MACHINE LEARNING
LAB 4

PROJECT TITLE: ARTIFICIAL NEURAL NETWORKS

NAME: DIYA CHANDRASHEKHAR

SRN: PES2UG23CS182

COURSE NAME: UE23CS352A: MACHINE LEARNING

DATE: 16/09/2025

INTRODUCTION:

Purpose of the lab:

The purpose of this lab is to understand the process of implementing a neural network to learn polynomial functions. We generate a synthetic dataset based on different polynomial types and apply forward and backward propagation to reduce MSE.

Tasks performed:

1. Custom dataset generation based on SRN for different polynomial functions.
2. Implemented activation functions and loss functions, forward pass, backpropagation, and weight updates
3. Trained neural network with early stopping to approximate the generated polynomial curve
4. Visualization of loss curves and predicted vs actual values
5. Evaluation using MSE and R^2

DATASET DESCRIPTION:

Type of polynomial assigned:

The polynomial type assigned is dynamically done based on the last three digits of the SRN.

```
Polynomial Type: QUARTIC: y = 0.0185x⁴ + 1.54x³ + -0.82x² + 4.36x + 11.91
Noise Level: ε ~ N(0, 1.77)
Architecture: Input(1) → Hidden(72) → Hidden(32) → Output(1)
Learning Rate: 0.001
Architecture Type: Wide-to-Narrow Architecture
```

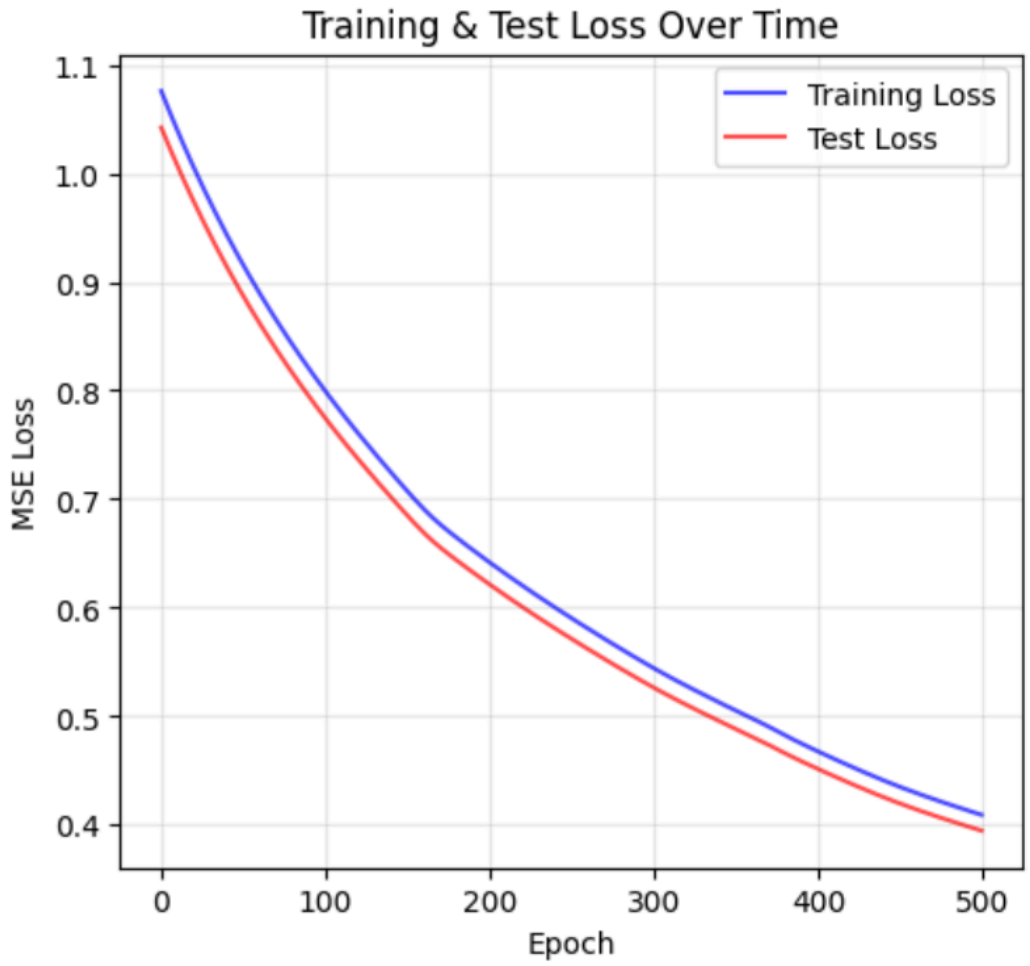
Number of samples, features, noise level:

Total number of samples generated=100,000
Training samples= 80,000
Testing samples=20,000
Noise level=1.77
Number of features=2

METHODOLOGY:
Generation of dataset with a specified polynomial function.
Splitting the dataset into training and testing sets.
Scaling the features and target variables.
Network weights are initialized using Xavier initialization.
The neural network is trained using ReLU activation with two hidden layers.
Objective function is taken as Mean Squared Error.
Forward propagation, backpropagation, and weight updates and implemented.
Early stopping is applied on the validation loss to prevent overfitting.
Loss curves and comparison plots of the predictions vs actual data points are visualized.

