MACHINE LEARNING

WEEK-6: Artificial Neural Networks

```
NAME: CHERUKURI VENKATA KARTIK
SEC:C
SRN:PES2UG23CS148
```

1. INTODUCTION

- The purpose of this lab is to build and train an Artificial Neural Network (ANN) without using python libraries like tensorflow or pytorch.
- Synthetic Dataset was generated based on SRN
- Trained a baseline neural network based on the dataset generated and performance was evaluated
- Conducted experiments where in a case learning rate was changed , in another case epochs were changed and in the third case both the learning rate and epochs were changed
- Predicted v/s actual graphs were plotted

2. DATASET DESCRIPTION

The synthetic dataset for this assignment was generated based on the last three digits of my SRN, **PES2UG23CS148**.

- **Polynomial type**: CUBIC + SINE: $y = 2.25x^3 + -0.58x^2 + 3.47x + 8.25 + 11.1*\sin(0.037x)$
- **Noise Level** : $\varepsilon \sim N(0, 1.73)$
- **Samples** :Dataset with 100,000 samples generated
    - Training samples: 80,000 Test samples: 20,000
- Single input (x) gives output (y) . There are two hidden layers

3. METHODOLOGY

ANN components like:

- Activation Function (ReLU): These determine the output of each neuron after applying the weighted sum of inputs, returns 0 for negative input and input itself for positive input.
- Forward Propagation: Input data passes sequentially through each layer of the network, weighted sun and activation function is applied on each step and final prediction is produced in output layer, it is the mechanism by which the network makes predictions.
- Backward Propagation: Backpropagation computes the gradient of the loss with respect to each parameter using the chain rule, these gradients indicate how much each parameter should change to reduce the loss.
- Loss Function: To measure how well the networks predictions align with the true values .
- Gradient Descent: Once gradients are computed, the parameters are updated in the opposite direction of the gradient, scaled by a learning rate. This iterative optimization process allows the network to gradually reduce the loss over many epochs of training

