5. Gradient Descent.

Have some function $J(w,b)$

Want $\min_{w,b} J(w,b)$.

We use Gradient Descent to minimize any cost function.

Our objective is to minimize.

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

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.

J May have more than 1 possible minimums.

An example can be given below, we see an example in 3D.

We can see that it has more than 1 minimum values.
We want to take a decision of steepest descent. and we take this step everytime. Slowly, we take these steps, to find ourselves at the bottom of J at a local minimum.

Depending on the initial position, we might end up on a different local minima.

I. Implementing Gradient Descent.

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

α is called a learning Rate

$0 < \alpha < 1$ → how big of a step we take.

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

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

We repeat these steps until convergence.

We need to simultaneously update both w and b.

Correct way:

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

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

$w = \text{temp\_w}$; $b = \text{temp\_b}$.

Incorrect way:

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

$w = \text{temp\_w}$

$$\text{temp\_b} = b - \alpha \frac{d}{db} J(w,b) \qquad b = \text{temp\_b}.$$

we are updating w first and then using it for calculating b. ☹ This is wrong.

II. Gradient Descent Intuition.

Algorithm: Repeat until convergence:

$$w = w - \alpha \frac{d}{dw} J(w,b) \qquad \alpha \rightarrow \text{Learning rate.}$$

$$0 < \alpha < 1$$

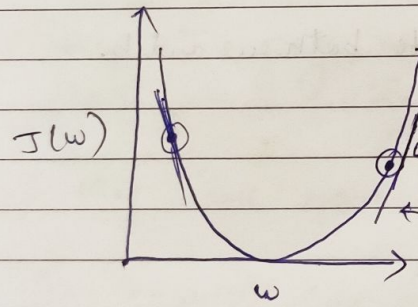
$$b = b - \alpha \frac{d}{db} J(w,b)$$

An example:

$J(w)$; if J is a number.

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

This our example, where we set $b=0$.



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

If $\frac{d}{dw} J(w) > 0$, $w = w - \alpha \times$ positive number.

so we are getting closer to min. value.

If have starting point in the left. The slope of this line will have negative slope.

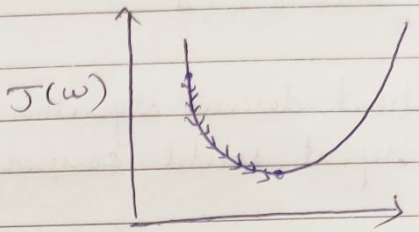
So, $w = w - \alpha \times$ (negative number).

So, subtracting a -ve no., ie adding a true number w. So, we are moving in the right direction.

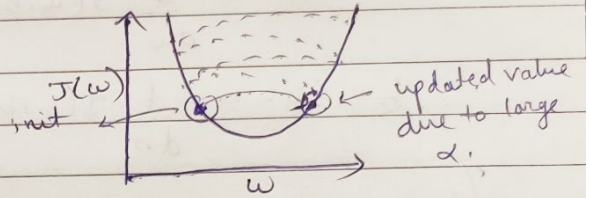
III. Learning Rate.

Now, we have $w = w - \alpha \frac{d}{dw} J(w)$.

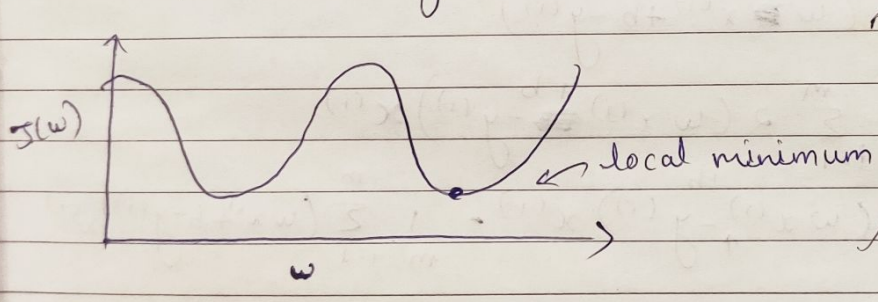
If α is too small, we take a very small step. We reach bottom but very slowly.



