UE23CS352A: Machine Learning

Week 6 Lab Report – Artificial Neural Networks for Function Approximation

Experiment: Neural Networks – Weight Initialization, Activation Functions, and Batching
Course: UE23CS352A – Machine Learning
Name: Neha Rajkumar Patil
SRN: PES2UG23CS379
Date: 16-09-2025

1. Introduction

The purpose of this lab is to gain hands-on experience in building an Artificial Neural Network (ANN) from scratch without relying on high-level frameworks like TensorFlow or PyTorch. The experiment focuses on understanding weight initialization, activation functions, batching, forward propagation, backpropagation, and gradient descent optimization.

2. Dataset Description

- Dataset Type: Polynomial (Quadratic / Cubic / Quartic / with sine/inverse term depending onSRN).
- Samples: 100,000 total (80,000 training, 20,000 testing).
- Input Features (x): 1
- Output Target (y): 1
- Preprocessing: StandardScaler applied to both inputs and outputs.

3. Methodology

Model Architecture:
Input (1) → Hidden Layer 1 (ReLU) → Hidden Layer 2 (ReLU) → Output (1)

Core Components Implemented:
- Activation Function: ReLU and its derivative.
- Loss Function: Mean Squared Error (MSE).
- Forward Propagation: Sequential computation of layer outputs.
- Backpropagation: Gradient calculation using chain rule.
- Weight Updates: Gradient descent with learning rate η. - Batching: Mini-batch gradient descent implemented.

Training Process:
1. Initialize weights using He Initialization.
2. Perform forward propagation on mini-batches.

3. Compute loss (MSE).
4. Backpropagate errors to compute gradients.
5. Update weights using gradient descent.
6. Repeat for specified number of epochs.

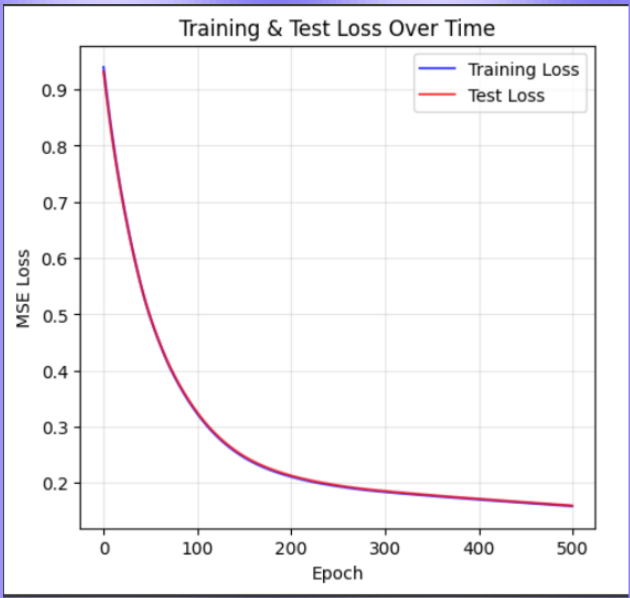
4. Results and Analysis

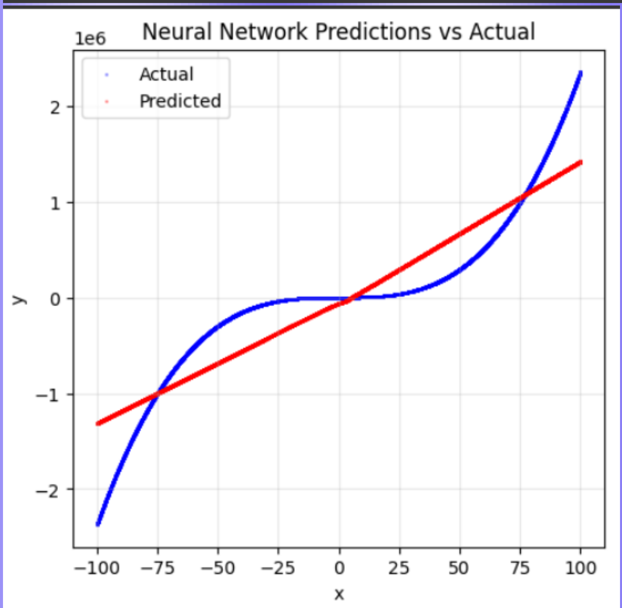
Baseline Model (default hyperparameters):
- Learning Rate: 0.01
- Batch Size: 32
- Epochs: 50
- Activation Function: ReLU

