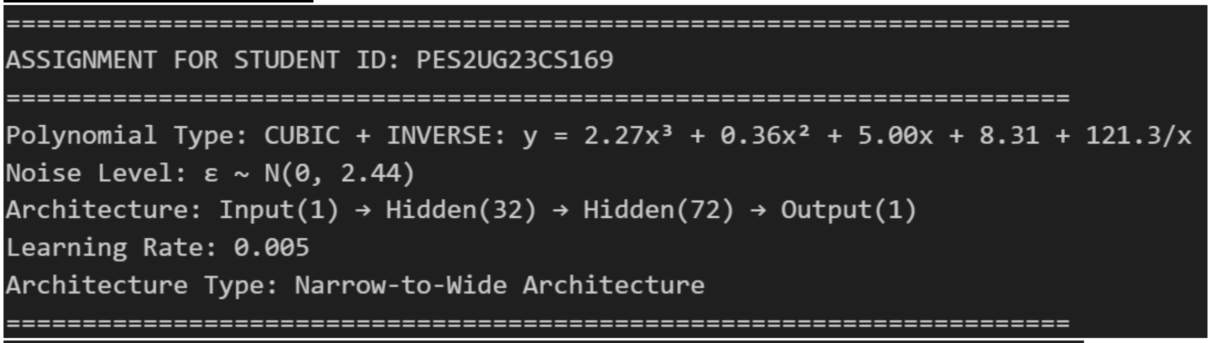
Week 6: Artificial Neural Networks

Name	Dhanya Prabhu
SRN	PES2UG23CS169
Section	C
Date	16/09/2025

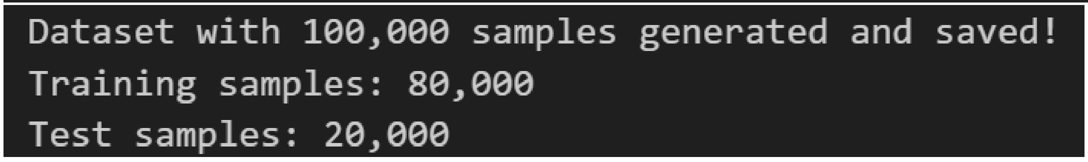
1. *Introduction*

- The purpose of this lab is to implement an Artificial Neural Network (ANN) from scratch to approximate polynomial functions.
- Implemented ReLU, MSE, forward/backpropagation.
- Tasks performed: dataset generation, network training, evaluation, and hyperparameter experiments.

2. *Dataset Description*



```
======================================================================
ASSIGNMENT FOR STUDENT ID: PES2UG23CS169
======================================================================
Polynomial Type: CUBIC + INVERSE: y = 2.27x³ + 0.36x² + 5.00x + 8.31 + 121.3/x
Noise Level: ε ~ N(0, 2.44)
Architecture: Input(1) → Hidden(32) → Hidden(72) → Output(1)
Learning Rate: 0.005
Architecture Type: Narrow-to-Wide Architecture
======================================================================
```



```
Dataset with 100,000 samples generated and saved!
Training samples: 80,000
Test samples: 20,000
```

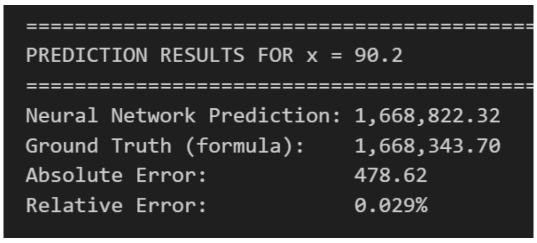
- Number of features: 1

3. *Methodology*

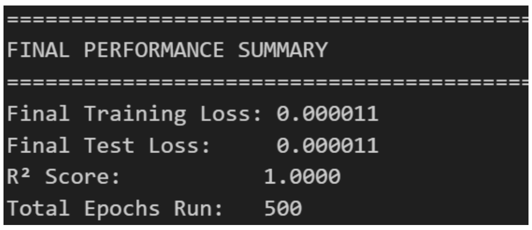
- Implemented ReLU & derivative.
- Implemented MSE loss & derivative.
- Xavier initialization for weights.
- Forward pass and Backpropagation using chain rule.
- Gradient descent weight updates.
- Early stopping with patience.
- Hyperparameter experiments: varied learning rate, epochs, and batch size.