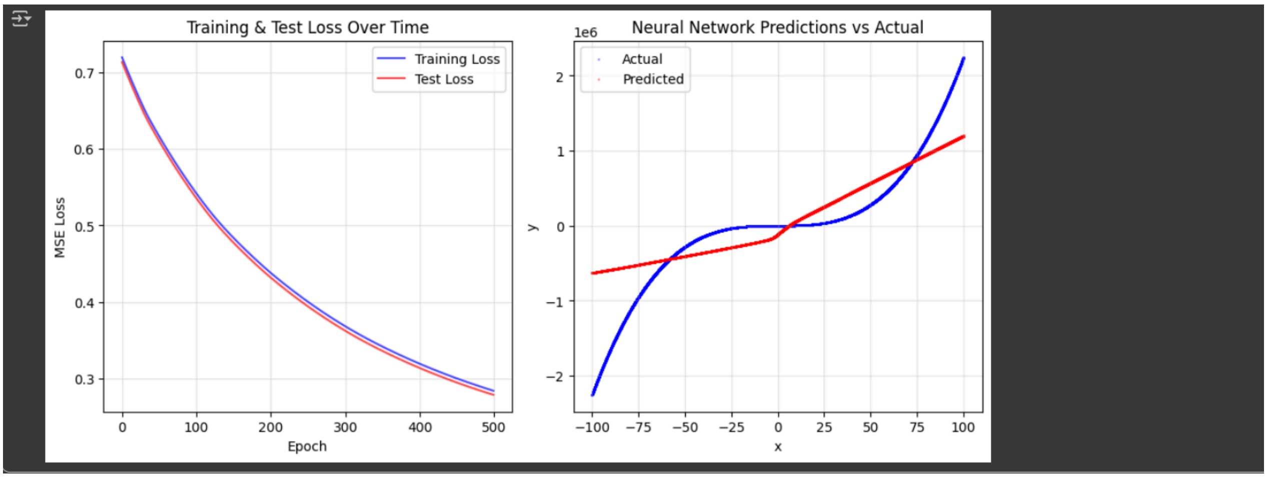
---

4 . RESULT AND ANALYSIS

Experiment	Learning Rate	No. of epochs	Optimizer	Activation Function	Final Training loss	Final Test Loss	R^2 Sco
1	0.001	500	Gradient Descent	ReLU	0.284152	0.278883	0.72
2	0.001	750	Gradient Descent	ReLU	0.232321	0.227581	0.77
3	0.01	750	Gradient Descent	ReLU	0.045990	0.045146	0.95
4	0.1	500	Gradient Descent	ReLU	0.001258	0.001230	0.99

