Assignment 3

Introduction

In this assignment, the objective is to build a neural network library that consists of various activation functions such as Sigmoid activation function, hyperbolic tangent function and Rectified linear unit. The assignment is split into two parts. Part 1 solves the XOR problem and Part 2 is working a taxi trip duration dataset with different models.

Part 1:

In this section, we have built the Layer class which acts as the parent class. Next, Linear layer class is implemented which inherits the layer class along with forward and backward functions. Next, we have created a class that implements sigmoid activation function. Rectified linear unit is implemented as well. Hyperbolic tangent function is created for the XOR problem. Binary Cross-entropy loss is implemented to classify the XOR problem. The sequential class is created to add the layers into the model as required. The save and load features are implemented. All these classes inherit the Layer class.

To test the library two neural networks are created based on sigmoid activation and hyperbolic tangent activation. Seed is used for reproducibility of the results. Learning rate is set to 0.1. The best performing weights will be saved as ‘XOR_solved.w’. After training the hyperbolic tangent network has shown the least loss which means that it is easier to train when compared to sigmoid network.

```
Epoch 0, Loss: 0.7149963164751073
Epoch 1000, Loss: 0.6955188132704107
Epoch 2000, Loss: 0.6938010260572636
Epoch 3000, Loss: 0.6927962064942359
Epoch 4000, Loss: 0.6919188405602257
Epoch 5000, Loss: 0.6907334991020461
Epoch 6000, Loss: 0.6885177103759698
Epoch 7000, Loss: 0.6836016820459935
Epoch 8000, Loss: 0.672594886784511
Epoch 9000, Loss: 0.6509168952137163
Epoch 0, Loss: 1.3935857335960962
Epoch 1000, Loss: 0.04501215732815503
Epoch 2000, Loss: 0.012063479878584646
Epoch 3000, Loss: 0.006536130460108653
Epoch 4000, Loss: 0.0044062645418950835
Epoch 5000, Loss: 0.003298435704827585
Epoch 6000, Loss: 0.002624994682560905
Epoch 7000, Loss: 0.002174435677319774
Epoch 8000, Loss: 0.0018527641617485672
Epoch 9000, Loss: 0.0016120743804308574
Minimum Loss with Sigmoid Activation: 0.6127742361951167
Minimum Loss with Hyperbolic Tangent Activation: 0.0014256357153351938
The Hyperbolic Tangent network performed better and was saved as 'XOR_solved.w'.
```

Part 2:

In this section, we will be working on the 'taxi trip duration' dataset. We train three models with different configurations to observe the performances of the model using RMSLE. The neural network library that was built before will be here.

Dataset:

The dataset was acquired from the canvas. Originally, this dataset was from a Kaggle challenge. This dataset contains various features such as pickup and dropoff date times, latitudes and longitudes, passenger count etc. The dataset on the canvas consists of an extra feature trip duration. It can be used by running the following code:

Load the dataset

```python
#Referenced from the assignment
dataset = np.load("nyc_taxi_data.npy", allow_pickle=True).item()
X_train, y_train, X_test, y_test = dataset["X_train"], dataset["y_train"], dataset["X_test"], dataset["y_test"]
```

