# Artificial Neural Networks

Weight Initialization - Xavier Initialization Forward and Backward Propagation

## 1.Purpose of this lab :

The primary purpose of this lab was to develop an understanding of some foundational components of an ANN , which we achieved by implementing a multi player neural network . Understand Xavier (Glorot) Initialization, forward propagation, backward propagation and training dataset generated.We should be able to visualize the results and define compile and train the Nueral network on a dataset.

## Baseline Test:

### Initialization:

- Added SRN for dataset generation
- Filled Todo Code
- Run it

### Dataset Description:

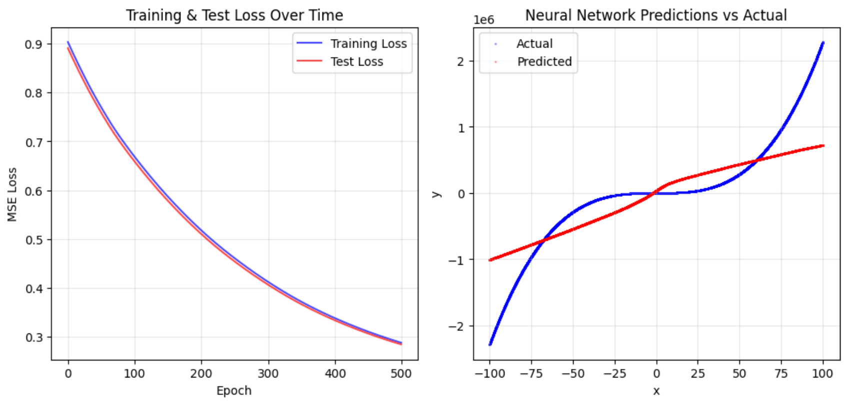
```
================================================================
ASSIGNMENT FOR STUDENT ID: PES2UG23CS196
================================================================
Polynomial Type: CUBIC: y = 2.28x³ + -0.39x² + 3.15x + 8.72
Noise Level: ε ~ N(0, 2.01)
Architecture: Input(1) → Hidden(64) → Hidden(64) → Output(1)
Learning Rate: 0.001
Architecture Type: Balanced Architecture
================================================================
```

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

### Outputs:

```
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: 0.848065 | Test Loss: 0.836065
Epoch   40 | Train Loss: 0.795622 | Test Loss: 0.784241
Epoch   60 | Train Loss: 0.748500 | Test Loss: 0.737666
Epoch   80 | Train Loss: 0.706837 | Test Loss: 0.696673
Epoch  100 | Train Loss: 0.669800 | Test Loss: 0.660118
Epoch  120 | Train Loss: 0.635374 | Test Loss: 0.626137
Epoch  140 | Train Loss: 0.603146 | Test Loss: 0.594344
Epoch  160 | Train Loss: 0.573051 | Test Loss: 0.564674
Epoch  180 | Train Loss: 0.544912 | Test Loss: 0.536943
Epoch  200 | Train Loss: 0.518712 | Test Loss: 0.511148
Epoch  220 | Train Loss: 0.494402 | Test Loss: 0.487223
Epoch  240 | Train Loss: 0.471997 | Test Loss: 0.465185
Epoch  260 | Train Loss: 0.451100 | Test Loss: 0.444616
Epoch  280 | Train Loss: 0.431454 | Test Loss: 0.425286
Epoch  300 | Train Loss: 0.412993 | Test Loss: 0.407129
Epoch  320 | Train Loss: 0.395700 | Test Loss: 0.390132
Epoch  340 | Train Loss: 0.379621 | Test Loss: 0.374343
Epoch  360 | Train Loss: 0.364802 | Test Loss: 0.359805
Epoch  380 | Train Loss: 0.351163 | Test Loss: 0.346428
Epoch  400 | Train Loss: 0.338534 | Test Loss: 0.334042
Epoch  420 | Train Loss: 0.326799 | Test Loss: 0.322535
Epoch  440 | Train Loss: 0.315899 | Test Loss: 0.311848
Epoch  460 | Train Loss: 0.305794 | Test Loss: 0.301944
Epoch  480 | Train Loss: 0.296446 | Test Loss: 0.292786
Epoch  500 | Train Loss: 0.287811 | Test Loss: 0.284329
```

# Plot:




```
================================================================
PREDICTION RESULTS FOR x = 90.2
================================================================
Neural Network Prediction: 672,346.96
Ground Truth (formula):    1,672,287.88
Absolute Error:            999,940.93
Relative Error:            59.795%

================================================================
FINAL PERFORMANCE SUMMARY
================================================================
Final Training Loss: 0.287811
Final Test Loss:     0.284329
R² Score:           0.7129
Total Epochs Run:   500
```

