

Gradient descent - 1

↳ It is an optimization algorithm used to minimize a fn, most commonly a loss fn in ML

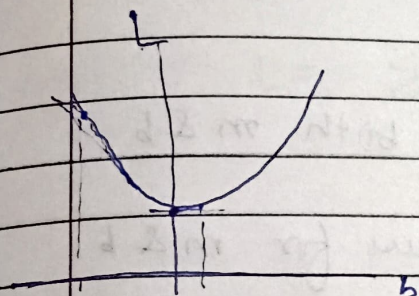
$$\rightarrow \hat{y}_i = m\hat{x}_i + b$$

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

assume $b = 78.35$ (we know)

$$L = \sum (y_i - 78.35 * x_i - b)^2$$

L depends on square of b



- I want b for which L is min

Step 1 - Select a random b_{min}

2 - find slope at that pt

→ If slope is -ve then move right

→ If slope is +ve then move left.

$$\rightarrow b_{new} = b_{old} - \text{slope}$$

$$b_{new} = b_{old} - \eta \text{ slope} \quad \eta \rightarrow \text{learning rate}$$

↳ we use it so that

there is no drastic changes in slope

generally $\eta = 0.01$

⇒ when to stop -

i) $b_{old} - b_{new} \approx 0.001$

ii) iterations

• Mathematical formulam -

Step-1 :- Start with random value of b

Step-2 :- for i in epochs: $\eta = 0.01$
 $b_{new} = b_{old} - \eta \text{ Slope}$

↳ finding slope -

$$\frac{d}{db} (\sum (y_i - \hat{y}_i)^2) = \sum \frac{d}{db} (y_i - mx_i - b)^2$$

$$\text{Slope} = \{2(y_i - mx_i - b)\} \quad (\text{initially } b=0)$$

$\eta \text{ Slope} \rightarrow \text{step size}$

→ Now we are considering both m & b .

1) initialize random value for m & b
 $m=1$ and $b=0$

2) epochs = 100, $\eta = 0.01$

for i in epochs:

$$b = b - \eta \text{ Slope}$$

$$m = m - \eta \text{ Slope}$$

$$L = \sum (y_i - \hat{y}_i)^2 =$$

$$L(m, b) = \sum (y_i - mx_i - b)^2$$

$$b\text{-Slope} = \frac{\partial L}{\partial b} \sum (y_i - mx_i - b)^2$$

$$= -2 \sum (y_i - mx_i - b)$$

$$m\text{-slope} = \frac{\partial L}{\partial m} = -2 \sum (y_i - mx_i - b) x_i$$

→ If learning rate is too low then step size will be small and it will take too long to converge
is model become slow

→ If learning rate is too high then it will do zigzag motion and may not reach to min pt

→ Effect of Loss fn -

$$L = \sum (y_i - \hat{y}_i)^2 \rightarrow \text{convex } f^n$$

↳ have only 1 min.
and that is global minima

Lecture-2

Gradient Descent

↳ Batch GD

↳ Stochastic GD

↳ minibatch GD

in batch GD for computing $\frac{\partial L}{\partial m}$ we see whole dataset
" Stochastic " " " " " 1 row
" minibatch " " " " " a batch

→ Mathematical formulaⁿ -
↳ assuming 3 columns & 2 rows

x_1	x_2	y
0.1	93	3.2
7.5	95	3.5

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

Step 1 - Choose Random values
 $\beta_0 = 0$, $\beta_1, \beta_2 = 1$

2) epoch = 100 , lr = 0.1

$$\begin{aligned} \beta_0 &= \beta_0 - \eta \text{ Slope} \rightarrow \frac{\partial L}{\partial \beta_0} \\ \beta_1 &= \beta_1 - \eta \text{ Slope} \rightarrow \frac{\partial L}{\partial \beta_1} \\ \beta_2 &= \beta_2 - \eta \text{ Slope} \rightarrow \frac{\partial L}{\partial \beta_2} \end{aligned}$$

$$L(\beta_0, \beta_1, \beta_2)$$

→ If we have n -dimensions then $(n+1)$ derivatives

$$\begin{aligned} L &= \frac{1}{n} \sum_{i=1}^m (y_i - \hat{y}_i)^2 \\ &= \frac{1}{2} [(y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2] \end{aligned}$$

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$\hat{y}_1 = \beta_0 + \beta_1 x_{11} + \beta_2 x_{12}$$

$$\hat{y}_2 = \beta_0 + \beta_1 x_{21} + \beta_2 x_{22}$$

$$L = \frac{1}{2} [(y_1 - \beta_0 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_2 - \beta_0 - \beta_1 x_{21} - \beta_2 x_{22})^2]$$

$$\begin{aligned} \frac{\partial L}{\partial \beta_0} &= \frac{1}{2} [2(y_1 - \hat{y}_1)(-1) + 2(y_2 - \hat{y}_2)(-1)] \\ &= -\frac{1}{2} [(y_1 - \hat{y}_1) + (y_2 - \hat{y}_2)] \end{aligned}$$

$$= -\frac{2}{n} [(y_1 - \hat{y}_1) + (y_2 - \hat{y}_2) + \dots + (y_n - \hat{y}_n)]$$

$$= \left[-\frac{2}{n} \sum_{i=1}^n (y_i - \hat{y}_i) = \frac{\partial L}{\partial \beta_1} \right]$$

$$-\frac{\partial L}{\partial \beta_1} = \frac{1}{2} [2(y_1 - \hat{y}_1)(-x_{11}) + 2(y_2 - \hat{y}_2)(-x_{21})]$$

$$= -\frac{2}{2} [(y_1 - \hat{y}_1)x_{11} + (y_2 - \hat{y}_2)x_{21}]$$

$$= \left[-\frac{2}{n} \sum_{i=1}^n (y_i - \hat{y}_i)x_{i1} = \frac{\partial L}{\partial \beta_1} \right]$$

$x_{i1} \Rightarrow$ represent values of 1st column

$$-\frac{\partial L}{\partial \beta_2} = -\frac{2}{n} \sum_{i=1}^n (y_i - \hat{y}_i)x_{i2}$$