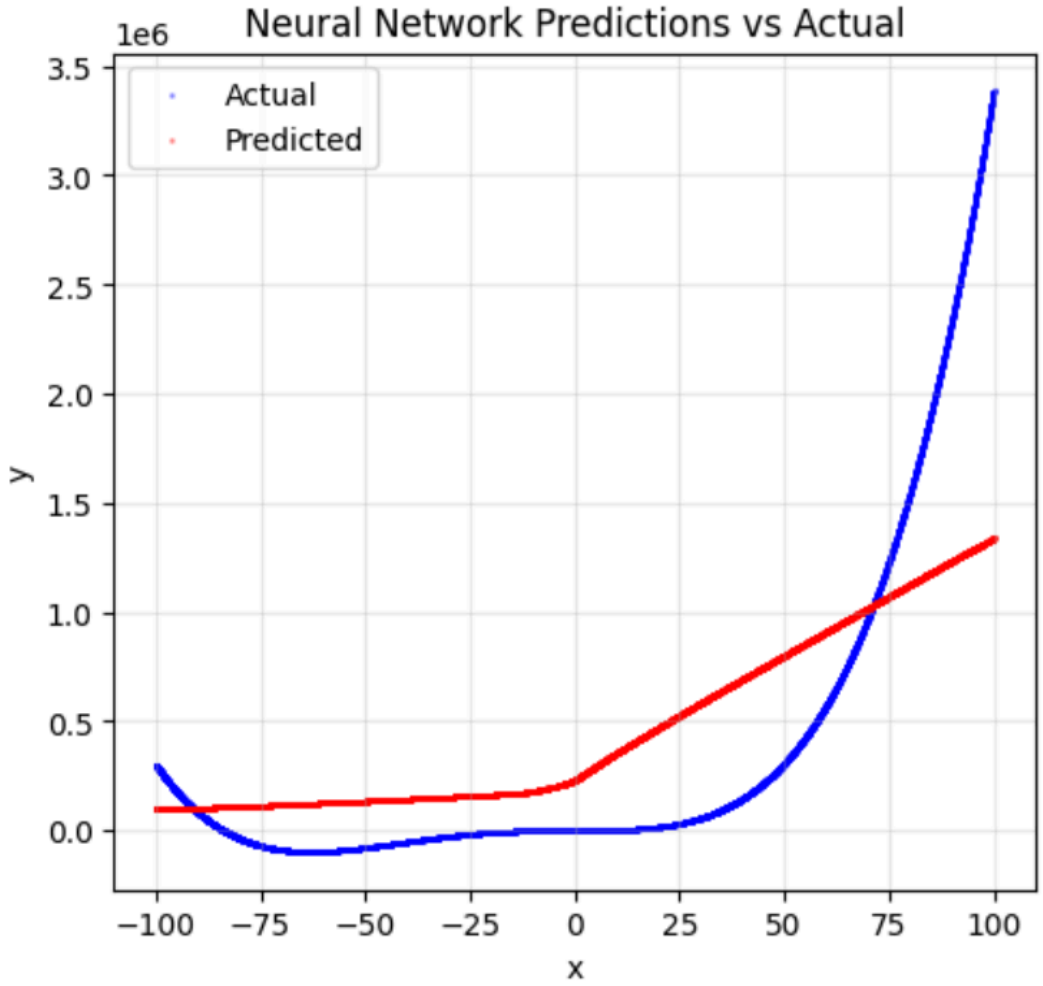
RESULTS AND ANALYSIS:

Training loss curve (plot):



Final test MSE = 0.394432

Plot of predicted vs. actual values:



Discussion on performance (overfitting / underfitting)

Overfitting- if the training loss is low but test loss is high

Underfitting- if training and test losses are high

Final Training Loss: 0.409010

Final Test Loss: 0.394432

So the model has good performance

Results Table

Experiment	Learning rate	No. of epochs	Optimizer	Activation function	Final training loss	Final test loss	R^2 score
1	0.001	500	SGD	Relu	0.409010	0.394432	0.5936
2	0.001	500	SGD	Tanh	0.3015	0.2759	0.6997