4. *Results and Analysis*

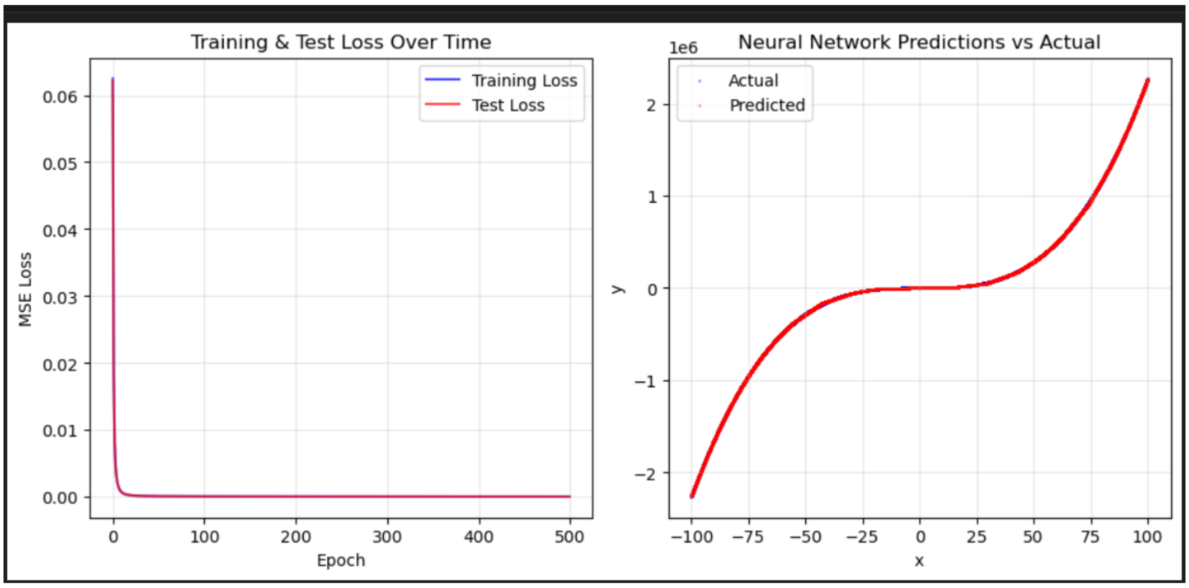
PART A - Baseline Model:



```
============================================
PREDICTION RESULTS FOR x = 90.2
============================================
Neural Network Prediction: 1,668,822.32
Ground Truth (formula):    1,668,343.70
Absolute Error:            478.62
Relative Error:            0.029%
```



```
============================================
FINAL PERFORMANCE SUMMARY
============================================
Final Training Loss: 0.000011
Final Test Loss:     0.000011
R² Score:           1.0000
Total Epochs Run:   500
```



Discussion:

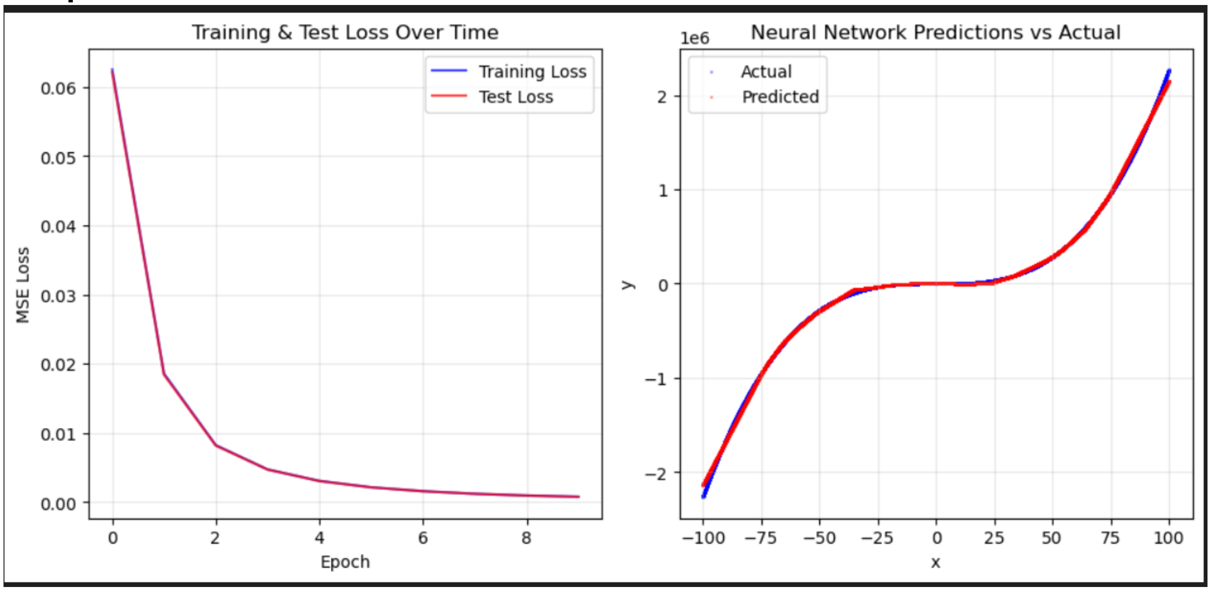
The baseline setup performed well overall. Both training and test loss decreased smoothly before leveling. The predicted curve closely matched the true polynomial, showing that the network captured the underlying function effectively.