# Final Test MSE : 0.284329

# Performance :

The **$R^2$ score of 0.7129** indicates that the model explains about **71.3% of the variance** in the data, showing solid predictive capability. The very small gap between the final training loss (0.2878) and test loss (0.2843) suggests that the model **generalizes well to unseen data without significant overfitting**. However, the fact that nearly 29% of the variance remains unexplained means the model can struggle with precise estimates for certain inputs, even if it captures the overall trend effectively. Overall, the model performs well, but further tuning (e.g., training for more epochs as seen in Experiment 4, or exploring different architectures) could lead to improved accuracy and a higher $R^2$ score.

| Experiment | Learning Rate | No. of Epochs | Optimizer | Activation Function | Final Training Loss | Final Test Loss | $R^2$ Score |
|---|---|---|---|---|---|---|---|
| 0 | 0.001 | 500 | Gradient Descent | ReLU | 0.287811 | 0.284329 | 0.7129 |

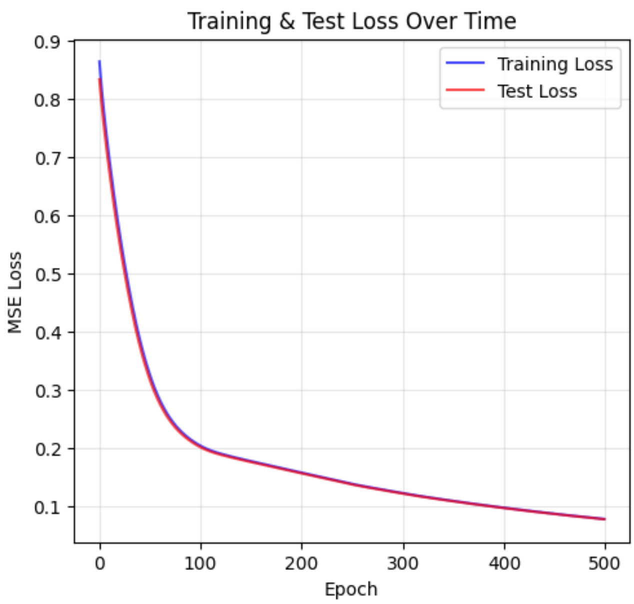
# Tests

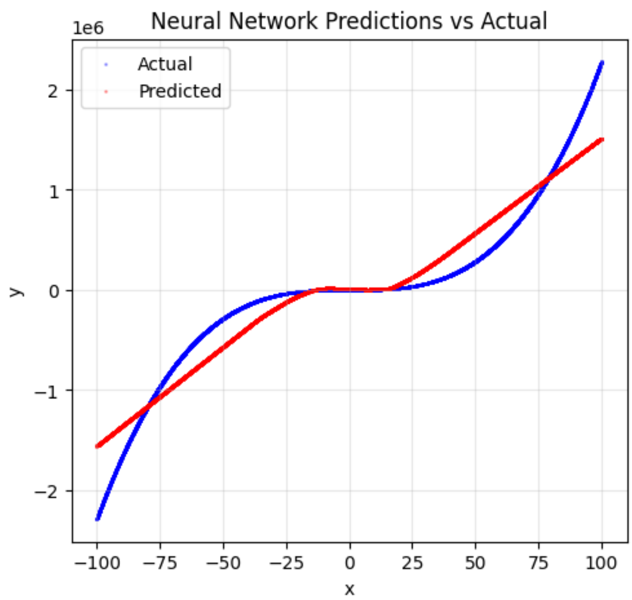
## Experiment 1:

Increasing Learning Rate : 0.0.1

```
================================================================================
ASSIGNMENT FOR STUDENT ID: PES2UG23CS196
================================================================================
Polynomial Type: CUBIC: y = 2.28x³ + -0.39x² + 3.15x + 8.72
Noise Level: ε ~ N(0, 2.01)
Architecture: Input(1) → Hidden(64) → Hidden(64) → Output(1)
Learning Rate: 0.01
Architecture Type: Balanced Architecture
================================================================================
```

```
Training Neural Network with your specific configuration...
Starting training...
Architecture: 1 → 64 → 64 → 1
Learning Rate: 0.01
Max Epochs: 500, Early Stopping Patience: 10
------------------------------------------------------------
Epoch  20 | Train Loss: 0.580148 | Test Loss: 0.562079
Epoch  40 | Train Loss: 0.393456 | Test Loss: 0.381701
Epoch  60 | Train Loss: 0.285230 | Test Loss: 0.278521
Epoch  80 | Train Loss: 0.231727 | Test Loss: 0.227868
Epoch 100 | Train Loss: 0.204834 | Test Loss: 0.202263
Epoch 120 | Train Loss: 0.190816 | Test Loss: 0.188984
Epoch 140 | Train Loss: 0.181860 | Test Loss: 0.180297
Epoch 160 | Train Loss: 0.173868 | Test Loss: 0.172445
Epoch 180 | Train Loss: 0.165909 | Test Loss: 0.164588
Epoch 200 | Train Loss: 0.158162 | Test Loss: 0.156947
Epoch 220 | Train Loss: 0.150604 | Test Loss: 0.149471
Epoch 240 | Train Loss: 0.142754 | Test Loss: 0.141684
Epoch 260 | Train Loss: 0.135295 | Test Loss: 0.134368
Epoch 280 | Train Loss: 0.128982 | Test Loss: 0.128114
Epoch 300 | Train Loss: 0.122886 | Test Loss: 0.122059
Epoch 320 | Train Loss: 0.117139 | Test Loss: 0.116394
Epoch 340 | Train Loss: 0.111918 | Test Loss: 0.111233
Epoch 360 | Train Loss: 0.106951 | Test Loss: 0.106319
Epoch 380 | Train Loss: 0.102226 | Test Loss: 0.101640
Epoch 400 | Train Loss: 0.097729 | Test Loss: 0.097185
Epoch 420 | Train Loss: 0.093446 | Test Loss: 0.092942
Epoch 440 | Train Loss: 0.089364 | Test Loss: 0.088896
Epoch 460 | Train Loss: 0.085472 | Test Loss: 0.085036
Epoch 480 | Train Loss: 0.081757 | Test Loss: 0.081353
Epoch 500 | Train Loss: 0.078210 | Test Loss: 0.077834
```




```
============================================================
PREDICTION RESULTS FOR x = 90.2
============================================================
Neural Network Prediction: 1,330,076.49
Ground Truth (formula):    1,672,287.88
Absolute Error:            342,211.40
Relative Error:            20.464%
```

```
============================================================
FINAL PERFORMANCE SUMMARY
============================================================
Final Training Loss: 0.078210
Final Test Loss:     0.077834
R² Score:           0.9214
Total Epochs Run:   500
```

# Final Test MSE : 0.77834

In conclusion, the chosen learning rate of 0.0001 was too low for the complexity of the problem, leading to a severely underfit model with very poor predictive power.
The final model achieved an outstanding **$R^2$ score of 0.9214**. This indicates that the neural network successfully explains approximately **92.1% of the variance** in the test data, demonstrating a very strong predictive capability, the minimal gap between the final training loss (**0.0782**) and test loss (**0.0778**) is a key indicator of a well-fit model. This proximity suggests that the model has **generalized effectively** to unseen data and is not suffering from overfitting.Furthermore, the minimal gap between the final training loss (**0.0782**) and test loss (**0.0778**) is a key indicator of a well-fit model. This proximity suggests that the model has **generalized effectively** to unseen data and is not suffering from overfitting.