If α is too large,
we might keep on increasing the cost function.
Gradient descent may overshoot and never reach the minimum.
It may fail to converge, or even diverge.



- What if we already reach a local minimum?.

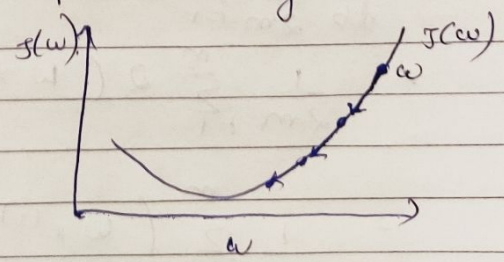


We have two local minimas here.

So, we have current value of w.
Since slope is 0, $w = w - \alpha \times 0$ → set w to w.
w value is unchanged.

Can reach local minimum with fixed learning rate (α).
~~Not α too large.~~

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



as we approach the minimum, the derivative approaches 0. Update steps becomes smaller.

IV. 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.

repeat until convergence {

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

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

}

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

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

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

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

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

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

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

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

So, the gradient descent algorithm for linear regression.

repeat until convergence {

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

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

we need to update w and b simultaneously.

So, as we already saw, $J(w,b)$ may have more than 1 local minima.

But when using squared error cost function, our J function is bowl shaped. So it will have only one local minima. It is a convex function.

Convex function is a bowl shaped function. It cannot ~~have a~~ have any local minima other than a single global minima.

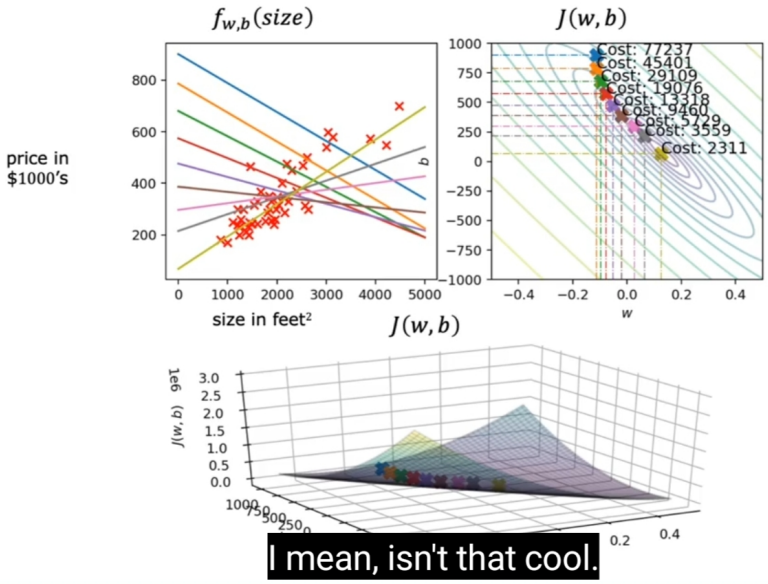
V. Running Gradient Descent.

We can have a look at our 3D plot for squared error cost function.

As we initialize it at some value, we can have a look at the figure.

This is actually called "Batch" Gradient Descent.

"Batch": Each step of gradient descent uses all the training examples.



I mean, isn't that cool.

for updating value of w and b, we are looking at the entire batch of training data.

$$\sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)})^2 \quad \{m = 47\}.$$

other gradient descent algorithms also uses subsets rather than the whole batch for training.

VI. Quiz.

1. Gradient descent is an algorithm for finding values of parameters w and b that minimize the cost function J.

repeat until convergence $\begin{cases} w = w - \alpha \frac{d}{dw} J(w,b) \\ b = b - \alpha \frac{d}{db} J(w,b). \end{cases}$

when $d/dw\ J(w,b)$ is a negative number, what happens to w after one update step?

a) w ↑ b) Not possible to tell c) w ↓
d) stays same

→ Value of w increases.

2. For linear regression, what is the update step for parameter b?

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

b) $b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{wb}(x^{(i)}) - y^{(i)}) x^{(i)}$

→ a.