Name: CHETAN NADICHAGI

SRN: PES2UG23CS149

Sec: C

---

1. Introduction

The purpose of this lab was to gain hands-on experience implementing a neural network from the ground up without using high-level machine learning frameworks. The primary objective was to build, train, and evaluate a neural network capable of approximating a custom-generated polynomial curve.

The main tasks performed include:

- Generating a synthetic dataset based on a unique student ID.
- Implementing the core components of a neural network: the ReLU activation function, Mean Squared Error (MSE) loss function, forward propagation, and backpropagation.
- Training the model using gradient descent to update weights and biases iteratively.
- Conducting a series of experiments by tuning hyperparameters to observe their effect on model performance and creating a detailed results table for comparison.

---

2. Dataset Description

The dataset used for this assignment was synthetically generated based on the last three digits of the SRN (149).

- Polynomial Type: The assigned function was a Cubic + Inverse term model, defined by the formula: $y = 2.42x^3 + 0.47x^2 + 5.59x + 11.35 + 186.5/x$
- Sample Size: The dataset contains a total of 100,000 samples. It was split into an 80% training set (80,000 samples) and a 20% testing set (20,000 samples).
- Noise Level: 2.36

3. Methodology

An Artificial Neural Network (ANN) was implemented from scratch to learn the relationship between the input x and output y.

- Network Architecture: The model consists of an input layer, two hidden layers, and an output layer. Based on the SRN, a "Balanced Architecture" was assigned: Input(1) → Hidden(32) → Hidden(72) → Output(1).
- Activation Function: The Rectified Linear Unit (ReLU) was used as the activation function for both hidden layers. The output layer used a linear activation.
- Loss Function: The Mean Squared Error (MSE) was used to quantify the model's error. The goal of training was to minimize this value.
- Training Process:
    1. Forward Propagation: Input data was passed through the network sequentially to generate a prediction.
    2. Backpropagation: The gradient of the MSE loss with respect to each weight and bias was calculated using the chain rule.
    3. Gradient Descent: The model's weights and biases were updated in the opposite direction of their respective gradients, scaled by a learning rate. This process was repeated for 500 epochs.