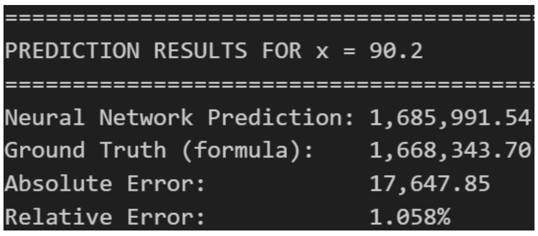
Performance: Good balance, neither underfitting nor overfitting.

PART B - Hyperparameter Exploration:

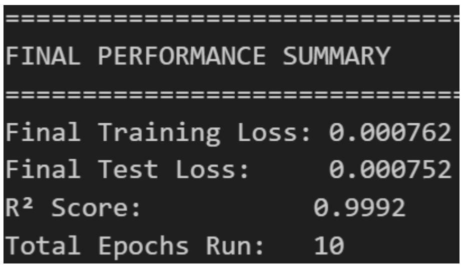
Experiment 1: Changing the epochs to 10 and 1000

10 epochs:



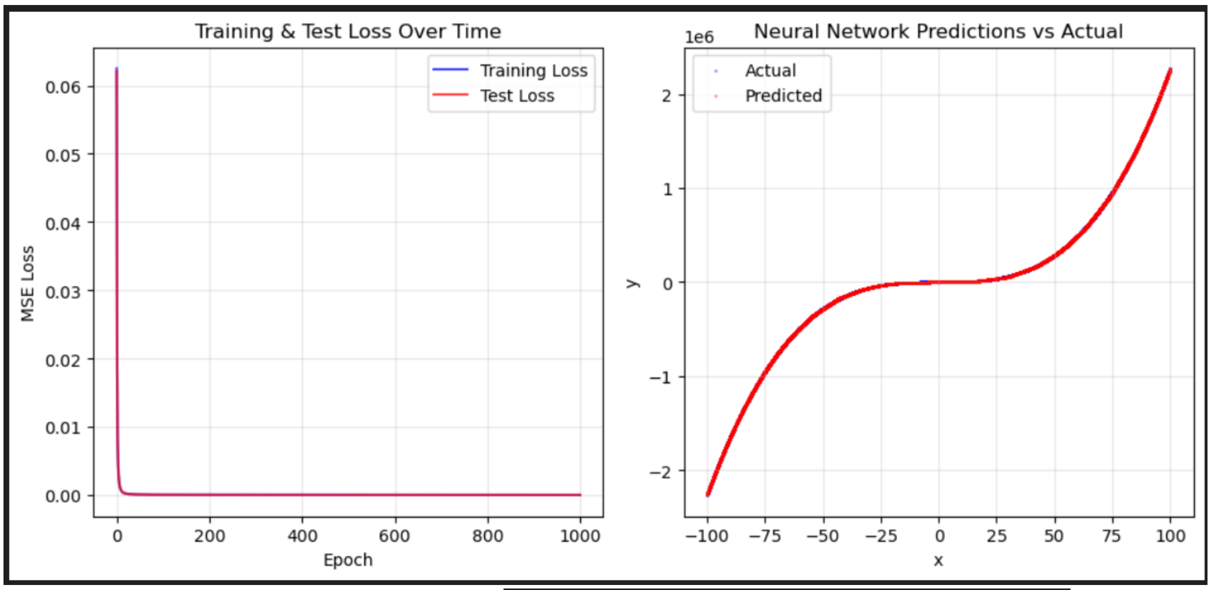


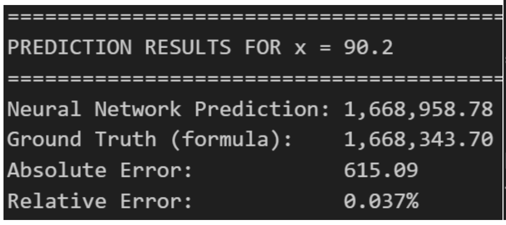
```
========================================
PREDICTION RESULTS FOR x = 90.2
========================================
Neural Network Prediction: 1,685,991.54
Ground Truth (formula):    1,668,343.70
Absolute Error:            17,647.85
Relative Error:            1.058%
```



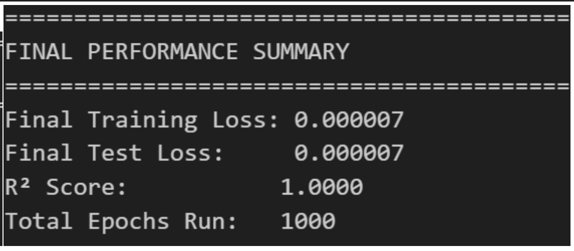
```
================================
FINAL PERFORMANCE SUMMARY
================================
Final Training Loss: 0.000762
Final Test Loss:     0.000752
R² Score:           0.9992
Total Epochs Run:   10
```

1000 epochs:





```