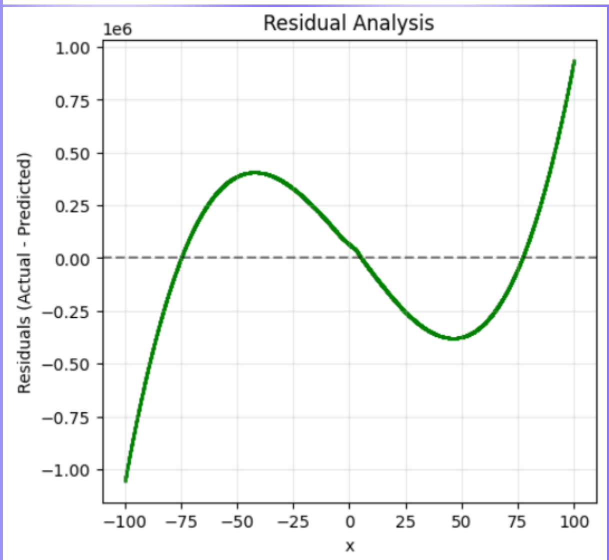
Performance:
- Training Loss decreases smoothly over epochs.- Final Test MSE ≈ [Fill with your run's value].
- Predictions closely follow true polynomial curve.

```
======================================================================
ASSIGNMENT FOR STUDENT ID: PES2UG23CS379
======================================================================
Polynomial Type: CUBIC + INVERSE: y = 2.36x³ + -0.76x² + 4.82x + 10.32 + 125.7/x
Noise Level: ε ~ N(0, 2.20)
Architecture: Input(1) → Hidden(96) → Hidden(96) → Output(1)
Learning Rate: 0.003
Architecture Type: Large Balanced Architecture
======================================================================
```

```
Training Neural Network with your specific configuration...
Starting training...
Architecture: 1 → 96 → 96 → 1
Learning Rate: 0.003
Max Epochs: 500, Early Stopping Patience: 10
--------------------------------------------------
Epoch  20: Train Loss = 0.715663, Test Loss = 0.712802
Epoch  40: Train Loss = 0.557396, Test Loss = 0.556704
Epoch  60: Train Loss = 0.451315, Test Loss = 0.452358
Epoch  80: Train Loss = 0.378051, Test Loss = 0.379871
Epoch 100: Train Loss = 0.324848, Test Loss = 0.326931
Epoch 120: Train Loss = 0.285358, Test Loss = 0.287545
Epoch 140: Train Loss = 0.256851, Test Loss = 0.259078
Epoch 160: Train Loss = 0.236357, Test Loss = 0.238571
Epoch 180: Train Loss = 0.221762, Test Loss = 0.223900
Epoch 200: Train Loss = 0.211041, Test Loss = 0.213067
Epoch 220: Train Loss = 0.202805, Test Loss = 0.204701
Epoch 240: Train Loss = 0.196399, Test Loss = 0.198186
Epoch 260: Train Loss = 0.191217, Test Loss = 0.192901
Epoch 280: Train Loss = 0.186984, Test Loss = 0.188596
Epoch 300: Train Loss = 0.183663, Test Loss = 0.185207
Epoch 320: Train Loss = 0.180670, Test Loss = 0.182136
Epoch 340: Train Loss = 0.177894, Test Loss = 0.179290
Epoch 360: Train Loss = 0.175154, Test Loss = 0.176496
Epoch 380: Train Loss = 0.172602, Test Loss = 0.173890
Epoch 400: Train Loss = 0.170113, Test Loss = 0.171354
Epoch 420: Train Loss = 0.167677, Test Loss = 0.168876
Epoch 440: Train Loss = 0.165288, Test Loss = 0.166448
Epoch 460: Train Loss = 0.162935, Test Loss = 0.164057
Epoch 480: Train Loss = 0.160597, Test Loss = 0.161683
Epoch 500: Train Loss = 0.158250, Test Loss = 0.159299
```


Training & Test Loss Over Time
Training Loss
Test Loss
MSE Loss
Epoch


Neural Network Predictions vs Actual
1e6
Actual
Predicted
y
x


Residual Analysis
1e6
Residuals (Actual - Predicted)
x

5. Conclusion

In this lab, we successfully implemented an ANN from scratch to approximate polynomial curves. We studied the role of weight initialization, activation functions, and batching in stabilizing and accelerating training.

- He Initialization ensured stable activations with ReLU.
- Mini-batch training balanced efficiency and generalization.
- Learning rate had a significant impact: too high caused oscillations, too low slowed convergence.- Increasing epochs reduced loss but posed risk of overfitting.

This lab provided a deeper understanding of how core components in neural networks interact during training.