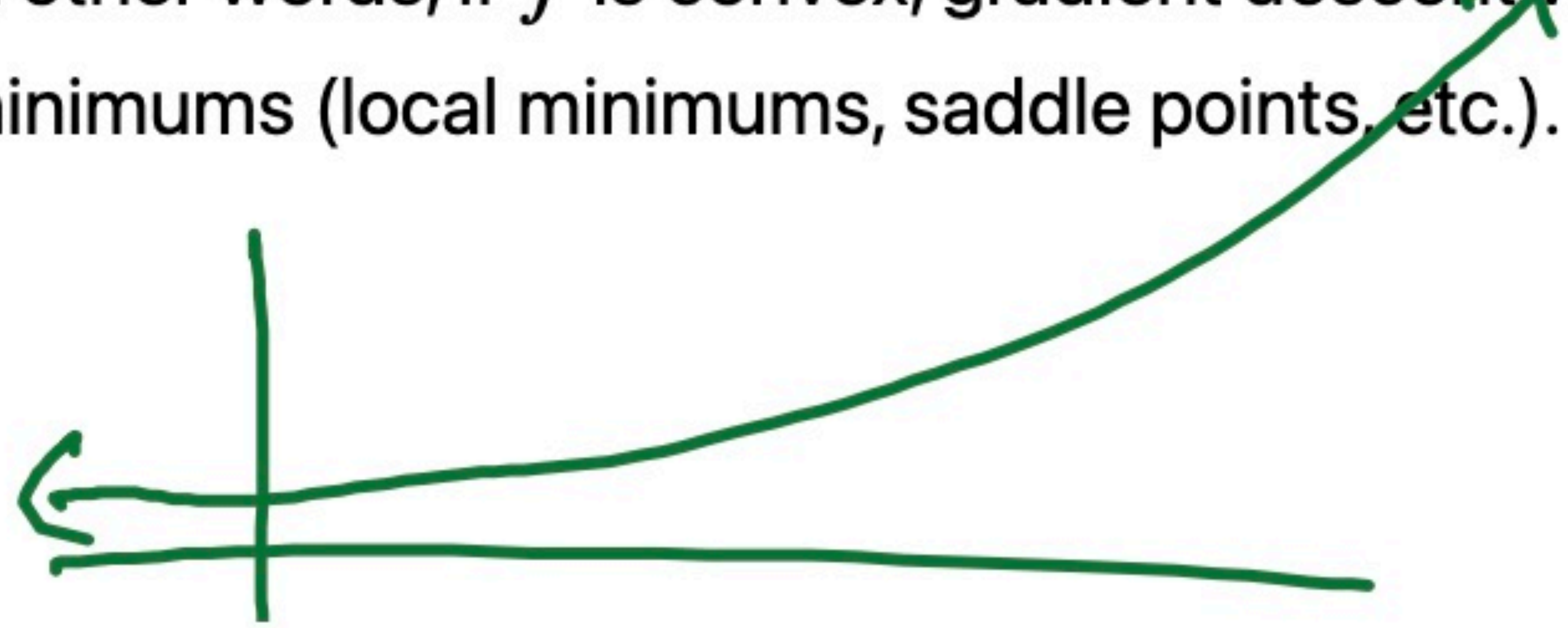
?