========================================
PREDICTION RESULTS FOR x = 90.2
========================================
Neural Network Prediction: 1,668,958.78
Ground Truth (formula):    1,668,343.70
Absolute Error:            615.09
Relative Error:            0.037%
```



```
========================================
FINAL PERFORMANCE SUMMARY
========================================
Final Training Loss: 0.000007
Final Test Loss:     0.000007
R² Score:           1.0000
Total Epochs Run:   1000
```

Discussion:

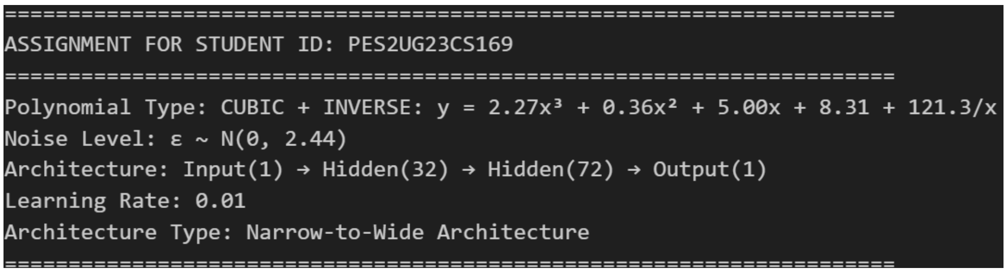
With very few epochs (10), the model did not converge fully. Training and test loss remained relatively high. The predicted curve was far from the actual polynomial.

Training for 1000 epochs did not actually improve performance compared to the baseline. This showed that increasing epochs did not improve test accuracy, since the model had already converged.