Python

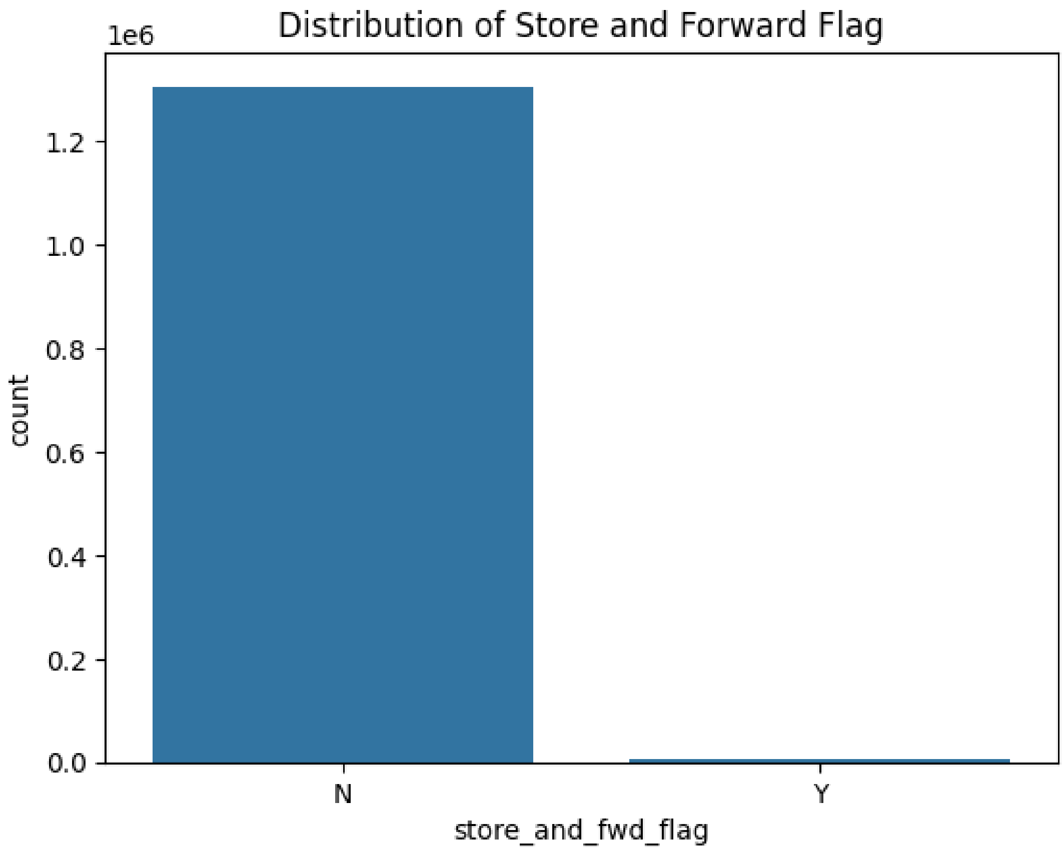
Working:

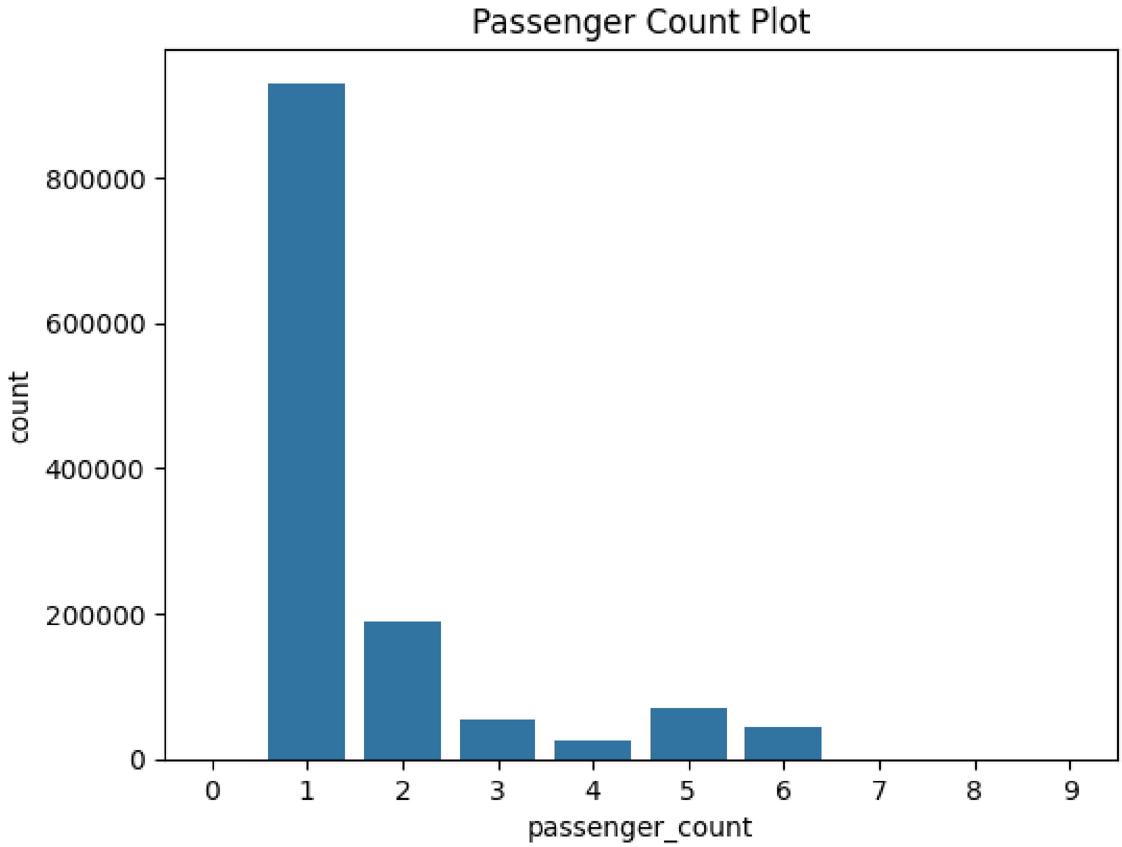
Import all the libraries such as numpy, pandas, etc., Next load the dataset using the code snippet above. The first step is to analyze the dataset even though it is filtered. It is a good practice to analze the data before we work with it.

