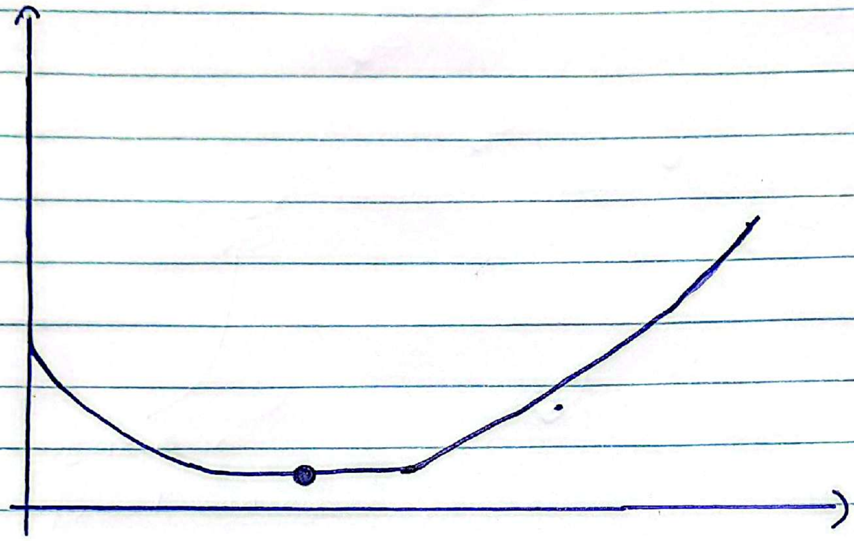
5) Optimization using Gradient Descent



$f(x) = e^x - \log(x)$

$f'(x) = 0$
(minimum)

$e^x - \frac{1}{x} = 0$

$e^x = \frac{1}{x}$

[Hard to Solve

1) Method 1
Try both direction
Gradient Descent

Function : $f(x)$ Goal: Find minimum of $f(x)$

*) Step 1 :
Define a learning rate α
Chose a starting point x_0

New point = old point − Slope

*) Step 2 :
Update → $\boxed{x_k = x_{k-1} - \alpha f'(x_{k-1})}$

*) Step 3 : Repeat until you are close enough to true minimum x^*

$f(x) = e^{x} - \log(x)$ $f'(x) = e^{x} - \frac{1}{x}$

Start $x = 0.05$ Rate: $\alpha = 0.005$

→ $f'(0.05) = -18.9$

$x = 0.05 - 0.005(-18.9)$
$(x) \Rightarrow 0.1447$

→ $f'(0.1447) = -5.7.552$

$x = 0.1447 - 0.005 f'(0.1447)$
$x = 0.1447 - 0.005(-5.7552)$
$(x) \Rightarrow 0.1735$