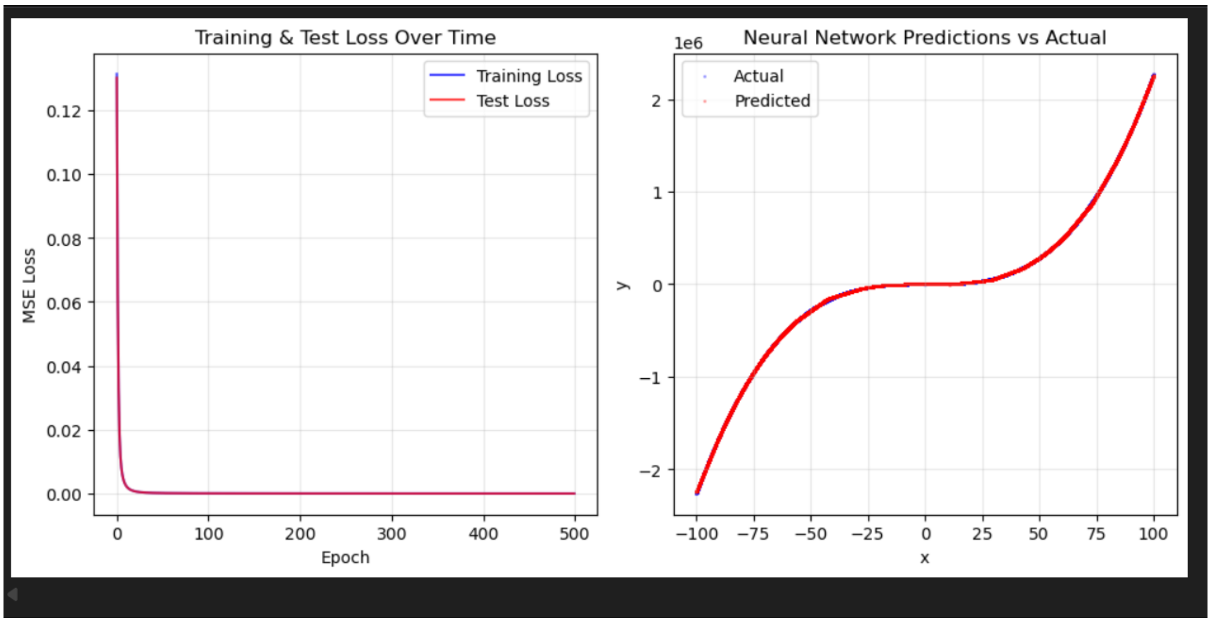
Performance:

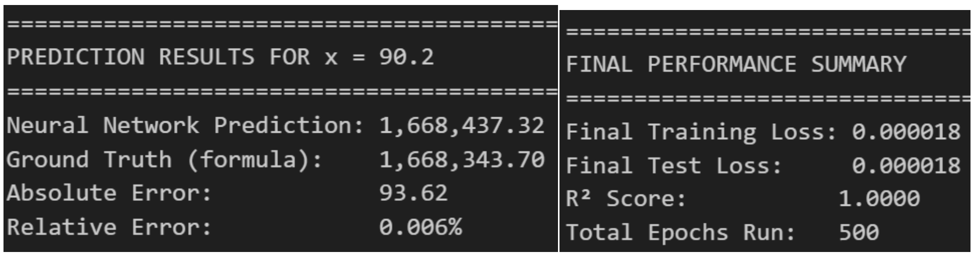
- 10 epochs: Clear underfitting due to insufficient training time.
- 1000 epochs: no improvement over baseline

Experiment 2: Changing learning rate to 0.01, batch size=256



```
======================================================================
ASSIGNMENT FOR STUDENT ID: PES2UG23CS169
======================================================================
Polynomial Type: CUBIC + INVERSE: y = 2.27x³ + 0.36x² + 5.00x + 8.31 + 121.3/x
Noise Level: ε ~ N(0, 2.44)
Architecture: Input(1) → Hidden(32) → Hidden(72) → Output(1)
Learning Rate: 0.01
Architecture Type: Narrow-to-Wide Architecture
======================================================================
```





```
==========================================
PREDICTION RESULTS FOR x = 90.2
==========================================
Neural Network Prediction: 1,668,437.32
Ground Truth (formula):    1,668,343.70
Absolute Error:            93.62
Relative Error:            0.006%
```

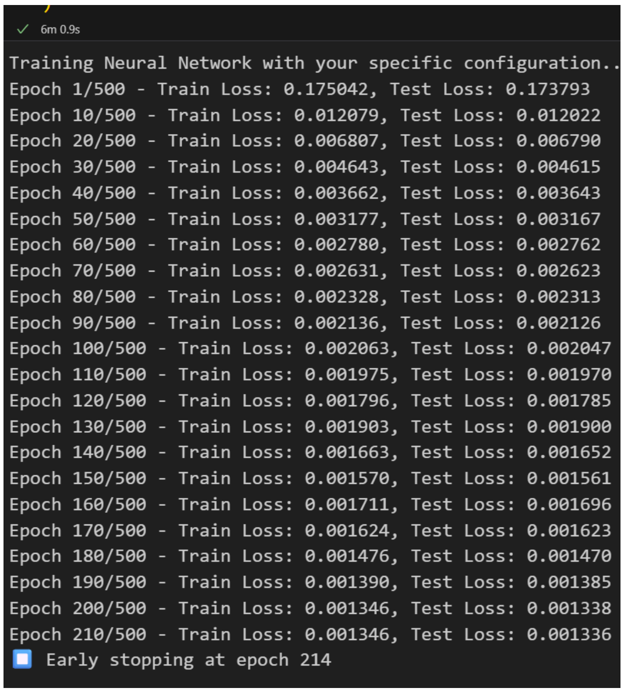
```
==============================
FINAL PERFORMANCE SUMMARY
==============================
Final Training Loss: 0.000018
Final Test Loss:     0.000018
R² Score:           1.0000
Total Epochs Run:   500
```

