MA3206 Artificial Intelligence

Assignment 2 – Task 1

Optimizer Performance on Non-Convex Functions

1. Introduction

This report presents a comparative study of different gradient-based optimization algorithms on non-convex objective functions. Non-convex optimization problems commonly arise in machine learning and artificial intelligence, where the presence of multiple local minima and saddle points makes convergence challenging.

2. Objectives

- To implement gradient-based optimization algorithms from scratch using Python.
- To study convergence behavior on non-convex objective functions.
- To analyze the impact of learning rate on convergence speed and stability.
- To compare robustness and efficiency of different optimizers.

3. Non-Convex Functions Considered

3.1 Rosenbrock Function

The Rosenbrock function is defined as:
$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$.
It contains a narrow curved valley leading to the global minimum at (1,1), which makes optimization particularly difficult for basic gradient-based methods.

3.2 sin(1/x) Function

The function $f(x) = \sin(1/x)$ is highly oscillatory near $x = 0$ and is non-convex with infinitely many local minima. To avoid numerical instability, the function is safely handled near $x = 0$ by defining $f(0) = 0$.

4. Optimization Algorithms

- Gradient Descent (GD)
- Gradient Descent with Momentum
- Adam Optimizer
- RMSprop
- Adagrad

5. Experimental Setup

Each optimizer was evaluated using learning rates $\alpha = 0.01$, 0.05, and 0.1. A gradient-norm–based stopping criterion was used, and divergence was detected when the gradient norm exceeded a predefined threshold. All experiments used identical initial conditions to ensure fair comparison.

6. Observations and Results

Rosenbrock Function:

- Gradient Descent converges very slowly due to the narrow curved valley.
- Momentum-based GD becomes unstable at higher learning rates and often diverges.
- Adam shows the fastest and most stable convergence across learning rates.
- RMSprop performs consistently but converges slower than Adam.
- Adagrad initially improves but stagnates due to rapid learning rate decay.

sin(1/x) Function:

- The oscillatory nature of the function leads to frequent gradient sign changes.
- Standard Gradient Descent struggles to converge reliably.
- Adaptive optimizers handle oscillations more effectively.
- Larger learning rates increase instability and divergence.

7. Impact of Hyperparameters

Learning rate plays a critical role in optimizer performance. Smaller learning rates ensure stability but slow convergence, while larger learning rates accelerate convergence at the cost of instability. Adaptive optimizers reduce sensitivity to learning rate selection and demonstrate greater robustness on non-convex problems.

8. Conclusion

This study demonstrates the challenges associated with non-convex optimization. Among the evaluated methods, Adam consistently outperforms others in terms of convergence speed and stability. The results highlight the importance of selecting appropriate optimizers and tuning hyperparameters in artificial intelligence applications.