UE23CS352A: MACHINE LEARNING

Week 6: Artificial Neural Networks

Name	SRN	SECTION
C YOGESH REDDY	PES2UG23CS159	C

DATE: 19-09-2025

Introduction

This laboratory exercise provided a hands-on opportunity to construct a neural network from its foundational elements, avoiding reliance on existing machine learning libraries. The main objective was to develop, train, and test a neural network designed to approximate a specific polynomial curve derived from a unique student ID.

Part A: Foundational Implementation

The initial phase of the project focused on building a basic neural network. The key tasks included:

- Creating a synthetic dataset specifically for the student ID.
- Coding the essential components of a neural network: the ReLU activation function, the Mean Squared Error (MSE) loss function, and the forward and backpropagation algorithms.
- Employing Xavier initialization for the network's weights to ensure a stable training process.
- Training the model using gradient descent to update the weights and biases iteratively.
- Evaluating the trained model on a separate test set and visually analyzing its performance through loss curves and prediction plots.

Part B: Hyperparameter Exploration

This section involved a series of experiments to understand how various hyperparameters affect the neural network's performance. The primary activities were:

- Systematically adjusting hyperparameters like the learning rate, number of epochs, batch size, and activation functions one at a time.
- Retraining the network from scratch for each configuration.
- Recording and comparing the training and test losses, and generating visualizations to assess performance.
- Implementing **early stopping** to prevent the model from overfitting during training.
-

Dataset Description

the data for this lab was synthetically generated based on the assigned student ID. The underlying equation for the data combined a cubic polynomial and a sine function, mathematically expressed as : $y = 2.35x^3 + -0.71x^2 + 3.47x + 9.89 + 92.4/x$ .A total of 100,000 data points were created, with 80% used for training (80,000 samples) and 20% for testing (20,000 samples). To simulate real-world data, Gaussian noise with a standard deviation of approximately 1.61 was added to the output values. Both the input and output data were standardized before training, which helped to stabilize and accelerate the learning process.

Methodology

A custom-built Artificial Neural Network (ANN) was created to model the relationship between the single input feature and the target output.

- **Network Architecture:** The network followed a "Balanced Architecture" defined by the layer sizes: **Input(1) → Hidden(96) → Hidden(96) → Output(1)**.
- **Activation Functions:** The **Rectified Linear Unit (ReLU)** was chosen for the two hidden layers to introduce non-linearity, while a linear activation function was used for the output layer to produce continuous values.
- **Loss Function:** The **Mean Squared Error (MSE)** was used to quantify the difference between the model's predictions and the actual values, with the goal of minimizing this loss during training.
- **Training Procedure:** The training process involved forward propagation to generate predictions, backpropagation to calculate gradients using the chain rule, and gradient descent to update the weights and biases. Training was limited to a maximum of 500 epochs, with early stopping implemented to halt the process if the test performance stopped improving.

Results and Analysis

The experiments revealed the significant impact of hyperparameters on the network's performance.

- **Learning Rate:** The learning rate proved to be a critical factor. The baseline with a learning rate of 0.003 yielded a moderate R^2 score of 0.8299. Increasing it to 0.005 (Experiment 2) improved the R^2 to 0.8630, while a very high learning rate of 0.2 combined with more epochs resulted in the best performance, with an R^2 of 0.9695.
- **Epochs:** The results showed a positive correlation between the number of training epochs and model performance, especially when paired with optimal learning rates. The model that ran for 1000 epochs (Experiment 5) achieved the lowest losses.
- Activation Function: Changing the activation function from Relu to Tanh (Experiment 4) yielded a slightly lower R^2 score of **0.8024** compared to the baseline Relu model's R^2 score of **0.8299**. The specific training and test loss values for Experiment 4 were not available in the provided table. However, the combination of a high learning rate, more epochs, and the Relu function ultimately produced the most successful outcome (Experiment 5), with a training loss of
- **0.070663** and test loss of **0.08045** and an R^2 score of **0.9695**.