Data Analysis:

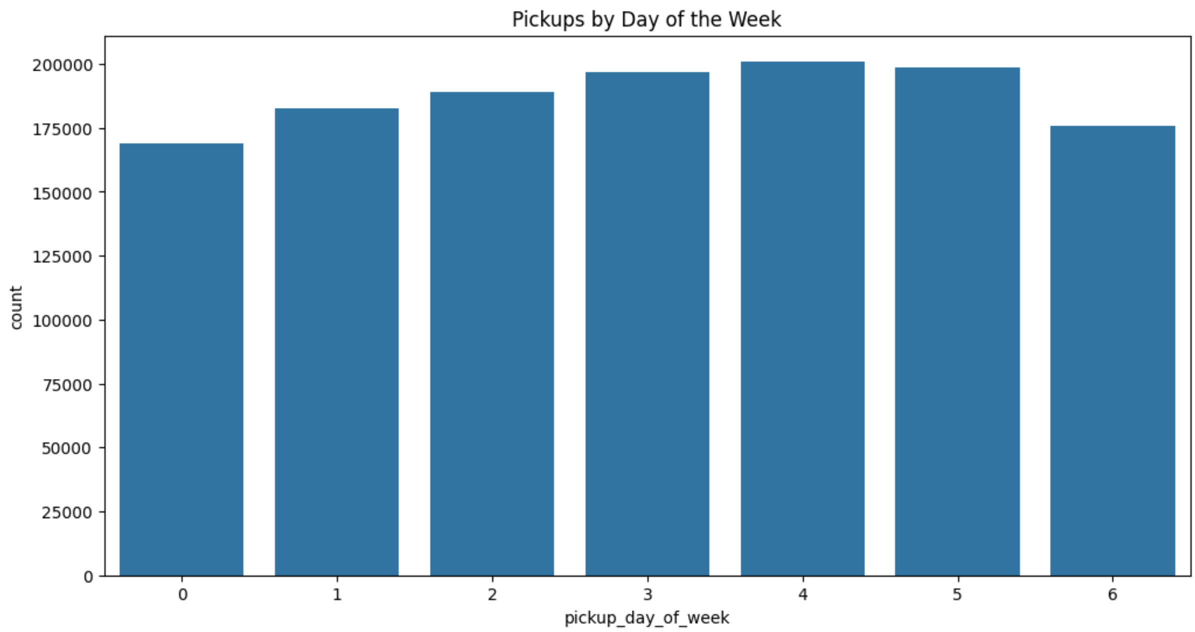
Refer to the DatasetAnalysis.ipynb file for detailed analysis. In this section, the data was loaded and analyzed to understand the data. Use .info() method to get the datatypes of the features. The pickup and dropoff times need to be changed to get hour, day and month. .describe() gives the summary statistics of the dataset. Next we check for null values. There were no null values. We have plotted various features as shown below:

The hour, day and day are extracted from pickup and dropoff datetime features. These features help to understand the influence of these features in predicting demand. These plots display the spread of data and variation.

