Numerical Optimization session 3 AI45 Mans. 29/1/2025

Review: Vanilla Gradient Descent
"Batch GD algorithm"

→ update parameters $\vec{\theta}$

$$\theta_{new} = \theta_{old} - \alpha \cdot \text{gradient of } J(\theta)_{old}$$

$$\theta_{new} = \theta_{old} - \alpha \nabla J(\theta_{old})$$

$$\frac{1}{2m} \sum_{j=1}^{m} (\quad)$$

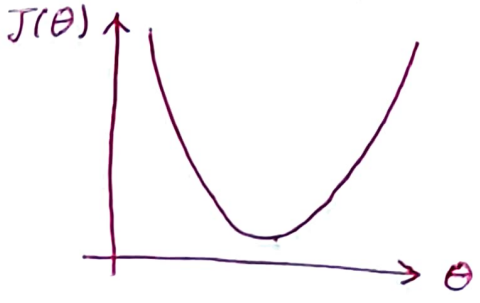
↖ summation over all data points (all examples in training dataset)

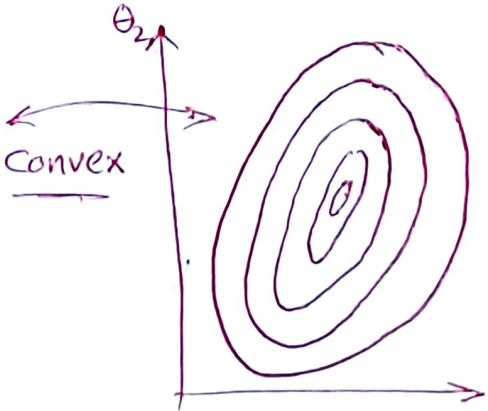
→ variations of GD

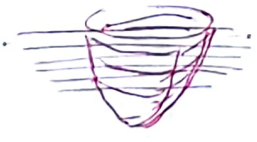
→ Problems associated with GD.

→ accelaration of GD; → • momentum method
• Nesterov method
• Adaptive GD
• Adam "ADAM" } next session

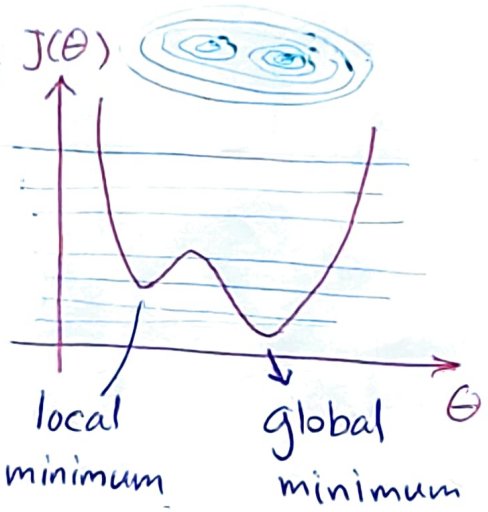
→ feature scaling

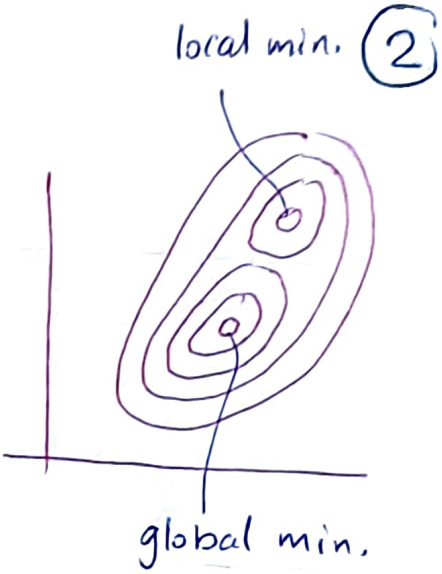


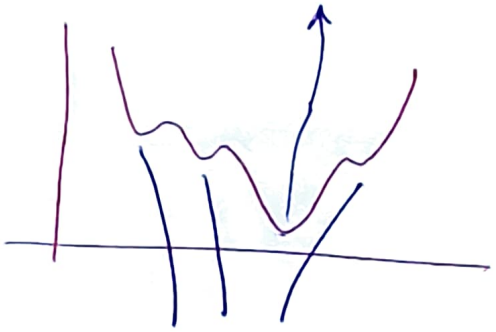




non convex







check feature scaling with logistic regression.

