**ASSIGNMENT 3**

**PART-1: NEURAL NETWORK LIBRARY- X-OR PROBLEM:**

**1)Activation Function: Sigmoid**

Epoch 0, Loss: 0.8071524112883687

Epoch 1000, Loss: 0.6789266327900263

Epoch 2000, Loss: 0.6010824636307509

Epoch 3000, Loss: 0.47038522712502484

Epoch 4000, Loss: 0.1959639054068461

Epoch 5000, Loss: 0.07977486951972554

Epoch 6000, Loss: 0.046665783068421514

Epoch 7000, Loss: 0.03241422394211989

Epoch 8000, Loss: 0.024659267402546217

Epoch 9000, Loss: 0.01982803218806773

**Final Accuracy**: **1.0**

**Weights & Biases:**

**Layer 0 weights:**

[[6.56906571 6.5751495]

[4.60149937 4.60253323]]

**Layer 0 biases:**

[-2.8873947 -7.04025048]

**Layer 2 weights:**

[[ 9.70372595 -10.41352544]]

**Layer 2 biases:**

[-4.44429475]

**Sigmoid layer 1 output:**

[[5.27802172e-02 8.75140596e-04]

[9.75582980e-01 8.03414159e-02]

[9.75437639e-01 8.02650602e-02]

[9.99964884e-01 8.96949659e-01]]

**Sigmoid layer 3 output:**

[[0.01905395]

[0.9850187]

[0.98500962]

[0.01660483]]

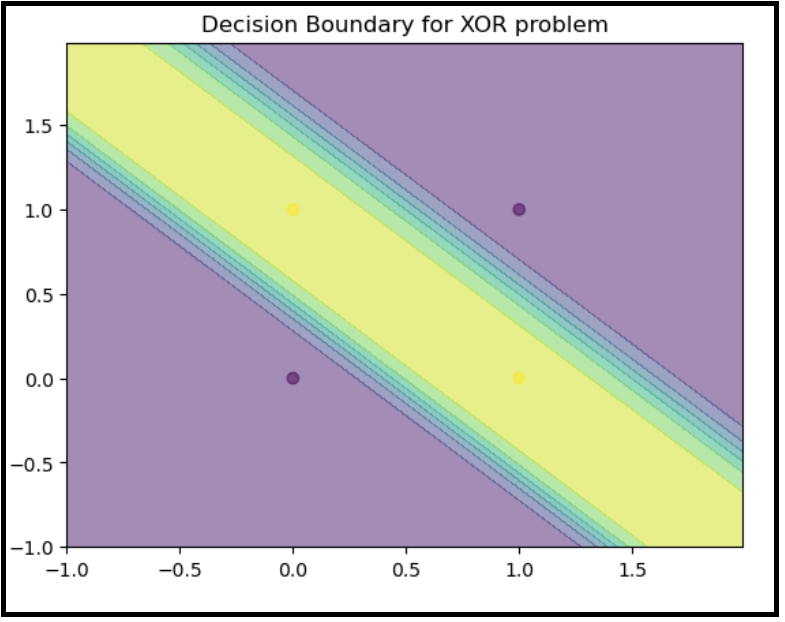
**Hidden layer values:**

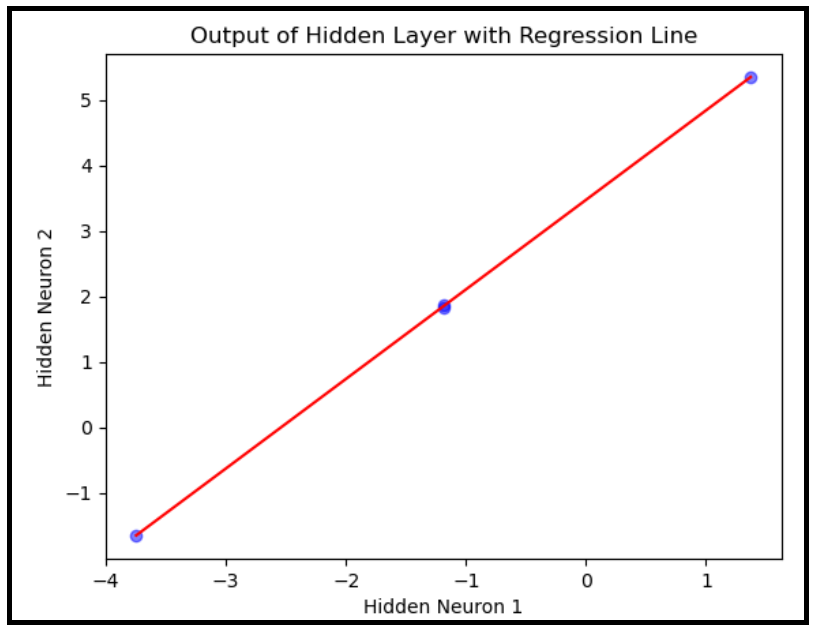
[[-2.8873947 -7.04025048]