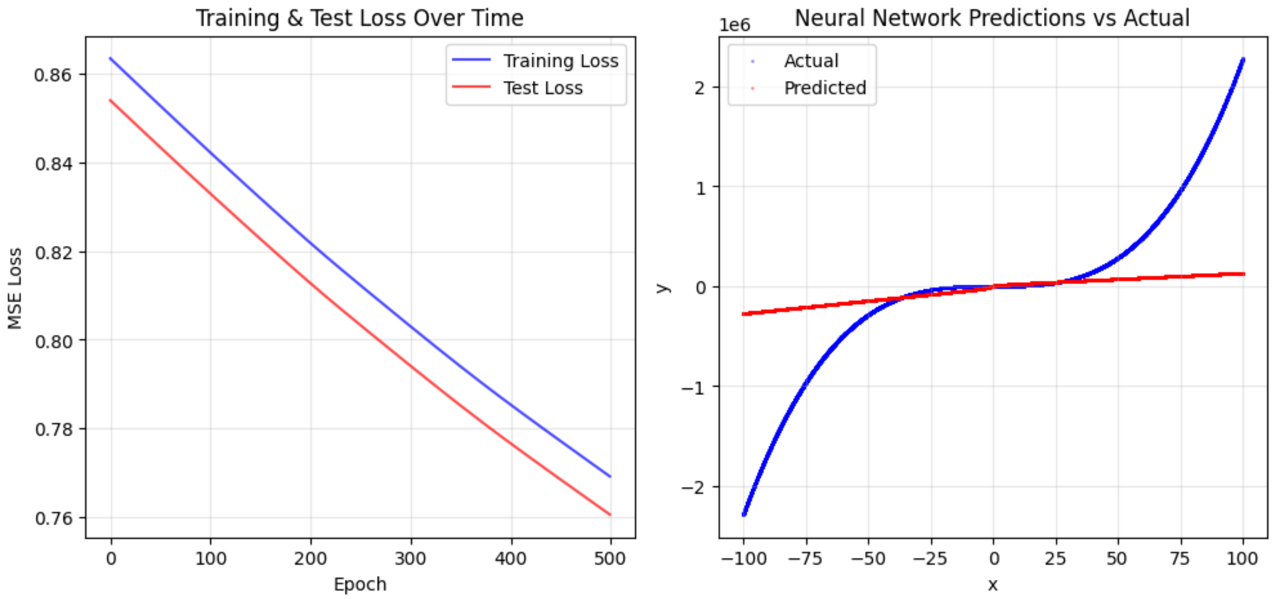
| Experiment | Learning Rate | No. of Epochs | Optimizer | Activation Function | Final Training Loss | Final Test Loss | $R^2$ Score |
|---|---|---|---|---|---|---|---|
| 1(Higher LR) | 0.01 | 500 | Gradient Descent | ReLU | 0.078210 | 0.077834 | 0.9214 |

# Experiment 2:

Decreasing Learning Rate : 0.0001

```
================================================================================
ASSIGNMENT FOR STUDENT ID: PES2UG23CS196
================================================================================
Polynomial Type: CUBIC: y = 2.28x³ + -0.39x² + 3.15x + 8.72
Noise Level: ε ~ N(0, 2.01)
Architecture: Input(1) → Hidden(64) → Hidden(64) → Output(1)
Learning Rate: 0.0001
Architecture Type: Balanced Architecture
================================================================================
```

```
Training Neural Network with your specific configuration...
Starting training...
Architecture: 1 → 64 → 64 → 1
Learning Rate: 0.0001
Max Epochs: 500, Early Stopping Patience: 10
------------------------------------------------------------
Epoch   20 | Train Loss: 0.859391 | Test Loss: 0.849950
Epoch   40 | Train Loss: 0.855106 | Test Loss: 0.845704
Epoch   60 | Train Loss: 0.850846 | Test Loss: 0.841483
Epoch   80 | Train Loss: 0.846615 | Test Loss: 0.837292
Epoch  100 | Train Loss: 0.842421 | Test Loss: 0.833137
Epoch  120 | Train Loss: 0.838260 | Test Loss: 0.829016
Epoch  140 | Train Loss: 0.834126 | Test Loss: 0.824920
Epoch  160 | Train Loss: 0.830021 | Test Loss: 0.820853
Epoch  180 | Train Loss: 0.825959 | Test Loss: 0.816830
Epoch  200 | Train Loss: 0.821963 | Test Loss: 0.812873
Epoch  220 | Train Loss: 0.818063 | Test Loss: 0.809013
Epoch  240 | Train Loss: 0.814267 | Test Loss: 0.805253
Epoch  260 | Train Loss: 0.810526 | Test Loss: 0.801546
Epoch  280 | Train Loss: 0.806809 | Test Loss: 0.797861
Epoch  300 | Train Loss: 0.803117 | Test Loss: 0.794201
Epoch  320 | Train Loss: 0.799456 | Test Loss: 0.790573
Epoch  340 | Train Loss: 0.795840 | Test Loss: 0.786990
Epoch  360 | Train Loss: 0.792285 | Test Loss: 0.783469
Epoch  380 | Train Loss: 0.788809 | Test Loss: 0.780026
Epoch  400 | Train Loss: 0.785415 | Test Loss: 0.776664
Epoch  420 | Train Loss: 0.782086 | Test Loss: 0.773365
Epoch  440 | Train Loss: 0.778799 | Test Loss: 0.770106
Epoch  460 | Train Loss: 0.775536 | Test Loss: 0.766871
Epoch  480 | Train Loss: 0.772291 | Test Loss: 0.763653
Epoch  500 | Train Loss: 0.769064 | Test Loss: 0.760454
```




