

Warsaw University of Technology
Faculty of Electronics and Information Technology
Specialization: Computer Science
Course: Introduction to Artificial Intelligence

PROJECT DOCUMENTATION
Simple Gradient Algorithm

Author:

Monika Jung

Student ID:

331384

This document describes the implementation of the gradient descent algorithm for function optimization. The algorithm minimizes two objective functions: a simple quadratic function and Matyas' function. The study involves analysing the impact of different learning rates (alpha values) on convergence and visualizing the optimization process for multiple starting points.

Implementation Details

The function gradient descent is an iterative optimisation algorithm, used to find a local minimum of a given function.

Parameters:

- `f` (callable): The objective function to minimize.
- `start` (tuple): The starting point (x1, x2).
- `alpha` (float): The learning rate.
- `max_iter` (int): Maximum number of iterations.
- `tol` (float): The tolerance for stopping based on gradient norm.

Returns:

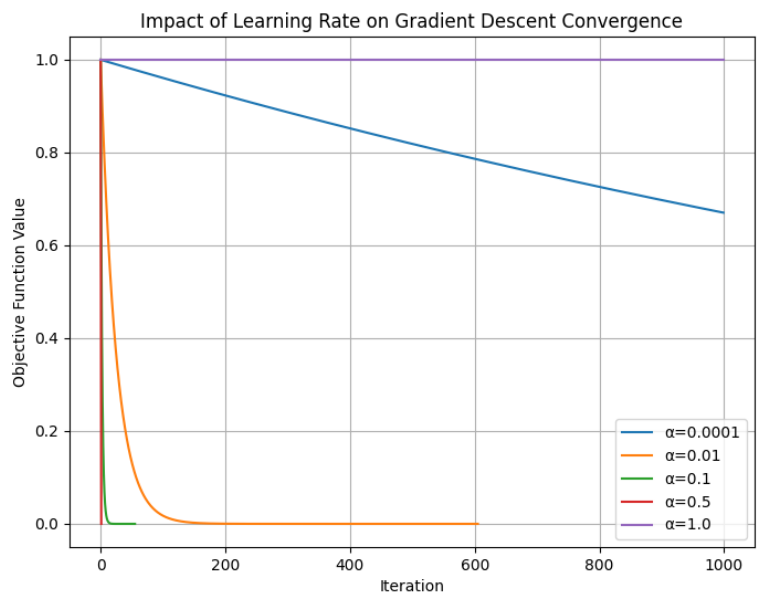
- `trajectory` (np.array): The trajectory of points visited during the descent.
- `function_values` (list): The values of the objective function at each step.

Two functions are used:

1. $f(x) = x_1^2 + x_2^2$
2. Matyas function (for 2 dimensions)

Observations

- A very small learning rate (e.g. $\alpha=0.0001$) means very slow progress towards the minimum
- A smaller learning rate (e.g. $\alpha = 0.01$) results in slow but stable convergence
- A large learning rate (e.g. $\alpha=0.5$) leads to rapid convergence but risks instability.
- An excessively large learning rate (e.g. $\alpha=1.0$) results in divergence, preventing convergence to the minimum.

