Gradient Descent:

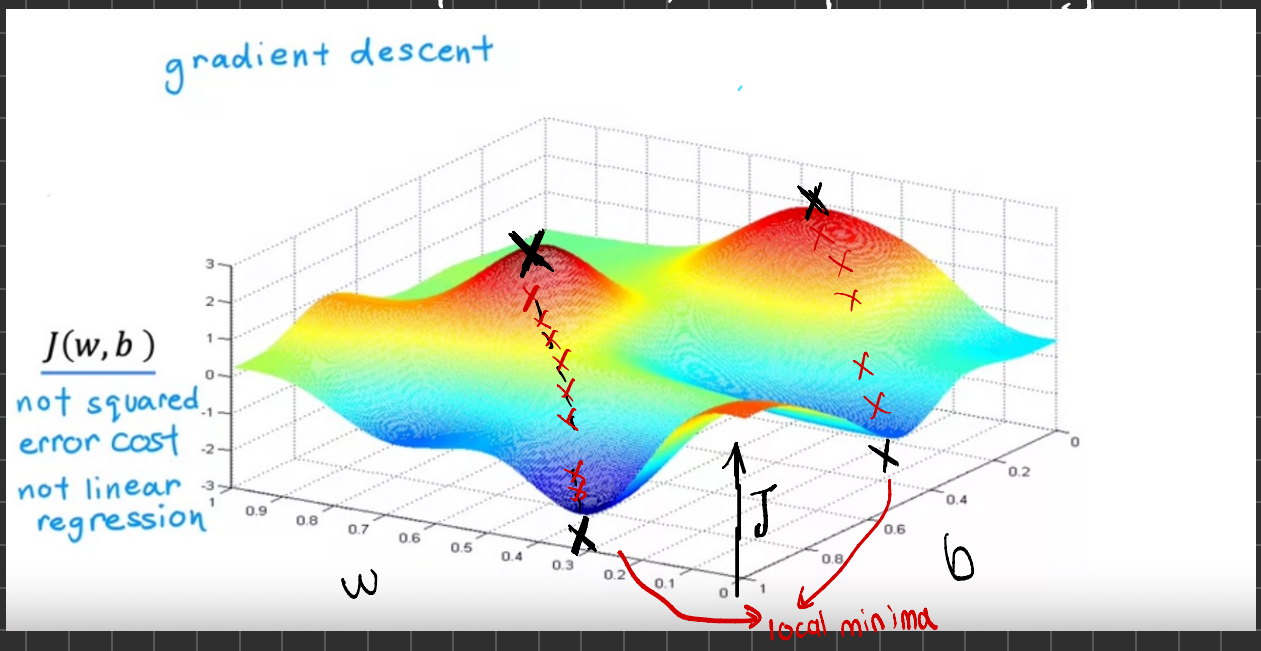
→ Minimize any function.

→ More general, multiple parameters.

→ Start with $w=0, m=0$. Keep reducing.

→ Since there can be multiple minimums, the shape can vary.

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



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

α → learning rate

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

Correct: Simultaneous update

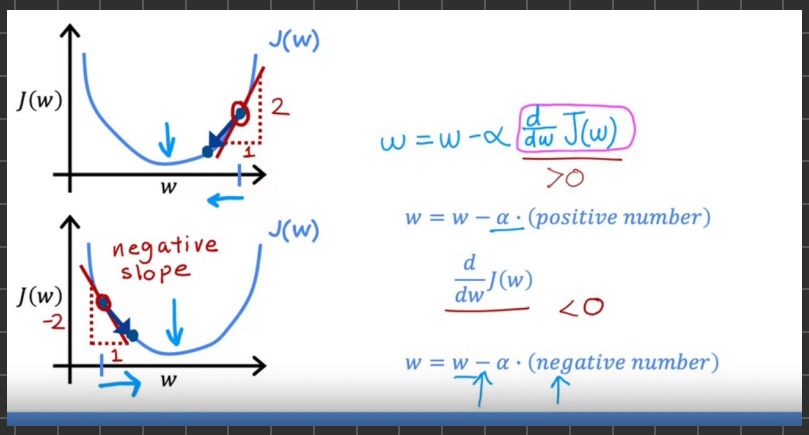
$$tmp_w = w - \alpha \frac{\partial}{\partial w} J(w,b)$$

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

$$w = tmp_w$$

$$b = tmp_b$$

Example



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

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

$$\underbrace{\frac{d}{dw} J(w)}_{<0}$$

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

Note:

/ → +

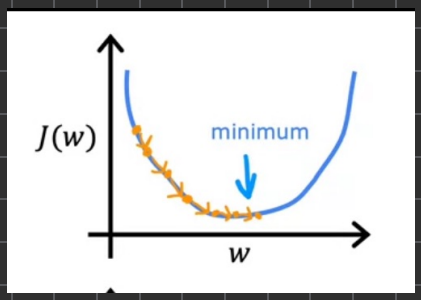
\ → −

Made with Goodnotes

Learning Rate: α

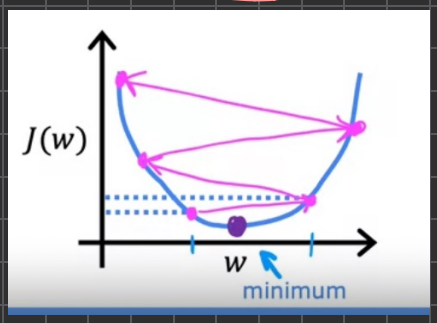
→ If α is too small:

GD may take a long time.