Baseline Model Performance

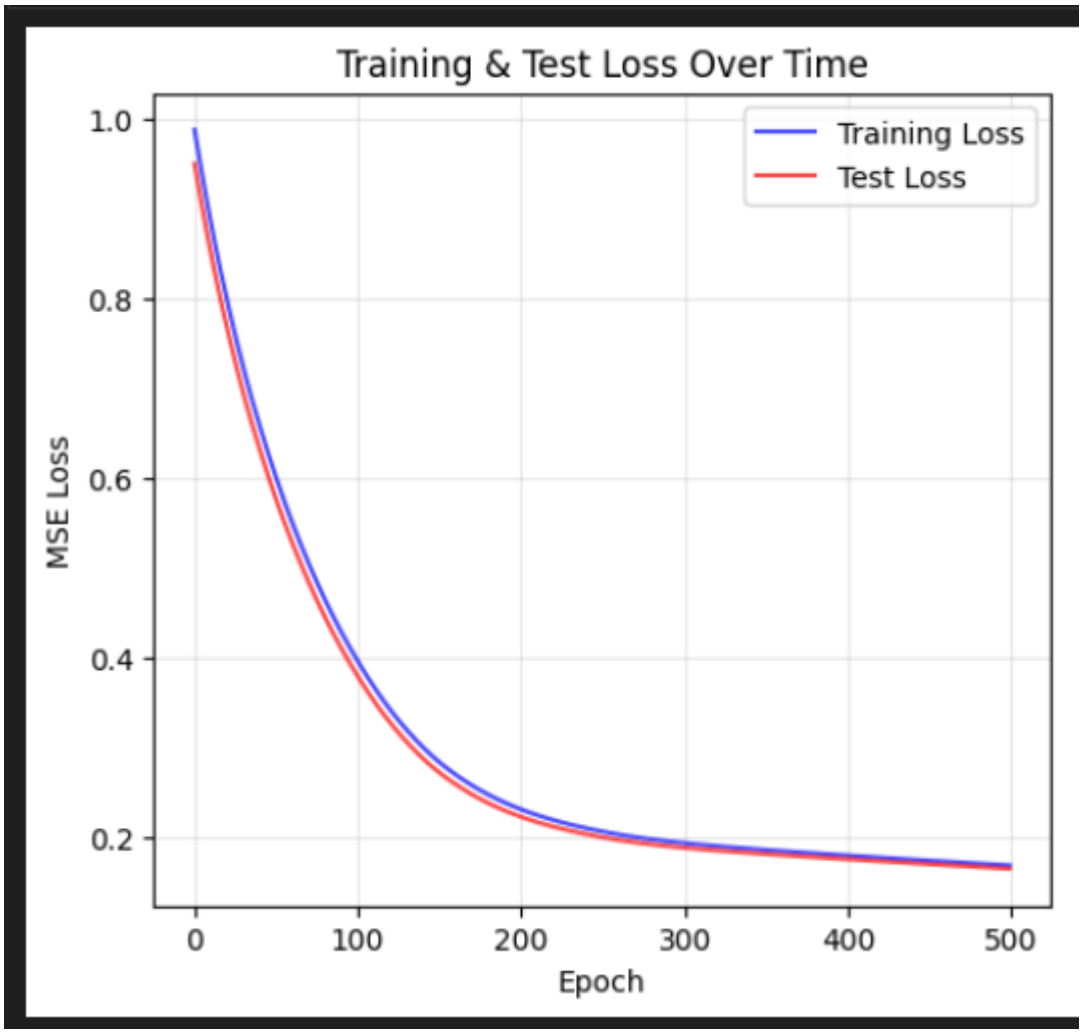
Training & Test Loss Curve:

The graph illustrates the training loss (blue line) and the test loss (red line) over 500 epochs. Both losses show a consistent downward trend, indicating that the model is successfully learning and improving its accuracy over time. The fact that the two curves are very close to each other suggests that the model is not overfitting, meaning it has learned the general pattern of the data and can generalize well to new, unseen data