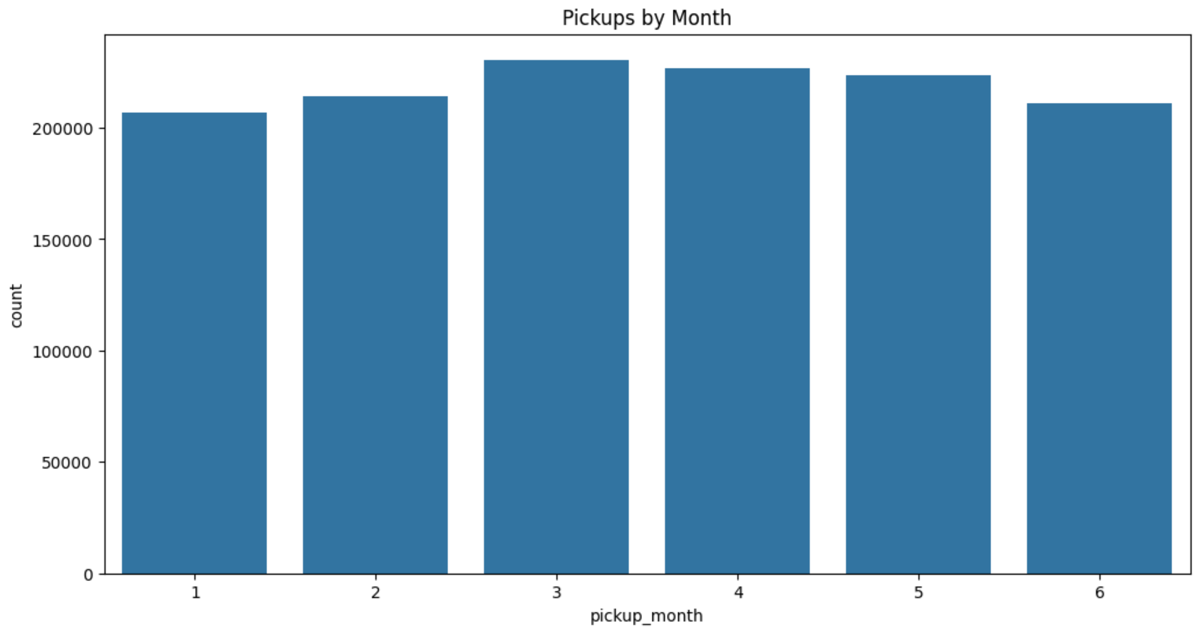




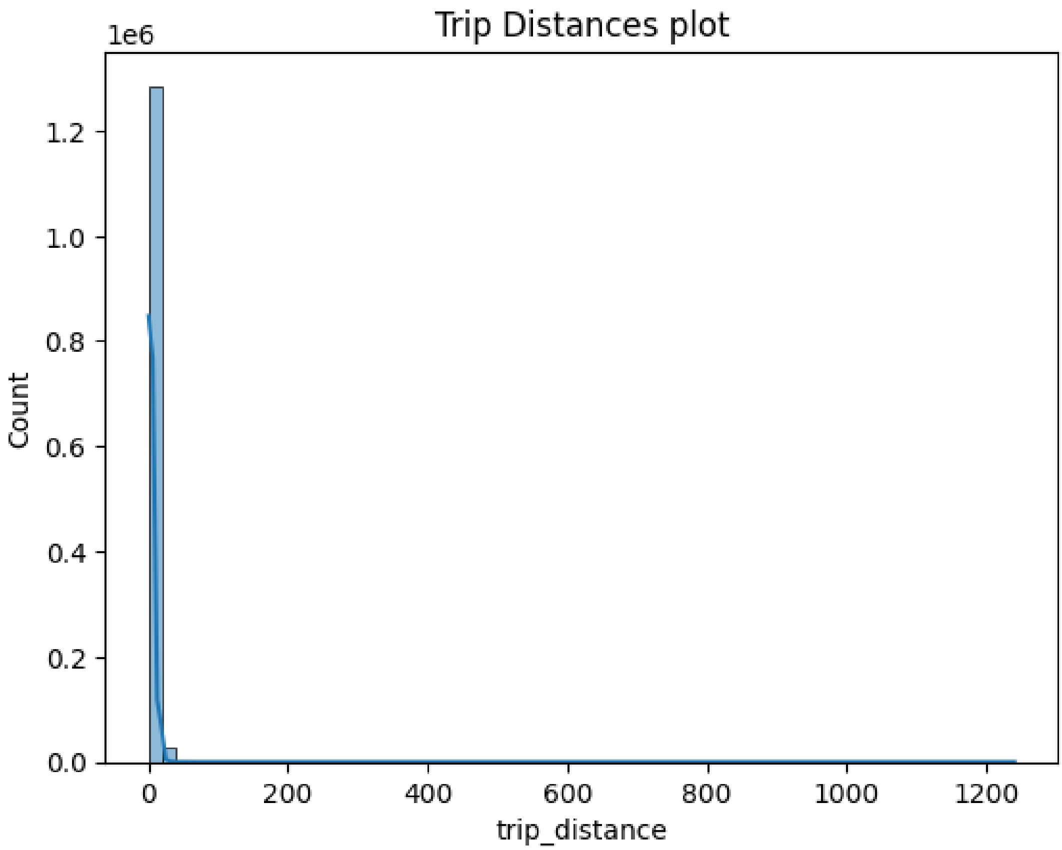
Passenger Count Plot
count
800000
600000
400000
200000
0
0 1 2 3 4 5 6 7 8 9
passenger_count



