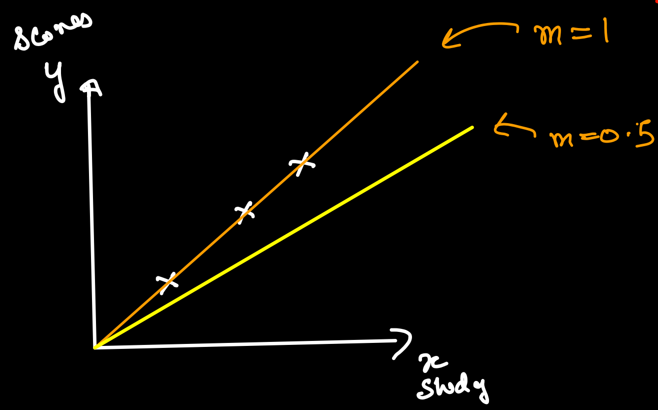
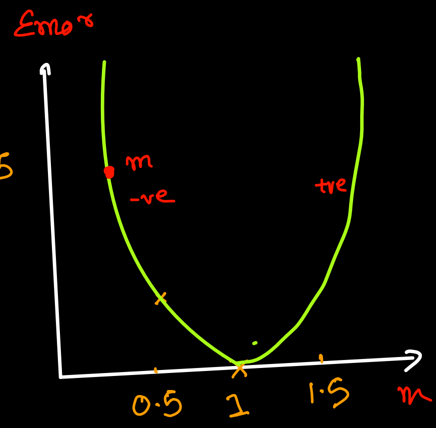
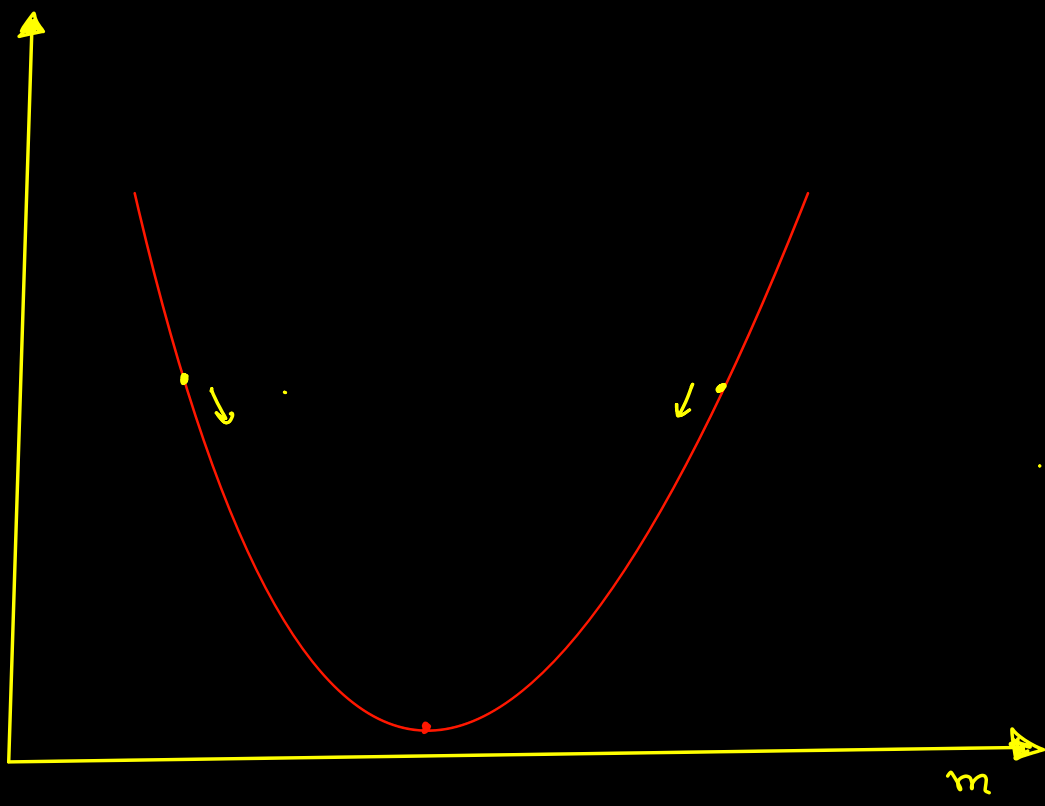
Gradient Descent





