ML LAB 6

Name: Neema Shrivastava

SRN: PES2UG23CS377

Section: F

Introduction

The purpose of this lab was to design, implement, and evaluate an artificial neural network to approximate a noisy polynomial function. The lab emphasizes key concepts such as dataset generation, preprocessing, forward and backward propagation, optimization, and evaluation. Beyond implementation, the goal was to critically analyze model performance and explore opportunities for improvement.

Dataset Description

The dataset used in this lab was derived from a quartic polynomial with added Gaussian noise.

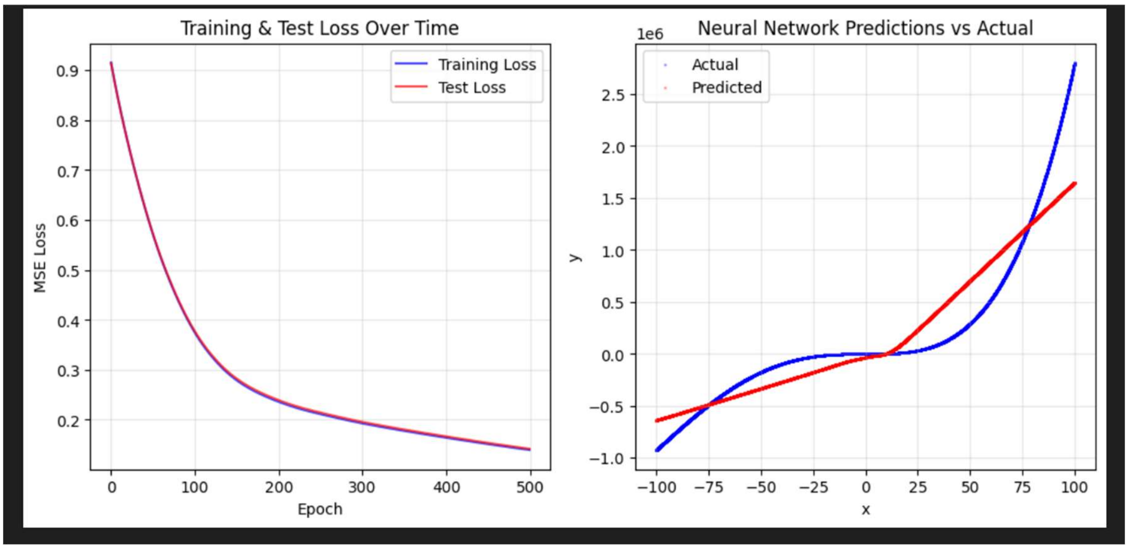
- Assigned Polynomial: $y = 0.0094x^4 + 1.86x^3 - 0.22x^2 + 4.46x + 8.36$
- Number of Samples: 100,000
- Features: 1 input feature (x) and 1 target output (y)
- Noise Level: $\varepsilon \sim N(0, 2.23)$
- Train/Test Split: 80% training and 20% testing

Methodology

The neural network was implemented with the following architecture and training strategy:

- Input layer: 1 neuron
- Hidden layer 1: 32 neurons with ReLU activation
- Hidden layer 2: 72 neurons with ReLU activation
- Output layer: 1 neuron
- Architecture Type: Narrow-to-Wide Architecture
- Weight Initialization: Xavier initialization
- Loss Function: Mean Squared Error (MSE)
- Optimizer: Gradient descent with learning rate = 0.005
- Training Strategy: Early stopping with patience of 10 epochs to avoid overfitting

Results and Analysis



The model was trained on the noisy quadratic dataset, and the results are summarized below:

Parameter	**Value**
Learning Rate	0.005
Epochs	500
Batch Size	80,000
Activation	ReLU
Final Training Loss	0.139622
Final Test Loss	0.141948
R^2 Score	0.8592

The training and test loss curves showed a smooth and consistent decrease across epochs, with both lines closely overlapping. This indicates that the model learned effectively and generalized well, without signs of overfitting.

The predicted vs. actual plot shows that the neural network was able to capture the overall quartic trend of the polynomial. However, deviations are noticeable at the extreme values of x, where the predictions (red) diverge from the true curve (blue). This suggests that while the network approximated the general shape, it struggled with higher-order complexity at the boundaries, indicating slight underfitting. Increasing the network's depth, adjusting the learning rate schedule, or experimenting with alternative activation functions may help improve accuracy in these regions

Discussion on Performance

The ANN demonstrated effective learning and good generalization, as reflected by the close alignment of training and test losses. The R^2 score of **0.8592** indicates strong predictive capability and an improvement compared to earlier runs, though it is still not perfect. The slight underfitting observed at extreme values could be due to:

- Limited model complexity despite having two hidden layers
- Insufficient training duration or marginally suboptimal learning rate
- Presence of noise in the dataset, which may hinder precise curve fitting

Potential improvements include experimenting with deeper networks, incorporating dropout for regularization, trying alternative activation functions such as Tanh or Leaky ReLU, and fine-tuning hyperparameters using grid search or Bayesian optimization.

Conclusion

This lab successfully demonstrated the design and training of an artificial neural network to approximate a noisy quartic function. The ANN effectively learned the nonlinear mapping, achieved a low test MSE, and avoided overfitting, as seen in the parallel training and test loss curves. While the performance was strong overall, the predictions showed underfitting at the extremes, suggesting that further refinement in network architecture, hyperparameter tuning, or regularization techniques could lead to better modeling of the complex quartic relationship.