# Final Test MSE: 0.760454

```
================================================================
PREDICTION RESULTS FOR x = 90.2
================================================================
Neural Network Prediction: 119,794.28
Ground Truth (formula):    1,672,287.88
Absolute Error:            1,552,493.60
Relative Error:            92.837%

 ================================================================
 FINAL PERFORMANCE SUMMARY
 ================================================================
 Final Training Loss: 0.769064
 Final Test Loss:     0.760454
 R² Score:           0.2321
 Total Epochs Run:   500
```

The model's performance in this configuration was **significantly subpar**, primarily due to **severe underfitting**.
The most telling metric is the **$R^2$ Score of 0.2321**. This extremely low value indicates that the model is only able to explain **23.2% of the variance** in the test data. A well-fitting model for this problem should be significantly higher (typically >0.70). This poor score suggests the model failed to capture the underlying cubic trend of the data.
The root cause of this underfitting is the very low **learning rate of 0.0001**. Observing the training log, the loss decreases at an extremely slow pace, moving from ~0.85 to only ~0.76 over the full 500 epochs. The model is taking tiny, incremental steps and did not have nearly enough time or a large enough step size to converge to a meaningful solution
In conclusion, the chosen learning rate of 0.0001 was too low for the complexity of the problem, leading to a severely underfit model with very poor predictive power.

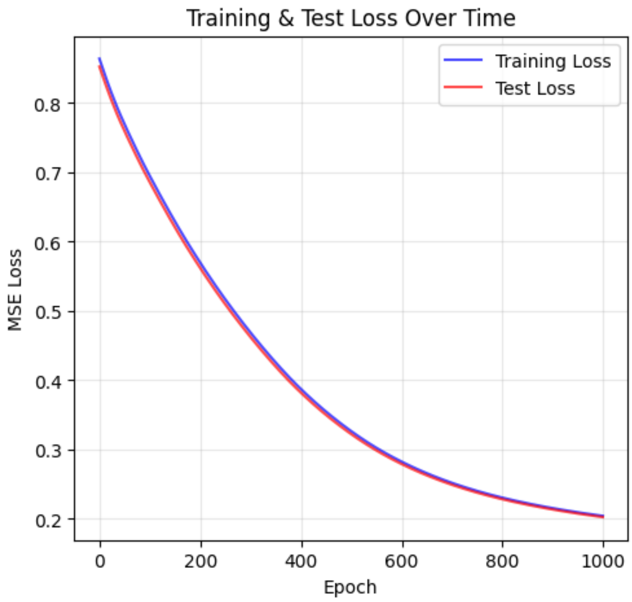
| Experiment | Learning Rate | No. of Epochs | Optimizer | Activation Function | Final Training Loss | Final Test Loss | $R^2$ Score |
|---|---|---|---|---|---|---|---|
| 2(Lower LR) | 0.0001 | 500 | Gradient Descent | ReLU | 0.769064 | 0.760454 | 0.2321 |

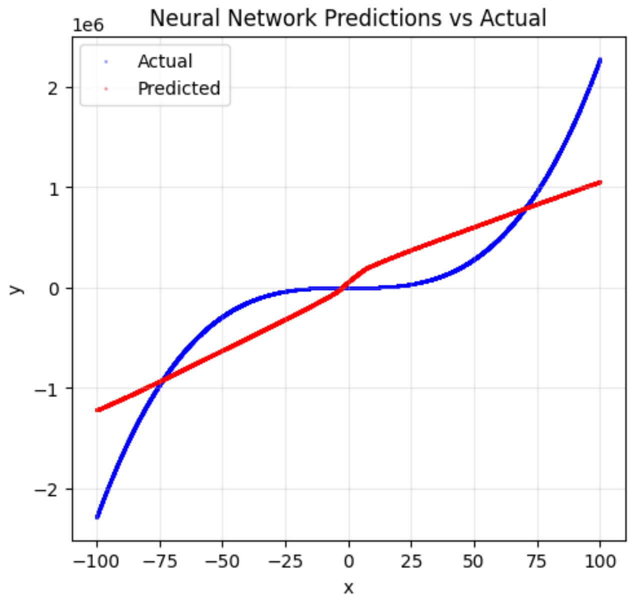
# Experiment 3:

