Week 2: Gradients and Gradient Descent:

$\partial \rightarrow$ di
$\nabla \rightarrow$ Nabla

For 3D graph the planes are having tangents as 2D planes. The tangent planes are described by two vectors that forms these planes. Then, these vectors are put in a gradient matrix.

Examples:

$$f(x, y) = x^2 + y^2$$

Solution:

step(1): $\frac{\partial f}{\partial x} = 2x \qquad \frac{\partial f}{\partial y} = 2y$

partial derivatives

step(2): The gradient of $f(x,y)$ is:

$$\nabla f = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

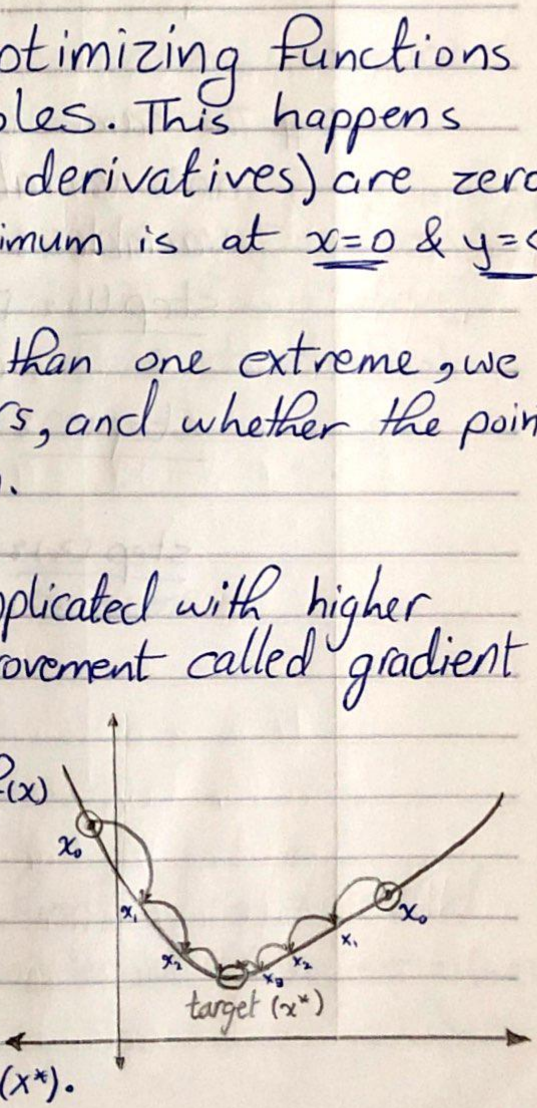
Gradient is useful in optimizing functions of two or more variables. This happens when all slopes (partial derivatives) are zeros.

step(3): The minimum is at $x=0$ & $y=0$.

In case of presence of more than one extreme, we need to check valid answers, and whether the point is a maximum or a minimum.

The gradient can get very complicated with higher dimensions. Therefore, an improvement called gradient descent was introduced:

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



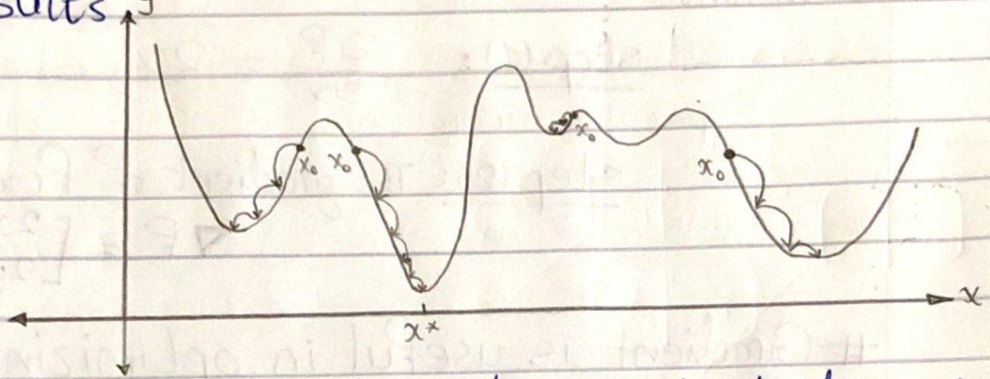
step(1): Define learning rate α
choose start point x_0

step(2): $x_k = x_{k-1} - \alpha f'(x_{k-1})$

step(3): repeat step 2 till you get close enough to the true minimum(x^*).

Unfortunately, there is no rule to give the best learning rate (α). The problem is for:
→ large learning rate, you can miss x^*.
→ small learning rate, you can take a lot of time and computing processes.

Another drawback is that gradient descent gets you to the local minimum not to the absolute minimum. The solution here is to redo it many times till you get better results.



The same way we used a gradient descent on one variable graph, we apply it to more variables: Function: $f(x,y)$ Goal: Find minimum $f(x,y)$

step (1): Define learning rate α
Choose a starting point (x_0, y_0)

step (2):
$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} - \alpha \nabla f(x_{k-1}, y_{k-1})$$

step (3): repeat step 2 until you are close enough to the true minimum (x^*, y^*).