$$m_{new} = m_{old} - \text{slope}$$



Overall Idea/ Intuition behind G.D

1. Start from a random value of m, b
2. We find the slope
3. Update $m_{new} = m_{old} - slope_m$

   $b_{new} = b_{old} - slope_b$

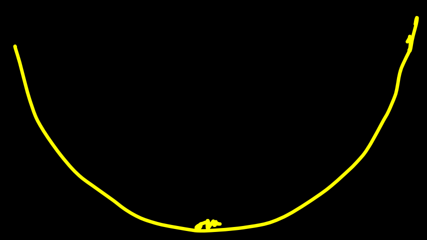
4. Find new m & b
5. We will repeat at the stage where

   2 things

   epochs

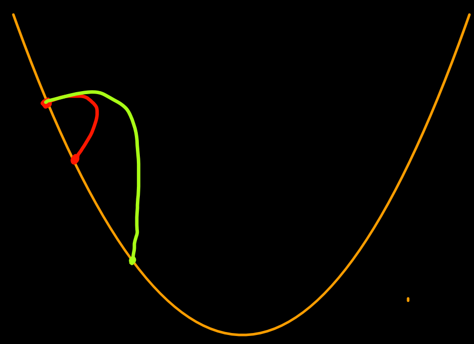
   fix no. of times

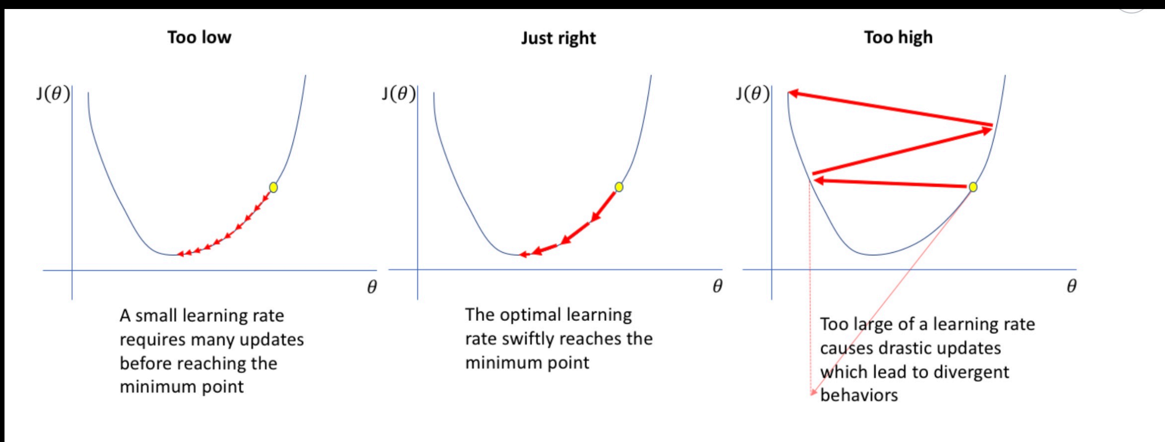
   till no much change in our cost fn.