[ 3.6877548 -2.43771724]

[ 3.681671 -2.43875111]

[10.2568205 2.16378213]]





**2).Activation Function: Tanh**

Epoch 0, Loss: 0.7880194612814513

Epoch 1000, Loss: 0.004599254127663894

Epoch 2000, Loss: 0.0018317707079696047

Epoch 3000, Loss: 0.001129437561770157

Epoch 4000, Loss: 0.0008128106366359926

Epoch 5000, Loss: 0.0006334410451012898

Epoch 6000, Loss: 0.0005182518857175114

Epoch 7000, Loss: 0.00043813938231224175

Epoch 8000, Loss: 0.0003792549212869441

Epoch 9000, Loss: 0.00033417858082036855

Epoch 10000, Loss: 0.00029858094556348006

Epoch 11000, Loss: 0.0002697677112874187

Epoch 12000, Loss: 0.0002459752956881268

**Final Accuracy: 1.0**

**Layer 0 weights:**

[[ 3.41566079 3.41207601]

[-2.44644127 -2.44565319]]

**Layer 0 biases:**

[-1.6028618 3.5750369]

**Layer 2 weights:**

[[4.84001841 4.96393143]]

**Layer 2 biases:**

[-4.41465789]

**Tanh layer 1 output:**

[[-0.9220982 0.99843162]

[ 0.94775194 0.81080823]

[ 0.94811552 0.81053807]

[ 0.99994209 -0.86605032]]

**Tanh layer 3 output:**

[[-0.9992153]

[ 0.99954791]

[ 0.99954829]

[-0.99913705]]