① feature scaling

in this example for simplicity, $\theta_0 = 0$

area x_1	# rooms x_2
100	2
⋮	⋮
200	5

ex: $J(\theta_1, \theta_2) = \frac{1}{2m}\sum_{i=1}^{m}\left(y - (\theta_0 + \theta_1 x_1 + \theta_2 x_2)\right)^2$

1
1.5
2

$\theta_0 x_0$, $x_0 = 1$

$x_1 \in [100, 200]$

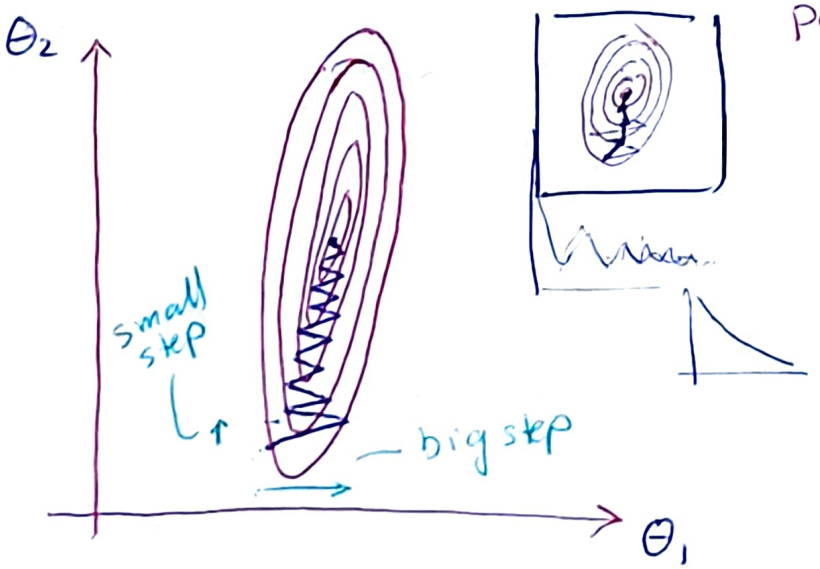
$x_2 \in [2, 5]$

remember; grad. ⇒ deriviative ⇒ (J) معدل تغير قيمة
الى التغير في ال parameter (θ)

$$\hat{y} = 0 + \theta_1 x_1 + \theta_2 x_2$$

1
2
7

area



x_1 has higher scale than x_2

$\Rightarrow \theta_2 \gg \theta_1$

$$Y = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2$$

$x_0 = 0$

$$\hat{Y} = 0.001\, x_1 + 10\, x_2$$

$\theta_1 \downarrow\downarrow$ $\theta_2 \uparrow$

Y	x_1	x_2
2	1000	0.1
3	1000	0.2
4	2000	0.2
	3000	

---

scaling : scaled features.

$x_1 \in [100, 200]$

$|\dot{x}_1| \sim 1 \sim$ $0 \to 1$, $-1 \to 1$

scaled feature

$\dot{x} \leftarrow x$ — $1000 \to 3000$

$\dot{x}$: scaled feature; x : original feature

$x \uparrow\uparrow \sim \theta_1$: 0.0010 → 0.0015

θ_1' : 10.0 → 15.01

θ x

x_k

$$\begin{bmatrix} 0 \\ \vdots \\ 2 \end{bmatrix} \quad \begin{bmatrix} 0.1 \\ \vdots \\ 0.8 \end{bmatrix}$$

