

# Labwork 2: Linear Regression

April 30, 2025

## 1 Introduction

**Linear Regression:** Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data. The simplest form of linear regression is simple linear regression, which models the relationship between two variables by fitting a linear equation to the observed data.

In this lab, we implemented a simple linear regression algorithm using gradient descent to minimize the loss function. The loss function used is the Mean Squared Error (MSE), which measures the difference between the predicted and actual values.

## 2 Implementation

The implementation of linear regression in this lab uses the gradient descent optimization algorithm to find the best-fitting line for a given set of data points. The algorithm starts with initial values for the weights  $w_0$  (intercept) and  $w_1$  (slope) and iteratively updates them to minimize the loss function.

The loss function for a single data point  $(x_i, y_i)$  is defined as:

$$L_i = \frac{1}{2} (w_1 x_i + w_0 - y_i)^2$$

The gradients of the loss function with respect to  $w_0$  and  $w_1$  are computed as:

$$\begin{aligned} \frac{\partial L}{\partial w_0} &= \frac{1}{n} \sum_{i=1}^n (w_1 x_i + w_0 - y_i) \\ \frac{\partial L}{\partial w_1} &= \frac{1}{n} \sum_{i=1}^n (w_1 x_i + w_0 - y_i) x_i \end{aligned}$$

The weights are updated iteratively using the learning rate  $r$ :

$$w_0 \leftarrow w_0 - r \frac{\partial L}{\partial w_0}$$

$$w_1 \leftarrow w_1 - r \frac{\partial L}{\partial w_1}$$

The algorithm stops when the loss function converges below a specified threshold  $\epsilon$ .

### 3 Evaluation

The program was tested with a dataset loaded from a CSV file. The user provides the learning rate  $r$ , convergence threshold  $\epsilon$ , and initial values for  $w_0$  and  $w_1$ . The program outputs the updated weights and the final loss after convergence.

For example, testing with a dataset where  $y = x^4$  (or any other dataset provided in the CSV file), the program iteratively adjusts the weights to minimize the loss. Below is an example of the output:

```
Enter learning rate r: 0.0001
Enter convergence threshold epsilon: 9.452
Enter initial w0: 0
Enter initial w1: 0
Current w0: 0.010100000000000001, Current w1: 0.507, Loss: 5632.5
...
Final w0: 47.91192093041467, Final w1: 1.2635122635778553, Final loss: 9.451999985986241
```

### 4 Conclusion

In this labwork, we implemented a simple linear regression algorithm using gradient descent. The algorithm successfully minimized the loss function and found the best-fitting line for the given dataset.

An interesting finding is that the choice of learning rate  $r$  significantly affects the convergence speed. A small  $r$  leads to slow convergence, while a large  $r$  may cause the algorithm to diverge. Proper tuning of  $r$  and  $\epsilon$  is crucial for achieving optimal results.