

Goal: minimize the cost function $J(\theta) = \min_{w,b} J(w, b)$

We need to have some function $J(w, b) \rightarrow$

For linear regression or
any function

$$\min_{w_1, \dots, w_n, b} J(w_1, w_2, \dots, w_n, b)$$

Outline:

- Start with some w, b (set $w=0, b=0$)
 - keep changing w, b to reduce $J(w, b)$
- Until we settle at or near a minimum
may have > 1 minimum

Implementation of Gradient descent algorithm

We want to repeat until it converge

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

$$b = b - \alpha \frac{d}{db} J(w, b)$$

where:

α : Learning rate ($0, 1$)

$\frac{d}{dw} J(w, b)$: Derivative term of
of the cost function J

Simultaneously
update w and b

Simultaneously
Correct way to: update w and b

$$\text{temp_}w = w - \alpha \frac{d}{dw} J(w, b)$$

$$\text{temp_}b = b - \alpha \frac{d}{db} J(w, b)$$

$$w = \text{temp_}w$$

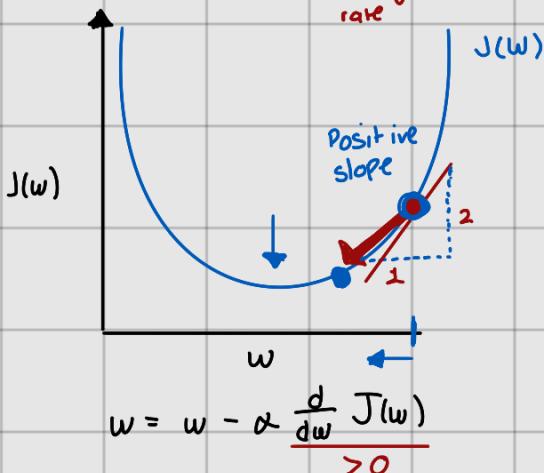
$$b = \text{temp_}b$$

Intuition

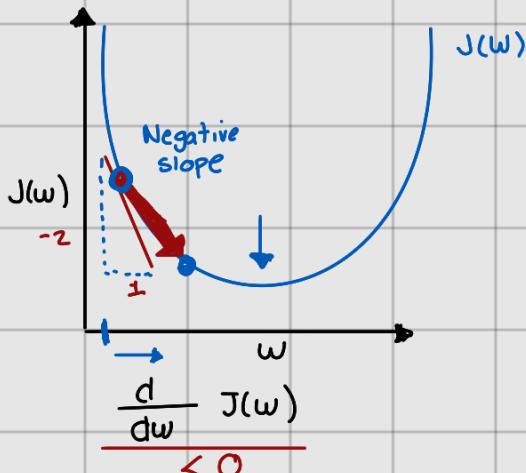
repeat until convergence {

$$\left. \begin{array}{l} w = w - \alpha \frac{\partial}{\partial w} J(w, b) \\ b = b - \alpha \frac{\partial}{\partial b} J(w, b) \end{array} \right\}$$

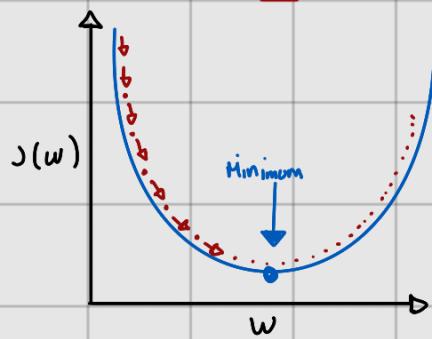
Learning rate



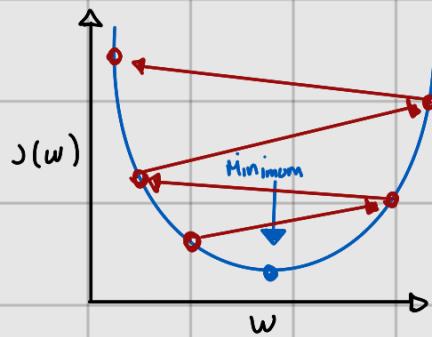
$$J(w)$$
 $w = w - \alpha \frac{d}{dw} J(w)$



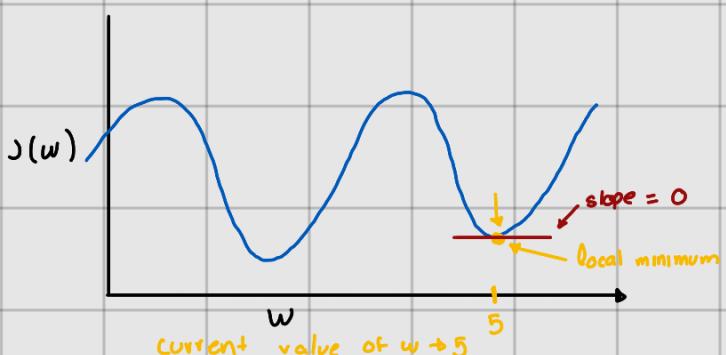
$$w = w - \alpha \frac{d}{dw} J(w)$$



- If α is too small, gradient may be slow



- If α is +∞ large, gradient descent may:
 - Overshot, never reach minimum
 - Fail to converge, diverge



If the slope is 0, then

$$w = w - \alpha \frac{d}{dw} J(w) = 0$$

$$w = w - \alpha \cdot 0$$

$$w = w$$

$$w = 5$$

$$w = 5 \quad \text{it's the same current value of } w$$

Near a local minimum

- Derivative become smaller
- Update steps become smaller

Can reach minimum without decreasing learning rate.