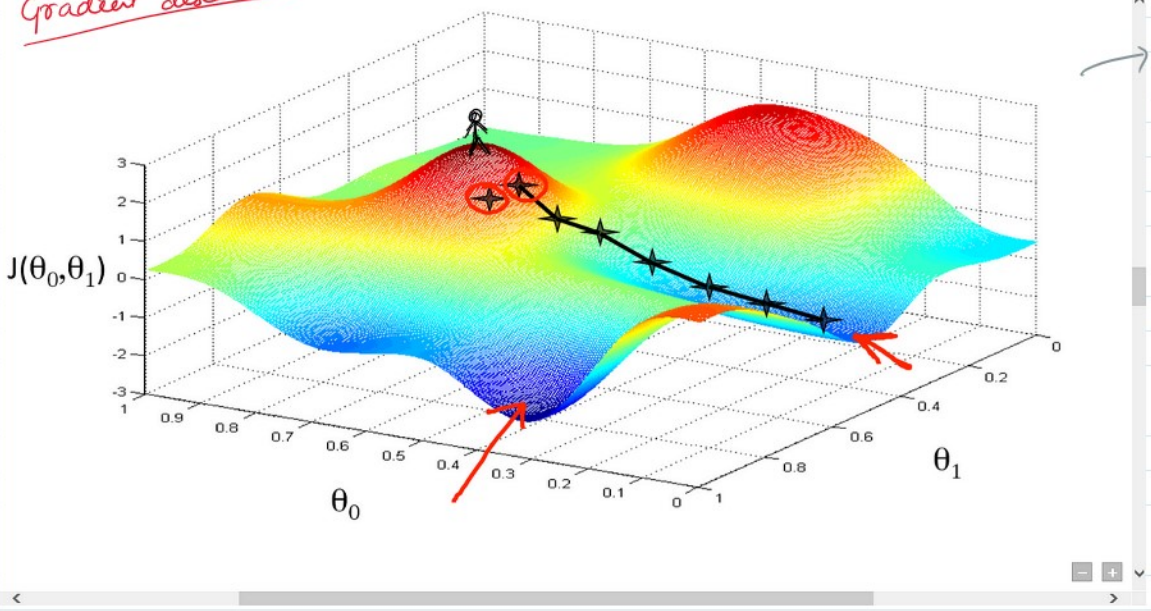
Gradient descendent:

Monday, August 28, 2023 11:44 AM

Gradient descendent :- WE USE GRADIENT DESCENDENT TO KEEP CHANGING OR ADJUSTING THE PARAMETER J(W,B) TO MINIMIZSE THE "ERROR" OR "COST FUNCTION".

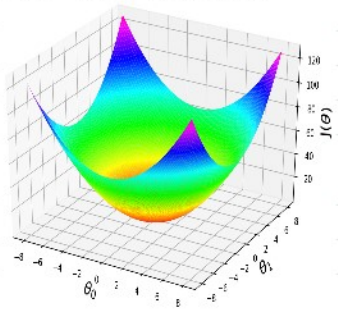
→ This helps the model get better making at predictions, classifications or other tasks based on the given data
→ We keep changing the parameters (w,b) to reduced J(w,b) until we settle at or near a minimum.

Gradient descendent



→ this type of Cost function you only get when you train a neural network.

→ Above function is not a squared error cost
For linear regression with Squared Error Cost function, you will always end up with graph of hammock shape like this.



✱ Gradient Descendent Algorithm ✱

$$w = w - \alpha \frac{\partial J(w,b)}{\partial w} \rightarrow ①$$

→ derivative

α → learning rate, small +ve number which 0 and 1.
- → What α basically does ?
- → it basically controls how big of a step you take downhill.
- → if α is big then it will be agressive steps downhill.
- → " " " small " " " " small baby steps downhill.

$$b = b - \alpha \frac{d}{db} J(w,b) \rightarrow ②$$

✱ we will be Constantly Repeat both the function until Convergence.
→ we will be keep updating the parameters w and b.

