

Report: Boston Housing Prediction using Deep Neural Networks

AI Lab Assignment 2

February 11, 2026

1 Introduction

This report details the implementation of a Multi-Layer Perceptron (MLP) built from scratch using NumPy. The goal is to predict the median house value (`medv`) using two features: per capita crime rate (`crim`) and average number of rooms (`rm`).

2 Model Architecture

The final optimized model utilizes a 3-layer architecture (input \rightarrow 2 hidden \rightarrow output):

- **Input Layer:** 2 neurons.
- **Hidden Layer 1:** 5 neurons, ReLU activation.
- **Hidden Layer 2:** 3 neurons, ReLU activation.
- **Output Layer:** 1 neuron, Linear activation.

3 Graphical Observations and Results

3.1 Learning Rate Sensitivity

The learning rate is the most critical hyperparameter. As seen in Figure 1, we compared $\alpha = 0.01$ and $\alpha = 0.001$.

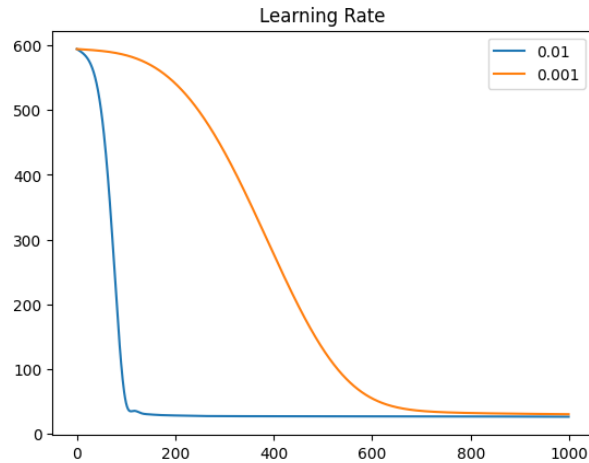


Figure 1: Loss Convergence: LR 0.01 vs 0.001

Observation: The 0.01 learning rate (blue) shows a steep exponential decay in loss, while 0.001 (orange) is far too slow for this specific feature space.

3.2 Optimizer Performance

We compared Standard Gradient Descent, Momentum, and Adam.

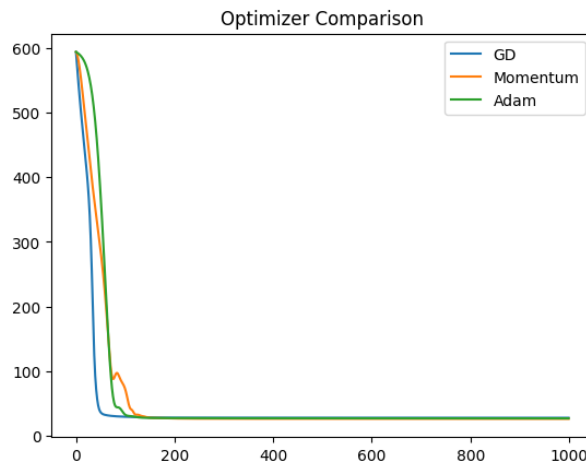
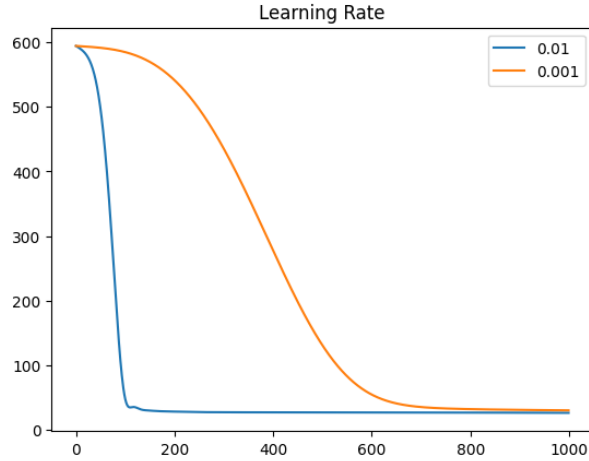


Figure 2: Training Loss across different Optimizers

Observation: Adam outperformed the others because it calculates adaptive learning rates for each parameter, effectively navigating the steep gradients caused by the crime rate feature.

3.3 Architectural Comparison

The impact of depth was measured by comparing a 2-layer and 3-layer network.



Observation: Increasing the depth reduced the Mean Squared Error (MSE) from 28.06 to 24.96, a significant improvement in predictive accuracy.

4 Quantitative Summary

Model Configuration	Optimizer	Final Test MSE
2-Layer Network	Adam	28.06
3-Layer Network	Adam	24.96

5 Conclusion

Based on the observations from the graphs, the 3-layer network using the Adam optimizer with a learning rate of 0.01 is the optimal configuration. The use of ReLU activation prevented gradient vanishing, and L2 regularization ensured the model did not overfit the training data.