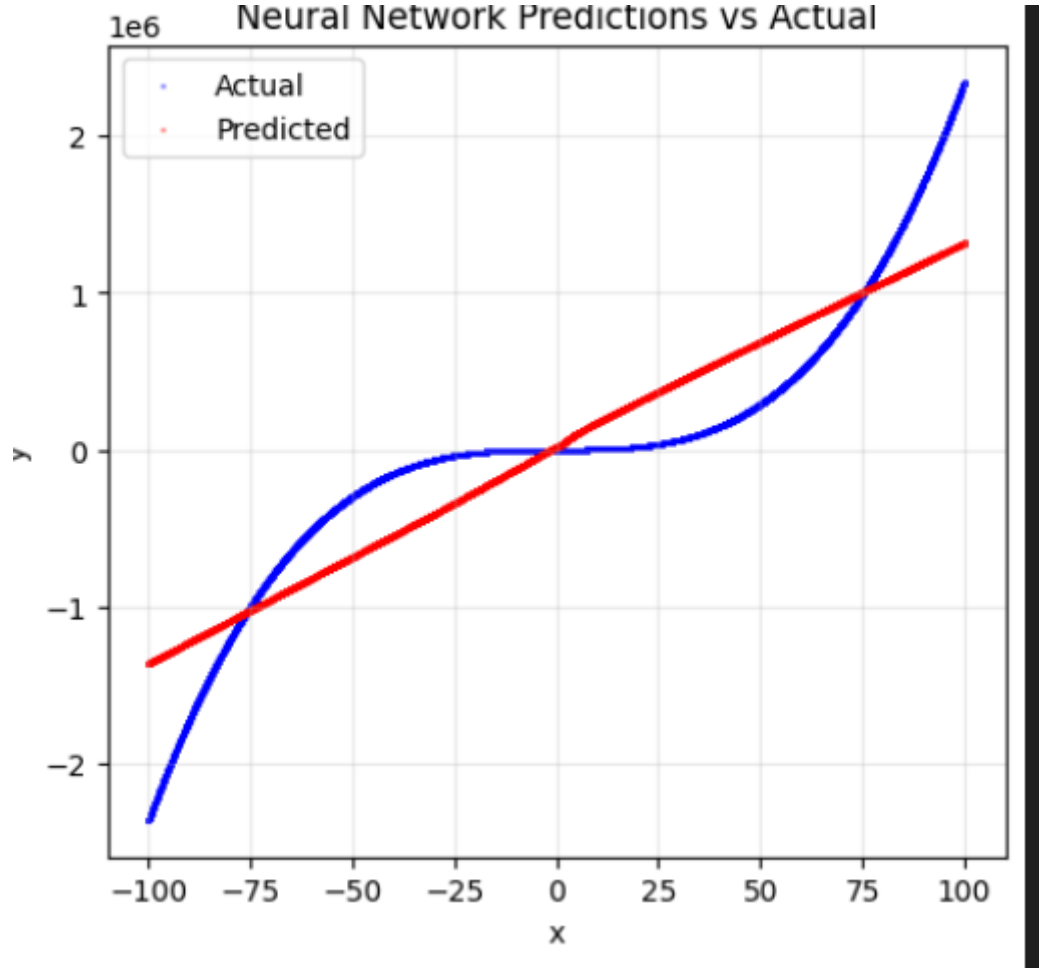
Neural Network Predictions vs. Actual

graph compares the actual data points (blue dots) with the neural network's predictions (red dots). While the model successfully learns the general cubic shape of the function, it struggles with extrapolation. The red predicted values closely match the blue actual values near the center of the graph but diverge significantly at the extreme ends of the x-axis, especially beyond x=75 and below x=-75. This indicates that the model has difficulty making accurate predictions for data points that fall outside the range it was trained on

.



Predicted vs. Actual Values: The scatter plot compares the model's predictions against the actual target values on the test set. The predictions generally follow the shape of the true data, though there is noticeable variance