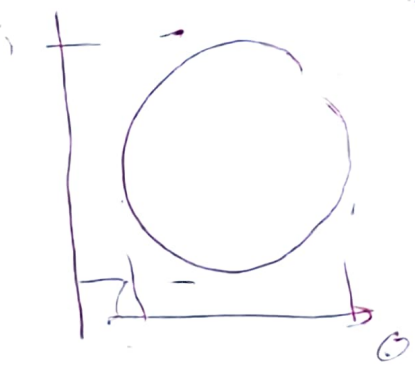
---

i) min-max normalization

$$0 \le \dot{x} = \frac{x - x_{min}}{x_{max} - x_{min}} \le 1$$

$\dot{x} \in [0,1]$

range



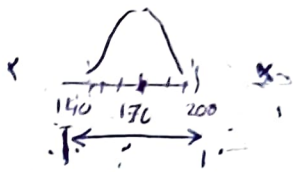
e.g.

$x_{min} = 100$

$x_{max} = 200$

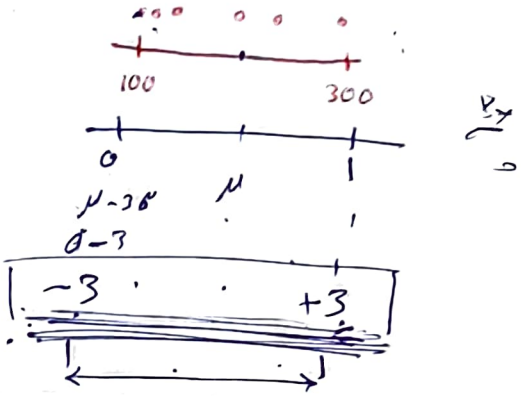
$x_{max} - x_{min} = 200 - 100 = 100$

x_1	$\dot{x}_1$	
$x_{min} \leftarrow 100$	0	☺
100	0	☺
150	0.5	0.05
$x_{max} \leftarrow 200$	1	
180	0.8	0.08
1000	1	



ii) mean-normalization (standardization)

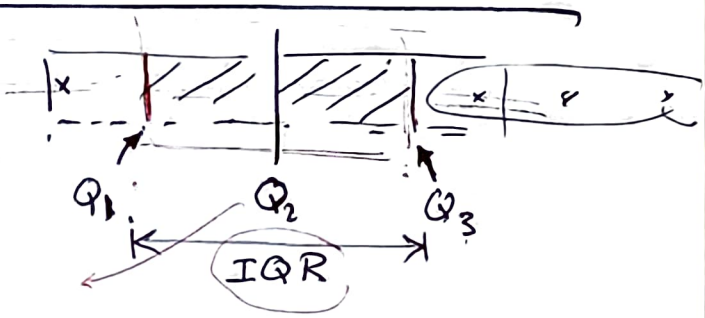
$$x' = \frac{x - \bar{x}}{\sigma} \quad \sim \text{mean}$$



---

iii) Quartiles

$$IQR = Q_3 - Q_1$$



Robust scaling

$$x' = \frac{x - Q_2}{Q_3 - Q_1}$$

→ measures of central tendency

→ measures of spread (variability)

---

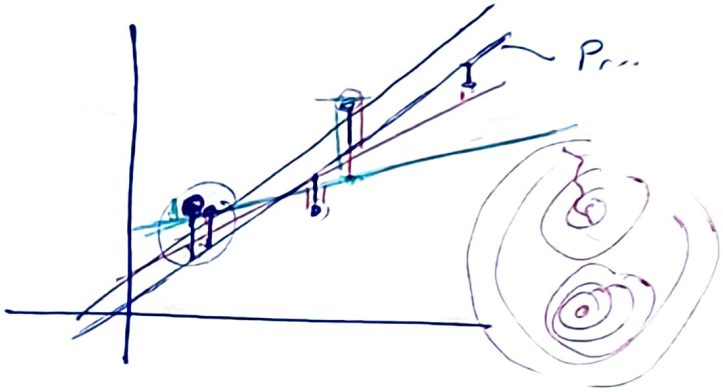
is feature scaling needed in all ML algorithms?