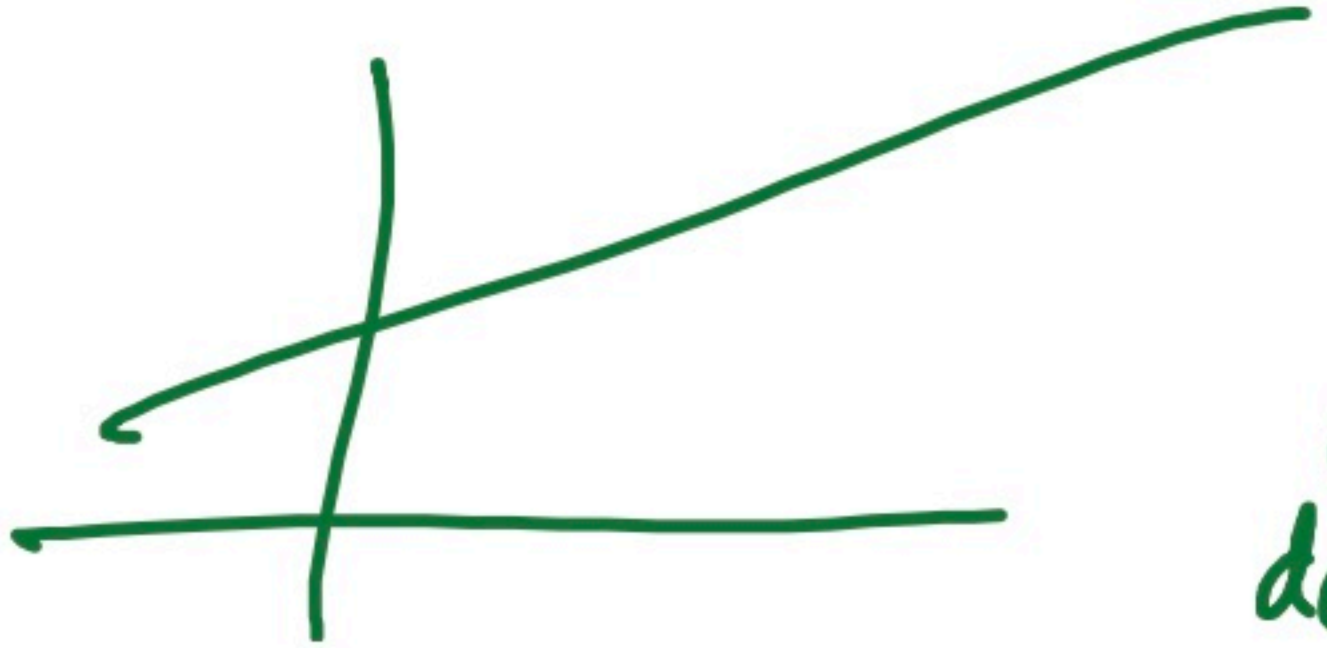


e.g. $f(w) = w^2$

Why does convexity matter?

- Convex functions are (relatively) easy to minimize with gradient descent.
- **Theorem**: If $f(w)$ 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.).





convex, but no derivative of 0.

$$\frac{\partial f}{\partial w_1} = 3 \cdot \cos(2w_1) \cdot 2 \cdot \cos(2w_2) + 2w_1$$

$$= 6\cos(2w_1)\cos(2w_2) + 2w_1$$

Minimizing functions of multiple variables

- Consider the function:

$$f(w_1, w_2) = 3\sin(2w_1)\cos(2w_2) + w_1^2 + w_2^2$$

- It has two **partial derivatives**: $\frac{\partial f}{\partial w_1}$ and $\frac{\partial f}{\partial w_2}$.