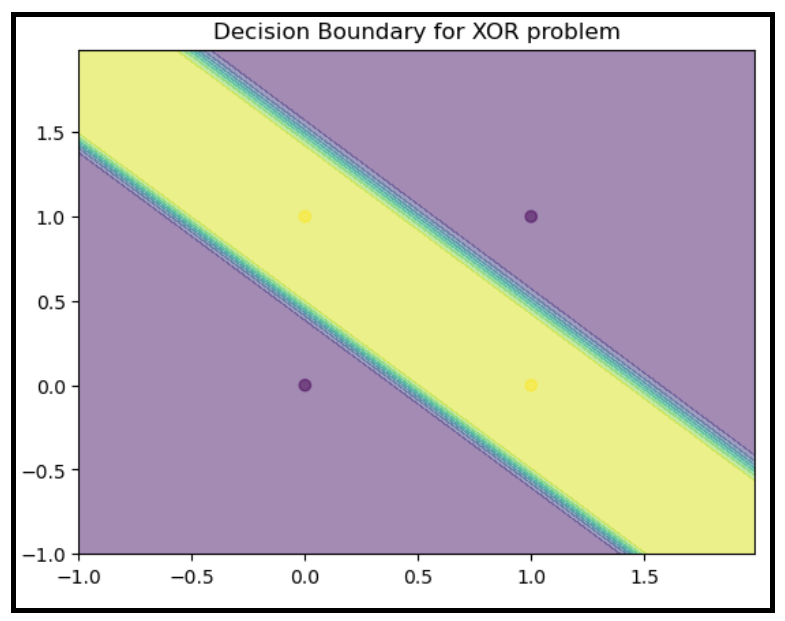
**Hidden layer values:**

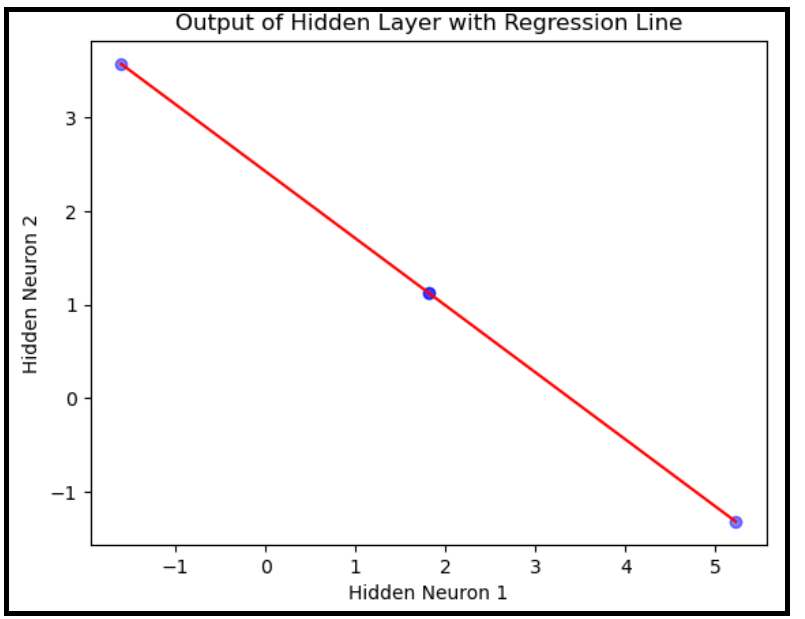
[[-1.6028618 3.5750369]

[ 1.80921421 1.12938371]

[ 1.812799 1.12859563]

[ 5.22487501 -1.31705757]]



****

**Conclusion:**

**1).Convergence Speed:**

* **Sigmoid**: Converged slower with significant loss reduction over 9000 epochs.
* **Tanh**: Converged much faster, with minimal loss by epoch 1000.

**2). Final Loss:**

* **Sigmoid**: Final loss of ~0.0198 after 9000 epochs.
* **Tanh**: Final loss of ~0.000245 after 12000 epochs.

**3). Accuracy:** Both models achieved a final accuracy of 1.0.

**4). Weight Values:** The weights and biases for both activation functions are significantly different, reflecting different optimization landscapes due to the nature of the activation functions.

**PART-2: PREDICTING NYC TAXI TRIP DURATION WITH NEURAL NETWORK LIBRARY**

**Feature implemented:**

Date, Time and Month of the pickup time and drop-off time.

**Normalization Techniques:**

1). Minmax Scaler

2). Standard Scaler

**Implementation:**

* Utilized 3-Cross validation technique.
* Also implemented Clipping to ensure only positive values while taking logarithms and utilizing different hyperparameters, batch size & learning rate.

