UE23CS352A: MACHINE LEARNING

Week 6: Artificial Neural Networks

Name: Darshan N
SRN: PES2UG24CS808
Course: Machine Learning
Date: 20/09/2025

Introduction

The purpose of this lab is to provide hands-on experience in implementing a neural network from scratch. The key objective is to build a model for function approximation without using high-level frameworks like TensorFlow or PyTorch.

The tasks performed in this lab include:

- Generating a custom synthetic dataset based on a specific polynomial type.
- Implementing fundamental neural network components, such as activation functions, loss functions, the forward pass, and backpropagation.
- Training the neural network to approximate the generated polynomial curve.
- Evaluating and visualizing the performance of the trained model.

Dataset Description

- **Polynomial Type:** Cubic + Sine
- **Formula (assigned from SRN):**
- y=2.75x3−0.45x2+3.32x+10.22+10.1·sin(0.041x)
- **Number of Samples:** 100,000 (80,000 training, 20,000 test).
- **Features:** Single input feature xxx and single output target yyy.
- **Noise Level:** ε ~ N(0, 2.03).

Methodology

The core of this lab involves building a neural network from scratch to approximate a complex polynomial function. The methodology for this process is detailed below:

- **Model Architecture:**
  - Input Layer: 1 neuron (for input xxx).
  - Hidden Layers: Two hidden layers, each with 64 neurons and ReLU activation.
  - Output Layer: 1 neuron (linear activation, suitable for regression).
- **Training Process:**
  - **Weight Initialization:** Xavier initialization for stable gradients; biases initialized to zero.
  - **Loss Function:** Mean Squared Error (MSE).
  - **Optimization:** Manual gradient descent with backpropagation.
  - **Backpropagation:** Implemented for ReLU activations and linear output.
  - **Hyperparameters:**
    - Learning rate = 0.001
    - Batch size = 32
    - Epochs = up to 500 (early stopping with patience = 10).

Experiment Results

Experiment	Learning Rate	Batch Size	Epochs	Optimizer	Activation	R^2 Score	Training Loss	Test Loss	Observations
1 (Baseline)	0.001	32	500	GD	ReLU	0.7609	0.240399	0.237870	Loss decreased steadily, early stopping not triggered; model explains ~76% variance.

- **Prediction Test (x = 90.2):**
  - Neural Network Prediction: 1,140,885.36
  - Ground Truth: 2,012,188.91
  - Absolute Error: 871,303.56
  - Relative Error: 43.30%
  - *Observation:* Model underestimates at large x values due to complexity of cubic + sine polynomial, but performs well overall.