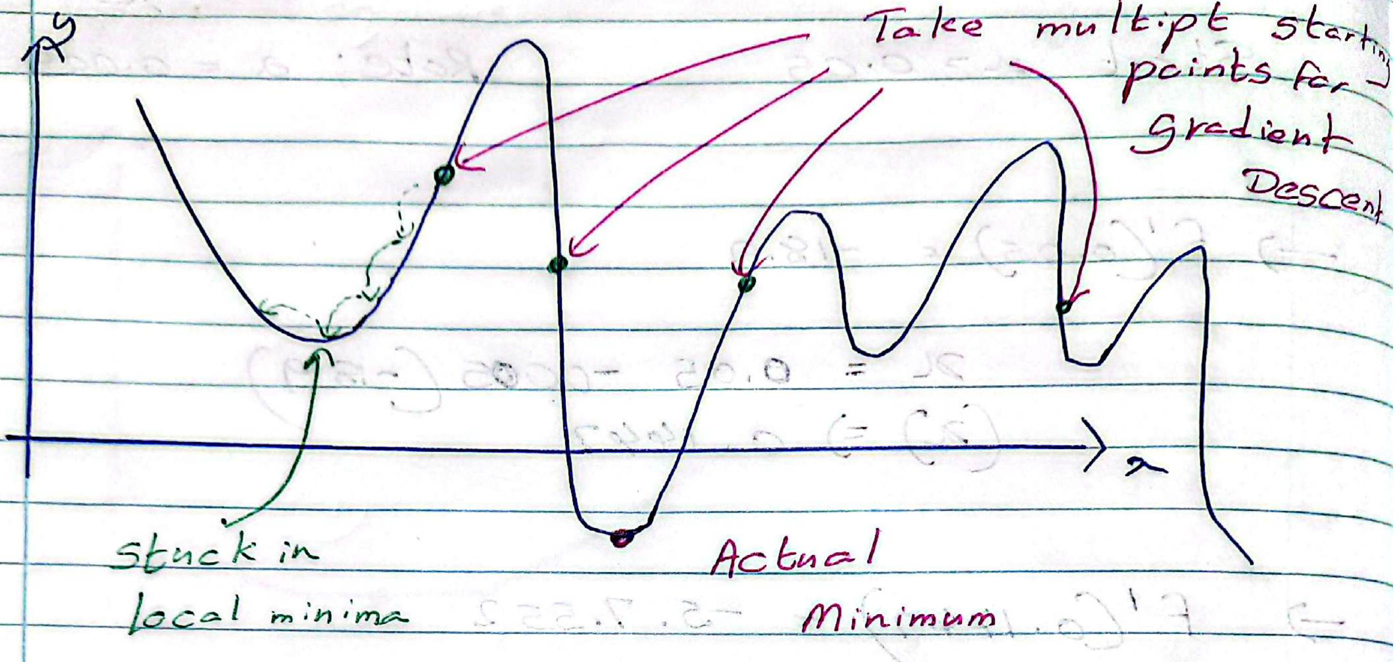
*) Repeat until you see no more changes to x

b] Learning Rate

*) If it is too large you may miss the minimum
*) If the learning rate is too small you may never reach the minimum

*) Unfortunately there is no definite rule to find the learning rate

Drawbacks of Gradient Descent



7] Optimization using Gradient Descent in two variables

function: $f(x,y)$ Goal: find minimum of $f(x,y)$

*] Step 1: •] Define a learning rate α
•] Chose a starting point (x_0, y_0)

*] Step 2: Update

$$\boxed{\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} - \alpha \nabla f(x_{k-1}, y_{k-1})}$$

*] Step 3:
Repeat step 2 until you are close enough to the true minimum

Method 2: Gradient Descent

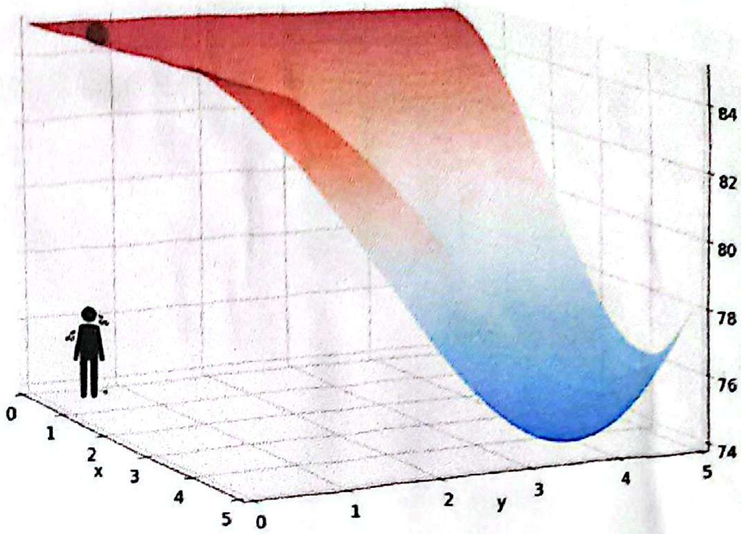
$T = f(x,y) = 85 - \frac{1}{90}x^2(x-6)y^2(y-6)$