Pickups by Day of the Week
count
200000
175000
150000
125000
100000
75000
50000
25000
0
0 1 2 3 4 5 6
pickup_day_of_week



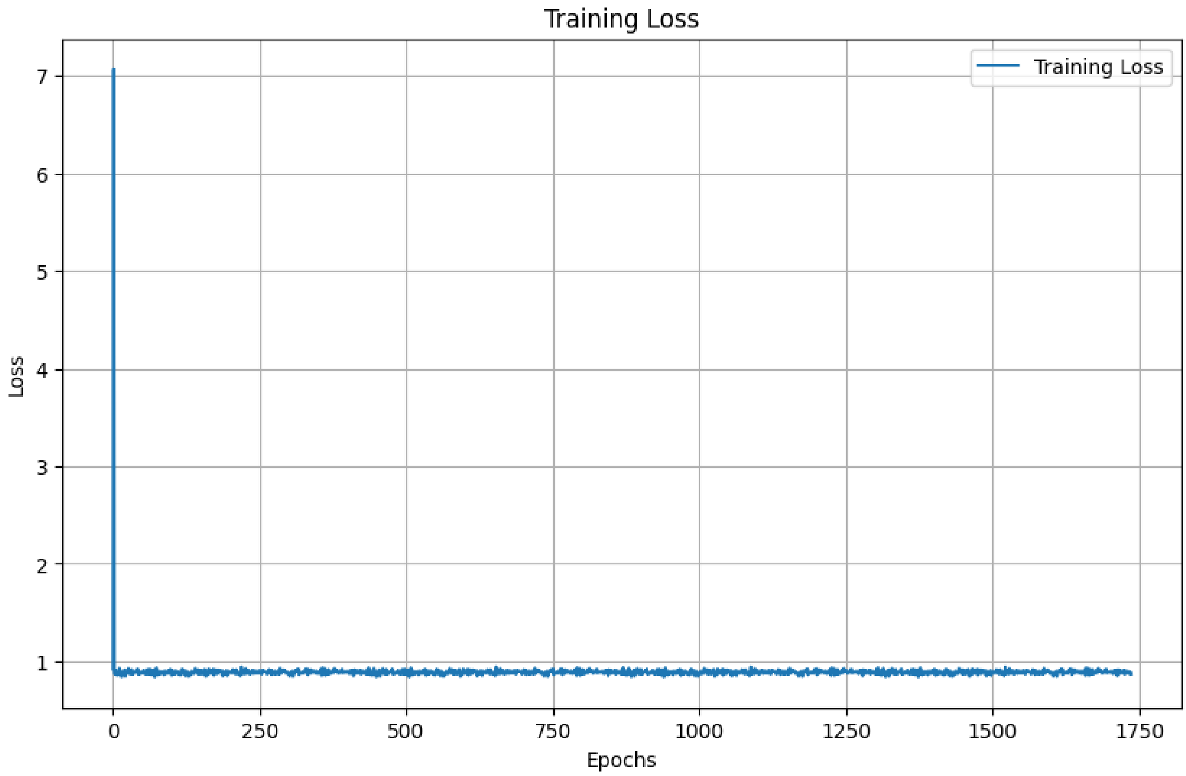
We can calculate trip distance from latitude and longitude features by using the haversine library. This feature helps to know the underlying patterns on how much is distance covered and helps to predict demand.