**I.Minmax Scaler:**

**Hyperparameters:** **layer1 neurons: 64, layer2 neurons: 32, layer3neurons: 16,**

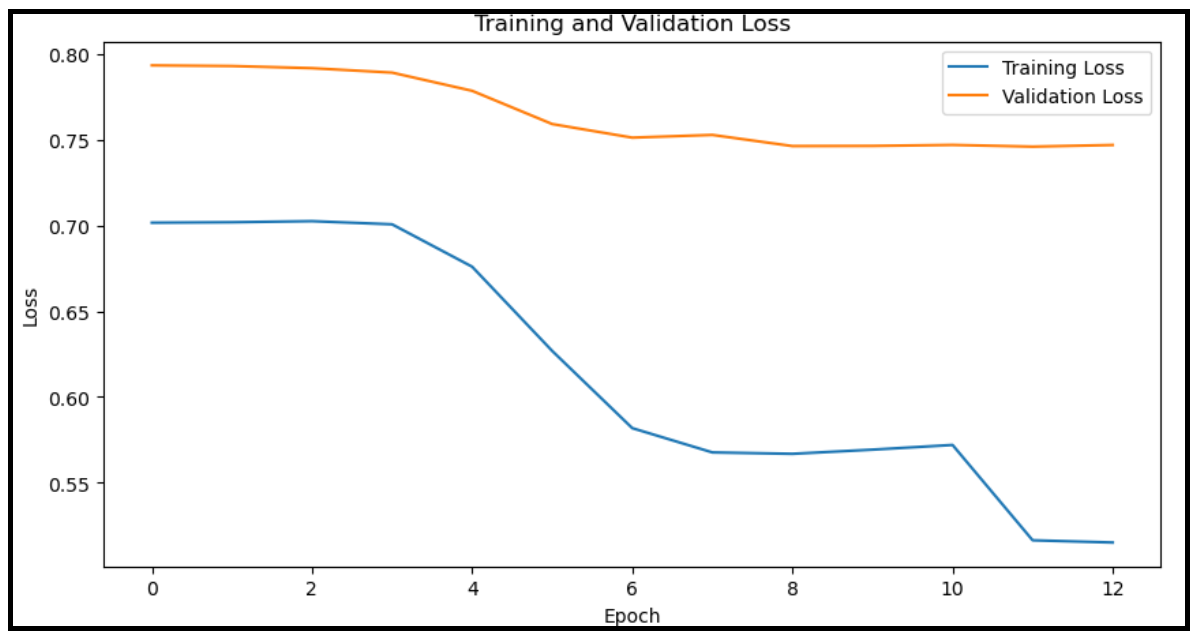
**1.learning rate: 0.1, batch size:32**

**Early stopping:**

Training fold 1: 10

Training fold 2: 12

Training fold 3: 9

****

**Test RMSLE: 0.7490715757267495**

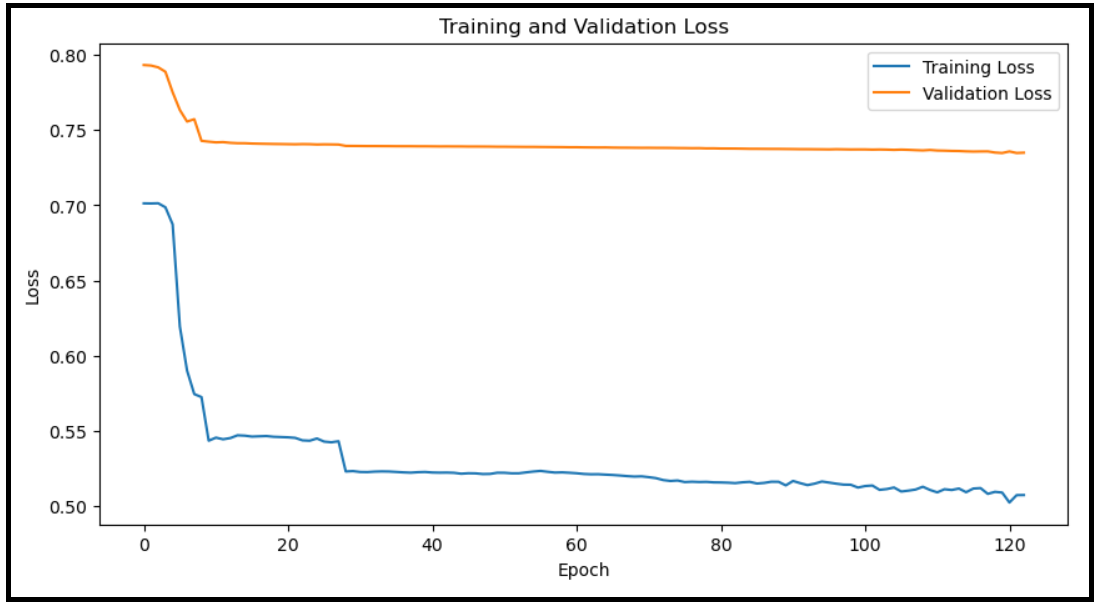
**Hyperparameters:** **layer1 neurons: 32, layer2 neurons: 16, layer3neurons: 8,**

