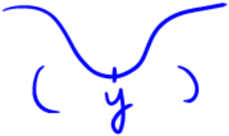
DSC 40A

Theoretical Foundations of Data Science I

Last Time: UCSD Loss

- We invented a new loss function that treated all outliers roughly the same:

$$L_{\text{ucsd}}(h, y) = 1 - e^{-(h-y)^2/\sigma^2}$$



- Our goal was to minimize the empirical risk:

$$R_{\text{ucsd}}(h) = \frac{1}{n} \sum_{i=1}^{n} L_{\text{ucsd}}(h, y_i)$$

- $R_{\text{ucsd}}(h)$ was differentiable, but we **couldn't solve** for the minimizer.

In This Video

We'll invent a general algorithm called **gradient descent** for minimizing a differentiable function like $R_{\text{ucsd}}(h)$.

Recommended Reading

Course Notes: Chapter 1, Section 3

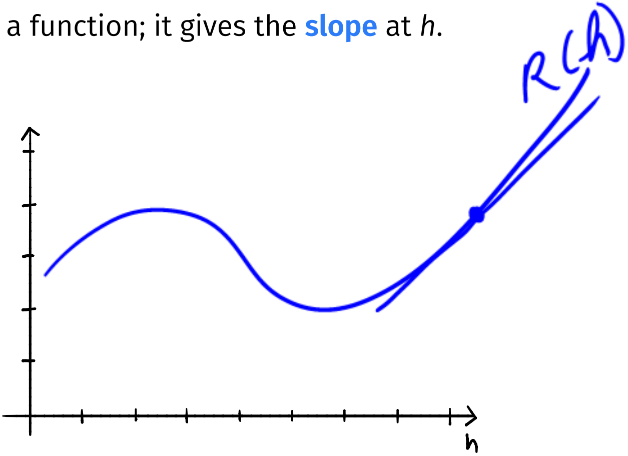
The General Problem

- **Given:** a differentiable function $R(h)$
- **Goal:** find the input h^* that minimizes $R(h)$

Meaning of the Derivative

- We're trying to minimize a **differentiable** function $R(h)$. Is calculating the derivative helpful?

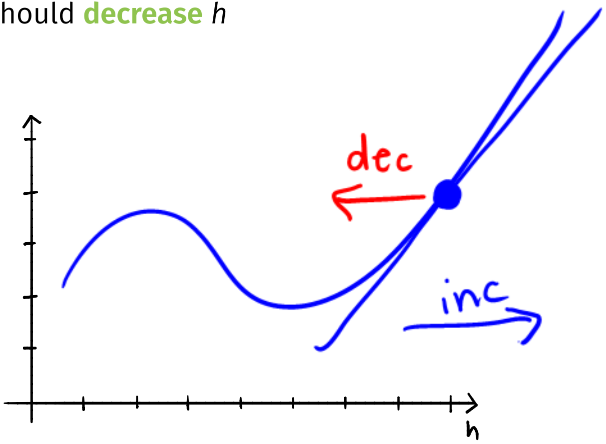
- $\frac{dR}{dh}(h)$ is a function; it gives the **slope** at h.



Key Idea Behind Gradient Descent

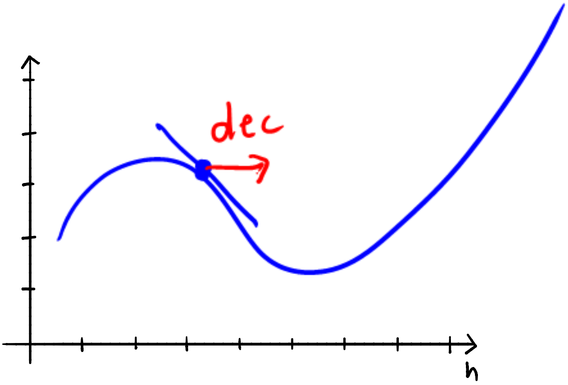
- If the slope of R at h is **positive** then moving to the **left** decreases the value of R.

- i.e., we should **decrease** h



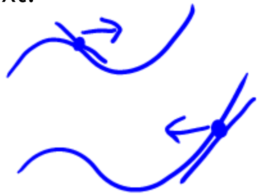
Key Idea Behind Gradient Descent

- If the slope of R at h is **negative** then moving to the **right** decreases the value of R.
- i.e., we should **increase** h



Key Idea Behind Gradient Descent

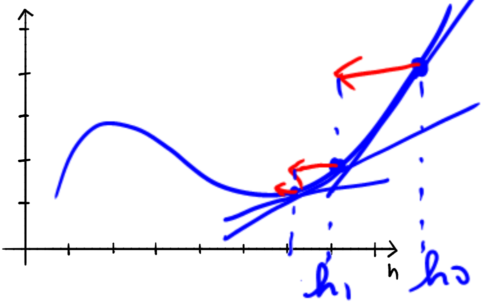
- Pick a starting place, h_0. Where do we go next?
- Slope at h_0 negative? Then increase h_0.
- Slope at h_0 positive? Then decrease h_0.
- This will work:



$$h_1 = h_0 - \underbrace{\frac{dR}{dh}(h_0)}_{\text{slope}}$$

Gradient Descent

- Pick α to be a positive number. It is the **learning rate**.
- Pick a starting prediction, h_0.
- On step i, perform update $h_i = h_{i-1} - \alpha \cdot \frac{dR}{dh}(h_{i-1})$
- Repeat until convergence (when h doesn't change much).



```python
def gradient_descent(derivative, h, alpha, tol=1e-12):
    """Minimize using gradient descent."""
    while True:
        h_next = h - alpha * derivative(h)
        if abs(h_next - h) < tol:
            break
        h = h_next
    return h
```

Example: Minimizing Mean Squared Error

- Recall the mean squared error and its derivative:

deriv

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

Question

Let $y_1 = -4, \quad y_2 = -2, \quad y_3 = 2, \quad y_4 = 4.$

Pick $h_0 = 4$ and $\alpha = 1/4$. What is h_1?

a) -1
b) 0
c) 1
d) 2

Solution

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

Data values are $-4, -2, 2, 4$. Pick $h_0 = 4$ and $\alpha = 1/4$. Find h_1.

$$h_1 = h_0 - \alpha \cdot \frac{dR}{dh}(h_0)$$
$$= 4 - \frac{1}{4} \cdot (8)$$
$$= 4 - 2$$
$$= 2$$

$$\frac{dR}{dh}(h_0) = \frac{2}{4} * (8 + 6 + 2 + 0)$$
$$= \frac{1}{2} * 16$$
$$= 8$$

Summary

- We invented **gradient descent**, which repeatedly updates our prediction by moving in the opposite direction of the derivative.

- **Next Time:** We'll look at gradient descent in action.