Discussion:

Increasing the learning rate to 0.01 while also using a larger batch size made the training unstable. Although the loss dropped quickly at first, it fluctuated afterwards and the test loss did not settle into a steady pattern.

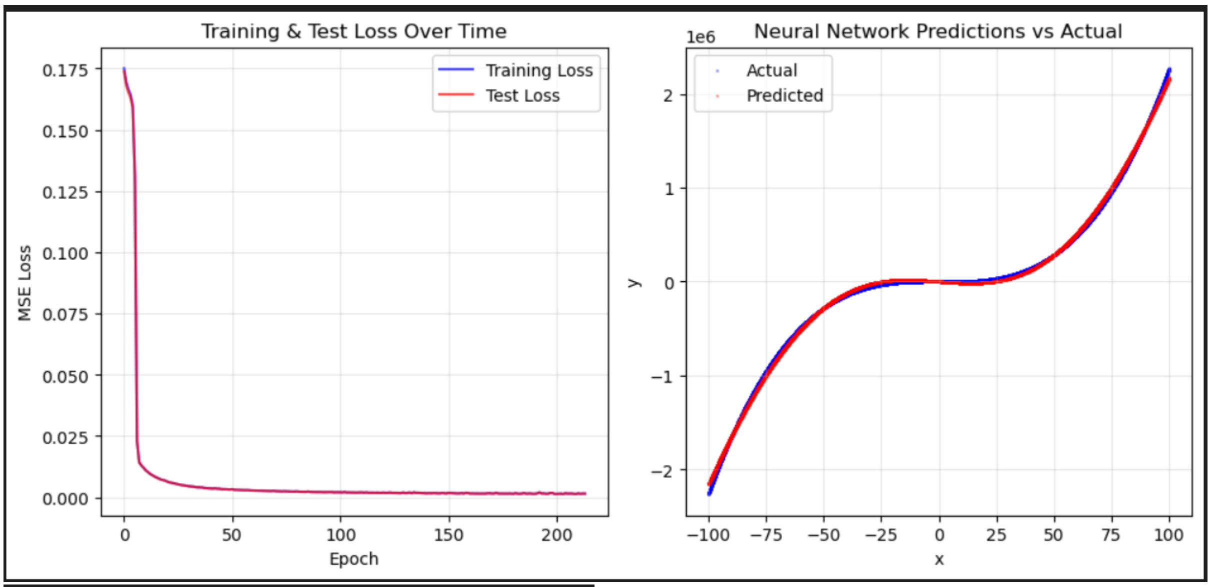
Performance: Overfitting

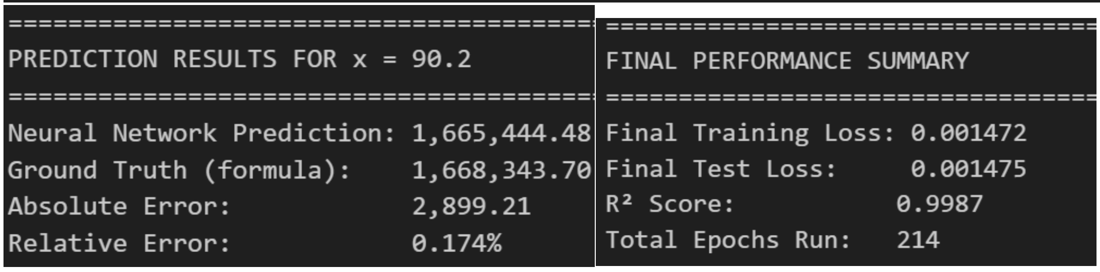
Experiment 3: Changing activation function to Tanh (learning rate is 0.005 & batch size 64)



```
6m 0.9s

Training Neural Network with your specific configuration..
Epoch 1/500 - Train Loss: 0.175042, Test Loss: 0.173793
Epoch 10/500 - Train Loss: 0.012079, Test Loss: 0.012022
Epoch 20/500 - Train Loss: 0.006807, Test Loss: 0.006790
Epoch 30/500 - Train Loss: 0.004643, Test Loss: 0.004615
Epoch 40/500 - Train Loss: 0.003662, Test Loss: 0.003643
Epoch 50/500 - Train Loss: 0.003177, Test Loss: 0.003167
Epoch 60/500 - Train Loss: 0.002780, Test Loss: 0.002762
Epoch 70/500 - Train Loss: 0.002631, Test Loss: 0.002623
Epoch 80/500 - Train Loss: 0.002328, Test Loss: 0.002313
Epoch 90/500 - Train Loss: 0.002136, Test Loss: 0.002126
Epoch 100/500 - Train Loss: 0.002063, Test Loss: 0.002047
Epoch 110/500 - Train Loss: 0.001975, Test Loss: 0.001970