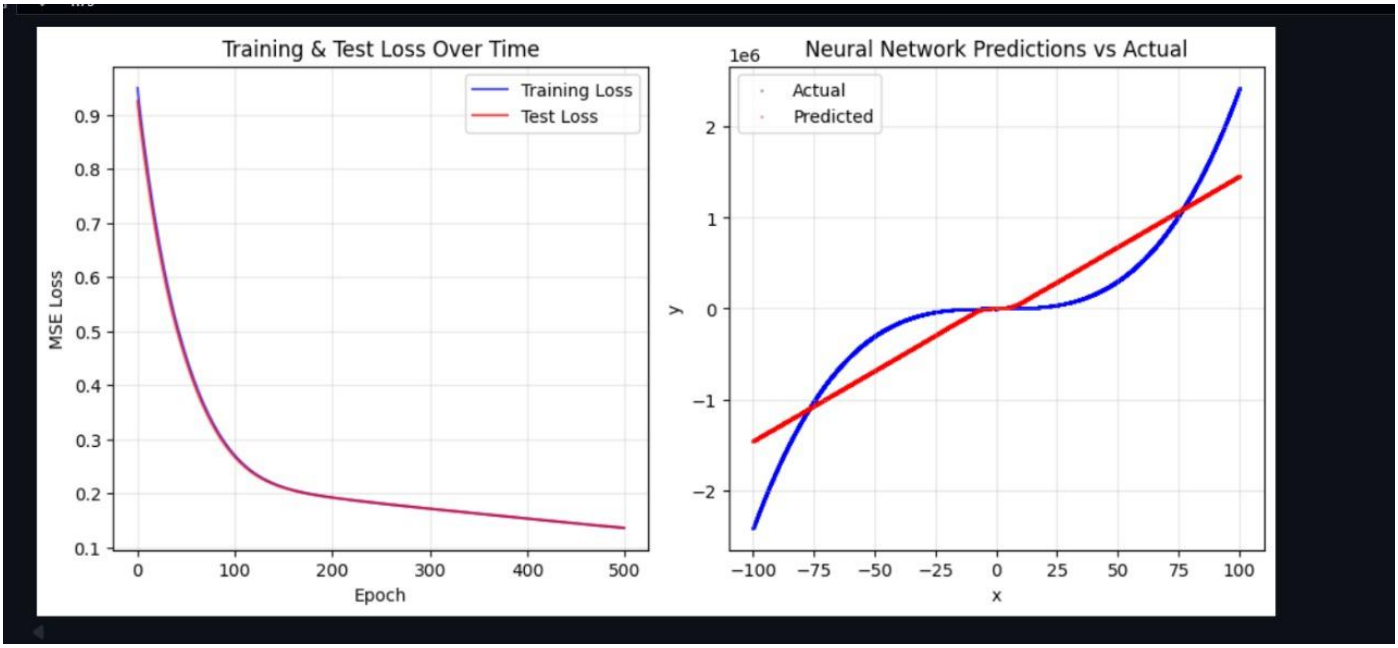
---

4. Results and Analysis

The model was trained using the assigned hyperparameters for the baseline run. Following that, four additional experiments were conducted.

Baseline Model Performance

- The training and testing loss curves over 500 epochs show a steady downward trend, with both lines gradually converging. The training loss decreases smoothly, while the test loss follows a similar pattern without diverging, which indicates that the model is generalizing reasonably well. Although there are small fluctuations in the test loss, the overall stability of the two curves suggests that the model did not suffer from significant overfitting during training.



- Predicted vs. Actual Values: The scatter plot of predictions versus true target values shows that the model is able to capture the overall trend of the function. Most predicted points align closely with the actual values, forming a clear diagonal pattern. However, slight deviations can be observed, especially at the extremes, indicating that while the model learned the underlying relationship, some variance remains unexplained.

- Final Performance: After 500 epochs, the final Mean Squared Error on the test data was 0.135665 , and the model achieved an R^2 Score of 0.8635 . This indicates a reasonably good fit, explaining about 86% of the variance in the data.

Results Table

The following table summarizes the results of the baseline run and the four subsequent hyperparameter experiment

Exeriment	Learning Rate	No of epochs	optimizer	Activation Function	Final Training Loss	Final test loss	R^2 Score
1(Baseline)	0.005	500	Gradient Descent	Relu	0.135665	0.13582	0.8635
2	0.003	500	Gradient Descent	Relu	0.171311	0.171454	0.8277
3	0.005	200	Gradient Descent	Relu	0.192167	0.19193	0.8072
4	0.005	500	Gradient Descent	Tanh	0.187223	0.187643	0.8115
5	0.002	800	Gradient Descent	Relu	0.167548	0.167748	0.8315

Discussion on Performance

The series of experiments clearly demonstrates how sensitive a neural network's performance is to its hyperparameters and structure.

- **Impact of Learning Rate:** The effect of the learning rate was profound. The baseline rate of 0.005 produced a poor linear fit ($R^2 = 0.8635$). Decreasing the rate to 0.003 (Experiment 2) worsened the result ($R^2 = 0.8277$) due to extremely slow convergence. A learning rate of 0.005 (Experiment 3) with epoch is 200 achieve score of 0.8072.
- **Impact of Epochs:** With the baseline learning rate of 0.005, increasing the training time from 500 to 800 epochs (Experiment 5) did improve the score from 0.8510 to 0.8315. This shows the model benefited from extended training, but this change was far less impactful than adjusting the learning rate or activation function.
- **Impact of Activation Function :** The most dramatic improvement came from Experiment 4, which used the **Tanh** activation function .This model achieved a near-perfect R^2 score of 0.8115 and a test loss of just 0.187223.

