Assignment 5

Pallakonda Sahithi - EE22B080

Gradient Descent based Optimization

Gradient Descent is an optimization algorithm for finding a local minimum of a differentiable function. Here, we deal with 1D and 2D polynomials. The python script `ee22b080_gradient_opt.py` implements the gradient descent algorithm for a few functions given. The code has a function **grad_opt()** which executes that algorithm and gives an animation showing the path converging to the minimum.

Tasks performed by the python code:

1. First, it imports all the required libraries such as `numpy`, `matplotlib` for implementation.
2. **grad_opt()**:

- Learning Rate(lr) controls the step size during each iteration of optimization,it is defined as 0.1 for all implementations.
- Then it checks whether it is a 1D polynomial or a 2D polynomial and performs the corresponding operation.
- **xall[], yall[], zall[]** are the lists to store the trajectories. Each subsequent point on the trajectory(Converging path while finding the local minima) can be found using the equations:

Consider any arbitary point in the trajectory, say (x_i, y_i), function(f).

1D Polynomials:

$-> x_{i+1} = x_i - lr \times \left(\frac{df1(x)}{dx}\right)_{x=x_i}$

2D Polynomials:

$-> x_{i+1} = x_i - lr \times \left(\frac{df(x,y)}{dx}\right)_{(x,y)=(x_i,y_i)}$

$-> y_{i+1} = y_i - lr \times \left(\frac{df(x,y)}{dy}\right)_{(x,y)=(x_i,y_i)}$

- Finally, it creates the animation and output the optimized values after 80 repetitions.

Polynomial 1

$f1(x) = x^2 + 3x + 8$.

It is a 1D polynomial(Quadratic) with positive co-efficients. Therefore, it has only one point of minimum in the specified range of coordinates. To implement optimization algorithm, lets assume the starting point to be `x = 5`, there is no specific reason to decide this starting point.

Optimized Values:

- x: -1.4999
- y: 5.75

These values are quite closer to the actual point of minimum. Improvements can be made by increasing the number of repetitions or changing the starting point.

Polynomial 2

$f3(x, y) = x^4 - 16x^3 + 96x^2 - 256x + y^2 - 4y + 262.$

It is 2D polynomial having only one point of minimum in the specified range of `x,y`. To implement optimization algorithm, lets assume the starting point to be `x = 5, y = 5`, the assumption should be made by trial and error method(trial and error is used to avoid overflow error).

Optimized Values:

- x: 4.1206, y: 2.0000
- z: 2.0002

The actual point of minimum is $(4, 2, 2)$, this is quite closer to the optimized values. Improvements can be made by increasing the number of repetitions.

Polynomial 3

$f4(x, y) = e^{-(x-y)^2} sin(y).$

It is 2D polynomial having one point of minima in the region $R1(x < 0, y < 0, z < 0)$ and one point of maxima in the region $R2(x > 0, y > 0, z > 0)$ within the specified range of coordinates. To find the minima, the starting point should be somewhere near the origin and in R1. Assuming it to be $(-0.5, -0.5, f4(-0.5, -0.5))$.

Optimized Values:

- x: -1.5454, y: -1.5516
- z: -0.9998

These values are quit nearer to the actual point of minima $(-\frac{\pi}{2}, -\frac{\pi}{2}, -1)$. Improvements can be made by increasing the number of repetitions.

Polynomial 4

$f5(x) = \cos^4 x - \sin^3 x - 4\sin^2 x + 4\cos x + 1.$

It is 1D poynomial(Trigonometric) having two local minima. Our goal is to optimize it to the lowest minima, so the assumption is made by plotting the given polynomial and decide which is the best starting point. The point which is just left to the maxima is suitable for it, therefore starting point is assumed to be `x = 3.04`.