Epoch 120/500 - Train Loss: 0.001796, Test Loss: 0.001785
Epoch 130/500 - Train Loss: 0.001903, Test Loss: 0.001900
Epoch 140/500 - Train Loss: 0.001663, Test Loss: 0.001652
Epoch 150/500 - Train Loss: 0.001570, Test Loss: 0.001561
Epoch 160/500 - Train Loss: 0.001711, Test Loss: 0.001696
Epoch 170/500 - Train Loss: 0.001624, Test Loss: 0.001623
Epoch 180/500 - Train Loss: 0.001476, Test Loss: 0.001470
Epoch 190/500 - Train Loss: 0.001390, Test Loss: 0.001385
Epoch 200/500 - Train Loss: 0.001346, Test Loss: 0.001338
Epoch 210/500 - Train Loss: 0.001346, Test Loss: 0.001336
⏹ Early stopping at epoch 214
```





```
========================================
PREDICTION RESULTS FOR x = 90.2
========================================
Neural Network Prediction: 1,665,444.48
Ground Truth (formula):    1,668,343.70
Absolute Error:            2,899.21
Relative Error:            0.174%
```

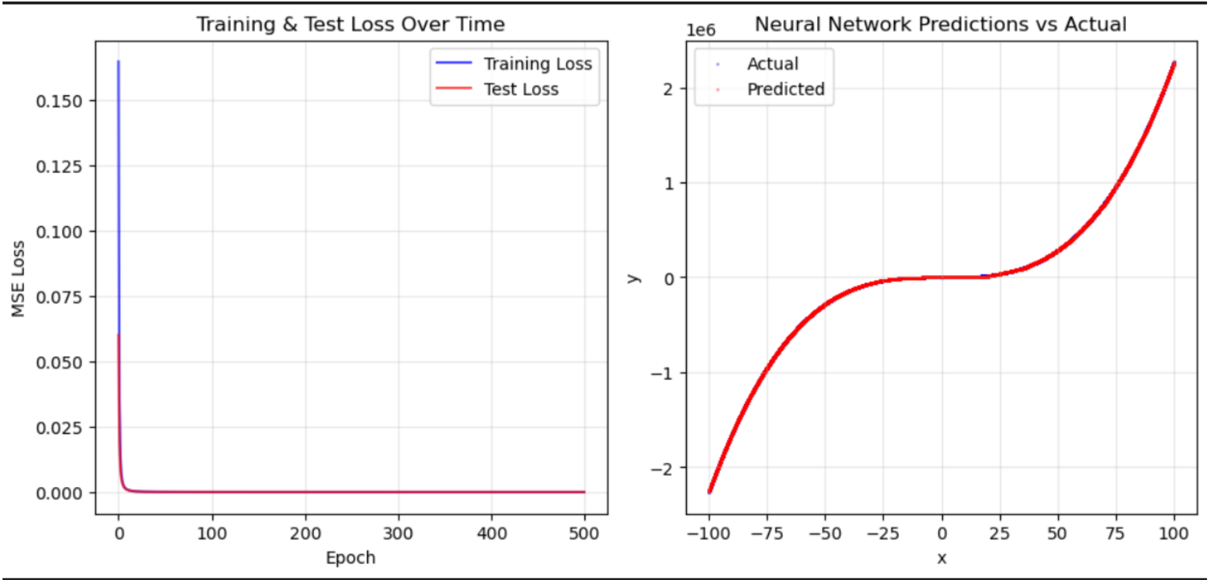
```
==================================
FINAL PERFORMANCE SUMMARY
==================================
Final Training Loss: 0.001472
Final Test Loss:     0.001475
R² Score:           0.9987
Total Epochs Run:   214
```