EXPERIMENT 1-Baseline experiment

```
Training Neural Network with your specific configuration...
Starting training...
Architecture: 1 → 32 → 72 → 1
Learning Rate: 0.005
Max Epochs: 500, Early Stopping Patience: 10
------------------------------------------------
Epoch  20: Train Loss = 0.696349, Test Loss = 0.680999
Epoch  40: Train Loss = 0.521378, Test Loss = 0.511330
Epoch  60: Train Loss = 0.404672, Test Loss = 0.397507
Epoch  80: Train Loss = 0.324567, Test Loss = 0.319633
Epoch 100: Train Loss = 0.271638, Test Loss = 0.268400
Epoch 120: Train Loss = 0.238290, Test Loss = 0.236253
Epoch 140: Train Loss = 0.218270, Test Loss = 0.217046
Epoch 160: Train Loss = 0.205938, Test Loss = 0.205210
Epoch 180: Train Loss = 0.197957, Test Loss = 0.197540
Epoch 200: Train Loss = 0.192167, Test Loss = 0.191930
Epoch 220: Train Loss = 0.187363, Test Loss = 0.187237
Epoch 240: Train Loss = 0.183028, Test Loss = 0.182979
Epoch 260: Train Loss = 0.179014, Test Loss = 0.179020
Epoch 280: Train Loss = 0.175162, Test Loss = 0.175205
Epoch 300: Train Loss = 0.171393, Test Loss = 0.171463
Epoch 320: Train Loss = 0.167680, Test Loss = 0.167770
Epoch 340: Train Loss = 0.164005, Test Loss = 0.164111
Epoch 360: Train Loss = 0.160355, Test Loss = 0.160473
Epoch 380: Train Loss = 0.156720, Test Loss = 0.156844
...
Epoch 440: Train Loss = 0.145774, Test Loss = 0.145913
Epoch 460: Train Loss = 0.142239, Test Loss = 0.142383
Epoch 480: Train Loss = 0.138912, Test Loss = 0.139061
Epoch 500: Train Loss = 0.135665, Test Loss = 0.135820
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```
============================================================
PREDICTION RESULTS FOR x = 90.2
============================================================
Neural Network Prediction: 1,304,035.62
Ground Truth (formula):    1,778,041.87
Absolute Error:            474,006.25
Relative Error:            26.659%
```