→ not in algroithms that are not distance-based.

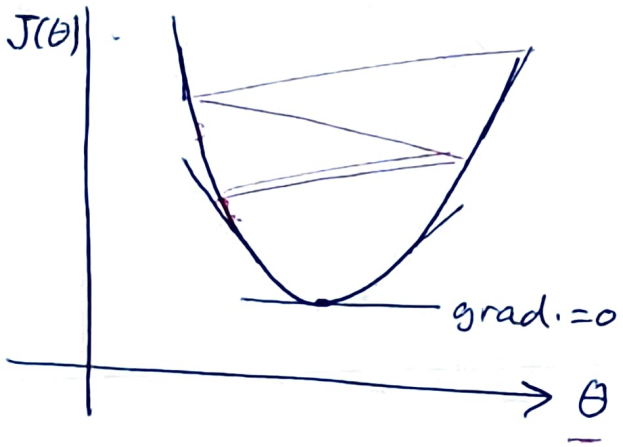
(2) Variants ~~Variations~~ of Batch GD "Vanilla GD";

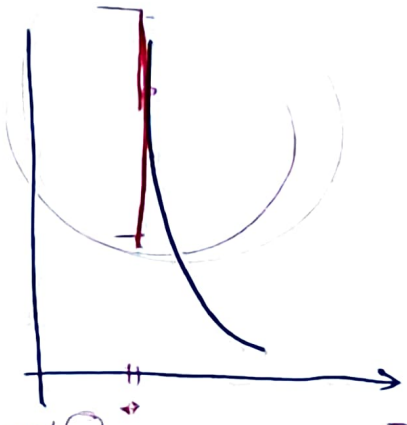
→ stochastic gradient descent

→ mini-batch gradient descent



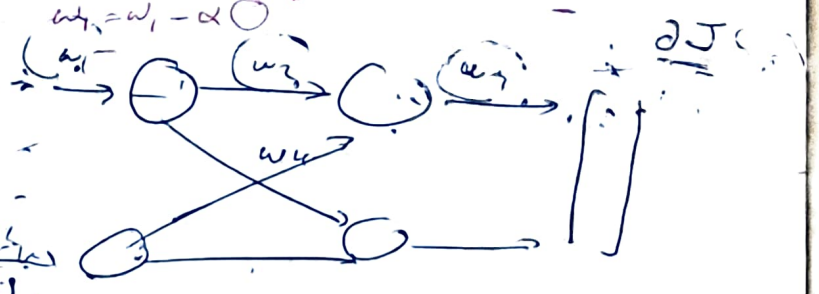
C.F





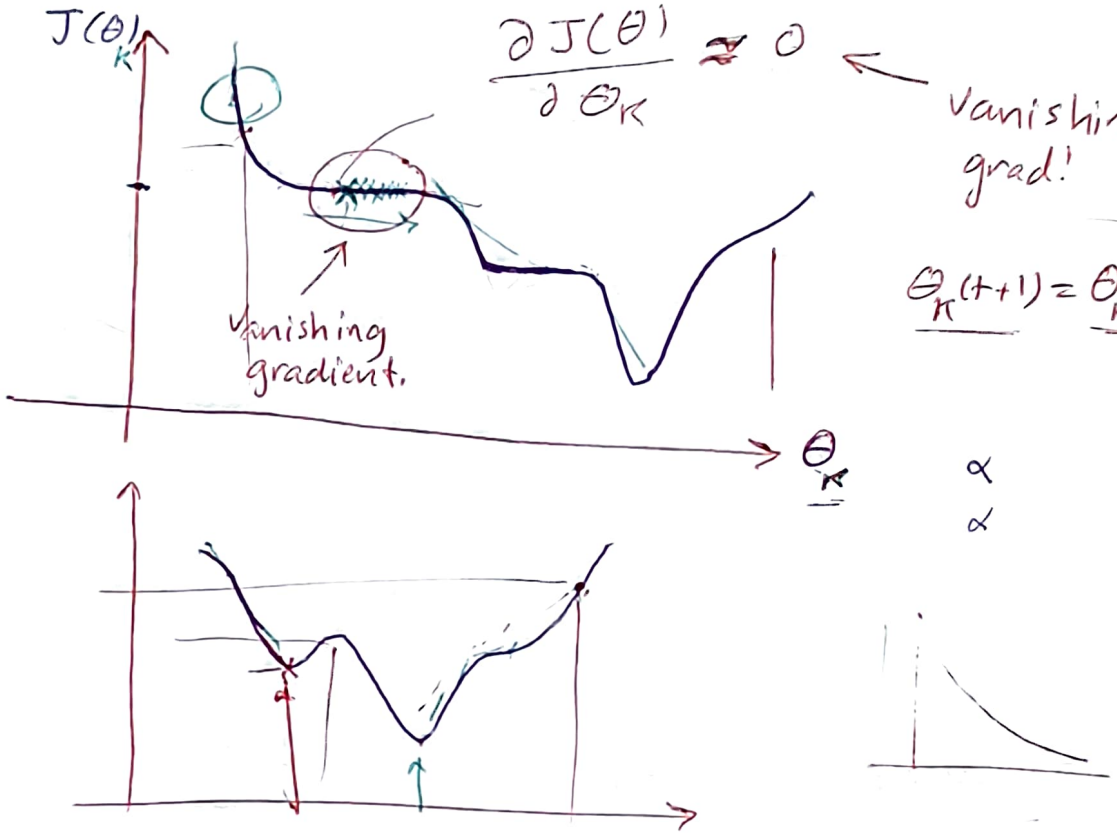
→ exploding grad.

→ vanishing grad.