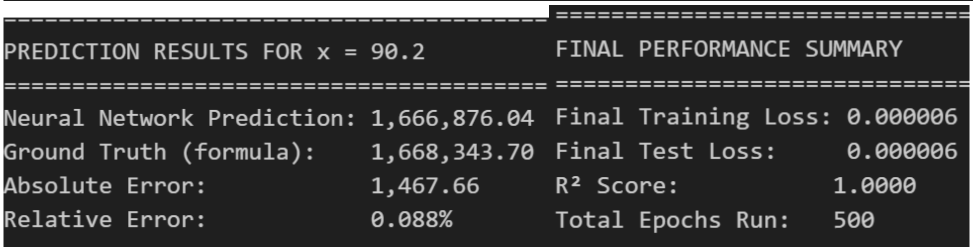
Discussion:

Switching the hidden layer activation from ReLU to Tanh slowed down learning. While the network eventually stabilized, it struggled more with vanishing gradients, which limited its ability to represent the polynomial accurately.

Performance: Underfitting

Experiment 4: Using Adam's Optimizer





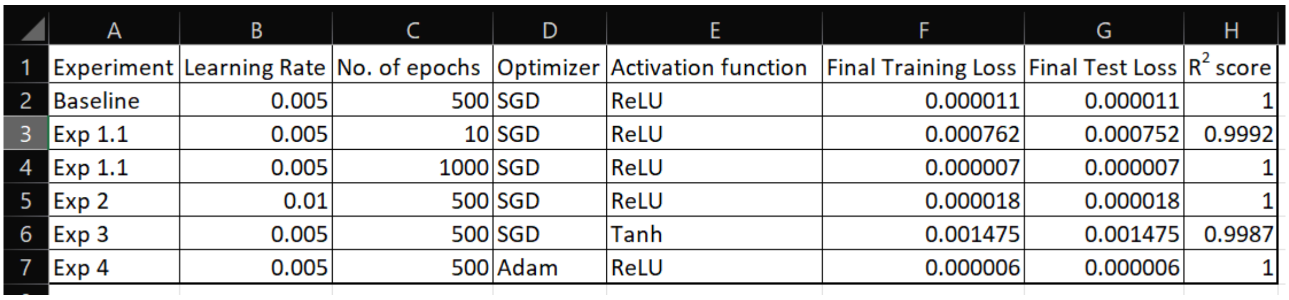
```
========================================
PREDICTION RESULTS FOR x = 90.2
========================================
Neural Network Prediction: 1,666,876.04
Ground Truth (formula):    1,668,343.70
Absolute Error:            1,467.66
Relative Error:            0.088%
```

```
==============================
FINAL PERFORMANCE SUMMARY
==============================
Final Training Loss: 0.000006
Final Test Loss:     0.000006
R² Score:           1.0000
Total Epochs Run:   500
```

Discussion:

Replacing vanilla SGD with the Adam optimizer significantly improved training stability and convergence. The training and test losses decreased smoothly and plateaued earlier, and the test loss reached a slightly better minimum than with SGD.

Results Table:



	A	B	C	D	E	F	G	H
1	Experiment	Learning Rate	No. of epochs	Optimizer	Activation function	Final Training Loss	Final Test Loss	R^2 score
2	Baseline	0.005	500	SGD	ReLU	0.000011	0.000011	1
3	Exp 1.1	0.005	10	SGD	ReLU	0.000762	0.000752	0.9992
4	Exp 1.1	0.005	1000	SGD	ReLU	0.000007	0.000007	1
5	Exp 2	0.01	500	SGD	ReLU	0.000018	0.000018	1
6	Exp 3	0.005	500	SGD	Tanh	0.001475	0.001475	0.9987
7	Exp 4	0.005	500	Adam	ReLU	0.000006	0.000006	1

5. Conclusion

The experiments demonstrated the importance of tuning hyperparameters and choosing the right optimizer. While the baseline performed well, Adam with ReLU activation provided the best trade-off between speed and accuracy, achieving the most reliable predictions of the polynomial function.