Name: Omar Rashad Salem

Course: CV - prof.Heba

Assignemnt No.: 3

> ## QUESTIONS

1)What is the purpose of gradient descent optimization in machine learning?

ans: Used to find the best set of parameters for a model to minimize the loss function and perform well on a given task. It is an iterative algorithm that repeatedly adjusts the parameters in the direction that reduces the loss function.

***2)* What is the learning rate in gradient descent, and why is it important?**

ans: It's important becuase it's the parameter that tells us How far should we go along the direction that loss function has steepest rate of increase. It's also called `step size`

***3)* What is the role of batch size in gradient descent optimization?**

ans: It desides the number of training examples that are used to calculate the gradient of the loss function in each iteration. Determining batch size is important for these specific issues:

- **Convergence speed**
- **Accuracy**
- **Stability**

***4)* Consider a quadratic function □(□) = □ – 4□ + 3. Use gradient descent to find the minimum value of the function?**

Use the following conditions:

- **Initial guess: x = 0**
- **Learning rate: 0.25**
- **Number of iterations: 5**

ans:

$f'(x) = 1 - 8x$

$x_{new} = x_{old} - rate * f'(x_{old})$

Initial guess (x_0) = 0 Learning rate = 0.25 Number of iterations = 5

Itr1:

$$x_1 = x_0 - rate * f'(x_0) = 0 - 0.25 * (1 - 8 * 0) = 0 - 0.25 * 1 = -0.25$$

Itr2:

$$x_2 = x_1 - rate * f'(x_1) = -0.25 - 0.25 * (1 - 8 * (-0.25)) = -0.25 - 0.25 * 3 = -0.25 - 0.75 = -1$$

Itr3:

$$x_3 = x_2 - rate * f'(x_2) = -1$$
$$-0.25 * (1 - 8 * (-1)) = -1$$
$$-0.25 * 9 = -1 - 2.25 = -3.25$$

Itr4:

$$x_4 = x_3 - rate * f'(x_3) = -3.25$$
$$-0.25 * (1 - 8 * (-3.25)) = -3.25$$
$$-0.25 * 27 = -3.25 - 6.75 = -10$$

Itr5:

$$x_5 = x_4 - rate * f'(x_4) = -10$$
$$-0.25 * (1 - 8 * (-10)) = -10$$
$$-0.25 * 81 = -10 - 20.25 =$$
$$-30.25$$

After 5 iterations, the estimated minimum value of the function is at x ~= -30.25.

Programming: **1. write a python code to perform gradient decent to find the value that gives the minimum value of the function ()() = – 4 + 3.**

In [1]:

```
def derivative_f(x):
    return 1 - 8*x

def gradient_descent(initial_x, learning_rate, num_iterations):
    x = initial_x

    for _ in range(num_iterations):
        gradient = derivative_f(x)
        x = x - learning_rate * gradient

    return x

# Define parameters
init_guess = 0
rate = 0.25
itr_no = 100

# Perform gradient descent
min_x = gradient_descent(init_guess, rate, itr_no)

# Calculate the minimum value of the function
min_value = min_x - 4*min_x**2 + 3

print(f"Best x  approximately = {min_x}")
print(f"minimum value of the function  approximately = {min_value}")
```

```
Best x  approximately = -6.442219009150142e+46
minimum value of the function  approximately = -1.6600874304742172e+94
```

programming: **2.If you learned that the value of to give the minimum value is: 2, could you tune your code so it gives a close value. What is the learning rate and number of iterations that you used.**

ans: tuned parameter would be :

initial_guess = 0

learning_rate = 0.01

num_iterations = 1000