**1.learning rate: 0.1, batch size:32**

Training fold 1: 8

Training fold 2: 122

Training fold 3: 27



**Test RMSLE: 0.7468815480585771**

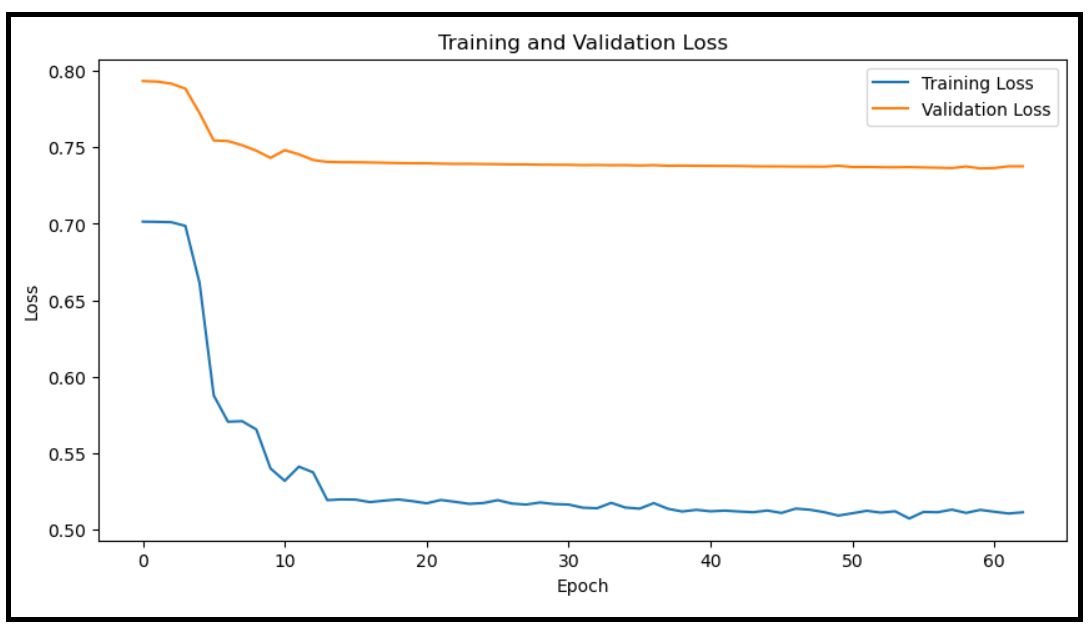
**Hyperparameters:** **layer1 neurons: 128, layer2 neurons: 64, layer3neurons: 32**

**1. learning rate: 0.1, batch size:32**

Training fold 1: 8

Training fold 2: 62

Training fold 3: 12

****

**Test RMSLE: 0.7472769858630088**

**Inference**

* The best RMSLE achieved is 0.746881548 (Configuration 2, 1-fold CV, Layer1: 32, Layer2: 16, Layer3: 8).
* RMSLE values range from 0.746881548 to 0.749071576.

**II. Standard Scaler:**

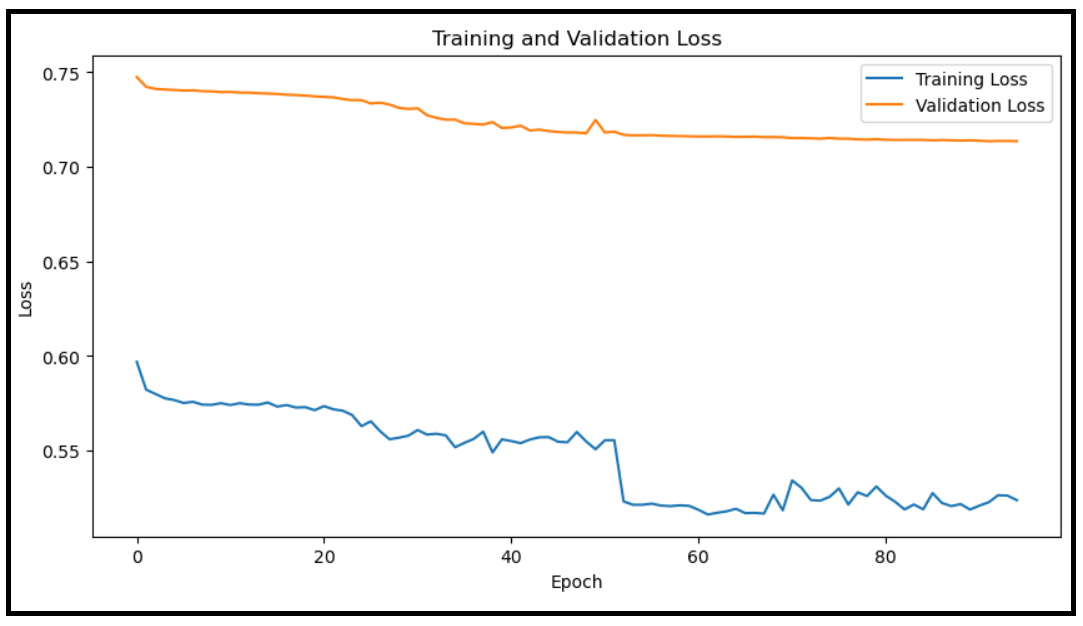
**Hyperparameters:** **layer1 neurons: 64, layer2 neurons: 32, layer3neurons: 16,**