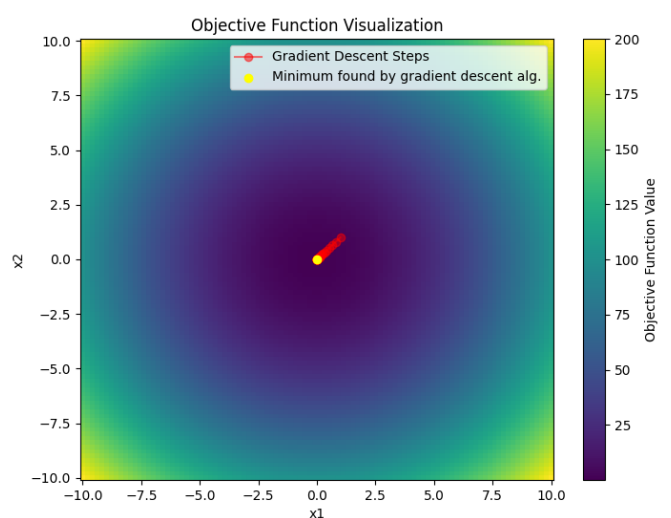
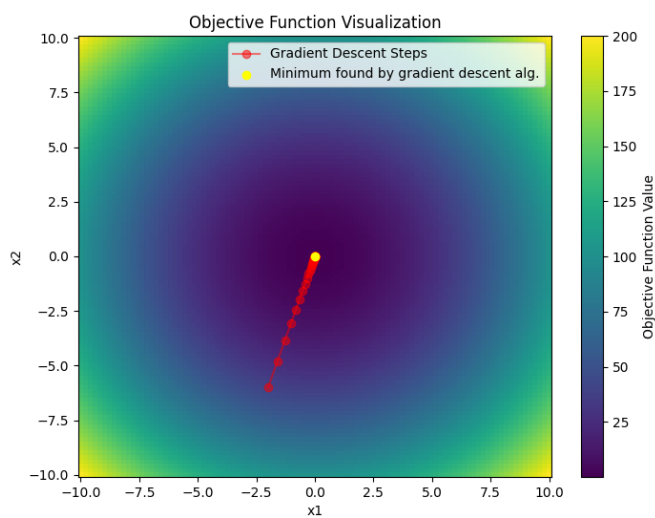
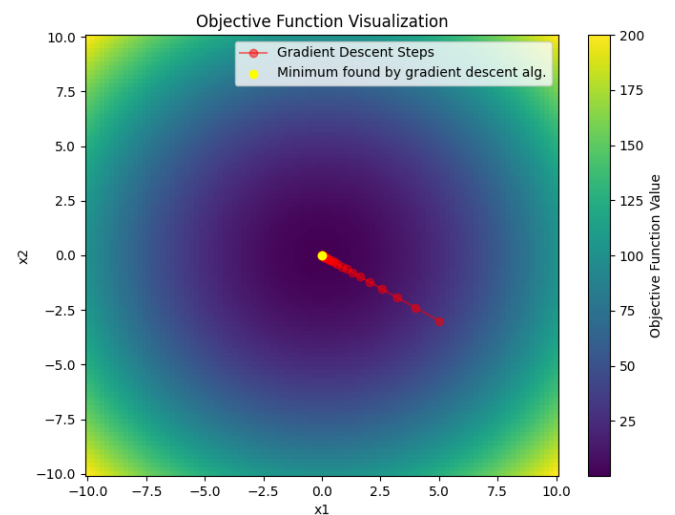
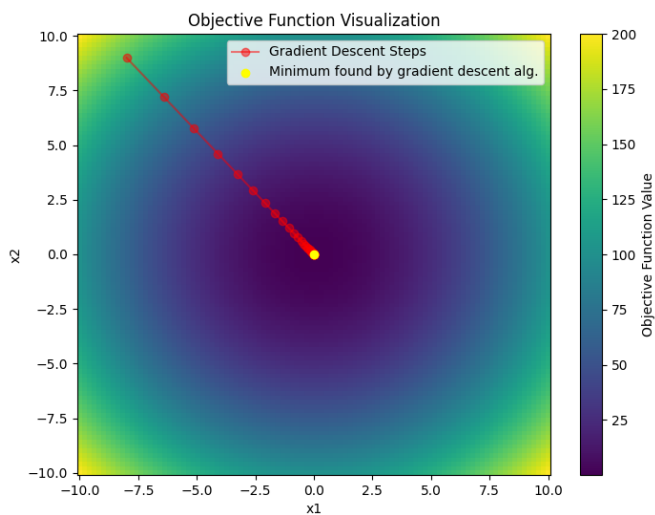


Based on the observations, the best learning rate should balance convergence speed and stability. That is why, for my algorithm I choose learning rate $\alpha=0.1$, as it provides efficient convergence .