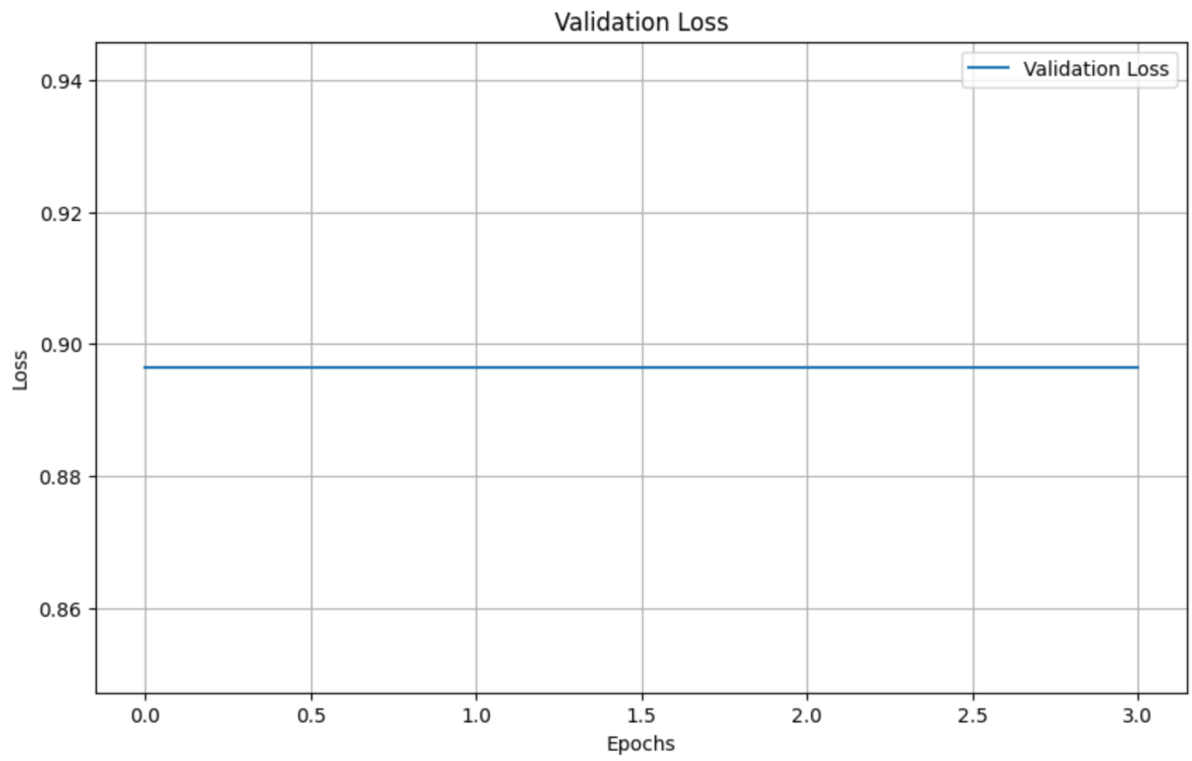


Back to the main code, the functions for data preprocessing is created which includes the hour, day and month extraction and trip distance calculations. Next the methods for training, testing and normalisation are implemented. The models are created with different configurations as shown below:

Model 1:

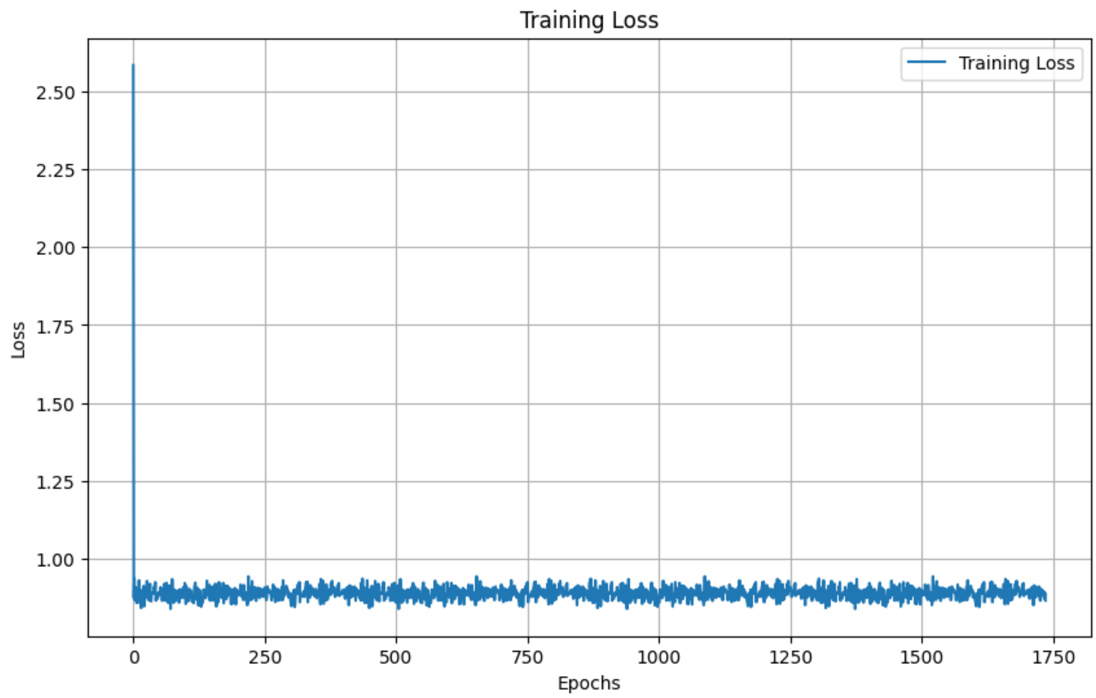
In this model, we have used these features: 'passenger_count', 'pickup_hour', 'dropoff_hour', 'pickup_day_of_week', 'pickup_month', 'trip_distance'. We have used three layers with Relu activations. The learning rate is set to 0.01, early stopping is 3. The data is scaled and normalised before training. The plots are shown below. The early stopping is triggered at epoch 4. The model gave a score of 0.893 on the test set.