**1. learning rate: 0.1, batch size:32**

Training fold 1: 51

Training fold 2: 69

Training fold 3: 94

****

**Test RMSLE: 0.7233504012133369**

**Hyperparameters:** **layer1 neurons: 32, layer2 neurons: 16, layer3neurons: 8,**

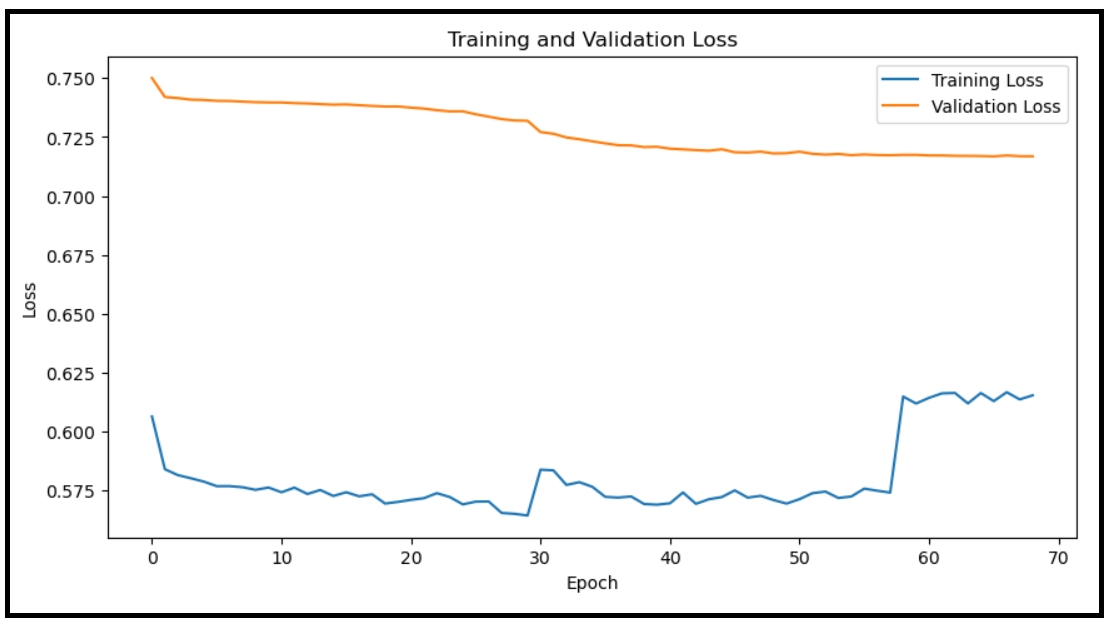
**1. learning rate: 0.1, batch size:32**

