ML – Gradient Descent

draft

Gradient descent

- An algorithm used to find the minimum value of cost.
  - It works for any function, not only just the cost function of linear regression that we discussed so far.
- Used in almost all ML modes e.g., regression, NN, Deep learning.

What we want to do with cost?

Have some function $J(w, b)$ for linear regression or any function

Want $\min\limits_{w,b} J(w, b)$ $\min\limits_{w_1, \ldots, w_n, b} J(w_1, w_2, \ldots, w_n, b)$

Outline:

Start with some w, b (set $w=0$, $b=0$)

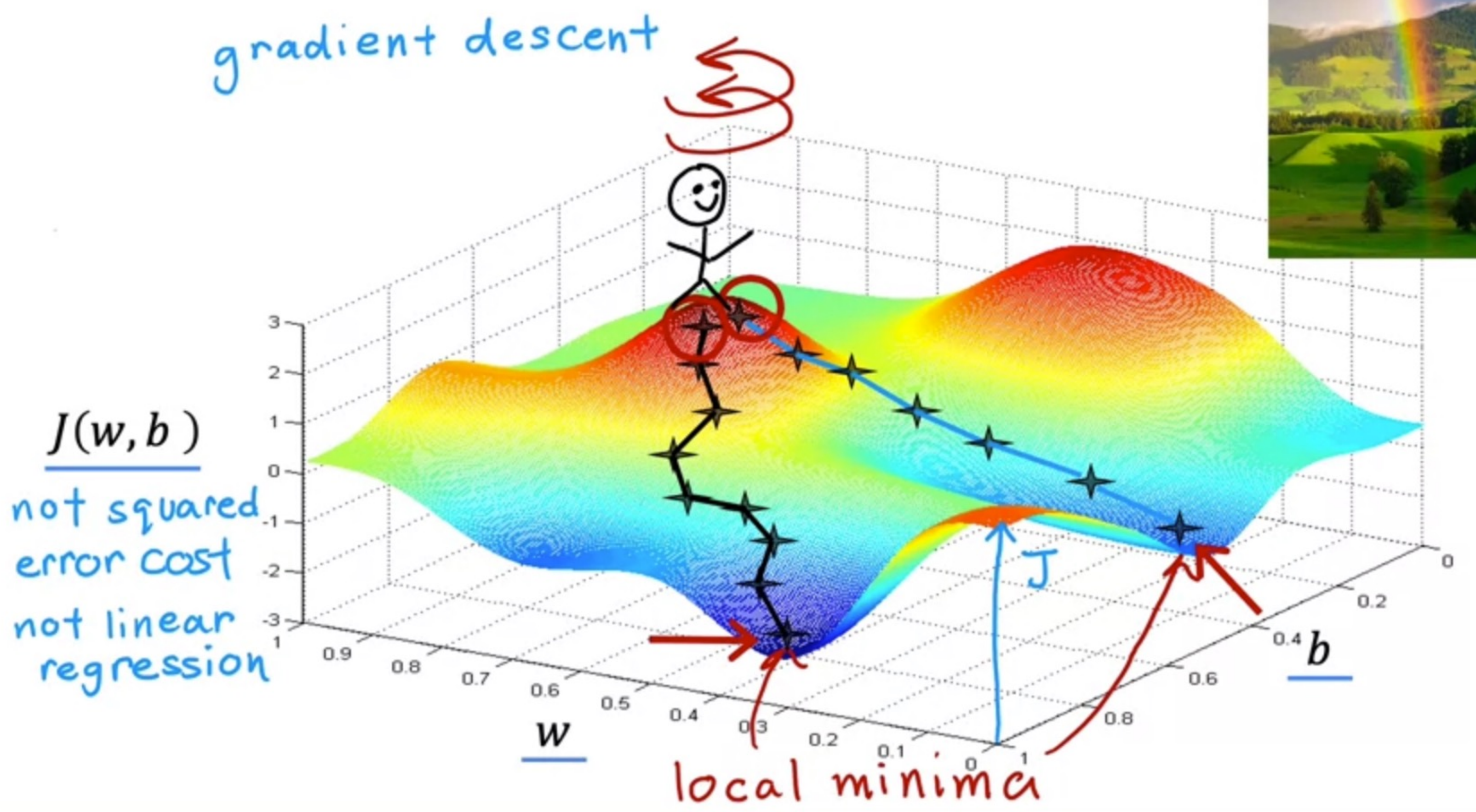
Keep changing w, b to reduce $J(w, b)$

Until we settle at or near a minimum

may have >1 minimum

J not always

Gradient Descent



Gradient descent algorithm

Repeat until convergence

$$\begin{cases} w = w - \alpha \boxed{\frac{\partial}{\partial w} J(w,b)} \\ b = b - \alpha \frac{\partial}{\partial b} J(w,b) \end{cases}$$