Learning rate (α, η)

$m_{new} = m_{old} - \alpha * slope_m$





Hyperparameter $\Rightarrow$ We need to find optimal value of our hyper parameter

A learning rate controls the jumping & crawling of our algo.

$$J/\mathcal{E}/\mathcal{L}(m,b) = \sum (y_i - \hat{y}_i)^2$$

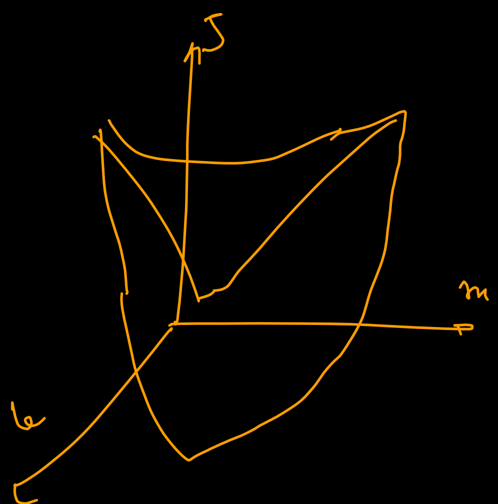
$$= \sum ((y - (m x_i + b)^{1})^2$$

$m, b \Rightarrow$ random

epochs, α

$$m_{new} = m_{old} - \alpha * slope_m$$

$$b_{new} = b_{old} - \alpha * slope_b$$



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

$$0 \qquad 0 \qquad (-1)$$

$$= \sum 2 (y_i - mx_i - b)(-1)$$

$$\boxed{= -2 \sum (y_i - mx_i - b)}$$

```
-2 * np.sum(y_train - self.m * X_train - self.b)
```

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

$$0 \qquad -x_i \qquad 0$$

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

$$\boxed{= -2 \sum (y_i - mx_i - b)(x_i)}$$

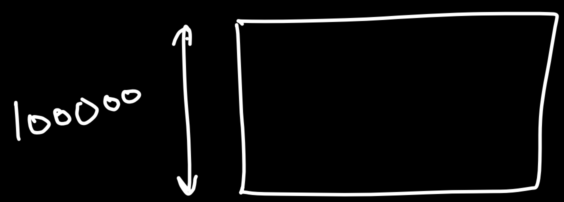
$$\frac{\partial L}{\partial m_1}$$

$$\frac{\partial L}{\partial m_2}$$

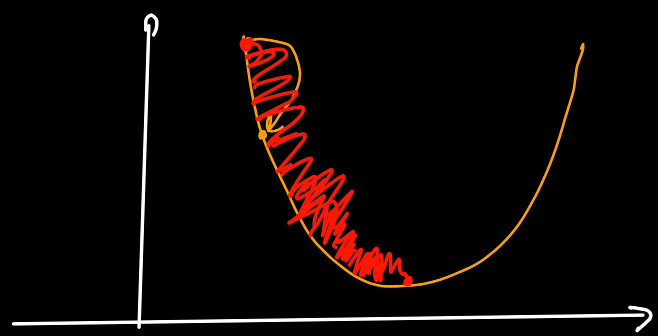
$$y = m_1 \text{study} + m_2 \text{sleep} + m_3 \text{exam} + m_4 \text{eating} + c$$

Types of Gradient Descent

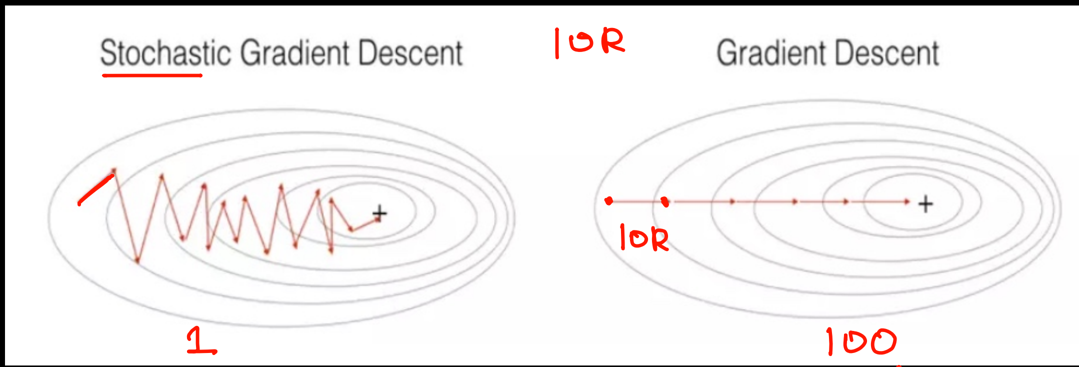
1. Batch Gradient Descent
2. Stochastic gradient descent