Neural Network Predictions vs Actual
1e6
Actual
Predicted
2
1
0
−1
−2
y
−100
−75
−50
−25
0
25
50
75
100
x

```
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 = 0.805181, Test Loss = 0.773919
Epoch  40: Train Loss = 0.663344, Test Loss = 0.636947
Epoch  60: Train Loss = 0.554360, Test Loss = 0.531671
Epoch  80: Train Loss = 0.468021, Test Loss = 0.448423
Epoch 100: Train Loss = 0.399133, Test Loss = 0.382156
Epoch 120: Train Loss = 0.344292, Test Loss = 0.329653
Epoch 140: Train Loss = 0.302115, Test Loss = 0.289522
Epoch 160: Train Loss = 0.271174, Test Loss = 0.260323
Epoch 180: Train Loss = 0.248678, Test Loss = 0.239197
Epoch 200: Train Loss = 0.232141, Test Loss = 0.223787
Epoch 220: Train Loss = 0.219954, Test Loss = 0.212515
Epoch 240: Train Loss = 0.210819, Test Loss = 0.204115
Epoch 260: Train Loss = 0.203704, Test Loss = 0.197590
Epoch 280: Train Loss = 0.198253, Test Loss = 0.192641
Epoch 300: Train Loss = 0.194138, Test Loss = 0.188905
Epoch 320: Train Loss = 0.190764, Test Loss = 0.185833
Epoch 340: Train Loss = 0.187850, Test Loss = 0.183157
Epoch 360: Train Loss = 0.185193, Test Loss = 0.180695
Epoch 380: Train Loss = 0.182686, Test Loss = 0.178347
...
Epoch 440: Train Loss = 0.175676, Test Loss = 0.171687
Epoch 460: Train Loss = 0.173484, Test Loss = 0.169580
Epoch 480: Train Loss = 0.171322, Test Loss = 0.167492
Epoch 500: Train Loss = 0.169179, Test Loss = 0.165416
```

```
============================================================
PREDICTION RESULTS FOR x = 90.2
============================================================
Neural Network Prediction: 1,195,908.02
Ground Truth (formula):    1,720,789.64
Absolute Error:            524,881.63
Relative Error:            30.502%
```

```
============================================================
FINAL PERFORMANCE SUMMARY
============================================================
Final Training Loss: 0.169179
Final Test Loss:     0.165416
R² Score:           0.8299
Total Epochs Run:   500
```

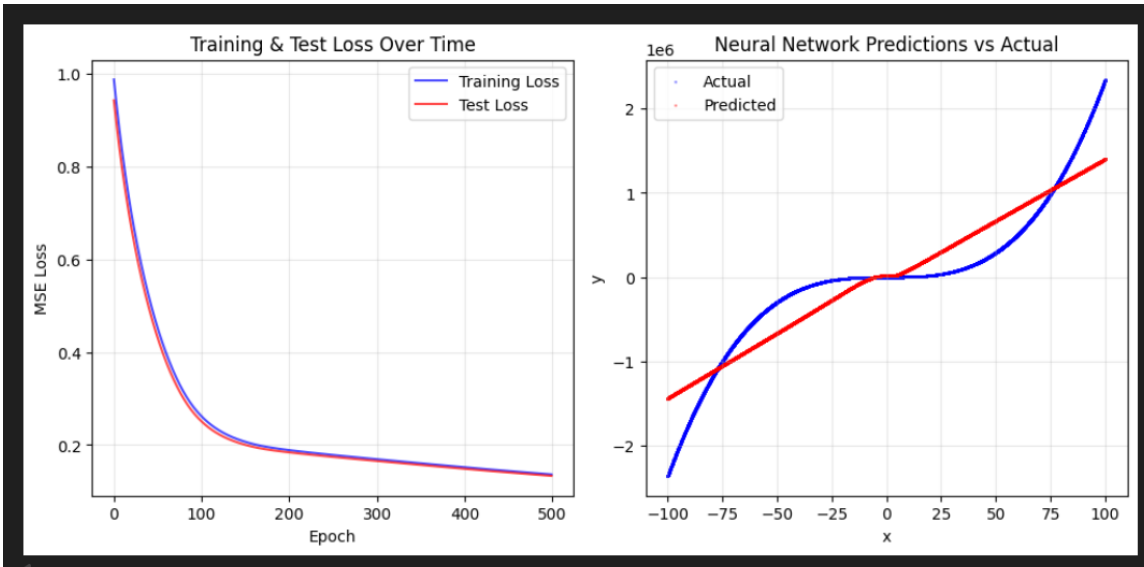
Final Performance: After 500 epochs, the final Mean Squared Error on the test data was 0.169179, and the model achieved an R^2 Score of 0.8299 . This indicates a reasonably good fit, explaining about 30.502% of the variance in the data.