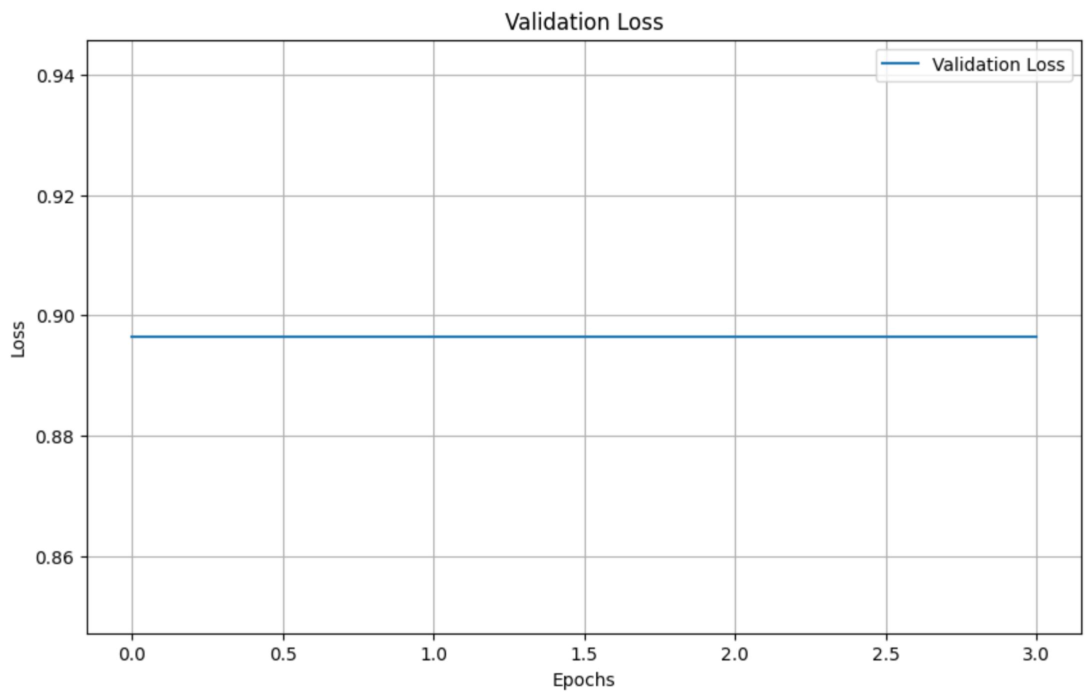




Model 2:

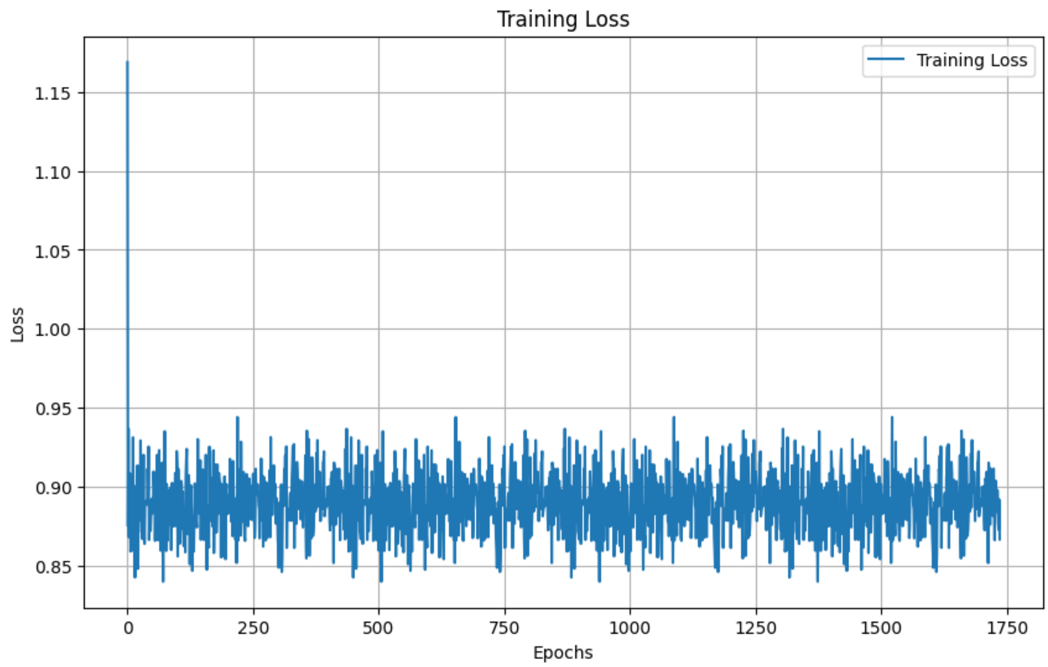
In this model, we have used these features: 'passenger_count', 'pickup_hour', 'pickup_day_of_week', 'pickup_month', 'trip_distance', 'dropoff_hour', 'dropoff_month'. We have used two layers with sigmoid and Relu activations. The learning rate is set to 0.01, early stopping is 3. The data is scaled and normalised before training. The plots are shown below. The early stopping is triggered at epoch 4. The model gave a score of 0.8933 RMSLE on the test set.