Training fold 1: 68

Training fold 2: 29

Training fold 3: 57

**Test RMSLE: 0.7179822970938742**

****

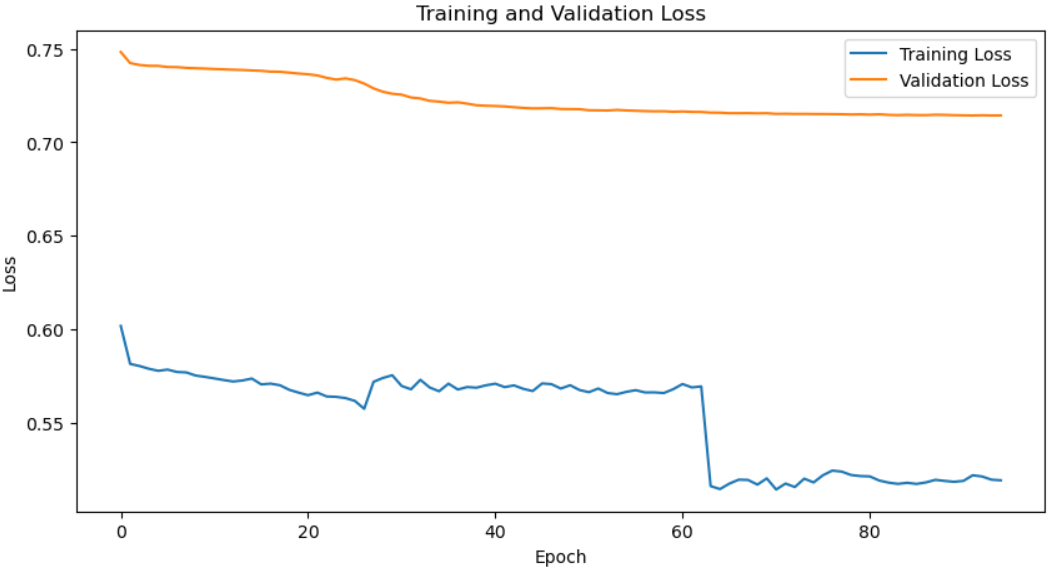
**Hyperparameters:** **layer1 neurons: 128, layer2 neurons: 64, layer3neurons: 32,**

1. **learning rate: 0.1, batch size:32**

Training fold 1: 62

Training fold 2: 94

Training fold 3: 26

****

**Test RMSLE: 0.7177436226296388**

**Inference**

**Standard Scaler:**

* The best RMSLE achieved is 0.717743623 (Configuration 3, 1-fold CV, Layer1: 128, Layer2: 64, Layer3: 32).
* RMSLE values range from 0.717743623 to 0.723350401.

**FEATURE: Date, Time and Month from pickup time and drop-off time.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Sl.No.** | **Normalization** | **Layer1 neurons** | **Layer2 neurons** | **Layer3 neurons** | **3 split CV** | **Learning Rate** | **Batch Size** | **Epoch Stopping** | **RMSLE** | |  | | **1** | **Min-Max Scaler** | **64** | **32** | **16** | **1 fold** | **0.1** | **32** | **10** | **0.749071576** |  | | **2 fold** | **12** |  | | **3 fold** | **19** |  | | **2** | **Min-Max Scaler** | **32** | **16** | **8** | **1 fold** | **0.1** | **32** | **8** | **0.746881548** |  | | **2 fold** | **122** |  | | **3 fold** | **27** |  | | **3** | **Min-Max Scaler** | **128** | **64** | **32** | **1 fold** | **0.1** | **32** | **8** | **0.747276986** |  | | **2 fold** | **62** |  | | **3 fold** | **12** |  | | **4** | **Standard Scaler** | **64** | **32** | **16** | **1 fold** | **0.1** | **32** | **51** | **0.723350401** |  | | **2 fold** | **69** |  | | **3 fold** | **94** |  | | **5** | **Standard Scaler** | **32** | **16** | **8** | **1 fold** | **0.1** | **32** | **68** | **0.717982297** |  | | **2 fold** | **29** |  | | **3 fold** | **57** |  | | **6** | **Standard Scaler** | **128** | **64** | **32** | **1 fold** | **0.1** | **32** | **62** | **0.717743623** |  | | **2 fold** | **94** |  | | **3 fold** | **26** |  | |
|  |
|  |
| **Conclusion**  **Standard Scaler performs better than Min-Max Scaler** based on the RMSLE values. The models trained with the Standard Scaler consistently achieve lower RMSLE values compared to those trained with the Min-Max Scaler. This suggests that standardizing the features to have zero mean and unit variance helps the model learn more effectively, likely due to better handling of varying feature distributions and improved numerical stability during training.   1. **Epoch Stopping:** The number of epochs at which training stopped due to early stopping based on validation loss. The Standard Scaler models typically require more epochs to converge, indicating a more gradual learning process. 2. **Layer Configurations:** Models with different layer configurations were tested, with the Standard Scaler models generally showing better performance across configurations. 3. **Cross-Validation:** Although 3-fold CV is mentioned, the provided results are for different fold setups. The consistency of the RMSLE across folds for the Standard Scaler indicates robust performance. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Conclusion:**