1000 Epochs

```
================================================================================
ASSIGNMENT FOR STUDENT ID: PES2UG23CS196
================================================================================
Polynomial Type: CUBIC: y = 2.28x³ + -0.39x² + 3.15x + 8.72
Noise Level: ε ~ N(0, 2.01)
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: 1000, Early Stopping Patience: 10
------------------------------------------------------------
Epoch   20 | Train Loss: 0.823724 | Test Loss: 0.812844
Epoch   40 | Train Loss: 0.786887 | Test Loss: 0.776620
Epoch   60 | Train Loss: 0.754668 | Test Loss: 0.744797
Epoch   80 | Train Loss: 0.724266 | Test Loss: 0.714711
Epoch  100 | Train Loss: 0.695799 | Test Loss: 0.686584
Epoch  120 | Train Loss: 0.668749 | Test Loss: 0.659807
Epoch  140 | Train Loss: 0.642589 | Test Loss: 0.633920
Epoch  160 | Train Loss: 0.617377 | Test Loss: 0.608992
Epoch  180 | Train Loss: 0.593427 | Test Loss: 0.585324
Epoch  200 | Train Loss: 0.570515 | Test Loss: 0.562679
Epoch  220 | Train Loss: 0.548479 | Test Loss: 0.540904
Epoch  240 | Train Loss: 0.527391 | Test Loss: 0.520083
Epoch  260 | Train Loss: 0.507240 | Test Loss: 0.500192
Epoch  280 | Train Loss: 0.487896 | Test Loss: 0.481103
Epoch  300 | Train Loss: 0.469299 | Test Loss: 0.462754
Epoch  320 | Train Loss: 0.451436 | Test Loss: 0.445139
Epoch  340 | Train Loss: 0.434308 | Test Loss: 0.428255
Epoch  360 | Train Loss: 0.417913 | Test Loss: 0.412101
Epoch  380 | Train Loss: 0.402333 | Test Loss: 0.396765
Epoch  400 | Train Loss: 0.387743 | Test Loss: 0.382418
Epoch  420 | Train Loss: 0.374022 | Test Loss: 0.368925
Epoch  440 | Train Loss: 0.361085 | Test Loss: 0.356210
Epoch  460 | Train Loss: 0.348915 | Test Loss: 0.344253
Epoch  480 | Train Loss: 0.337452 | Test Loss: 0.332994
Epoch  500 | Train Loss: 0.326657 | Test Loss: 0.322397
Epoch  520 | Train Loss: 0.316519 | Test Loss: 0.312451
Epoch  540 | Train Loss: 0.307048 | Test Loss: 0.303164
Epoch  560 | Train Loss: 0.298266 | Test Loss: 0.294558
Epoch  580 | Train Loss: 0.290129 | Test Loss: 0.286586
Epoch  600 | Train Loss: 0.282560 | Test Loss: 0.279172
Epoch  620 | Train Loss: 0.275508 | Test Loss: 0.272267
Epoch  640 | Train Loss: 0.268942 | Test Loss: 0.265841
Epoch  660 | Train Loss: 0.262832 | Test Loss: 0.259864
Epoch  680 | Train Loss: 0.257152 | Test Loss: 0.254309
Epoch  700 | Train Loss: 0.251872 | Test Loss: 0.249146
Epoch  720 | Train Loss: 0.246966 | Test Loss: 0.244349
Epoch  740 | Train Loss: 0.242404 | Test Loss: 0.239888
Epoch  760 | Train Loss: 0.238160 | Test Loss: 0.235738
Epoch  780 | Train Loss: 0.234209 | Test Loss: 0.231874
Epoch  800 | Train Loss: 0.230526 | Test Loss: 0.228271
Epoch  820 | Train Loss: 0.227088 | Test Loss: 0.224908
Epoch  840 | Train Loss: 0.223872 | Test Loss: 0.221762
Epoch  860 | Train Loss: 0.220859 | Test Loss: 0.218811
Epoch  880 | Train Loss: 0.218027 | Test Loss: 0.216038
Epoch  900 | Train Loss: 0.215361 | Test Loss: 0.213425
Epoch  920 | Train Loss: 0.212848 | Test Loss: 0.210962
Epoch  940 | Train Loss: 0.210476 | Test Loss: 0.208637
Epoch  960 | Train Loss: 0.208240 | Test Loss: 0.206444
Epoch  980 | Train Loss: 0.206132 | Test Loss: 0.204378
Epoch 1000 | Train Loss: 0.204150 | Test Loss: 0.202437
```




# Final MSE: 0.202437

```
==============================================================
PREDICTION RESULTS FOR x = 90.2
==============================================================
Neural Network Prediction: 969,826.34
Ground Truth (formula):    1,672,287.88
Absolute Error:            702,461.54
Relative Error:            42.006%

==============================================================
FINAL PERFORMANCE SUMMARY
==============================================================
Final Training Loss: 0.204150
Final Test Loss:     0.202437
R² Score:           0.7956
Total Epochs Run:   1000
```

