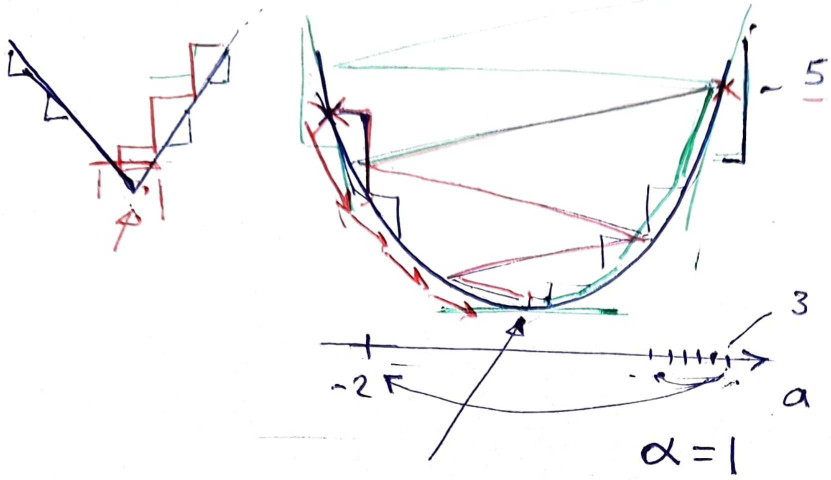
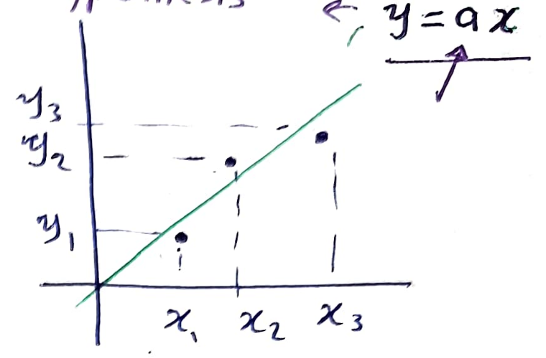
Optimization "Mansoura" session 2 AI45 21/1/2025



$\alpha = 1$

hypothesis $\Leftarrow \hat{y} = ax$



$$a_{new} = a_{old} \boxed{- \alpha \times \text{gradient}}$$

α: learning rate

$$a_{new} = a_{old} - 1 \times 5$$

- → cost function
- → objective
- → loss

$\Leftarrow$ hypothesis

$$h(x) = \hat{y} = ax + b$$

$10{,}000x + 100{,}000$

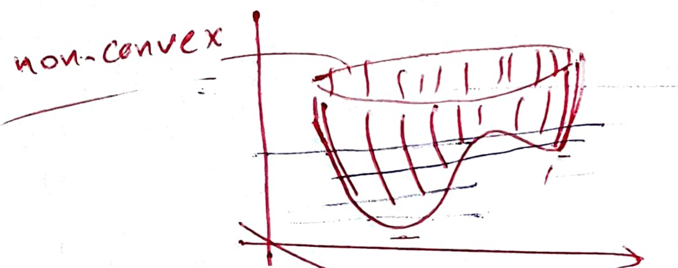
a: slope, b: intercept

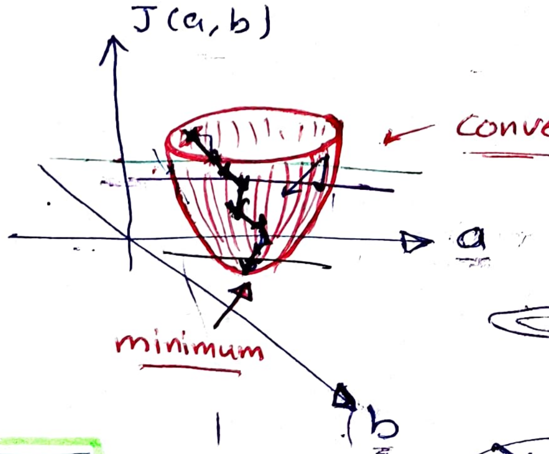
$$\text{MSE} \Rightarrow \frac{1}{m} \sum_i (y_i - \hat{y}_i)^2 = \frac{1}{m} \sum_{i=1}^{m} (y_i - (ax_i + b))^2 = J(a,b)$$

m: عدد ال data points

$\hat{y}_i$: $h_i(a,b)$

non-convex





contour plot

②

Gradient

$$\frac{1}{2m}\sum_{i=1}^{m}\left(y_i-(ax_i+b)\right)^2$$

$$\frac{\partial}{\partial a}J(a,b)=\frac{1}{\cancel{2}m}\sum_{i=1}^{m}\cancel{2}\left(y_i-(ax_i+b)\right)(-x_i)$$