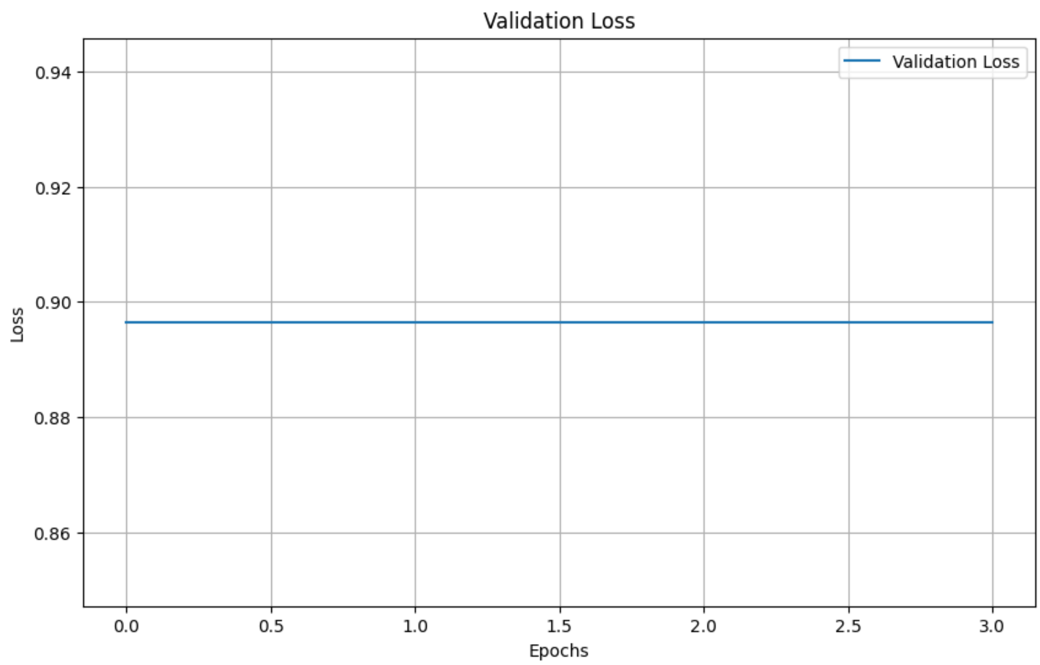




Model 3:

In this model, we have used these features: 'passenger_count', 'pickup_hour', 'dropoff_hour', 'pickup_day_of_week' ,'pickup_month', 'dropoff_month', 'dropoff_day_of_week', 'trip_distance'. We have used two layers with Relu activations. The learning rate is set to 0.01, early stopping is 3. The data is scaled and normalised before training. The plots are shown below. The early stopping is triggered at epoch 4. The model gave a score of 0.893307 RMSLE on the test set.





Conclusion:

In this assignment, we have performed various tasks such as building a neural network from scratch. Solving the XOR problem. From part 1 result we can say that the Hyperbolic Tangent activation network (loss: 0.001425) is easier to train than the Sigmoid. In second part of the assignment, we have trained models on the taxi trip dataset. The dataset was analysed thoroughly and pre-processed for the model training. We have created three models with different configurations. The models have given a good performance on test data. The plots for training and validation loss are plotted.