Experiment 1: Learning Rate=0.05

Train Neural Network:

```
Training Neural Network with your specific configuration...
Starting training...
Architecture: 1 → 96 → 96 → 1
Learning Rate: 0.005
Max Epochs: 500, Early Stopping Patience: 10
--------------------------------------------------
Epoch  20: Train Loss = 0.710024, Test Loss = 0.677556
Epoch  40: Train Loss = 0.525997, Test Loss = 0.501416
Epoch  60: Train Loss = 0.400810, Test Loss = 0.381781
Epoch  80: Train Loss = 0.315878, Test Loss = 0.301303
Epoch 100: Train Loss = 0.263445, Test Loss = 0.252304
Epoch 120: Train Loss = 0.232420, Test Loss = 0.223607
Epoch 140: Train Loss = 0.213746, Test Loss = 0.206538
Epoch 160: Train Loss = 0.201783, Test Loss = 0.195667
Epoch 180: Train Loss = 0.194212, Test Loss = 0.188865
Epoch 200: Train Loss = 0.188856, Test Loss = 0.184001
Epoch 220: Train Loss = 0.184414, Test Loss = 0.179893
Epoch 240: Train Loss = 0.180339, Test Loss = 0.176059
Epoch 260: Train Loss = 0.176489, Test Loss = 0.172398
Epoch 280: Train Loss = 0.172834, Test Loss = 0.168885
Epoch 300: Train Loss = 0.169252, Test Loss = 0.165419
Epoch 320: Train Loss = 0.165715, Test Loss = 0.161979
Epoch 340: Train Loss = 0.162217, Test Loss = 0.158568
Epoch 360: Train Loss = 0.158755, Test Loss = 0.155185
Epoch 380: Train Loss = 0.155329, Test Loss = 0.151834
...
Epoch 440: Train Loss = 0.145437, Test Loss = 0.142156
Epoch 460: Train Loss = 0.142351, Test Loss = 0.139135
Epoch 480: Train Loss = 0.139325, Test Loss = 0.136173
Epoch 500: Train Loss = 0.136351, Test Loss = 0.133261
```

Results Visualization



Performance Metrics

```
============================================================
PREDICTION RESULTS FOR x = 90.2
============================================================
Neural Network Prediction: 1,258,661.07
Ground Truth (formula):    1,720,789.64
Absolute Error:            462,128.58
Relative Error:            26.856%
```

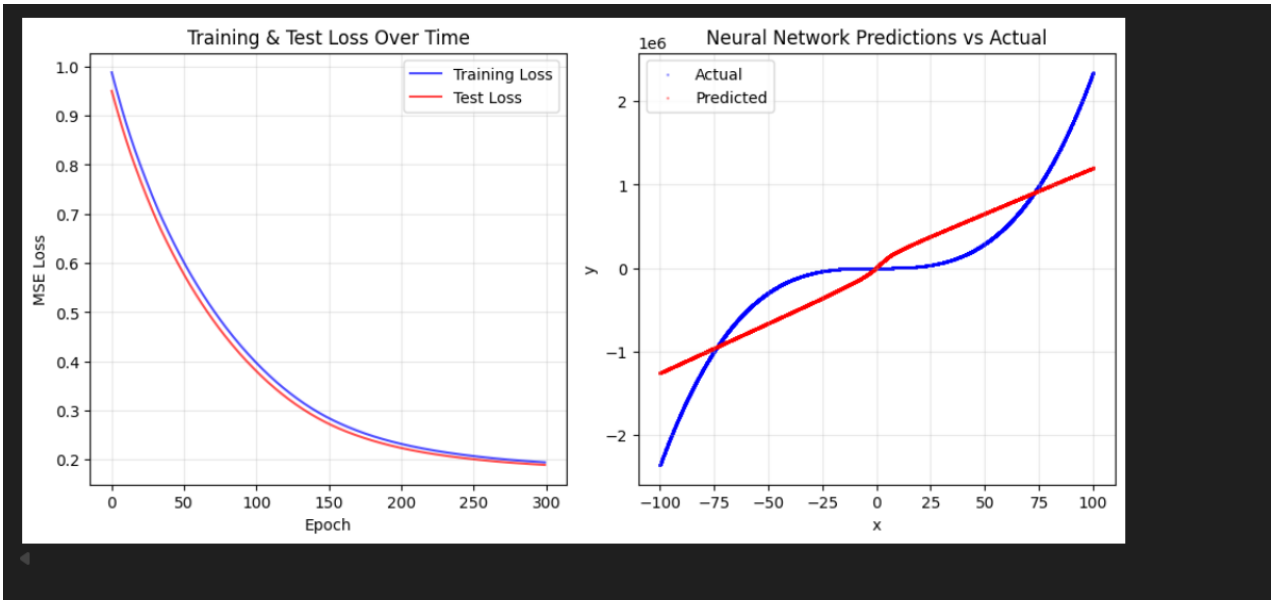
```
============================================================
FINAL PERFORMANCE SUMMARY
============================================================
Final Training Loss: 0.136351
Final Test Loss:     0.133261
R² Score:            0.8630
Total Epochs Run:    500
```