Visualization of Gradient Descent for Multiple Starting Points

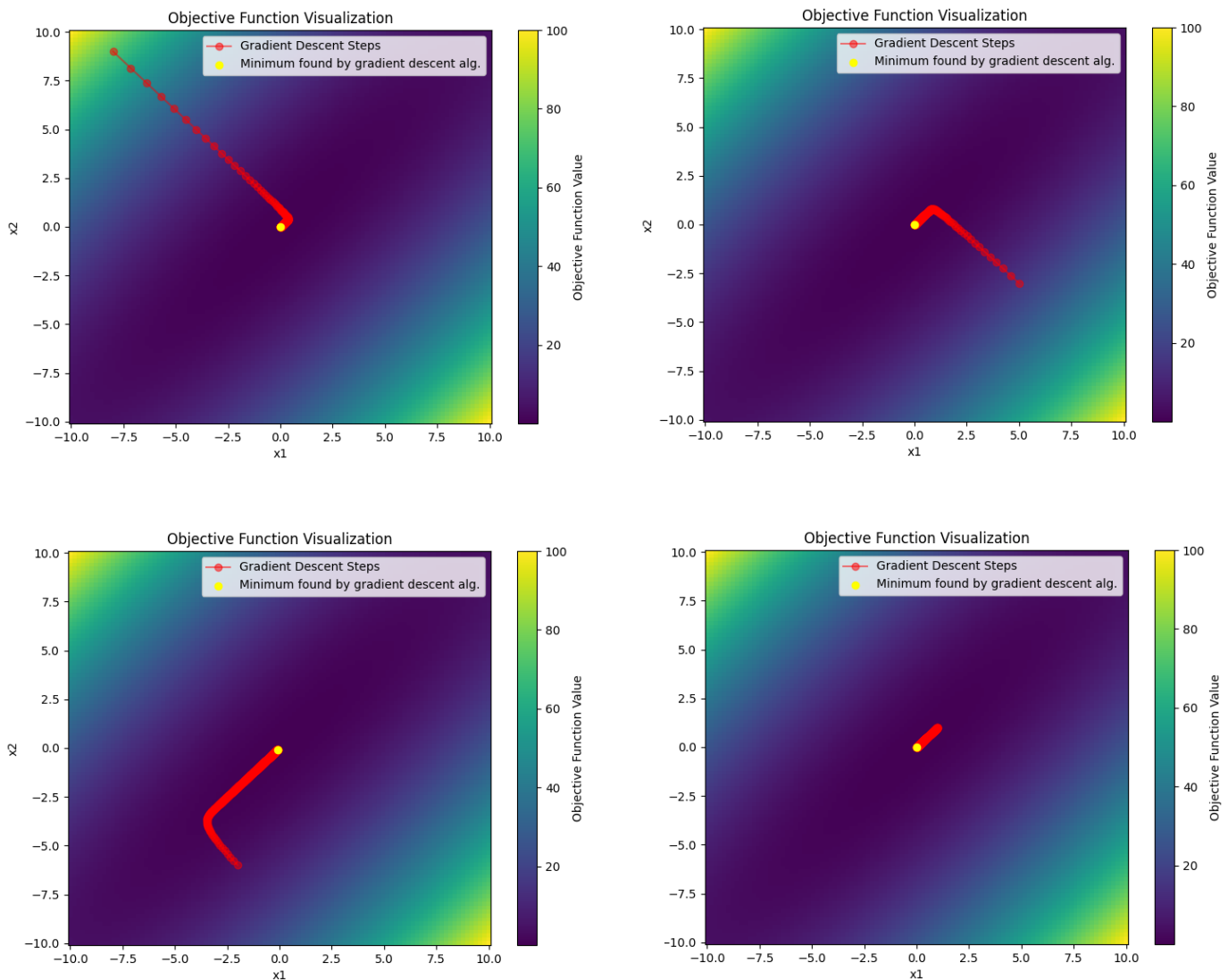
To analyse how the gradient descent algorithm behaves for different initial conditions, I tested three distinct starting points: $(-8, 9)$, $(5, -3)$, $(-2, -6)$, $(1, 1)$

Objective function plots:



The gradient descent paths are straight lines directed toward the minimum, confirming the expected behaviour of a quadratic function.

Matys function plot:



The descent paths show curvature due to the interaction between x_1 and x_2 , indicating the influence of the $-0.48x_1x_2$ term.

Despite differences in path shapes, all starting points successfully reach the minimum.

Conclusion

This was a successful implementation of the gradient descent algorithm for minimizing both the quadratic and Matyas functions. The choice of an appropriate learning rate allowed for optimized convergence, balancing speed and stability.

Overall, gradient descent proves to be an effective optimization method widely used in machine learning and numerical computing, demonstrating its reliability in finding function minimum.