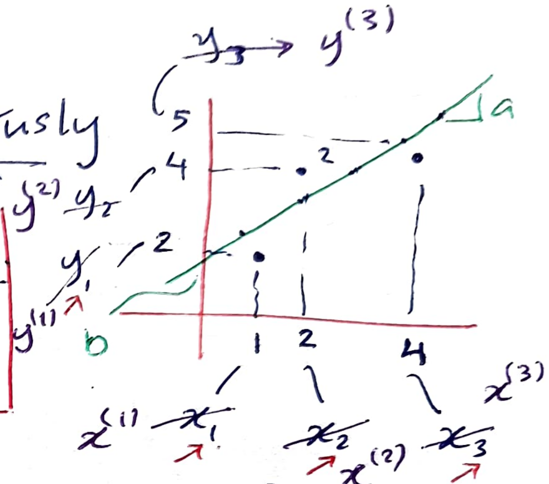
$$\frac{\partial}{\partial b}J(a,b)=\frac{1}{\cancel{2}m}\sum_{i=1}^{m}\cancel{2}\left(y_i-(ax_i+b)\right)(-1)$$

update a, b simultaneously

Repeat until convergence

e.g., 10 → times
100
1000

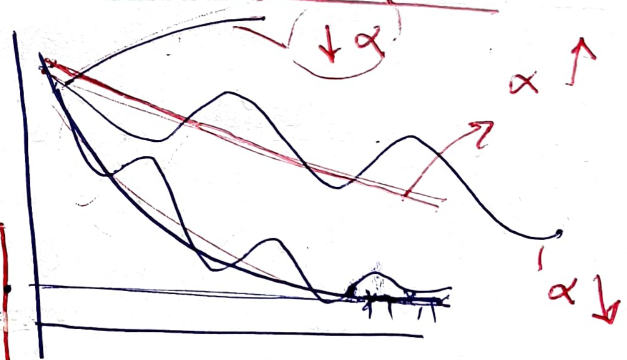
$$\boxed{\begin{aligned} a_{new} &= a_{old} - \alpha\ \text{grad} \\ b_{new} &= b_{old} - \alpha\ \text{grad.} \end{aligned}}$$



m=3

y_i, x_i : data point number

convergence and error →



$$\boxed{\text{next, we consider fitting a model with n-parameters, datset with m-point}}$$

→ **Multivariate linear regression**

$x^{(j)}$

$$h(a_i)=\hat{y}=\cancel{b}\,a_0+a_1x_1+a_2x_2+\cdots+a_nx_n$$

a_0 ↑ intercept /bias

$$h(\theta_0,\ldots,\theta_n)=\hat{y}=\theta_0+\theta_1x_1+\theta_2x_2+\cdots+\theta_nx_n$$

area, # of rooms, floor#

ex area : 1st feature = θ

$x_1^{(2)}$ area of apartment #2

ex. سعر الشقة price

y	x_0	x_1	x_2	...	x_n
1	1	150	4	---	3
0.8	1	120	3	---	4
	1				

m data points

→ x_i = variable # i = i^{th} variable

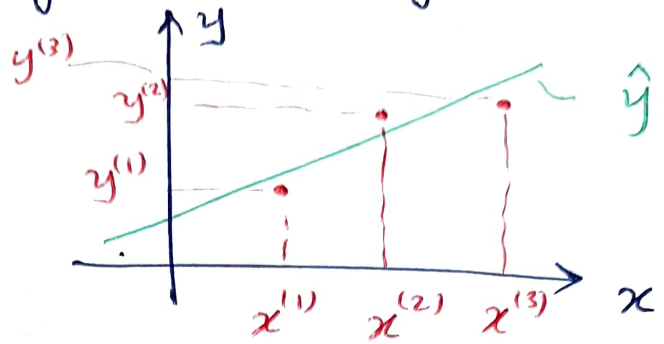
→ $x_i^{(j)}$ ⇒ i^{th} variable, data point # j

linear regression (single variable)