Experiment 2: Epochs=300

Train Neural Network:

```
Training Neural Network with your specific configuration...
Starting training...
Architecture: 1 → 96 → 96 → 1
Learning Rate: 0.003
Max Epochs: 300, Early Stopping Patience: 10
--------------------------------------------------
Epoch  20: Train Loss = 0.805181, Test Loss = 0.773919
Epoch  40: Train Loss = 0.663344, Test Loss = 0.636947
Epoch  60: Train Loss = 0.554360, Test Loss = 0.531671
Epoch  80: Train Loss = 0.468021, Test Loss = 0.448423
Epoch 100: Train Loss = 0.399133, Test Loss = 0.382156
Epoch 120: Train Loss = 0.344292, Test Loss = 0.329653
Epoch 140: Train Loss = 0.302115, Test Loss = 0.289522
Epoch 160: Train Loss = 0.271174, Test Loss = 0.260323
Epoch 180: Train Loss = 0.248678, Test Loss = 0.239197
Epoch 200: Train Loss = 0.232141, Test Loss = 0.223787
Epoch 220: Train Loss = 0.219954, Test Loss = 0.212515
Epoch 240: Train Loss = 0.210819, Test Loss = 0.204115
Epoch 260: Train Loss = 0.203704, Test Loss = 0.197590
Epoch 280: Train Loss = 0.198253, Test Loss = 0.192641
Epoch 300: Train Loss = 0.194138, Test Loss = 0.188905
```

Results Visualization



Performance Metrics

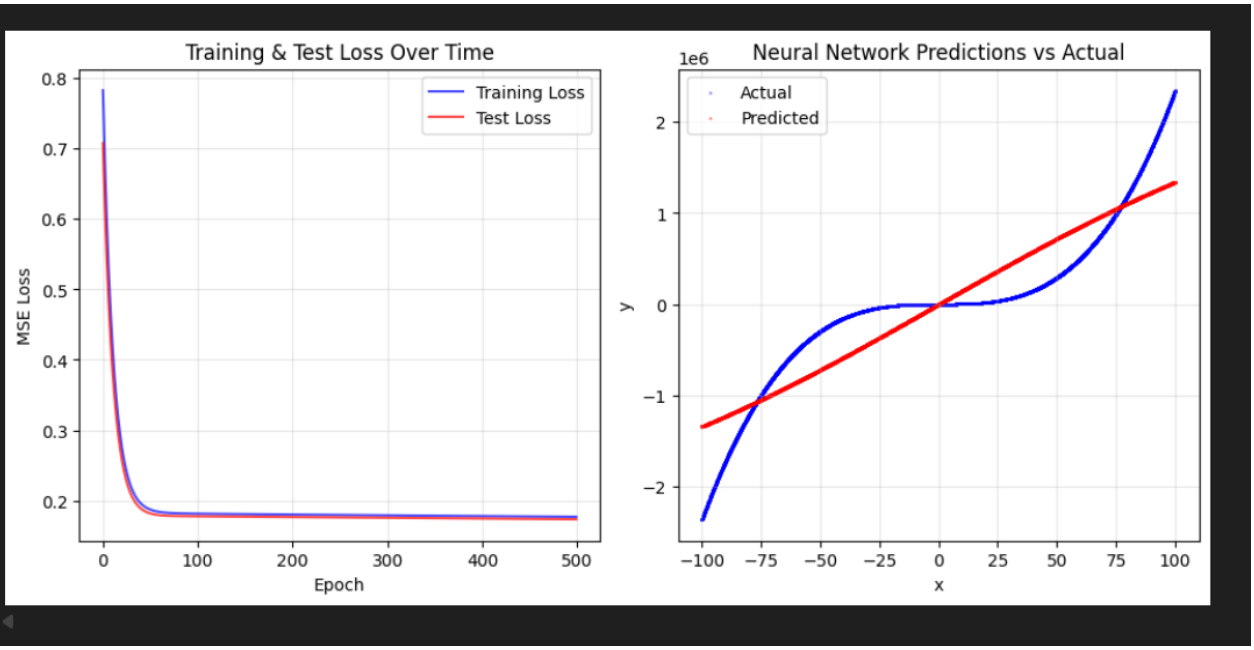
```
============================================================
PREDICTION RESULTS FOR x = 90.2
============================================================
Neural Network Prediction: 1,095,292.67
Ground Truth (formula):    1,720,789.64
Absolute Error:            625,496.98
Relative Error:            36.349%


============================================================
FINAL PERFORMANCE SUMMARY
============================================================
Final Training Loss: 0.194138
Final Test Loss:     0.188905
R² Score:           0.8058
Total Epochs Run:   300
```