Update m, b



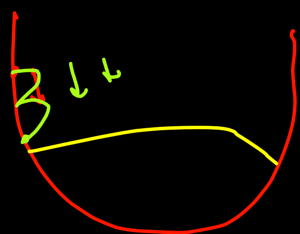
we update m & b after a single point



m , b after a single random point

10,000 Updates

100 updates



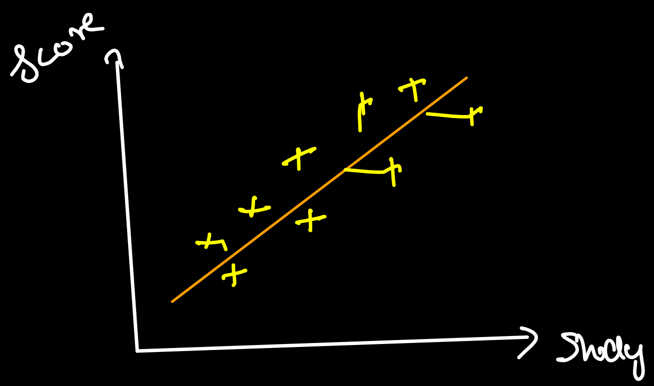
$m_{new} = m_{old} - \text{slope}$

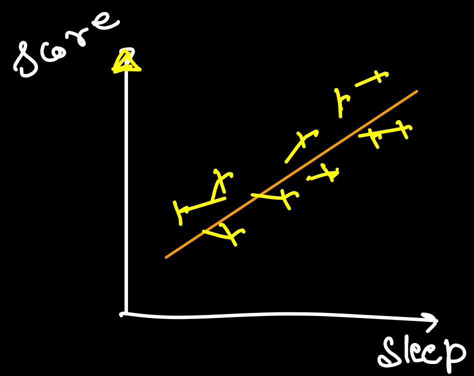
b

3. Mini batch after 30 data points

we will update m & b

Regression metrics

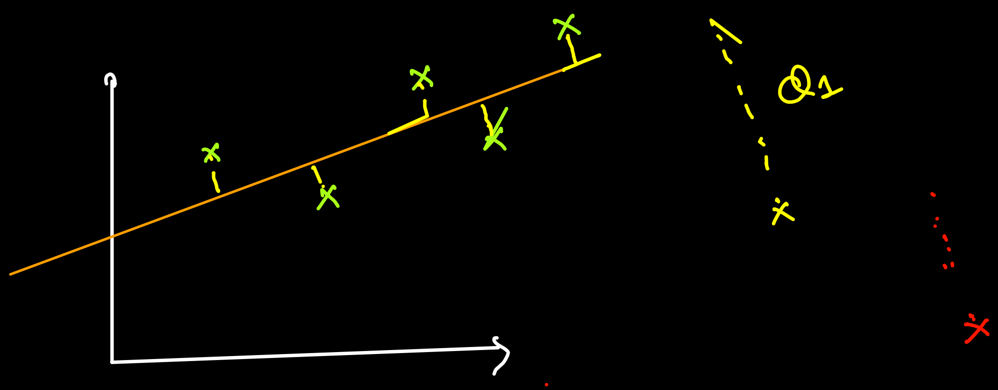




MAE
MSE
RMSE

$r2$ (coeff of determ)
Adjusted $r2$

MAE (mean absolute error)