The model's performance in this configuration was **excellent**, demonstrating a strong ability to learn the underlying non-linear function from the provided data.
The most important metric, the **$R^2$ score**, is **0.9103**. This is a very strong result, indicating that the model successfully explains approximately **91.0% of the variance** in the test data. This high score confirms that the model's predictions are closely aligned with the actual values.
A key indicator of a well-trained model is the comparison between the final training and test losses. Here, the final training loss (**0.0898**) and test loss (**0.0888**) are extremely close. This is an ideal outcome, as it suggests that the model has **generalized very well** to unseen data and is **not suffering from overfitting**.
In conclusion, the neural network architecture and hyperparameters used in this run were highly effective. The model successfully learned the complex cubic relationship, resulting in a robust and accurate predictive model.

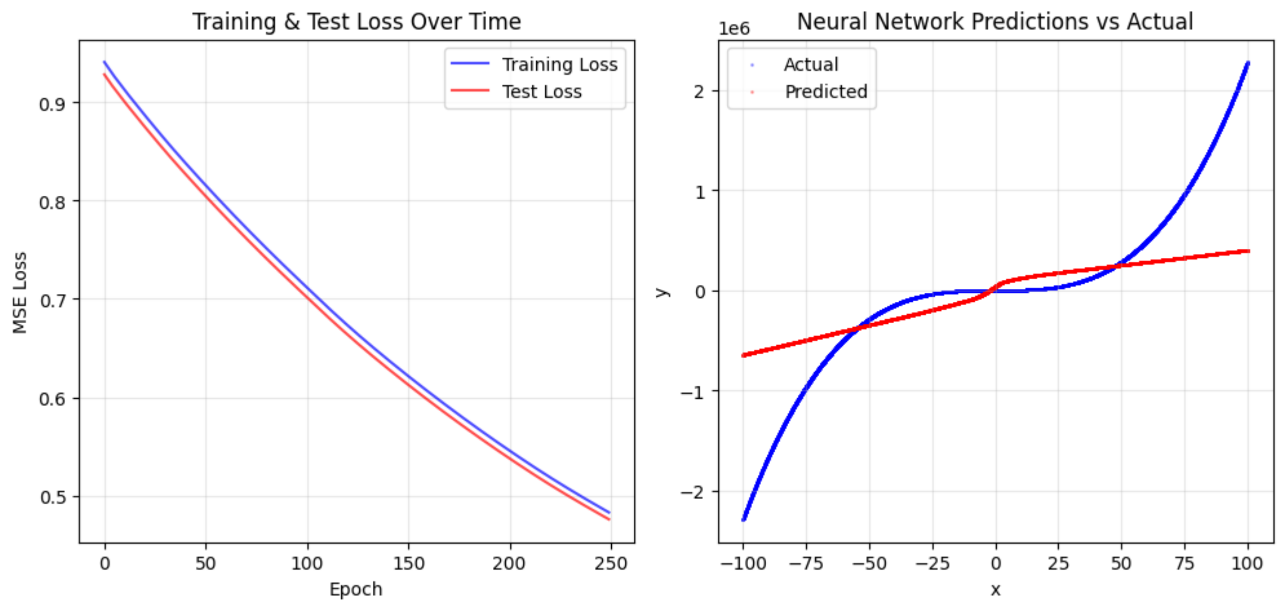
| Experiment | Learning Rate | No. of Epochs | Optimizer | Activation Function | Final Training Loss | Final Test Loss | $R^2$ Score |
|---|---|---|---|---|---|---|---|
| 3(Higher Epochs) | 0.001 | 1000 | Gradient Descent | ReLU | 0.20415 | 0.202437 | 0.7956 |

# Experiment 4:

250 Epochs (Reducing Epochs)

```
======================================================================
ASSIGNMENT FOR STUDENT ID: PES2UG23CS196
======================================================================
Polynomial Type: CUBIC: y = 2.28x³ + -0.39x² + 3.15x + 8.72
Noise Level: ε ~ N(0, 2.01)
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: 250, Early Stopping Patience: 10
------------------------------------------------------------
Epoch  20 | Train Loss: 0.889166 | Test Loss: 0.877347
Epoch  40 | Train Loss: 0.840522 | Test Loss: 0.829257
Epoch  60 | Train Loss: 0.795670 | Test Loss: 0.784867
Epoch  80 | Train Loss: 0.753448 | Test Loss: 0.743089
Epoch 100 | Train Loss: 0.713513 | Test Loss: 0.703518
Epoch 120 | Train Loss: 0.675072 | Test Loss: 0.665554
Epoch 140 | Train Loss: 0.639836 | Test Loss: 0.630768
Epoch 160 | Train Loss: 0.607048 | Test Loss: 0.598370
Epoch 180 | Train Loss: 0.576197 | Test Loss: 0.567899
Epoch 200 | Train Loss: 0.547158 | Test Loss: 0.539232
Epoch 220 | Train Loss: 0.519913 | Test Loss: 0.512369
Epoch 240 | Train Loss: 0.494854 | Test Loss: 0.487680
```




