

1) Why Stochastic Gradient Descent (SGD)?

→ In Gradient Descent:

$N \Rightarrow$ Total no. of Training Examples

$$W_1 = W_0 - \frac{1}{N} \sum_{i=1}^N \left(\frac{\partial L}{\partial W} \right)_{W_0}$$

→ $L \Rightarrow$ loss function (MSE)

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

→ Since finding gradient for whole Dataset takes lot of time.

→ Stochastic Gradient Descent

2) SGD:

$$W_1 = W_0 - \alpha \sum_{i=1}^n \left(\frac{\partial L}{\partial W} \right)_{W_0}$$

$\alpha \Rightarrow$ learning rate

$L \Rightarrow$ loss function

$n \Rightarrow$ samples from "N"

Step 1) Initialize the weight & bias matrices with random values & weight matrices shape = (1, no. of dim)

$$W = \text{np.random.randn}(1, \text{no. of dim})$$

$$b = \text{np.random.randn}(1, 1)$$

Step 2) Pick randomly "n" no. of samples from total Dataset & of "N" data points

Step 3) Separate X-to & y-to.

Step 4) Differentiate the loss function (MSE)

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

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

(MSE)
Linear Regn
 $\hat{y}_i = w^T x_i + b$

$$\frac{\partial L}{\partial w} = \sum_{i=1}^n (-2x_i)(y_i - w^T x_i - b)$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^n (-2)(y_i - w^T x_i - b)$$

Here x_i, w are vectors

Eg: $x_1 \Rightarrow$ For 1st training example datapoint
(Boston Data) 13 columns

$$x_1 = [x_{11} \ x_{12} \ x_{13} \ \dots \ x_{1,13}] (1 \times 13)$$

$$w_0 = [w_{11} \ w_{12} \ w_{13} \ \dots \ w_{1,13}] (1 \times 13)$$

$\rightarrow y_i \Rightarrow$ scalar value

Steps **Epoch 1**
S-1) Initialize sum of $\frac{\partial L}{\partial w}$ & sum of $\frac{\partial L}{\partial b}$
to zero values initially

S-2) Iterate over "n" training examples
with first $w = w_0$ & find sum of $\frac{\partial L}{\partial w}$
over "n" examples
& also $\frac{\partial L}{\partial b}$ sum over "n" examples.

Eg: $\boxed{n=3}$ $\text{sum}(\frac{\partial L}{\partial w}) = [(-2x_1)(y_1 - w_0^T x_1 - b) + (-2x_2)(y_2 - w_0^T x_2 - b) + (-2x_3)(y_3 - w_0^T x_3 - b)]$

1/10 $\text{sum}(\frac{\partial L}{\partial b})$

step 5.3) Find next "w" $W_{old} = W_{previous\ step}$

$$w_{1\ new} = W_{old} - \alpha \left[\frac{\text{sum of } \partial L / \partial w}{n} \right]$$

$$b_{1\ new} = b_{old} - \alpha \left[\frac{\text{sum of } \partial L / \partial b}{n} \right]$$

step 5.4) Find $(y_{pred})_{i=1}^n \Rightarrow W^T X_i$

eg: if ~~$n=3$~~

if $n=3$

& $i=1, 2, 3$

$$(y_{pred})_1 = W_1^T X_1 + b_1$$

$(y_{pred})_2 \rightarrow$ append to list y_{pred}

step 6) Iterating over "n", now "i" takes 2 & so;

$$n=3 \quad \text{sum} \left(\frac{\partial L}{\partial w} \right) = [(-2 \times 1) (W_1^T -$$

$$6.1) \quad \text{sum} \left(\frac{\partial L}{\partial w} \right) = (-2 \times 1) [y_1 - W_1^T X_1 - b] + (-2 \times 2) [y_2 - W_1^T X_2 - b] + (-2 \times 3) [y_3 - W_1^T X_3 - b]$$

11) sum $(\partial L / \partial b)$

$$6.2) \quad w_2 = W_1 - \alpha \left[\frac{\text{sum of } \partial L / \partial w}{n} \right]$$

$$b_2 = b_1 - \alpha \left[\frac{\text{sum of } \partial L / \partial b}{n} \right]$$

step 6.3) Find $(y_{pred})_2 = w_2^T x_2 + b_2$

$\therefore i = 2$

2nd Training example. & w_2

\rightarrow append $(y_{pred})_2$ to the list y_{pred} .

step 6.7) Repeat step 6 till $(y_{pred})_n = w_n^T x_n$

so; we find " w ", " b " times

if $n = 3 \Rightarrow$ we find " w " 3 times.

step 7.2) calculate MSE \Rightarrow $mse(y_{pred}, y_{tr})$

step 8) epoch = 2

$W = W_{prev. epoch}$
Use " w " calculated in previous Epoch.
 \rightarrow Repeat steps 6, 7.

step 9) Find $mse(y_{pred}, y_{tr})$

with $(w_0)_2, (w_1)_2, (w_2)_2, \dots, (w_n)_2$

Epoch number
nth example

$(w_n)_2$ ← separately calculated

step 10) returns "Final" w & b , y_{pred} .

step 11) Predict on Test dataset
Use w, b of (Train data calculated)

eg Test Data has 5 examples:

~~$y_1 = w^T x_1 + b$~~ $y_1 = w^T x_1 + b$

$y_2 = w^T x_2 + b$

$y_3 = w^T x_3 + b$

$y_5 = w^T x_5 + b$

" w " is the final & last " w " of last Epoch