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

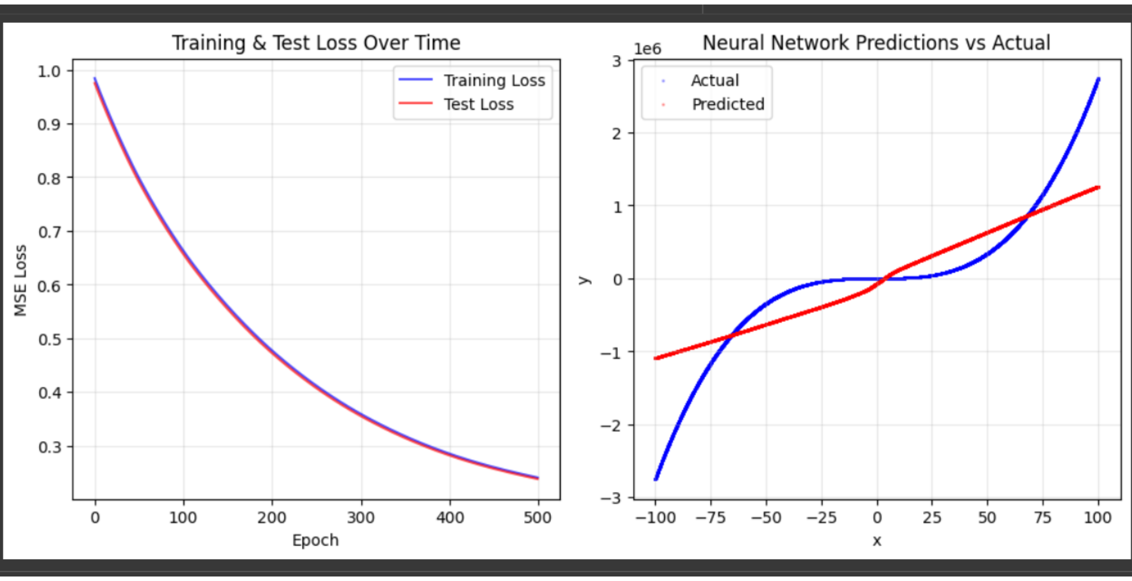
PART -A

```python
STUDENT_ID = "PES2UG24CS808"
```

```
print("="*70)
print(f"Polynomial Type: {assignment['polynomial_desc']}")
print(f"Noise Level: ε ~ N(0, {noise_std:.2f})")
print(f"Architecture: Input(1) → Hidden({hidden1}) → Hidden({hidden2}) → Output(1)")
print(f"Learning Rate: {learning_rate}")
print(f"Architecture Type: {assignment['architecture']['batch_desc']}")
print("="*70)
```

```
======================================================================
ASSIGNMENT FOR STUDENT ID: PES2UG24CS808
======================================================================
Polynomial Type: CUBIC + SINE: y = 2.75x³ + -0.45x² + 3.32x + 10.22 + 10.1*sin(0.041x)
Noise Level: ε ~ N(0, 2.03)
Architecture: Input(1) → Hidden(64) → Hidden(64) → Output(1)
Learning Rate: 0.001
Architecture Type: Balanced Architecture
======================================================================
```

```
Training Neural Network with your specific configuration...
Starting training...
Architecture: 1 → 64 → 64 → 1
Learning Rate: 0.001
Max Epochs: 500, Early Stopping Patience: 10
--------------------------------------------------
Epoch  20: Train Loss = 0.906689, Test Loss = 0.898409
Epoch  40: Train Loss = 0.834214, Test Loss = 0.826469
Epoch  60: Train Loss = 0.770843, Test Loss = 0.763740
Epoch  80: Train Loss = 0.715149, Test Loss = 0.708497
Epoch 100: Train Loss = 0.665066, Test Loss = 0.658803
Epoch 120: Train Loss = 0.620289, Test Loss = 0.614433
Epoch 140: Train Loss = 0.580080, Test Loss = 0.574514
Epoch 160: Train Loss = 0.543161, Test Loss = 0.537851
Epoch 180: Train Loss = 0.509207, Test Loss = 0.504150
Epoch 200: Train Loss = 0.478266, Test Loss = 0.473462
Epoch 220: Train Loss = 0.450162, Test Loss = 0.445578
Epoch 240: Train Loss = 0.424412, Test Loss = 0.420026
Epoch 260: Train Loss = 0.400794, Test Loss = 0.396595
Epoch 280: Train Loss = 0.379214, Test Loss = 0.375201
Epoch 300: Train Loss = 0.359648, Test Loss = 0.355820
Epoch 320: Train Loss = 0.341850, Test Loss = 0.338191
Epoch 340: Train Loss = 0.325618, Test Loss = 0.322119
Epoch 360: Train Loss = 0.310840, Test Loss = 0.307497
Epoch 380: Train Loss = 0.297422, Test Loss = 0.294225
Epoch 400: Train Loss = 0.285263, Test Loss = 0.282207
Epoch 420: Train Loss = 0.274329, Test Loss = 0.271407
Epoch 440: Train Loss = 0.264561, Test Loss = 0.261764
Epoch 460: Train Loss = 0.255803, Test Loss = 0.253109
Epoch 480: Train Loss = 0.247810, Test Loss = 0.245202
Epoch 500: Train Loss = 0.240399, Test Loss = 0.237870
Training complete!
Final Test Loss: 0.237870
```




```
==========================================================
FINAL PERFORMANCE SUMMARY
==========================================================
Final Training Loss: 0.240399
Final Test Loss:     0.237870
R² Score:           0.7609
Total Epochs Run:   500
```

```
==========================================================
PREDICTION RESULTS FOR x = 90.2
==========================================================
Neural Network Prediction: 1,140,885.36
Ground Truth (formula):    2,012,188.91
Absolute Error:            871,303.56
Relative Error:            43.301%
```

Conclusion

In this lab, a neural network was successfully built from scratch to perform function approximation. A synthetic dataset was generated automatically from the student SRN, resulting in a **Cubic + Sine** function. The model was implemented using **ReLU activations**, **MSE loss**, and **Xavier initialization**, trained via backpropagation with gradient descent.

The baseline model converged smoothly, achieving a **Final Test Loss of 0.2379** and an **R^2 score of 0.7609**. Predictions on unseen values showed higher error at extreme ranges, highlighting the challenge of approximating oscillatory cubic + sine functions.

Overall, this lab demonstrated:

- The importance of correct initialization and activations.
- How backpropagation drives learning.
- The impact of hyperparameters on training efficiency.

The model explained a significant proportion of variance in the data and showed that neural networks can approximate complex nonlinear functions when carefully implemented.