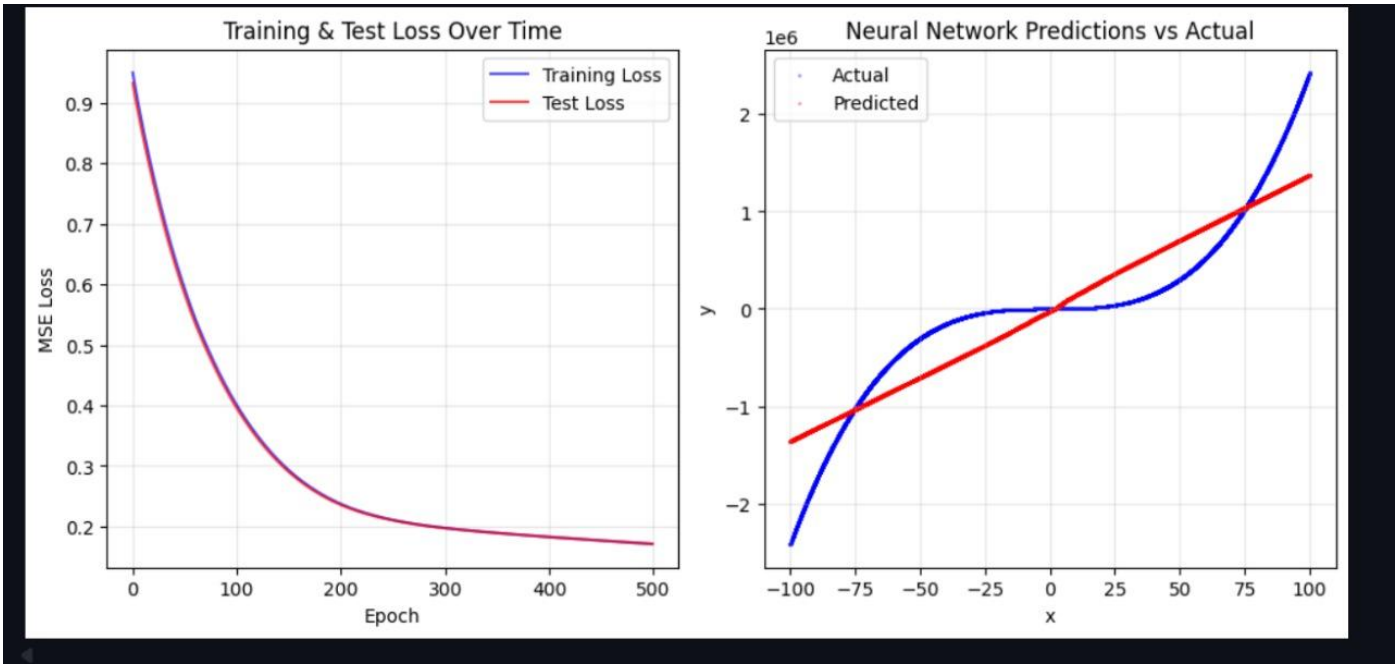
EXPERIMENT 2: LR = 0.003

```
Training Neural Network with your specific configuration...
Starting training...
Architecture: 1 → 32 → 72 → 1
Learning Rate: 0.003
Max Epochs: 500, Early Stopping Patience: 10
--------------------------------------------------
Epoch  20: Train Loss = 0.785056, Test Loss = 0.772123
Epoch  40: Train Loss = 0.651719, Test Loss = 0.641581
Epoch  60: Train Loss = 0.547706, Test Loss = 0.539695
Epoch  80: Train Loss = 0.467820, Test Loss = 0.461367
Epoch 100: Train Loss = 0.403090, Test Loss = 0.397847
Epoch 120: Train Loss = 0.351717, Test Loss = 0.347492
Epoch 140: Train Loss = 0.311149, Test Loss = 0.307785
Epoch 160: Train Loss = 0.279768, Test Loss = 0.277139
Epoch 180: Train Loss = 0.255910, Test Loss = 0.253862
Epoch 200: Train Loss = 0.237970, Test Loss = 0.236440
Epoch 220: Train Loss = 0.224839, Test Loss = 0.223708
Epoch 240: Train Loss = 0.215172, Test Loss = 0.214339
Epoch 260: Train Loss = 0.207858, Test Loss = 0.207255
Epoch 280: Train Loss = 0.202255, Test Loss = 0.201833
Epoch 300: Train Loss = 0.197872, Test Loss = 0.197584
Epoch 320: Train Loss = 0.194241, Test Loss = 0.194052
Epoch 340: Train Loss = 0.191074, Test Loss = 0.190961
Epoch 360: Train Loss = 0.188200, Test Loss = 0.188147
Epoch 380: Train Loss = 0.185510, Test Loss = 0.185503
...
Epoch 440: Train Loss = 0.178157, Test Loss = 0.178248
Epoch 460: Train Loss = 0.175847, Test Loss = 0.175958
Epoch 480: Train Loss = 0.173566, Test Loss = 0.173695
Epoch 500: Train Loss = 0.171311, Test Loss = 0.171454
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```
============================================================
PREDICTION RESULTS FOR x = 90.2
============================================================
Neural Network Prediction: 1,243,590.66
Ground Truth (formula):    1,778,041.87
Absolute Error:            534,451.21
Relative Error:            30.058%
```

```
============================================================
FINAL PERFORMANCE SUMMARY
============================================================
Final Training Loss: 0.171311
Final Test Loss:     0.171454
R² Score:            0.8277
Total Epochs Run:    500
```

