Gradient Deseent: An optimization algorithm for Regression

Why we need gradient descent?

In ~~a~~ Linear Regression, we can train the models using **OLS method**. But the problem is in higher dimension, the matrix calculation becomes very complex and for that the time complexity is very high in terms of OLs.

Gradient descent is an algorithm, it takes input any differentiable function and output the minima of that function. It can be used in

- Linear Regression
- Logistic Regression
- ~~Grad~~ Deep learning

Cost Function of LR:

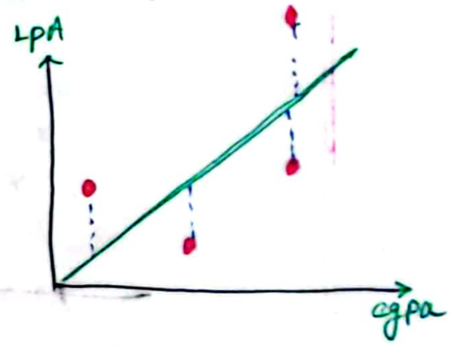
$$L = \sum_{i=1}^{n} (\cancel{x}y_i - \cancel{z}\hat{y_i})^2$$

$$= \sum_{i=1}^{n} (y_i - mx_i \cancel{+b} - b)^2 \quad \Big| \quad \hat{y_i} = mx_i + b$$

So, the changes depending on (m and b)

For eovy discussion, let us say we know the value of $m = 78.35$

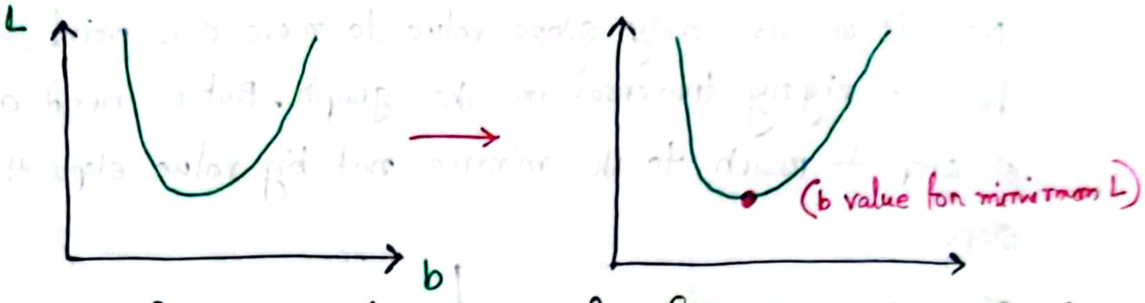
$$\therefore L = \sum_{i=}^{4} (y_i - 78.35 \cdot x_i - b)^2$$



Now, the changes only depend upon b. which is upper or lower movement of the bfl.

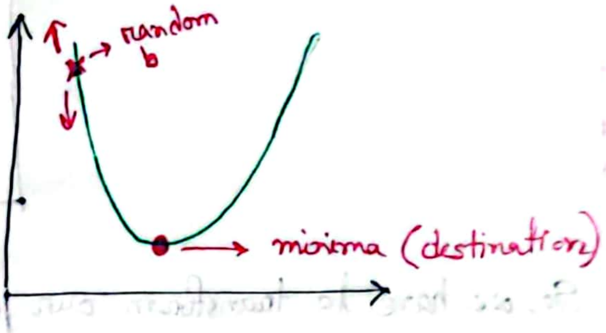
So, we need such value for b, using which we can get minimum L.

Now, if we plot L and b graph, it would be parabolic.



Here, we have to find such value for b for which will get min value for L.

Select a random value for b.



After taking a random b value it can be anywhere in the parabola Then we have to increase or decrease the value of b to reach to the minima. Whatever point you are, you will find the slope in that point.

How we calculate slope?

- We find the $\frac{d}{dx}$ of the function
- then put the x value and get the result ($x = b$ in our case)

If slope = -ve?

then increment b value

if slope = +ve?

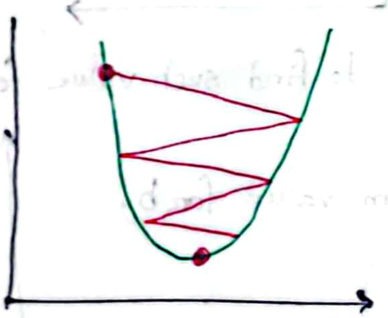
then decrement value of b

$$b_{new} = b_{old} - \text{slope}$$ (iterative)

~~Now, how would we know where to stop?~~

Now, if we use only slope value to move our point, then we will have a zigzag traversal in the graph. But we need a small amount of step to reach to the minima not big value steps that we get from slope.

If we use only slope →

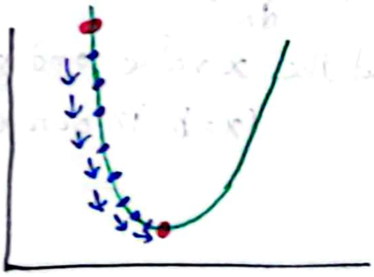


So, we have to transform our equation so that we can take small steps towards the minima.

Equation ⇒ $b_{new} = b_{old} - \eta \times slope$

↳ Learning rate

Using this we can reach to our minima value. Generally value = 0.01



Now, How would we know that when to stop?

If bnew - bold giving us really small value like → 0.00001, then we have to know that, we are not moving that much from the previous possible, the movement is like almost 0. Then we have to understand that we have found our minimum value of L for b.

Mathematical Formulation of the Algorithm:

```
→ start with a random value for b

for i in epochs:          epochs = number of iterations (100, 1000, ...)
    $b_{new} = b_{old} - \underbrace{\eta}_{0.01} \text{Slope}$
```

How to find slope?

→ $\frac{dL}{db}$ ~~for~~ function

→ put the random value in the result, and that would be the slope

$$L = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$\frac{dL}{db}\left(\sum_{i=1}^{n} (y_i - \hat{y}_i)^2\right)$$

$$\Rightarrow \frac{dL}{db}\left(\sum_{i=1}^{n} (y_i - mx_i - b)^2\right)$$

$$\Rightarrow 2\sum_{i=1}^{n} (y_i - mx_i - b)(-1)$$

$$\Rightarrow -2\sum_{i=1}^{n} (y_i - mx_i - b)$$

Suppose, random value taken for b = 0

$$\Rightarrow -2\sum_{i=1}^{n} (y_i - mx_i)$$

Previously, we assumed m value = 78.35

$$\therefore \Rightarrow -2\sum_{i=1}^{n}(y_i - 78.35x_i)$$

iteratively, we will keep calculating b_{new} and last we will find that b value for which L function value will be minimum.