Implementing Linear Regression Using a Multi-Layer Neural Network from Scratch

Mohd Darish Khan, Hardik Singh, Suryansh Jaiswal (Group - 4)

2101MC29, 2101MC19, 2101MC41

Abstract

This report details the implementation of a multi-layer neural network for linear regression using the Boston Housing dataset. The focus is on predicting the median value of homes (‘MEDV‘) based on the number of rooms (‘RM‘) and crime rate (‘CRIM‘). We cover data preprocessing, neural network architecture, training, evaluation, and additional experiments.

Introduction

Linear regression is used to predict a continuous outcome based on input features. In this task, we develop a multi-layer neural network from scratch to predict the median value of homes ('MEDV') using two features from the Boston Housing dataset: the number of rooms ('RM') and the crime rate ('CRIM').

Dataset Description and Preprocessing

The Boston Housing dataset contains multiple features related to housing prices. For this analysis, we focus on:

- Number of Rooms (`RM`)
- Crime Rate (`CRIM`)

We preprocess the data by normalizing `RM` and `CRIM` to ensure they are on a similar scale. The dataset is then split into a training set (80%) and a test set (20%).

Model Architecture

The neural network is designed with the following architecture:

- **Input Layer**: 2 neurons (corresponding to `RM` and `CRIM`).
- **First Hidden Layer**:
  - Number of Neurons: 5
  - Activation Function: ReLU (Rectified Linear Unit)
- **Second Hidden Layer**:
  - Number of Neurons: 3
  - Activation Function: ReLU
- **Output Layer**:
  - Number of Neurons: 1 (predicting the median value of homes)

Training the Network

The network is trained using Gradient Descent with various learning rates (e.g., 0.01, 0.001). We train the network for 1000 epochs and track the loss throughout the training process. Different optimizers are compared, including Basic Gradient Descent, Momentum, and Adam. The effects of these optimizers on model convergence and performance are analyzed.

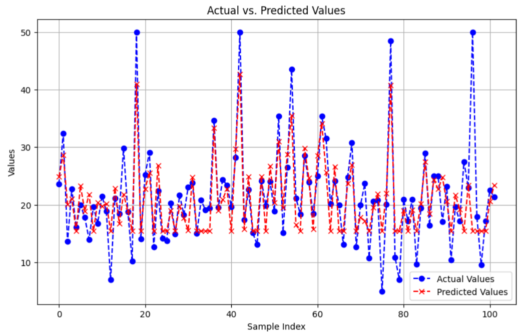

Figure 1: Gradient Descent

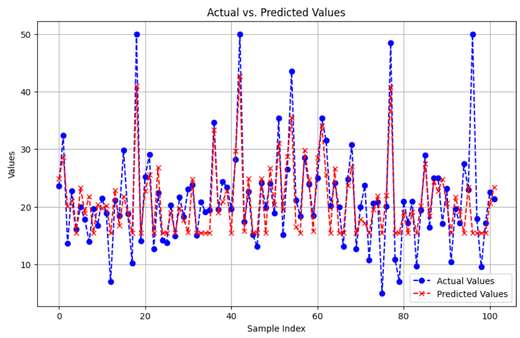

Figure 2: Momentum

Evaluation

The trained model is evaluated on the test set. Predictions are compared with actual values, and the results are visualized.

Evaluation metrics such as Mean Squared Error (MSE) are calculated to assess the model's performance on the test data.

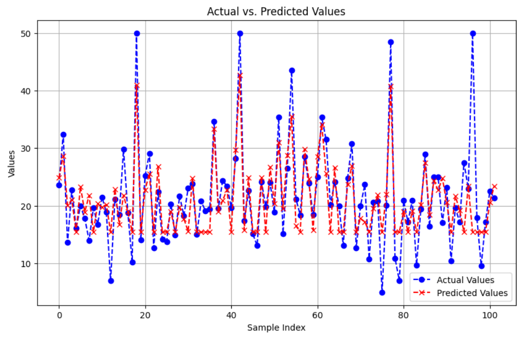

Figure 3: Adam

Bonus Experiments

To explore the impact of additional hidden layers, we add a third hidden layer with 2 neurons and compare the performance of networks with different numbers of hidden layers. The loss reduced from 91 to 87.

Additionally, we implement L2 regularization (weight decay) to observe its effect on overfitting. But, we see no considerable changes in the results. Hence we conclude that there was no overfitting.

Conclusion

The report presents the implementation and evaluation of a multi-layer neural network for linear regression. The impact of various optimizers, additional hidden layers, and regularization techniques on model performance are discussed.