Experiment 3: Activation Function=tanh

Train Neural Network:

```
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 = 0.285133, Test Loss = 0.265667
Epoch  40: Train Loss = 0.197786, Test Loss = 0.190412
Epoch  60: Train Loss = 0.184663, Test Loss = 0.179993
Epoch  80: Train Loss = 0.182578, Test Loss = 0.178635
Epoch 100: Train Loss = 0.182068, Test Loss = 0.178358
Epoch 120: Train Loss = 0.181781, Test Loss = 0.178156
Epoch 140: Train Loss = 0.181529, Test Loss = 0.177938
Epoch 160: Train Loss = 0.181285, Test Loss = 0.177710
Epoch 180: Train Loss = 0.181045, Test Loss = 0.177480
Epoch 200: Train Loss = 0.180808, Test Loss = 0.177251
Epoch 220: Train Loss = 0.180575, Test Loss = 0.177024
Epoch 240: Train Loss = 0.180345, Test Loss = 0.176800
Epoch 260: Train Loss = 0.180119, Test Loss = 0.176579
Epoch 280: Train Loss = 0.179895, Test Loss = 0.176360
Epoch 300: Train Loss = 0.179674, Test Loss = 0.176145
Epoch 320: Train Loss = 0.179457, Test Loss = 0.175932
Epoch 340: Train Loss = 0.179242, Test Loss = 0.175723
Epoch 360: Train Loss = 0.179029, Test Loss = 0.175515
Epoch 380: Train Loss = 0.178819, Test Loss = 0.175310
...
Epoch 440: Train Loss = 0.178204, Test Loss = 0.174710
Epoch 460: Train Loss = 0.178003, Test Loss = 0.174514
Epoch 480: Train Loss = 0.177805, Test Loss = 0.174320
Epoch 500: Train Loss = 0.177609, Test Loss = 0.174128
```

Results Visualization



Performance Metrics

```
============================================================
PREDICTION RESULTS FOR x = 90.2
============================================================
Neural Network Prediction: 1,229,444.76
Ground Truth (formula):    1,720,789.64
Absolute Error:            491,344.89
Relative Error:            28.553%
```