$$\frac{\partial O}{\partial g} \times \frac{\partial g}{\partial \theta} \times \frac{\partial \theta}{\partial w}$$

← Chain rule



$\theta_K(t+1) = \theta_K(t) - \alpha \cdot \text{grad.}$

if zero

Gradient $\nabla J(\theta) = \begin{bmatrix} \partial J/\theta_0 \\ \vdots \\ \partial J/\theta_n \end{bmatrix}$

cost function : scalar function

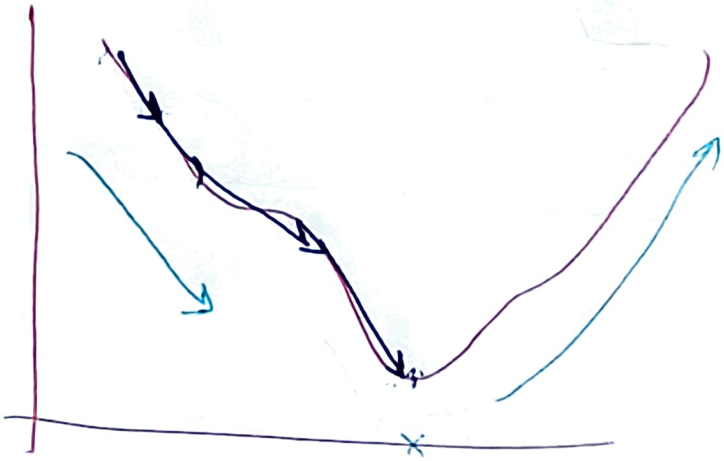
Jacobian → $\mathcal{J}$ (vector function)