Optimized Values:

- x: 1.6617 (Actual = 1.662)
- y: -4.0454 (Actual = 4.05)

Plots

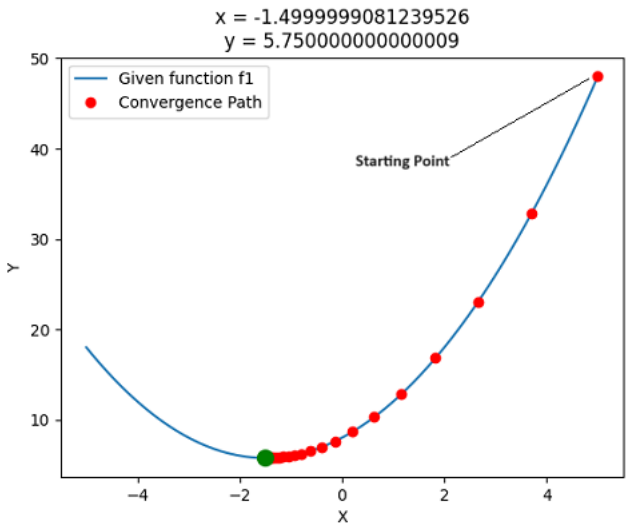

Figure 1: f1(x)

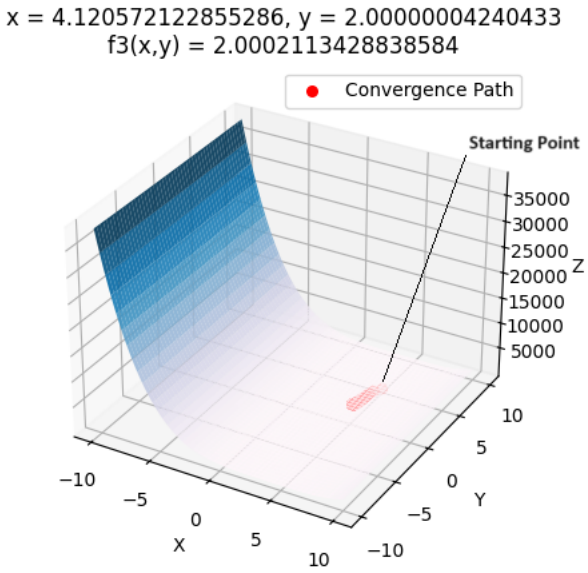

Figure 2: f3(x,y)

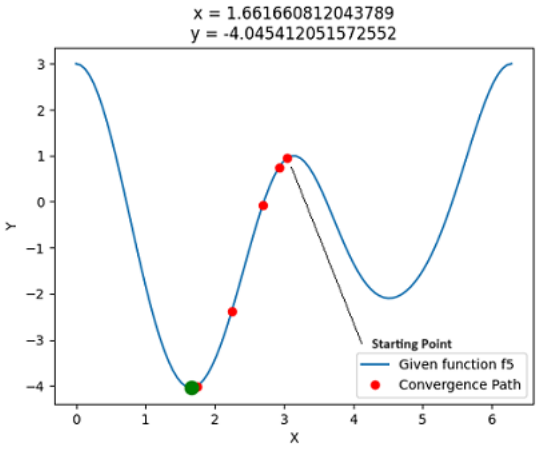

Figure 3: f5(x)

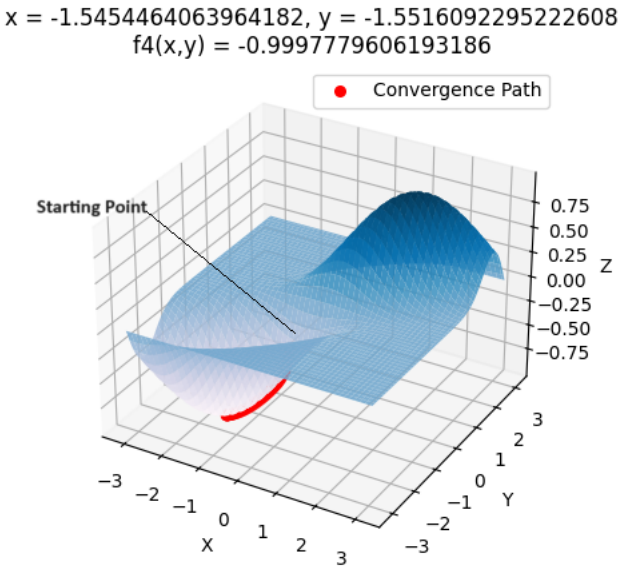

Figure 4: f4(x,y)