EE2703: Assignment 5

Nikhil Deepak - ee22b020

October 2023

1 Introduction

This report presents the results of implementing gradient descent optimization for a set of functions. The assignment required optimizing various functions using gradient descent and creating visualizations to demonstrate the convergence. The following sections detail the approach, functions, and results for each problem.

2 Problem 1: 1D Simple Polynomial

2.1 Function

- $f1(x) = x^2 + 3x + 8$
- Derivative: $df1\_dx(x) = 2x + 3$
- Optimization Range: $[-5, 5]$

2.2 Approach

- Implemented a generic gradient descent function.
- Chose the initial point as $x = 4$, learning rate as 0.1, and number of iterations = 100.
- Applied gradient descent to optimize the function.
- Visualized the convergence using Matplotlib.

2.3 Results

The function takes minimum value of 5.75 at x = -1.49999999887963

3 Problem 2: 2-D Polynomial

3.1 Function

- $f_3(x, y) = x^4 - 16x^3 + 96x^2 - 256x + y^2 - 4y + 262$
- Derivatives: $df3\_dx(x, y)$, $df3\_dy(x, y)$
- Optimization Ranges: $[-10, 10]$ for both x and y

3.2 Approach

- Adapted the generic gradient descent function for 2-D optimization.
- Took initial points as (5, 5), learning rate as 0.1, and the number of iterations as 1000
- Optimized the function using gradient descent.
- Visualized the convergence with Matplotlib.

3.3 Results

- Optimal x: 4.035235174181332
- Optimal y: 2.000000000000001
- Minima: 2.0000015413659185

4 Problem 3: 2-D Trigonometric

4.1 Function

- $f4(x, y) = e^{-(x-y)^2} \sin(y)$
- Derivatives: $df4\_dx(x, y)$, $df4\_dy(x, y)$
- Optimization Ranges: $[-\pi, \pi]$ for both x and y

4.2 Approach

- Adapted the generic gradient descent function for 2-D optimization.
- Set initial points as $(0, 0)$, learning rate $= 0.1$, and the number of iterations $= 100$
- Conducted optimization using gradient descent.
- Visualized the convergence with Matplotlib.

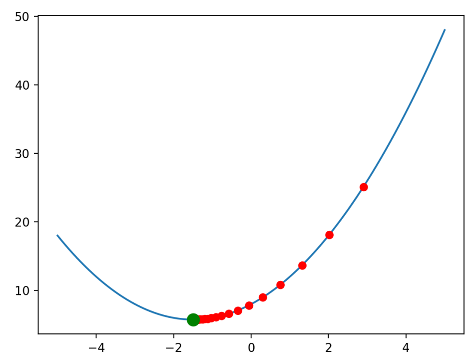

Figure 1: Convergence of Gradient Descent for Problem 1

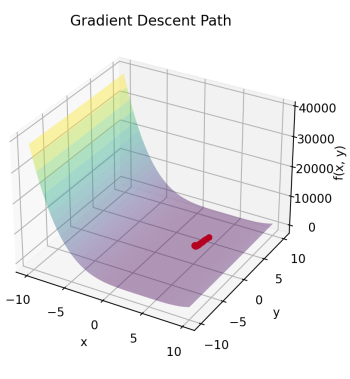

Figure 2: Convergence of Gradient Descent for Problem 2

4.3 Results

- Optimal x: -1.5425062289017262
- Optimal y: -1.5487091735363
- Minima: -0.9997176223493089

5 Problem 4 - 1-D Trigonometric

5.1 Function

- $f_5(x) = \cos(x)^4 - \sin(x)^3 - 4\sin(x)^2 + \cos(x) + 1$
- Derivative: $df5\_dx(x)$
- Optimization Range: $[0, 2\pi]$

5.2 Approach

- Adapted the generic gradient descent function for 1-D optimization.
- Defined initial point as $x = 3$, learning rate as 0.05, and number of iterations = 1000
- Performed optimization with gradient descent.
- Visualized the convergence using Matplotlib.

5.3 Results

- Optimal x: 1.661660812043789
- Minima: -4.045412051572552

6 Conclusion

In this assignment, we successfully implemented gradient descent optimization for a variety of functions. The generic gradient descent function allowed us to optimize functions in both 1-D and 2-D domains. The results and visualizations demonstrate the convergence of gradient descent for each problem.

The code and results can be found in the accompanying Python file. Overall, this assignment provided valuable experience in optimizing functions using gradient descent and creating informative visualizations.

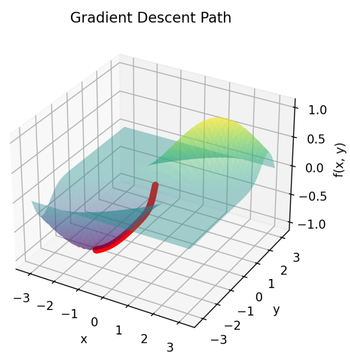

Figure 3: Convergence of Gradient Descent for Problem 3

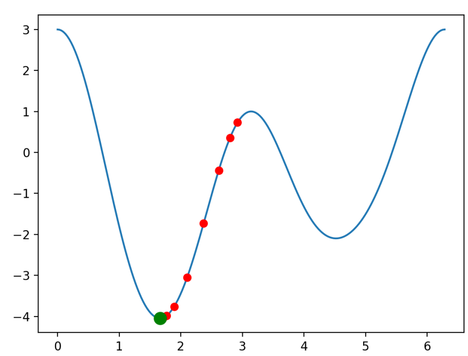

Figure 4: Convergence of Gradient Descent for Problem 4