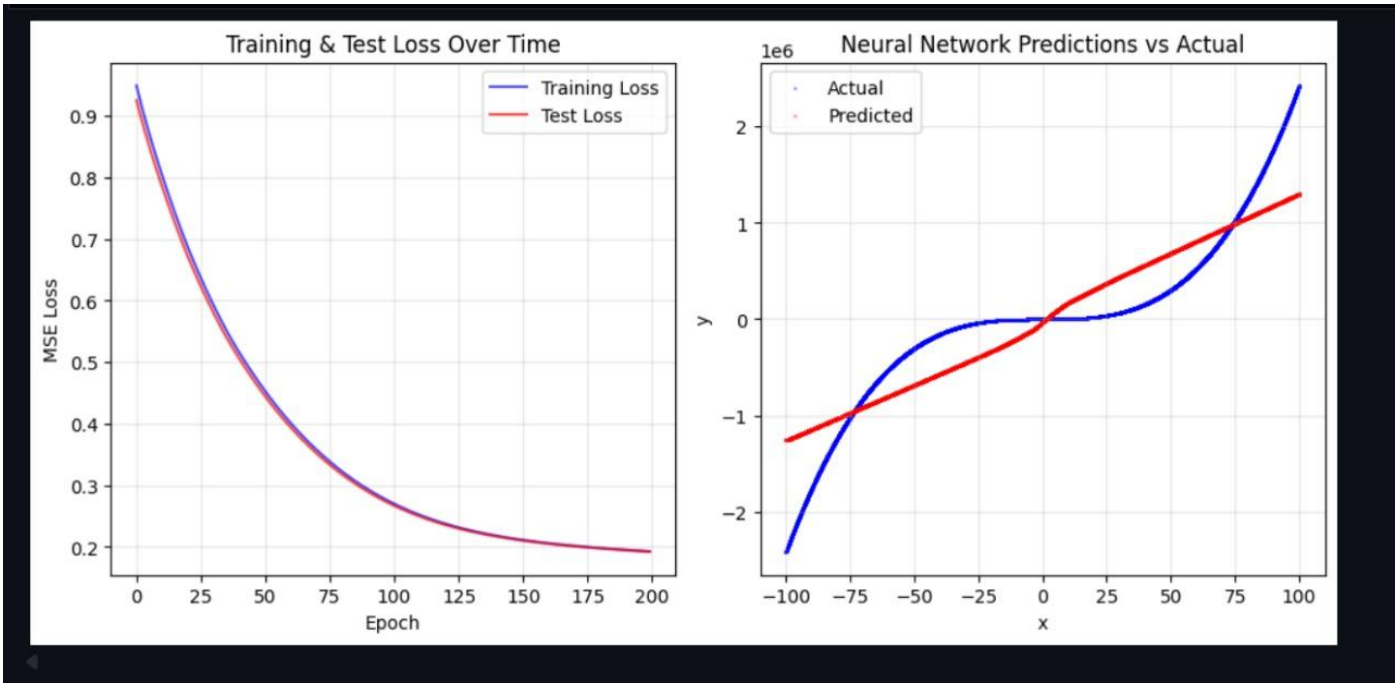


EXPERIMENT 3 : LR = 0.005

```
Training Neural Network with your specific configuration...
Starting training...
Architecture: 1 → 32 → 72 → 1
Learning Rate: 0.005
Max Epochs: 200, Early Stopping Patience: 10
--------------------------------------------------
Epoch  20: Train Loss = 0.696349, Test Loss = 0.680999
Epoch  40: Train Loss = 0.521378, Test Loss = 0.511330
Epoch  60: Train Loss = 0.404672, Test Loss = 0.397507
Epoch  80: Train Loss = 0.324567, Test Loss = 0.319633
Epoch 100: Train Loss = 0.271638, Test Loss = 0.268400
Epoch 120: Train Loss = 0.238290, Test Loss = 0.236253
Epoch 140: Train Loss = 0.218270, Test Loss = 0.217046
Epoch 160: Train Loss = 0.205938, Test Loss = 0.205210
Epoch 180: Train Loss = 0.197957, Test Loss = 0.197540
Epoch 200: Train Loss = 0.192167, Test Loss = 0.191930
```

```
============================================================
PREDICTION RESULTS FOR x = 90.2
============================================================
Neural Network Prediction: 1,178,890.38
Ground Truth (formula):    1,778,041.87
Absolute Error:            599,151.49
Relative Error:            33.697%
```

```
============================================================
FINAL PERFORMANCE SUMMARY
============================================================
Final Training Loss: 0.192167
Final Test Loss:     0.191930
R² Score:           0.8072
Total Epochs Run:   200
```