Learning rate

Derivative

Simultaneously update w and b

Assignment	Truth assertion
$a = c$	$a = c$
$a = a + 1$	$\cancel{a = a + 1}$
Code	Math
	a==c

Gradient descent algorithm

Repeat until convergence

$$\begin{cases} w = w - \alpha \boxed{\frac{\partial}{\partial w} J(w,b)} \\ b = b - \alpha \frac{\partial}{\partial b} J(w,b) \end{cases}$$

Learning rate

Derivative

Simultaneously update w and b

Assignment	Truth assertion
$a = c$	$a = c$
$a = a + 1$	$\cancel{a = a + 1}$
Code	Math
	a==c

Correct: Simultaneous update

$$tmp\_w = w - \alpha \frac{\partial}{\partial w} J(w,b)$$

$$tmp\_b = b - \alpha \frac{\partial}{\partial b} J(w,b)$$

$$w = tmp\_w$$

$$b = tmp\_b$$

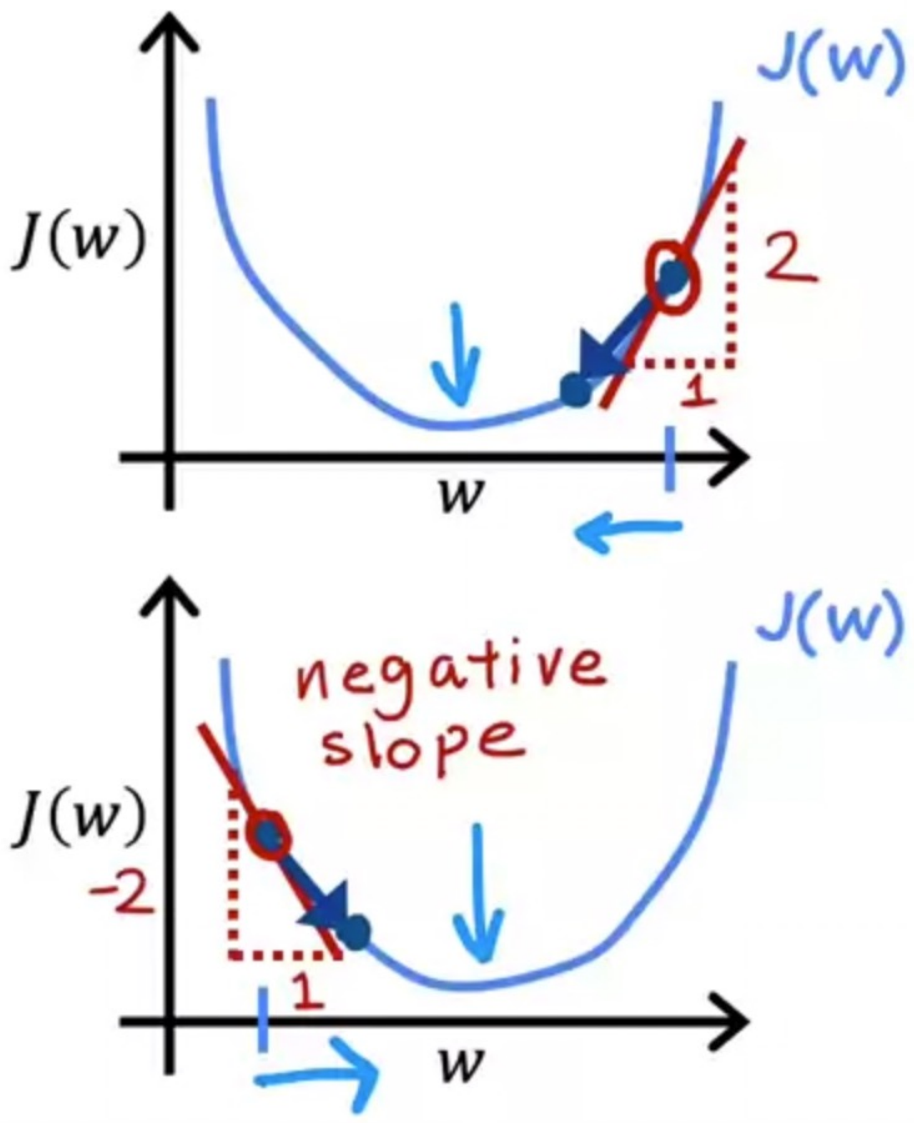
Incorrect

$$tmp\_w = w - \alpha \frac{\partial}{\partial w} J(w,b)$$

$$w = tmp\_w$$

$$tmp\_b = b - \alpha \frac{\partial}{\partial b} J(w,b)$$

$$b = tmp\_b$$



$$w = w - \alpha \underbrace{\boxed{\frac{d}{dw}J(w)}}_{>0}$$

$$w = w - \alpha \cdot (\text{positive number})$$

$$\underline{\frac{d}{dw}J(w)} \quad < 0$$

$$w = w - \alpha \cdot (\text{negative number})$$

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

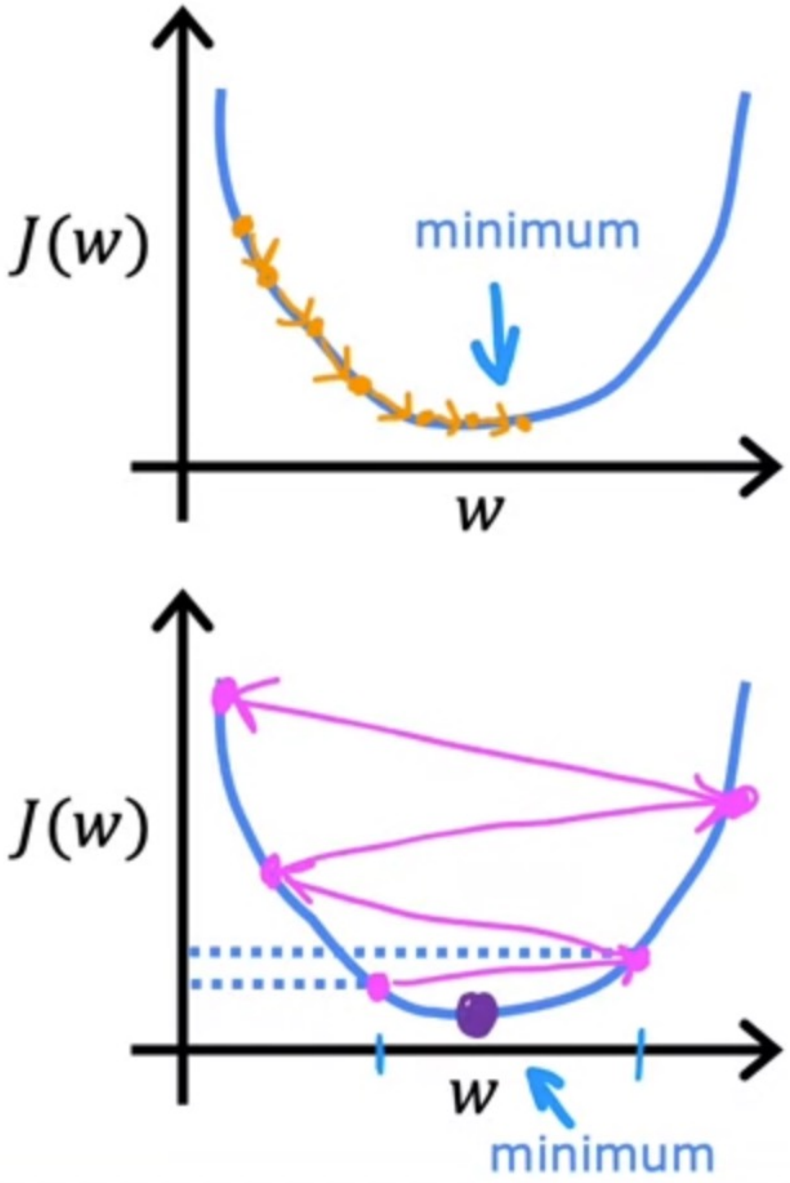
If α is too small...