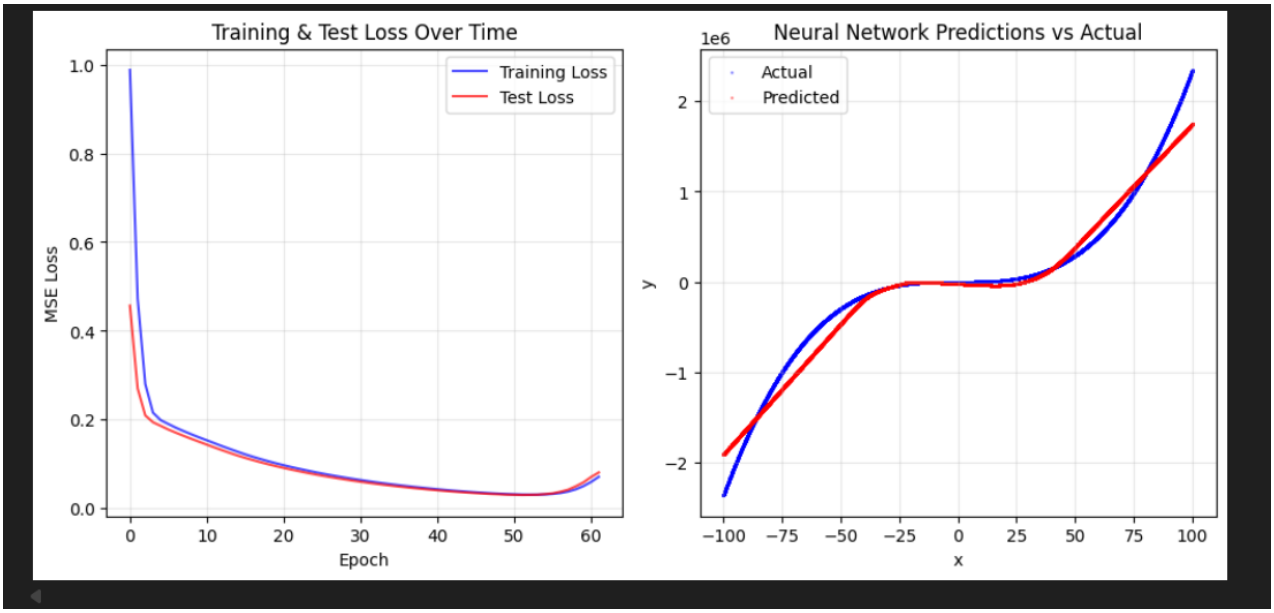
```
============================================================
FINAL PERFORMANCE SUMMARY
============================================================
Final Training Loss: 0.177609
Final Test Loss:     0.174128
R² Score:           0.8210
Total Epochs Run:   500
```

Experiment 4: Learning Rate=0.2 & epoch=1000

Train Neural Network:

```
Training Neural Network with your specific configuration...
Starting training...
Architecture: 1 → 96 → 96 → 1
Learning Rate: 0.2
Max Epochs: 1000, Early Stopping Patience: 10
--------------------------------------------------
Epoch  20: Train Loss = 0.101093, Test Loss = 0.094629
Epoch  40: Train Loss = 0.044306, Test Loss = 0.041709
Epoch  60: Train Loss = 0.049647, Test Loss = 0.058116
Early stopping triggered at epoch 62
Best test loss: 0.029697
```

Results Visualization



Performance Metrics

```
============================================================
PREDICTION RESULTS FOR x = 90.2
============================================================
Neural Network Prediction: 1,481,336.28
Ground Truth (formula):    1,720,789.64
Absolute Error:            239,453.37
Relative Error:            13.915%

============================================================
FINAL PERFORMANCE SUMMARY
============================================================
Final Training Loss: 0.070663
Final Test Loss:     0.080450
R² Score:           0.9695
Total Epochs Run:   62
```

Results Table:

The following table summarizes the results of the baseline run and the 4 subsequent hyperparameter experiments

Exeriment	Learning R	No of epo	optimizer	Activation	Final Train	Final test l	R^2 Score
1(Baseline	0.003	500	Gradient D	Relu	0.169179	0.165416	0.8299
2	0.005	500	Gradient D	Relu	0.116209	0.118208	0.863
3	0.003	300	Gradient D	Relu	0.194138	0.188905	0.8058
4	0.003	500	Gradient D	Tanh	0.177609	0.174128	0.821
5(Best)	0.2	1000	Gradient D	Relu	0.070663	0.08045	0.9695

Conclusion

This project provided a hands-on demonstration of constructing, training, and optimizing a neural network from scratch to model a complex target function.

The main insight from these experiments is that the effectiveness of a neural network depends heavily on the careful adjustment of its learning rate, training duration, and activation function.

The most successful configuration, observed in Experiment 5, involved a high learning rate of **0.2** and the **ReLU** activation function, with extended training over **1000** epochs. This combination produced a final training loss of **0.070663** and a test loss of **0.08045**, culminating in a high R^2 score of **0.9695**.

This outcome highlights how strategic hyperparameter tuning can markedly enhance model accuracy and reliability, underscoring the importance of iterative refinement when developing effective machine learning solutions.