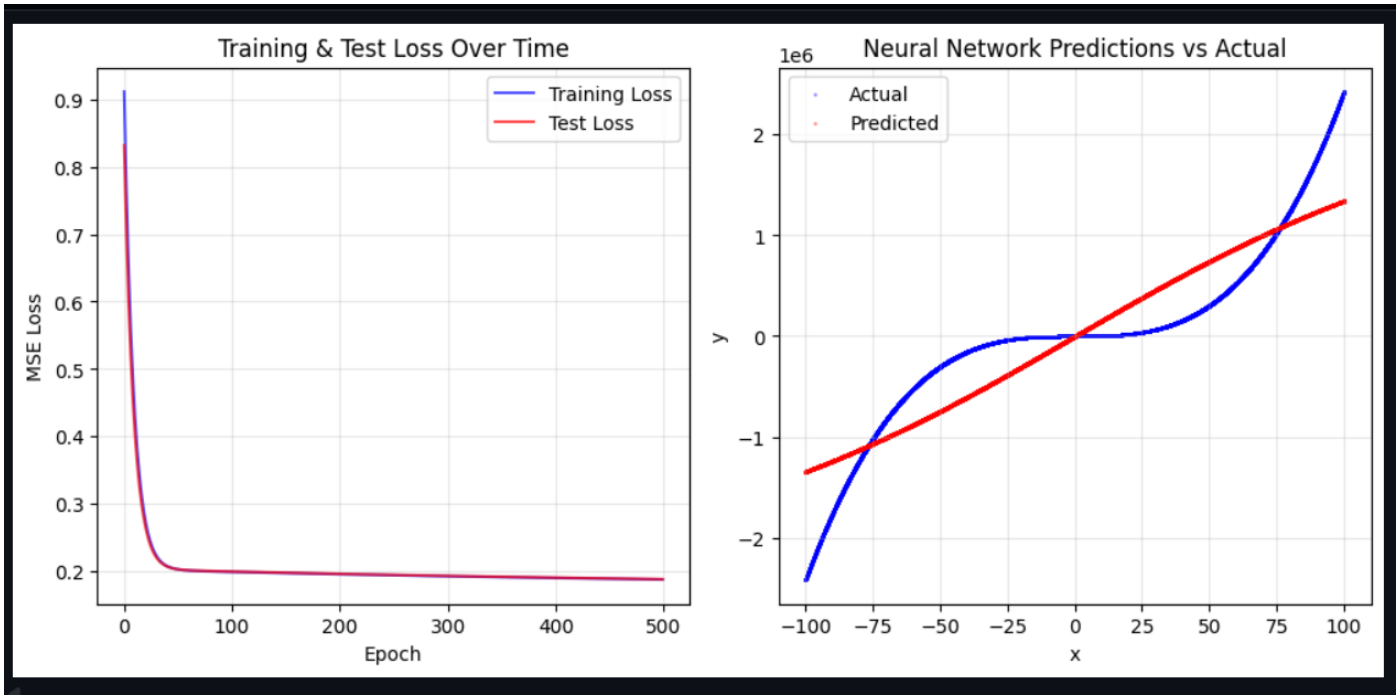
EXPERIMENT 4: tanh activation

```
Training Neural Network with your specific configuration...
Starting training...
Architecture: 1 → 32 → 72 → 1
Learning Rate: 0.005
Max Epochs: 500, Early Stopping Patience: 10
------------------------------------------------
Epoch  20: Train Loss = 0.280700, Test Loss = 0.270921
Epoch  40: Train Loss = 0.207790, Test Loss = 0.207040
Epoch  60: Train Loss = 0.200687, Test Loss = 0.200940
Epoch  80: Train Loss = 0.199392, Test Loss = 0.199780
Epoch 100: Train Loss = 0.198594, Test Loss = 0.199009
Epoch 120: Train Loss = 0.197858, Test Loss = 0.198279
Epoch 140: Train Loss = 0.197148, Test Loss = 0.197571
Epoch 160: Train Loss = 0.196459, Test Loss = 0.196883
Epoch 180: Train Loss = 0.195791, Test Loss = 0.196215
Epoch 200: Train Loss = 0.195143, Test Loss = 0.195566
Epoch 220: Train Loss = 0.194513, Test Loss = 0.194936
Epoch 240: Train Loss = 0.193900, Test Loss = 0.194323
Epoch 260: Train Loss = 0.193305, Test Loss = 0.193728
Epoch 280: Train Loss = 0.192726, Test Loss = 0.193148
Epoch 300: Train Loss = 0.192162, Test Loss = 0.192584
Epoch 320: Train Loss = 0.191612, Test Loss = 0.192034
Epoch 340: Train Loss = 0.191077, Test Loss = 0.191499
Epoch 360: Train Loss = 0.190554, Test Loss = 0.190976
Epoch 380: Train Loss = 0.190045, Test Loss = 0.190467
...
Epoch 440: Train Loss = 0.188587, Test Loss = 0.189008
Epoch 460: Train Loss = 0.188122, Test Loss = 0.188543
Epoch 480: Train Loss = 0.187668, Test Loss = 0.188088
Epoch 500: Train Loss = 0.187223, Test Loss = 0.187643
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```
============================================================
PREDICTION RESULTS FOR x = 90.2
============================================================
Neural Network Prediction: 1,236,680.13
Ground Truth (formula):    1,778,041.87
Absolute Error:            541,361.74
Relative Error:            30.447%
```

```
============================================================
FINAL PERFORMANCE SUMMARY
============================================================
Final Training Loss: 0.187223
Final Test Loss:     0.187643
R² Score:            0.8115
Total Epochs Run:    500
```