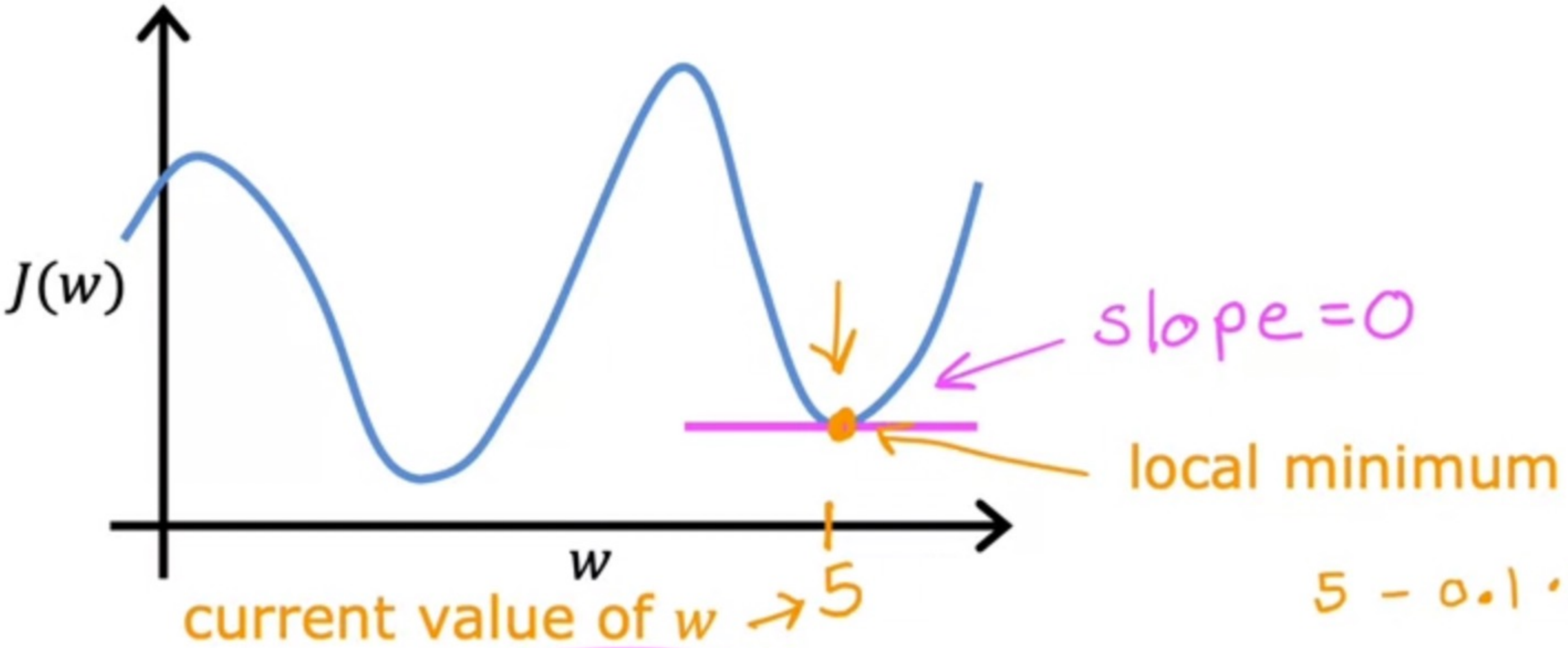
Gradient descent may be slow.

If α is too large...

Gradient descent may:

- Overshoot, never reach minimum
- Fail to converge, diverge





$$w = w - \underset{0.1}{\alpha}\underbrace{\boxed{\frac{d}{dw}J(w)}}_{=0}$$

$$5 - 0.1 \cdot 0$$

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

$$\overset{5}{w} = \overset{5}{w}$$

Can reach local minimum with fixed learning rate

α

smaller

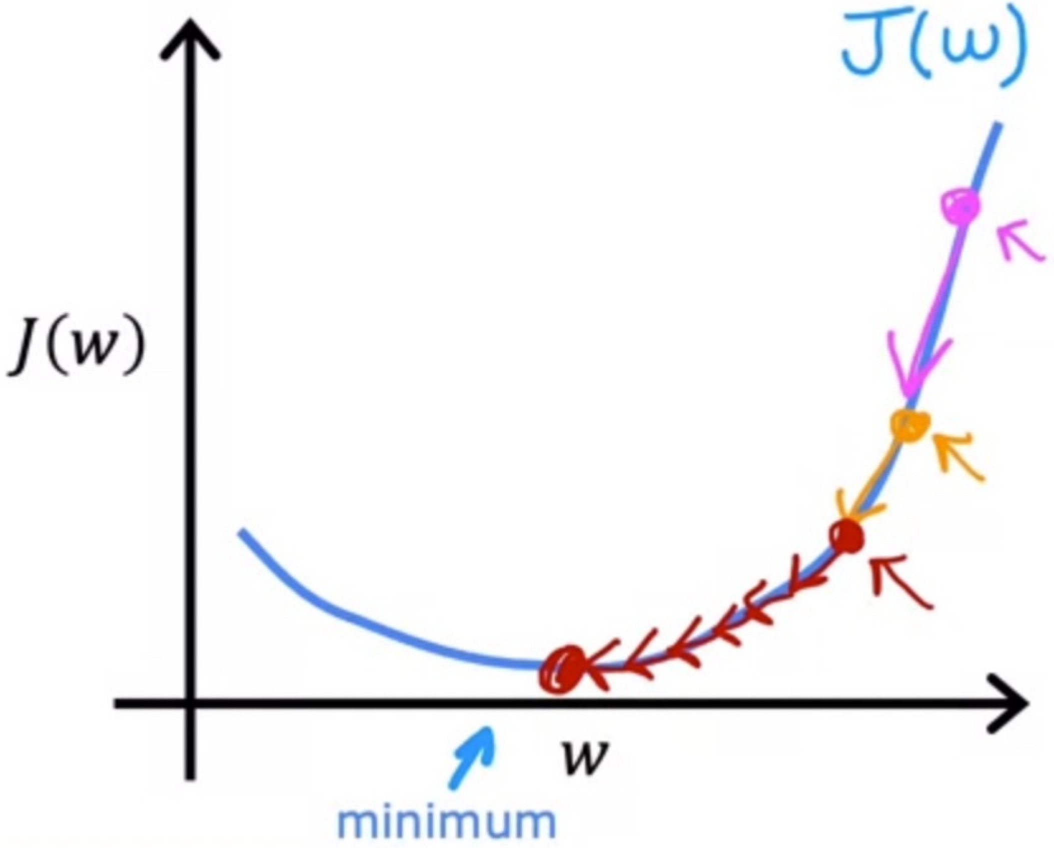
not as large

large

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

Near a local minimum,
- Derivative becomes smaller
- Update steps become smaller

Can reach minimum without decreasing learning rate α



Derived using calculus

Linear regression model

$$f_{w,b}(x) = wx + b$$

Cost function

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})^2$$

Gradient descent algorithm

repeat until convergence {

$$w = w - \alpha \frac{\partial}{\partial w}J(w,b) \rightarrow \frac{1}{m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})x^{(i)}$$

$$b = b - \alpha \frac{\partial}{\partial b}J(w,b) \rightarrow \frac{1}{m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})$$

}

Derivative

$$\frac{\partial}{\partial w} J(w,b) = \frac{d}{dw} \frac{1}{2m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2 = \frac{d}{dw} \frac{1}{2m} \sum_{i=1}^{m} \left(wx^{(i)} + b - y^{(i)} \right)^2$$

$$= \frac{1}{\cancel{2}m} \sum_{i=1}^{m} \left(wx^{(i)} + b - y^{(i)} \right) \cancel{2} x^{(i)} = \boxed{\frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right) x^{(i)}}$$

$$\frac{\partial}{\partial b} J(w,b) = \frac{d}{db} \frac{1}{2m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2 = \frac{d}{db} \frac{1}{2m} \sum_{i=1}^{m} \left(wx^{(i)} + b - y^{(i)} \right)^2$$

$$= \frac{1}{\cancel{2}m} \sum_{i=1}^{m} \left(wx^{(i)} + b - y^{(i)} \right) \cancel{2} = \boxed{\frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)}$$

no $x^{(i)}$

Gradient descent algorithm

$\frac{\partial}{\partial w} J(w,b)$

repeat until convergence {

$$w = w - \alpha \boxed{\frac{1}{m} \sum_{i=1}^{m} \left(\underline{f_{w,b}\left(x^{(i)}\right)} - y^{(i)}\right) x^{(i)}}$$

$$b = b - \alpha \boxed{\frac{1}{m} \sum_{i=1}^{m} \left(\underline{f_{w,b}\left(x^{(i)}\right)} - y^{(i)}\right)}$$

}

$\frac{\partial}{\partial b} J(w,b)$

Update
w and b
simultaneously

$f_{w,b}(x^{(i)}) = wx^{(i)} + b$

squared error cost