Start: $x = 0.5,\ y = 0.6$

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

$$\nabla f = \begin{bmatrix} -\frac{1}{90}x(3x-12)y^2(y-6) \\ -\frac{1}{90}x^2(x-6)y(3y-12) \end{bmatrix}$$

$$\nabla f(0.5, 0.6) = \begin{bmatrix} -0.1134 \\ -0.0935 \end{bmatrix}$$

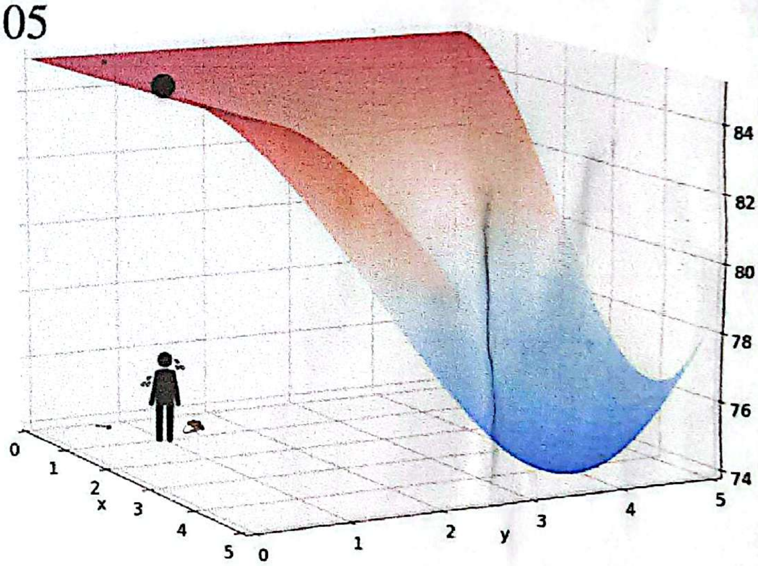


Start: $x = 0.5,\ y = 0.6$ Rate: $\alpha = 0.05$

$$\nabla f(0.5, 0.6) = \begin{bmatrix} -0.1134 \\ -0.0935 \end{bmatrix}$$

Move by
$-0.05\ \nabla f(0.5, 0.6)$

$x \mapsto 0.5057$
$y \mapsto 0.6047$



Start: $x = 0.5,\ y = 0.6$ Rate: $\alpha = 0.05$

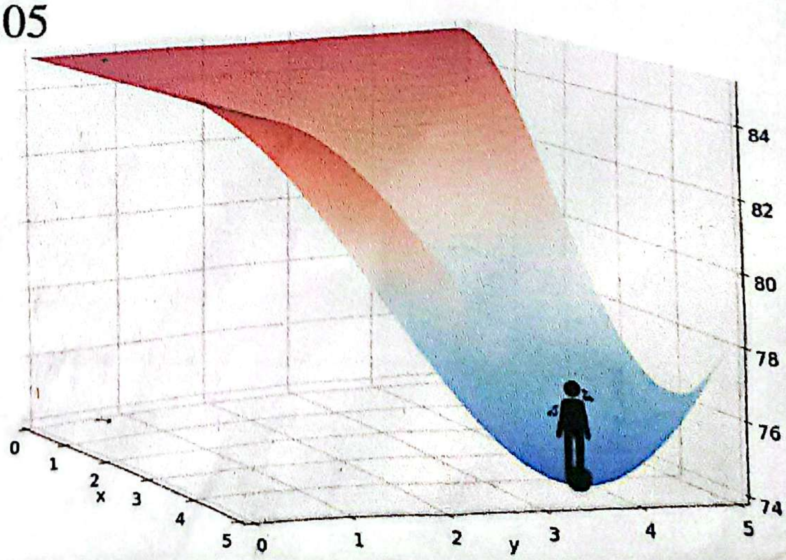
Find:

$$\nabla f(0.5057, 0.6047) = \begin{bmatrix} -0.1162 \\ -0.0961 \end{bmatrix}$$

Move by
$-0.05\ \nabla f(0.5057, 0.6047)$

$x \mapsto 0.5115$
$y \mapsto 0.6095$

Repeat!