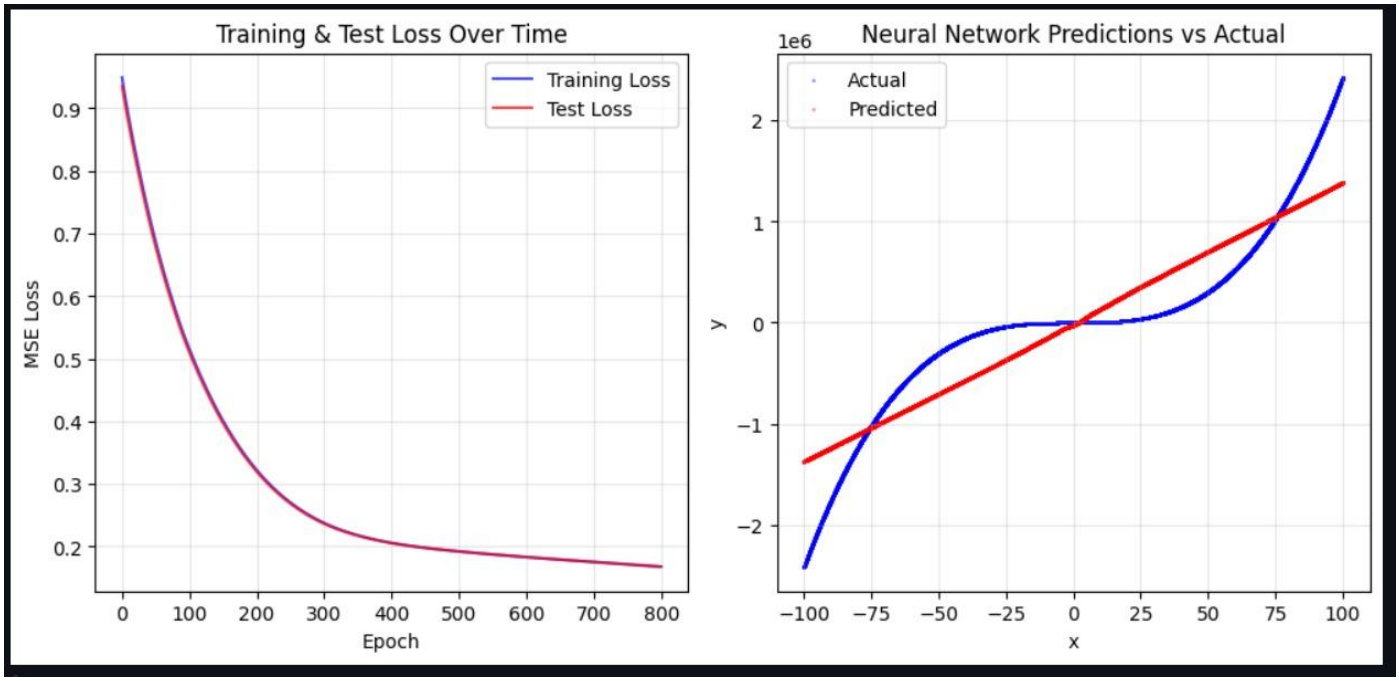
EXPERIMENT 5 : learning rate=0.002, Epochs=800

```
Training Neural Network with your specific configuration...
Starting training...
Architecture: 1 → 32 → 72 → 1
Learning Rate: 0.002
Max Epochs: 800, Early Stopping Patience: 10
--------------------------------------------------
Epoch  20: Train Loss = 0.834619, Test Loss = 0.823309
Epoch  40: Train Loss = 0.734807, Test Loss = 0.725148
Epoch  60: Train Loss = 0.649979, Test Loss = 0.641775
Epoch  80: Train Loss = 0.578360, Test Loss = 0.571296
Epoch 100: Train Loss = 0.517815, Test Loss = 0.511852
Epoch 120: Train Loss = 0.466801, Test Loss = 0.461542
Epoch 140: Train Loss = 0.422213, Test Loss = 0.417614
Epoch 160: Train Loss = 0.383909, Test Loss = 0.379914
Epoch 180: Train Loss = 0.351121, Test Loss = 0.347652
Epoch 200: Train Loss = 0.323089, Test Loss = 0.320096
Epoch 220: Train Loss = 0.299345, Test Loss = 0.296782
Epoch 240: Train Loss = 0.279453, Test Loss = 0.277275
Epoch 260: Train Loss = 0.262897, Test Loss = 0.261048
Epoch 280: Train Loss = 0.249103, Test Loss = 0.247560
Epoch 300: Train Loss = 0.237809, Test Loss = 0.236531
Epoch 320: Train Loss = 0.228636, Test Loss = 0.227587
Epoch 340: Train Loss = 0.221196, Test Loss = 0.220341
Epoch 360: Train Loss = 0.215088, Test Loss = 0.214393
Epoch 380: Train Loss = 0.210010, Test Loss = 0.209451
...
Epoch 740: Train Loss = 0.172018, Test Loss = 0.172194
Epoch 760: Train Loss = 0.170521, Test Loss = 0.170706
Epoch 780: Train Loss = 0.169031, Test Loss = 0.169224
Epoch 800: Train Loss = 0.167548, Test Loss = 0.167748
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settin

```
============================================================
PREDICTION RESULTS FOR x = 90.2
============================================================
Neural Network Prediction: 1,250,513.39
Ground Truth (formula):    1,778,041.87
Absolute Error:            527,528.48
Relative Error:            29.669%
```

```
============================================================
FINAL PERFORMANCE SUMMARY
============================================================
Final Training Loss: 0.167548
Final Test Loss:     0.167748
R² Score:           0.8315
Total Epochs Run:   800
```



---

5. Conclusion

This lab successfully demonstrated the end-to-end process of building, training, and tuning a neural network from scratch. By implementing the core components of an ANN, a functional model was created to approximate the assigned

"Cubic + Inverse" function.

The key takeaway is that performance is a direct result of systematic tuning. The baseline model was functional but suboptimal. Through methodical experimentation, a significantly better model was developed.