→ If α is too large:

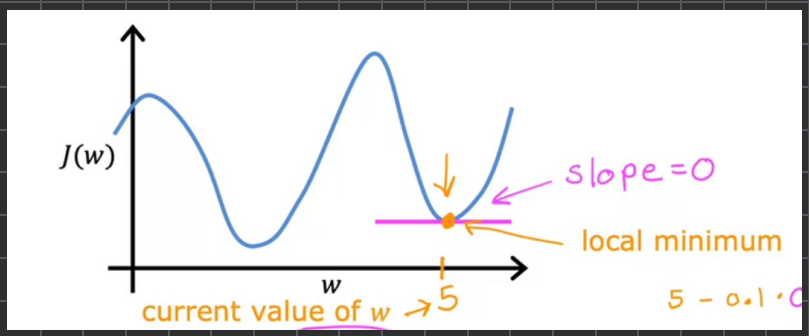
✗ GD may overshoot, never reach minimum.



Might fail to converge
||
Diverge

→ If w is already at local minimum, since slope is 0,

W remains unchanged as $w := w$, $w := w - \alpha \frac{d}{dw} J(w)$ $= 0$



→ Can reach local minimum with fixed learning rate.

→ As we go closer, slope (→ derivative) becomes less steep.
Meaning the steps get smaller and smaller.

Made with Goodnotes

Gradient Descent - Linear Regression

Linear Regression Model

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

Cost Function

$$J(w,b) = \frac{1}{2m} \sum_{i=1}^{m} \left(\underbrace{f_{w,b}(x^{(i)})}_{\text{prediction}} - \underbrace{y^{(i)}}_{\text{actual value}}\right)^2$$

Gradient Descent Algorithm:

Repeat until convergence {

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

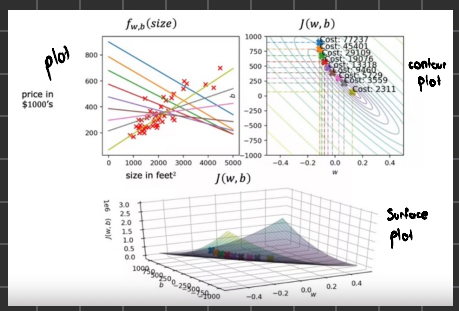
$$b = b - \alpha \frac{\partial}{\partial b} J(w,b) \longrightarrow \frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)}\right)$$

}

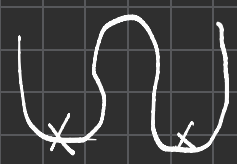
→ Linear regression's squared error cost function can not have multiple minimas. This is because it's a convex function.

↳ Can draw a straight line between two points.

Running GD



Ex



Can't draw straight line
Line not contained in same set
Concave

Convex

→ Batch GD: Model uses all the training examples in each step of GD.

Summary:

→ Linear Regression utilizes training data to fit parameters w,b by minimizing a measure of the error between our predictions $f_{w,b}(x^{(i)})$ and $y^{(i)}$

↳ Cost Function. $J(w,b)$

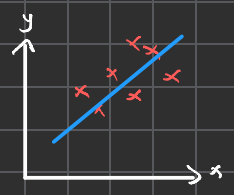
GD helps minimize the cost function.

Made with Goodnotes

Linear Regression:

→ Regression Model predicts numbers.

(x, y) = single target example.



→ x = input variable, feature

→ y = output variable, target.

$(x^{(i)}, y^{(i)})$ = i^{th} training example.

→ m = number of training examples.

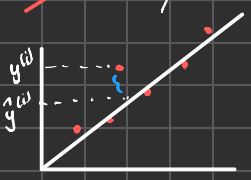
$$\boxed{f_{\vec{w},b}(\vec{x}) = \vec{w}\vec{x} + b}$$

→ w, b are parameters.

→ Linear Regression with one variable is called a univariate linear regression.

IMPORTANT

$$\hat{y}^{(i)} = f_{w,b}(x^{(i)})$$



→ Find w, b : $\hat{y}^{(i)}$ is close to $y^{(i)}$ for all $(x^{(i)}, y^{(i)})$

Cost Function:

Cost function measures how well a line fits our function.

→ $$J(w,b) = \frac{1}{2m}\sum_{m}^{1}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$$ → Squared error cost function.

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

Summary:

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

Parameters : w, b

Made with Goodnotes