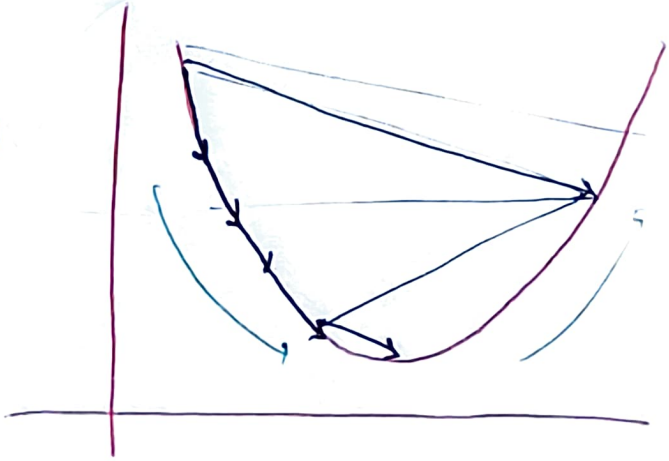
$\mathcal{J}(\vec{F}(\vec{x}))$ $\quad x_1, x_2 \quad \vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

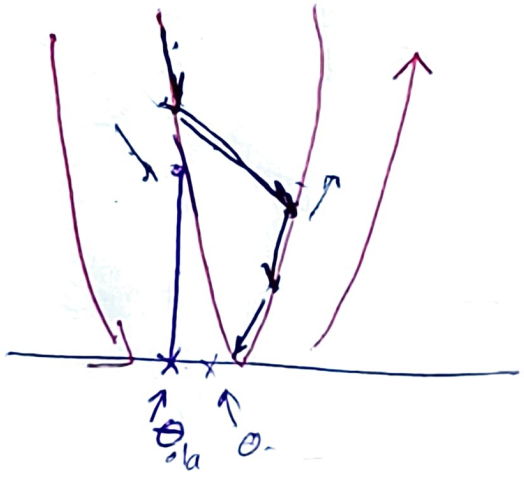
$\vec{F} = \begin{bmatrix} x_1 + x_2 \\ x_1 \times x_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$

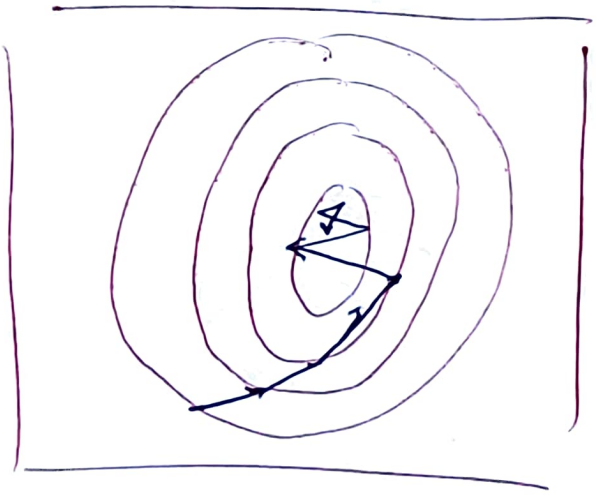
$$\mathcal{J}(\vec{F}) = \begin{bmatrix} \partial F_1/\partial x_1 & \partial F_1/\partial x_2 & \cdots & \partial F_1/\partial x_n \\ \vdots & & & \\ \partial F_2/\partial x_1 & \partial F_2/\partial x_2 & \cdots & \partial F_n/\partial x_n \end{bmatrix}$$

acceleration methods of GD Algorithm









Momemntum GD

$$\theta(t+1) = \theta_{new} = \theta_{old} - \underbrace{\alpha \text{ gradient}}_{\alpha \nabla J(\theta_{old})}$$