$y_1 = a x_1 + b$

$y_2 = a x_2 + b$

$y_3 = a x_3 + b$



→ two approaches

① using Gradient descent algorithm (using mean squared error)

data points تعداد

$$J(a,b) = \frac{1}{2m} \sum_{j=1}^{m} (\hat{y}_i - y_i)^2$$

gradient of cost function

recursive solution "optimization problem"

Repeat until convergence:

$$a := a - \alpha . \text{gradient}$$

$$b := b - \alpha . \text{gradient}$$

---

② using Pseudo-inverse (Moore-Penrose inverse) (generalized inverse) (The normal equation)

⇒ least squares solution (LS):

$y_1 = a x_1 + b$

$y_2 = a x_2 + b$

$y_3 = a x_3 + b$

$$\Rightarrow \underbrace{\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\vec{\theta}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}}_{\vec{y}}$$

$$\underbrace{X^T X}_{\text{square matrix}} \vec{\theta} = X^T \vec{y} \Rightarrow X^T X \vec{\theta} = X^T \vec{y}$$

$$I \leftarrow \cancel{(X^T X)^{-1} (X^T X)} \vec{\theta} = (X^T X)^{-1} X^T \vec{y} \Rightarrow \vec{\theta} = \begin{bmatrix} a \\ b \end{bmatrix}$$

4

hypothesis

Parameters

$$\hat{y} = h(\underbrace{\theta_0, \theta_1, \cdots \theta_n}_{\text{Parameters}}) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

$x_0 = 1$

$$\vec{\theta} = [\theta_0 \ \theta_1 \ \theta_2 \cdots \theta_n]^T$$

linear regression "multivariate"

$x_0 = 1$

$$\hat{y} = h_\theta(\vec{X}) = [\theta_0 \ \theta_1 \ \theta_2 \cdots \theta_n] \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$\vec{x} \in \mathbb{R}^{n+1}$

$\vec{\theta} \in \mathbb{R}^{n+1}$

$$\vec{x} = [x_0 \ x_1 \ x_2 \cdots x_n]^T$$

$h_\theta(\vec{x})$

$$\cancel{h(\vec{\theta})} = \hat{y} = \vec{\theta}^T \vec{X} = \vec{\theta} \cdot \vec{X}$$

n features

n+1 vector

Gradient

∇

del, nabla

divergance $(\vec{x})$ $\vec{\nabla} \cdot \vec{x}$

curl $(\vec{x})$ $\vec{\nabla} \times \vec{x}$

$$\nabla \omega = \frac{\partial \omega}{\partial x_1} \hat{x}_1 + \frac{\partial \omega}{\partial x} \hat{x}_2 + \cdots = \begin{bmatrix} \frac{\partial \omega}{\partial x_1} \\ \frac{\partial \omega}{\partial x_2} \\ \vdots \\ \frac{\partial \omega}{\partial x_n} \end{bmatrix}$$

$$\frac{\partial \bigcirc}{\partial x_1} \hat{x}_1 + \frac{\partial \bigcirc}{\partial x_2} \hat{x}_2$$

Cost /objective /loss function

$$J(\theta_0, \theta_1, \cdots \theta_n)$$

$$J(\vec{\theta}) = \frac{1}{2m} \underbrace{\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2}_{}$$

MSE

m-data points

sum of squared errors

for all data points

"Vanilla" (Batch)

Gradient descent algorithm

all model parameters from θ_0 to θ_n

{ update (simultaneously) all θ

$$\theta_0 := \theta_0 - \alpha \frac{\partial J(\vec{\theta})}{\partial \theta_0}$$

new old

$$\theta_1 := \theta_1 - \alpha \frac{\partial J(\vec{\theta})}{\partial \theta_1}$$

$$\vdots$$

$$\theta_n := \theta_n - \alpha \frac{\partial J(\vec{\theta})}{\partial \theta_n}$$

update "Repeat untill convergance"

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\vec{\theta})$$

gradient of cost function

$$\boxed{\vec{\theta} := \vec{\theta} - \alpha \nabla J(\vec{\theta})}$$

in vector form

update

vector of model parameters

cost function

learning rate

this can be also expressed as:

$$\theta \leftarrow \theta - \alpha \nabla J(\theta)$$

update