

Homework 2

Objective

The goal of this assignment is to implement gradient descent to minimize a loss function over a 100×50 matrix x using NumPy. Only use a for loop to update at each step.

Problem

Given a matrix $x \in \mathbb{R}^{100 \times 50}$, implement gradient descent to minimize the following loss function:

$$f(X) = \frac{1}{2} \sum_{i,j} (X_{i,j} - A_{i,j})^2$$

Where A is a known matrix of the same size as x . This is essentially a **mean squared error** (MSE) loss function, where the goal is to make x as close as possible to a given matrix A .

Tasks

1. **Define the Loss Function:**

$$f(X) = \frac{1}{2} \sum_{i,j} (X_{i,j} - A_{i,j})^2$$

2. **Gradient of the Loss Function:**

$$\nabla f(X) = X - A$$

3. **Gradient Descent Algorithm:**

- Implement gradient descent to minimize the loss function.
- Use the update rule:

$$X_{\text{new}} = X - \alpha \cdot \nabla f(X)$$

where α is the learning rate.

4. **Convergence Criteria:**

- Stop the algorithm when the difference between the loss values in two consecutive iterations is smaller than a threshold (e.g., 10^{-6}), or after 1000 iterations.

5. **Initialization:**

- Initialize the matrix X with random values.
- Initialize the matrix A with random values.

6. **Track Progress:**

- Track the loss values over each iteration.

7. **Visualize Results:**

- Print the loss values.