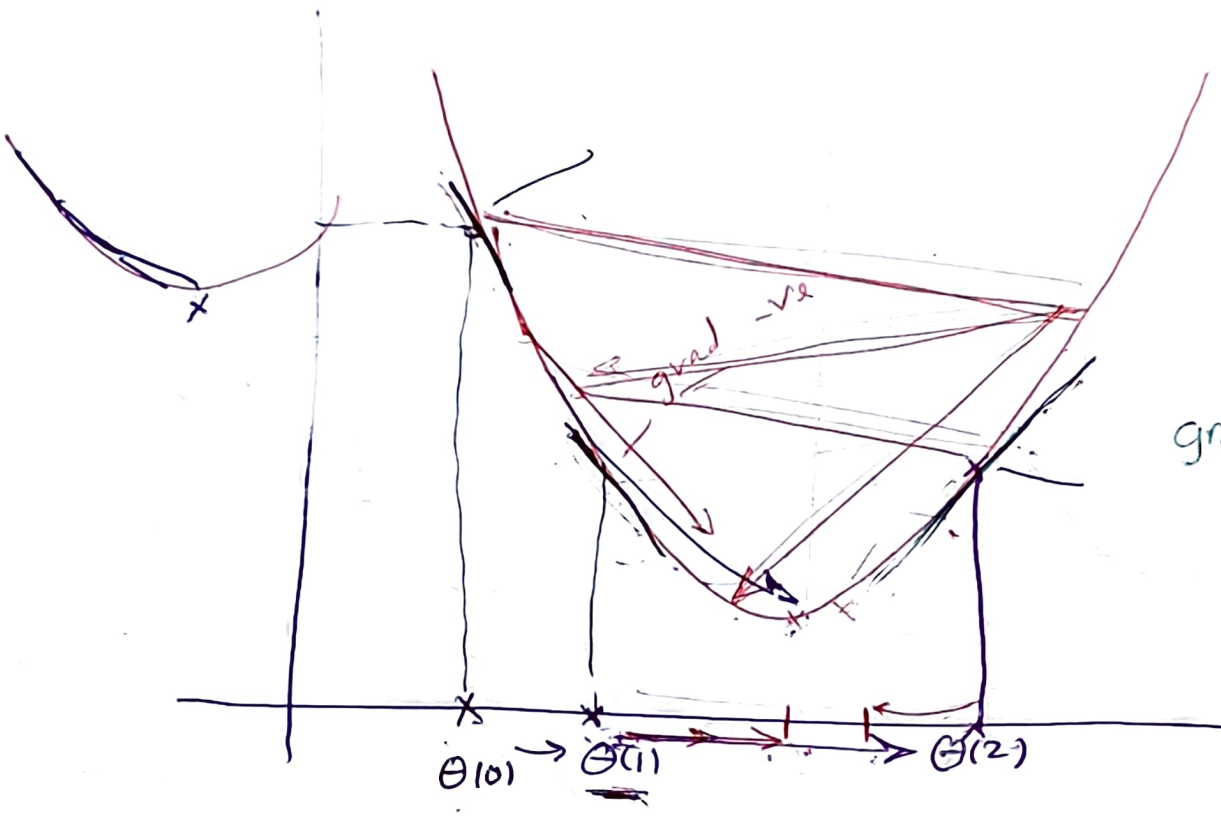
$\theta_{old} \Rightarrow$ gradient $(\theta_{old}) \Rightarrow$ update parameters

$$\theta_{new} = \theta_{old} - \alpha \text{ gradient}(\theta_{old})$$

→ momentum

$$\theta_{new} = \theta_{old} - \underbrace{\text{new gradient}}_{\text{current gradient}}$$

$$\theta_{old} - (\underbrace{\gamma \times \text{oldgradient}}_{} + \alpha \nabla J(\theta_{old}))$$



$$\theta(1) = \theta(0) - \alpha \, \text{grad}(\theta_0)$$ — old grad.

$$\theta(2) = \theta(1) - \alpha \, (\text{grad}(\theta_1))$$

momentum

$$\text{new gradient} = \gamma \cdot \text{old grad.} + \alpha \nabla J(\theta_1)$$

$$\theta(3) = \theta(2) - \left(\gamma \, \text{old grad.} + \alpha \nabla J(\theta_2) \right)$$

γ : $\theta(1)$; < 1 ; $\alpha \nabla J(\theta_2)$: new grad. evaluated at θ_2

Nesterov

$$\theta_{temp} = \theta_{old} - \gamma \times \text{old grad}$$

$$\theta_{new} = \theta_{temp} - \alpha \nabla J(\theta_{temp})$$

$$\text{newgradient} = \gamma \times \text{old gradient} + \alpha \nabla J(\theta_{temp})$$