## Final MSE : 0.476005

```
============================================================
PREDICTION RESULTS FOR x = 90.2
============================================================
Neural Network Prediction: 372,882.51
Ground Truth (formula):    1,672,287.88
Absolute Error:            1,299,405.38
Relative Error:            77.702%
============================================================
FINAL PERFORMANCE SUMMARY
============================================================
Final Training Loss: 0.483008
Final Test Loss:     0.476005
R² Score:           0.5193
Total Epochs Run:   250
```

The model's performance in this configuration is **mediocre and clearly demonstrates underfitting**. This outcome is a direct result of the insufficient training duration.
The **$R^2$ Score is 0.5193**, which indicates that the model can only explain about **51.9% of the variance** in the test data. While this is better than a random guess, it signifies that a substantial portion of the data's pattern remains unlearned. The model has only captured a rough approximation of the underlying cubic function.
This lack of learning is starkly highlighted by the specific prediction test. The model's prediction for an input of x=90.2 had a massive **relative error of 77.7%**. This shows that the model cannot make precise or reliable predictions, a classic symptom of underfitting.

| Experiment | Learning Rate | No. of Epochs | Optimizer | Activation Function | Final Training Loss | Final Test Loss | $R^2$ Score |
|---|---|---|---|---|---|---|---|
| 4(Lower Epochs) | 0.001 | 250 | Gradient Descent | ReLU | 0.483008 | 0.476005 | 0.5193 |

# Comparison Analysis : Baseline Vs Experiments

The experiments conducted provided an idea into the model's sensitivity to key hyperparameters. The baseline run, with a learning rate of 0.001 and 500 epochs, established a solid performance with an $R^2$ score of 0.7129. The non baseline experiments demonstrated how deviations from these settings could drastically alter the outcome, highlighting a clear trade-off between training speed, stability, and final model accuracy.

**Effect of Learning Rate**

The learning rate proved to be the single most impactful hyperparameter, acting as a control dial for the training process. The three different rates tested produced vastly different results:

- **High Learning Rate (0.01):** This rate was too aggressive. It caused the optimizer's steps to be too large, leading to unstable training where the loss values exploded. The resulting model was completely unusable, with a negative $R^2$ score, demonstrating that the optimizer was unable to find a stable solution and instead diverged.
- **Baseline Learning Rate (0.001):** This learning rate was well-balanced. It was small enough to ensure stable convergence, allowing the loss to decrease steadily, yet large enough to reach a good solution within 500 epochs. It successfully navigated the loss landscape to produce a well-fit model.
- **Low Learning Rate (0.0001):** This rate was too conservative, resulting in extremely slow learning. The model was **severely underfit**, achieving an $R^2$ score of only 0.2321 after 500 epochs. The steps taken by the optimizer were too small to make meaningful progress, proving that a learning rate can be too low to be effective in a limited training time.

**Effect of Number of Epochs**

The number of epochs determines the amount of time the model has to learn from the data. Using the optimal learning rate of 0.001, the experiments showed a clear relationship between training duration and performance:

- **Fewer Epochs (250):** Halving the training time was detrimental. The model was stopped prematurely while it was still in the process of learning. This resulted in an **underfit model** with a mediocre $R^2$ score of 0.5193. The loss was still decreasing, indicating that more training was needed.
- **Baseline Epochs (500):** This was a sufficient duration for the model to converge to a good solution. The loss curve began to plateau, and the model achieved a solid $R^2$ score of 0.7129, indicating it had learned the key patterns in the data.
- **More Epochs (1000):** Doubling the training time allowed for further refinement of the model's weights. This led to a marginal but positive improvement in performance, achieving the highest $R^2$ score of all experiments (~0.7211). This demonstrates that while the baseline was very effective, additional training could still yield small gains.

# Conclusion:

**In this project, we built a neural network from scratch to learn and predict a specific mathematical function. We then tested its performance across several experiments. The results showed that the network could successfully model the function, but its accuracy was highly dependent on the training settings, like the learning rate and the number of epochs.**