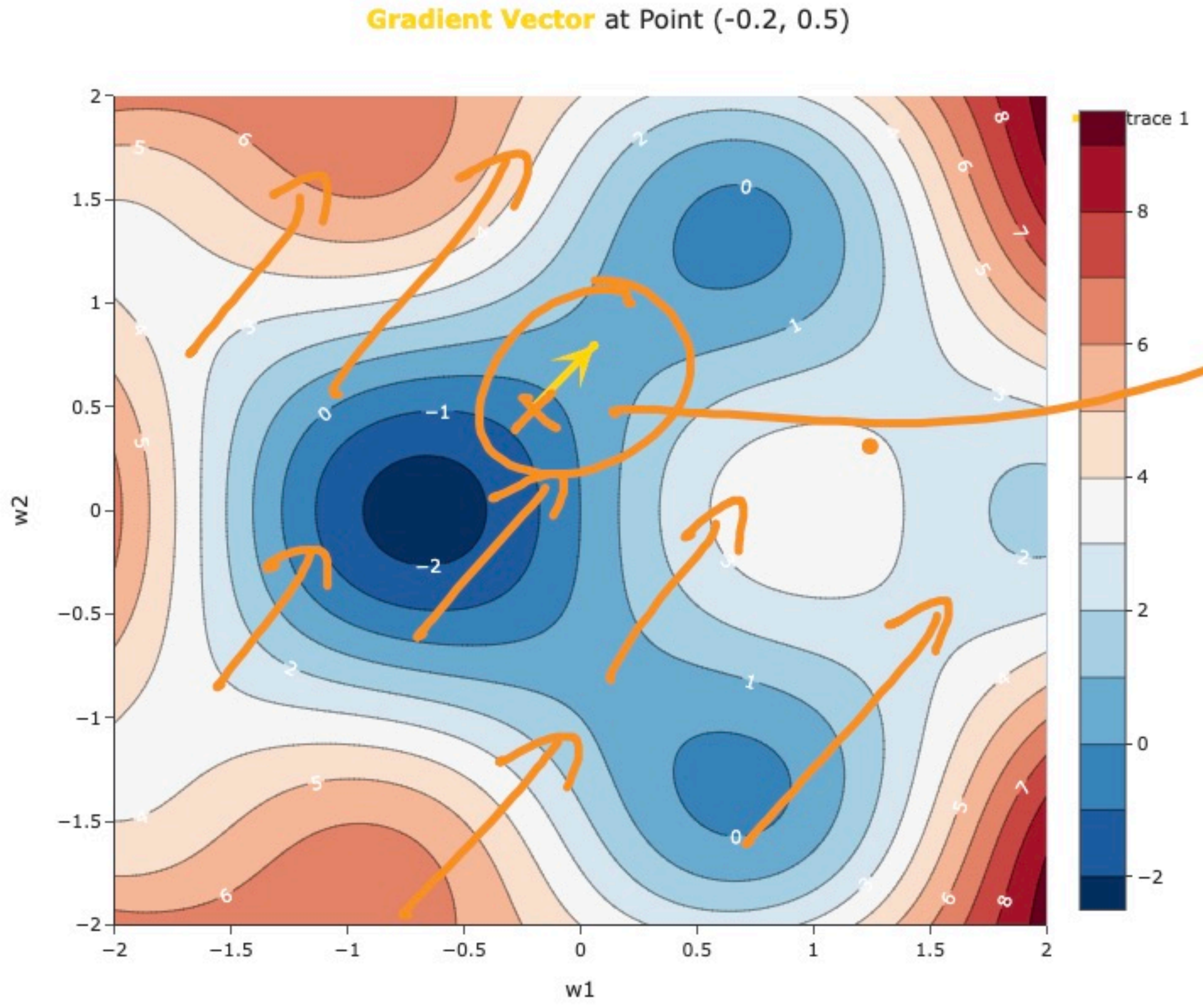
See the annotated slides for what they are and how we find them.

$$\frac{\partial f}{\partial w_2} = -6\sin(2w_1)\sin(2w_2) + 2w_2$$

$$f(\vec{w}) = f(w_1, w_2) = 3\sin(2w_1)\cos(2w_2) + w_1^2 + w_2^2$$

$$\nabla f(\vec{w}) = \begin{bmatrix} 6\cos(2w_1)\cos(2w_2) + 2w_1 \\ -6\sin(2w_1)\sin(2w_2) + 2w_2 \end{bmatrix}$$

In [24]:
```
util.make_3D_contour(with_gradient=True, w1_start=-0.2, w2_start=0.5)
```



to draw arrow, we plugged in $w_1 = -0.2$ and $w_2 = 0.5$ into

$$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

Gradient descent for functions of multiple variables

- Example:

$$f(\vec{w}) = f(w_1, w_2) = 3\sin(2w_1)\cos(2w_2) + w_1^2 + w_2^2$$

$$\nabla f(\vec{w}) = \begin{bmatrix} 6\cos(2w_1)\cos(2w_2) + 2w_1 \\ -6\sin(2w_1)\sin(2w_2) + 2w_2 \end{bmatrix}$$

- The global minimizer* of f is a vector, $\vec{w}^* = \begin{bmatrix} w_1^* \\ w_2^* \end{bmatrix}$.

*If one exists.

- We start with an initial guess, $\vec{w}^{(0)}$, and step size α, and update our guesses using:

$$\vec{w}^{(t+1)} = \vec{w}^{(t)} - \alpha \nabla f(\vec{w}^{(t)})$$

general gradient descent update rule!

Example: Gradient descent for simple linear regression

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

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

- 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 $\vec{w}^* = \begin{bmatrix} w_0^* \\ w_1^* \end{bmatrix}$, we *could* use gradient descent!

- Why would we, when closed-form solutions exist?

$$\underbrace{X^T X \vec{w} = X^T \vec{y}}$$

$\Rightarrow$ solving system of $d+1$ equations, $d+1$ unknowns