$$|y_1 - \hat{y}_1| + |y_2 - \hat{y}_2| + \dots$$

total error $= \sum_{i=1}^{n} |y_i - \hat{y}_i|$

$$\boxed{MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}}$$

+

1] Same units

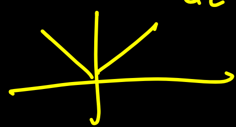
in terms of y only

0.5

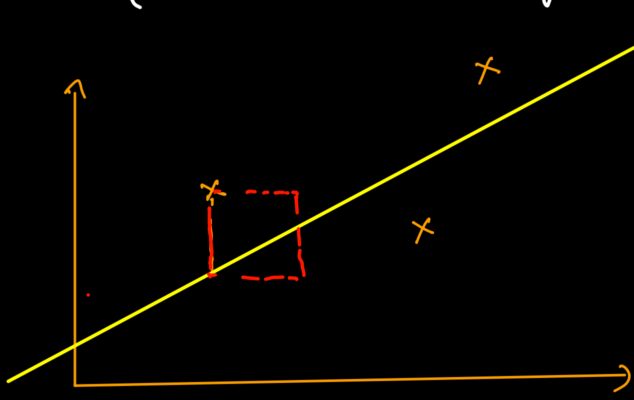
1] robust to outliers

−

1] graph is not differentiable at 0



MSE (mean squared error)



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

+	−
1. differential	1. unit is square
	2 m^2
	2. robust to outliers

RMSE (Root mean square error)

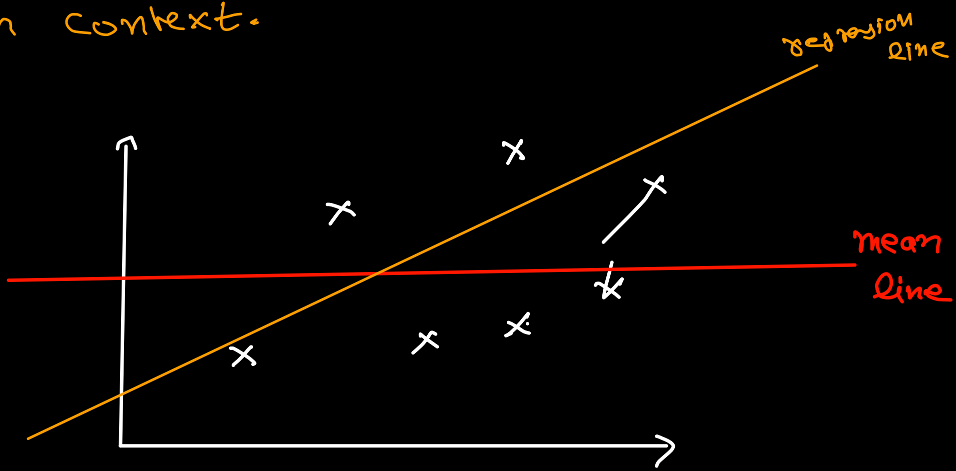
$$rmse = \sqrt{\frac{(y_i - \hat{y}_i)^2}{n}}$$

Deep learning

R^2 score

Tell how good / bad our model is.

All above were errors and they depend a lot on context.



by default we give the mean of our data

r2 score is how good our line is wrt mean line.

$$r2 = 1 - \frac{SS_r}{SS_m} = \frac{\text{Sum of square of errors in reg.}}{\text{Sum of square of error in mean line.}}$$

$$r2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

$r^2 = 0$ $\hat{y}_i = \bar{y}$

our regression line is equal to my mean line.

$r2 = 1$ $\sum (y_i - \hat{y}_i) = 0$

this means that our line is a perfect line

$r2 \longrightarrow 1$

Can my r2 score be -ve?

Study Score

$r2 = 0.90$

that means that my input (study) is able to explain 90% of my output (score)

very good general metric

free from context

Adjusted R2

Complete linear

logistic

Naive Bayes

DT, RF