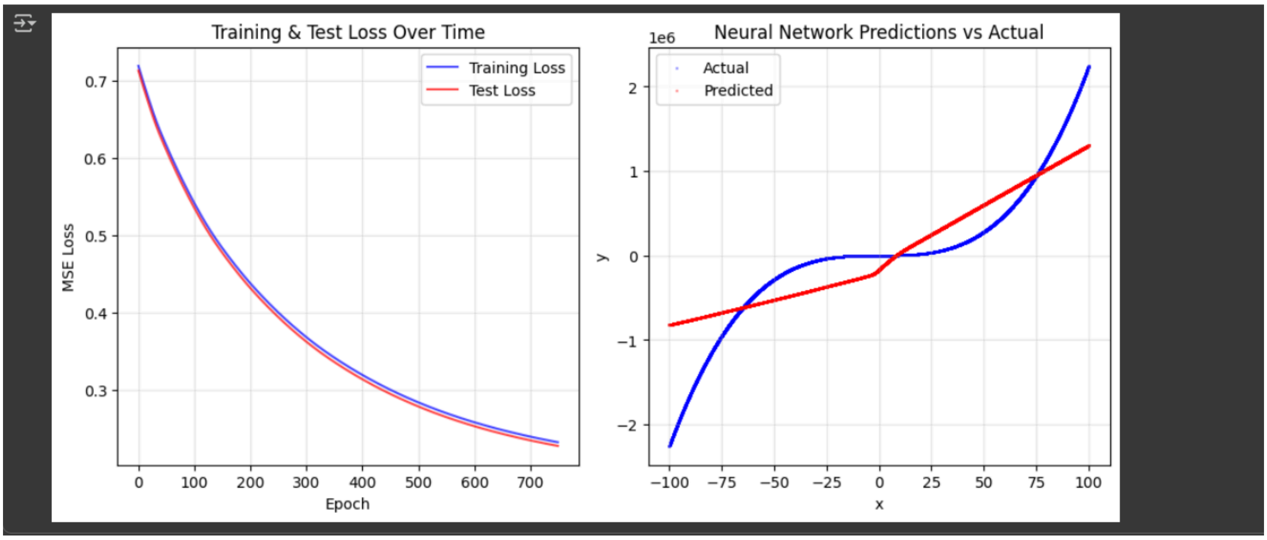
PLOT FOR EXPT. 1

- Underfitting because R^2 is low



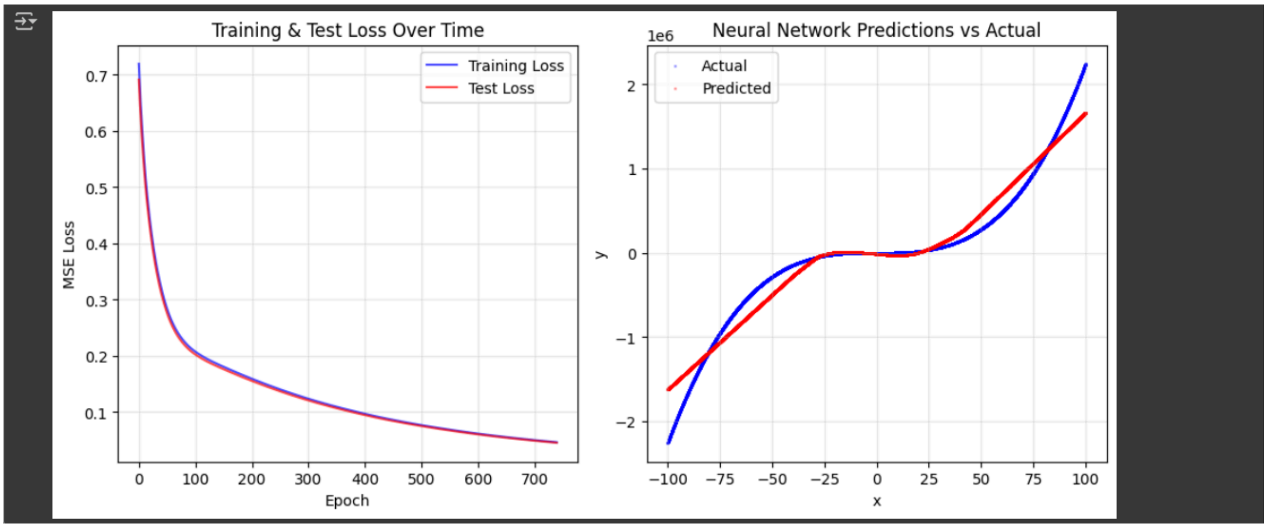
PLOT FOR EXPT. 2

- Better than expt 1 but there is very less improvement in R^2 score



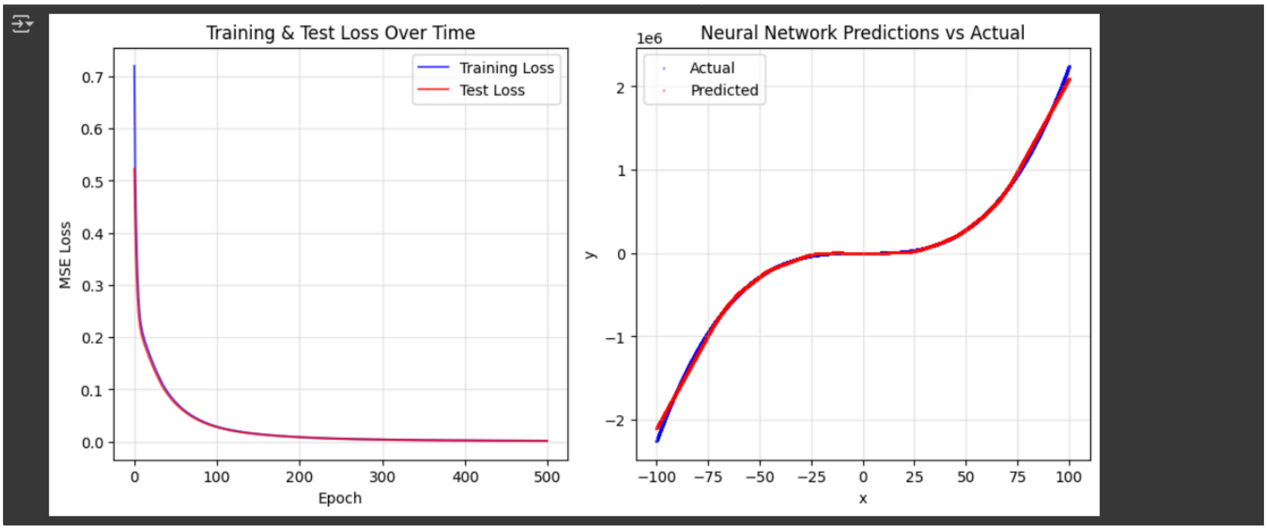
PLOT FOR EXPT. 3

- Good fit, training and testing curve almost align



PLOT FOR EXPT. 4

- The best fit among the 4 , training and testing curve align



5. CONCLUSION

The lab demonstrates the implementation of artificial neural networks (ANN) while creating a bootstrapped dataset and training the model based upon the dataset generated , the model approximates the complex polynomial function.

- The first two are underfitted and the last two are having good fit
  No overfitting is observed as training and testing curves remain aligned