Date ___ | ___ | ___ Nº ___

8] Optimization using Gradient Descent - Least squares

Goal: Minimize sum of squares cost

$$\nabla E = [28m + 12b - 42,\ 6b + 12m - 20]$$

$m =$ **?**
$b =$

The points m,b such that the cost is minimum

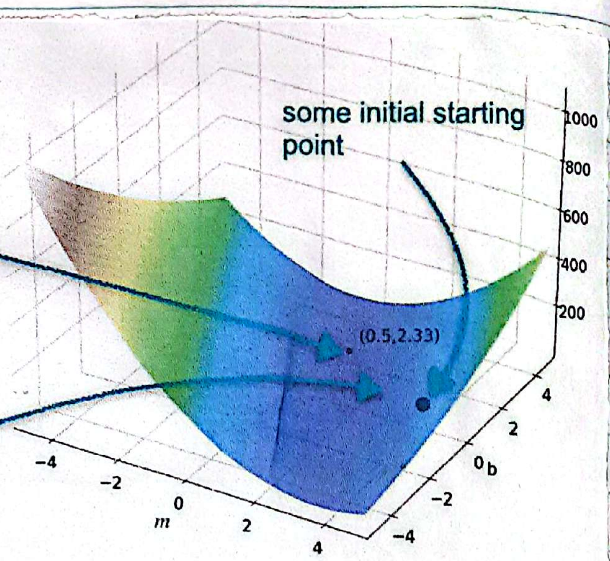
Steps:

Start with (m_0, b_0)

Iterate

$$(m_{k+1}, b_{k+1}) = (m_k, b_k) - \alpha \nabla E(m_k, b_k)$$

descend until you find the minimum



9] Least squares with multiple observations

Tv budget → Sales

230.1 → 22.1
44.5 → 10.4
17.2 → 9.3

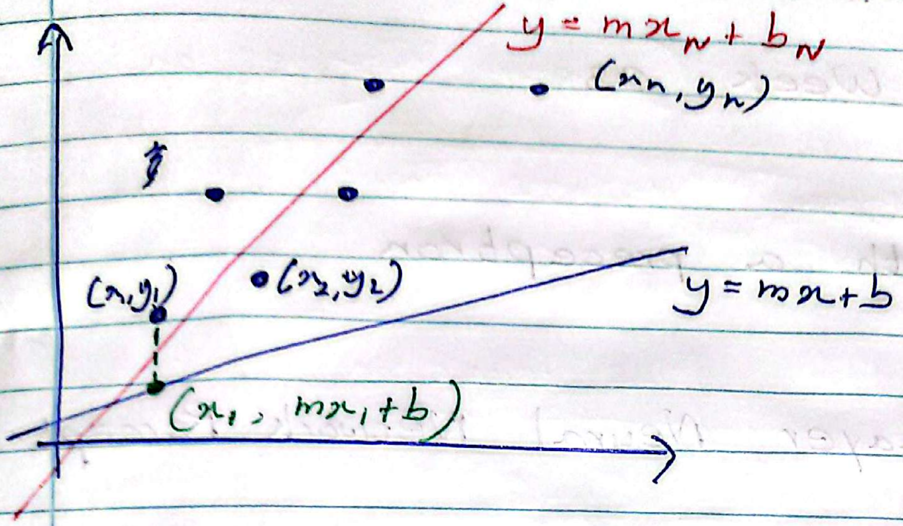
Goal : predict sales in terms of TV budget

Tool : Linear Regression

$y = mx + b$ (Multiple Observations)

RATHNA

Date



$$\text{Distance}^2\ (\text{Loss}) = (mx_1 + b - y_1)^2$$

$$L(m, b) = \frac{1}{2n}\left[(mx_1 + b - y_1)^2 - \dots + (mx_n + b - y_n)^2\right]$$

$$\begin{bmatrix} m_0 \\ b_0 \end{bmatrix} \longrightarrow \begin{bmatrix} m_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} m_0 \\ b_0 \end{bmatrix} - \alpha \nabla L_1(m_1, b_1)$$

$$\begin{bmatrix} m_N \\ b_N \end{bmatrix} \longrightarrow \begin{bmatrix} m_N \\ b_N \end{bmatrix} = \begin{bmatrix} m_{N-1} \\ b_{N-1} \end{bmatrix} - \alpha \nabla L_1(m_{N-1}, b_{N-1})$$