✱ Correct Simultaneous update ✱

✤ Correct Simultaneous update ✤

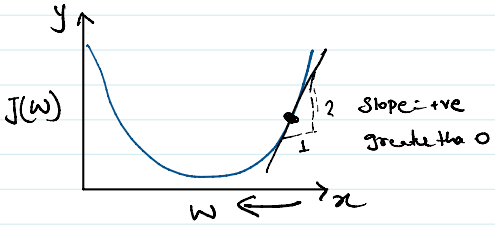
$tmp\_w = w - \alpha \frac{\partial}{\partial w} J(w,b)$

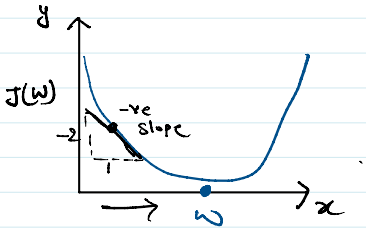
$tmp\_b = b - \alpha \frac{\partial}{\partial b} J(w,b)$

$w = tmp\_w$

$b = tmp\_b$

✤ Gradient descendent Intuitions ✤





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

$w = w - \alpha \cdot (+ve\ number)$

✤ you will moving to left while performing this. and you decreasing cost of J(w) and you are getting closed to minima.

$\frac{dJ(w)}{dw}$

$w = w - \alpha (negative\ number)$

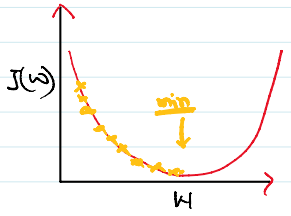
⇒ you will end up increasing w which mean you moving to the right to the graph and your cost J decreasing which gets you closer to minimum.

✤ learning Rate (α) ✤

✤ Two cases of learing Rate:-

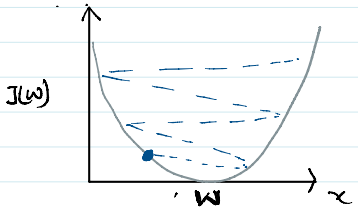
1. If α is too small.
2. If α is too big.

✤ If learing rate is small (α)



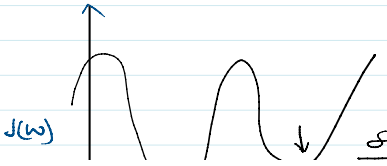
⇒ If the learing rate is small then the G.T will work but it will be slow. because here tiny steps takes long time to get to the minima.

✤ If α is too larg (learing rate).



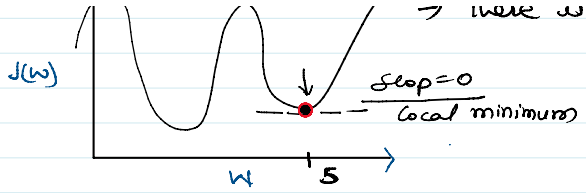
⇒ If α is large then G.T may overshoot and may never reach the minimum.

⇒ Fail to Converge and may even diverge



⇒ There is two local minima so, this isn't Squared error Cost Function.

→ there is two local minima so, this isn't Squared error Cost Function.



What IF Parameter W is already at local minima.

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

$$W = W - \alpha (0)$$

$\Rightarrow W = W \Rightarrow$ unchanged

Can reach local minimum with fixed learning rate.

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

Near a local minimum.

→ Derivative becomes smaller.
→ update steps become small.

Gradient descent For linear Regression

* linear regression model

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

* Cost Function

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

* Gradient descent algorithm.

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

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

→ If will use these function instead of derivative term it will work as well

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

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

$$\Rightarrow \frac{\partial}{\partial w} \frac{1}{2m} \sum_{i=1}^{m} \left(wx^{(i)} + y - y^{(i)}\right) / 2(x^{(i)})$$

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

* Derivative of $\frac{d}{db} J(w,b)$

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

$$\Rightarrow \frac{\partial}{\partial b} \frac{1}{2m} \sum_{i=1}^{m} \left(wx^{(i)} + y - y^{(i)}\right) / 2$$

$$\Rightarrow \frac{\partial}{\partial b} \frac{1}{2m} \sum_{i=1}^{m} \left(wx^{(i)} + y - y^{(i)} \right)^2$$

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

Batch $\Rightarrow$ Each step of gradient descent uses all the training examples.