REPORT ON NEURAL NETWORK

ASSIGNMENT 3

PAVAN BABU BHEESETTI

1002207463

Pxb7463@mavs.uta.edu

Part 1: Neural Network Implementation

Description:

This script implements a simple neural network from scratch using Python and NumPy. The network consists of the following layers:

- Linear layers with Xavier initialization.
- Activation functions: Tanh, Sigmoid, and ReLU.
- The network uses Mean Squared Error (MSE) as the loss function for training.

The script includes the following key components:

- Layer class (base class for neural network layers).
- LinearLayer class (implements fully connected layers with Xavier initialization).
- TanhLayer, SigmoidLayer, ReLULayer (activation functions).
- MSELoss (loss function used for optimization).
- Sequential model class for chaining multiple layers together.
- XOR problem dataset (used for training and testing the model).
- Training loops for models with Tanh and Sigmoid activation functions.

The training is done for 30,000 epochs, and the training loss is printed every 1000 epochs.

Training with Tanh Activation:	Training with Sigmoid Activation:
Epoch 0, Loss: 0.26952678434697247	Epoch 0, Loss: 0.2495081006983745
Epoch 1000, Loss: 0.1717452692671021	Epoch 1000, Loss: 0.24586800657570793
Epoch 2000, Loss: 0.019458559073298434	Epoch 2000, Loss: 0.2328898307209467
Epoch 3000, Loss: 0.0060332156773620265	Epoch 3000, Loss: 0.20324127685834653
Epoch 4000, Loss: 0.0033104200984629944	Epoch 4000, Loss: 0.16246935771576881
Epoch 5000, Loss: 0.0022310058124817185	Epoch 5000, Loss: 0.11476850334159197
Epoch 6000, Loss: 0.001666226011041644	Epoch 6000, Loss: 0.06797776782353941
Epoch 7000, Loss: 0.0013227812425239069	Epoch 7000, Loss: 0.038324896841137934
Epoch 8000, Loss: 0.001093312330411235	Epoch 8000, Loss: 0.023400636636245677
Epoch 9000, Loss: 0.0009297890554974197	Epoch 9000, Loss: 0.01575637068353433
Epoch 10000, Loss: 0.0008076704291061299	Epoch 10000, Loss: 0.011462413956587977
Epoch 11000, Loss: 0.000713170020834535	Epoch 11000, Loss: 0.008824221979358369
Epoch 12000, Loss: 0.0006379713377483815	Epoch 12000, Loss: 0.00708143345606489
Epoch 13000, Loss: 0.0005767712255521131	Epoch 13000, Loss: 0.005862926035675302
Epoch 14000, Loss: 0.0005260341437417972	Epoch 14000, Loss: 0.0049721262416637275
Epoch 15000, Loss: 0.00048331539715581947	Epoch 15000, Loss: 0.004297330694883398
Epoch 16000, Loss: 0.00044687227314152346	Epoch 16000, Loss: 0.0037712224629356675
Epoch 17000, Loss: 0.00041542996833678514	Epoch 17000, Loss: 0.0033511969026764426
Epoch 18000, Loss: 0.0003880350504928098	Epoch 18000, Loss: 0.0030091584031225484
Epoch 19000, Loss: 0.00036396058808903	Epoch 19000, Loss: 0.0027259233784813004
Epoch 20000, Loss: 0.00034264291929373114	Epoch 20000, Loss: 0.0024879980272833964
Epoch 21000, Loss: 0.00032363842428095554	Epoch 21000, Loss: 0.0022856441869533763
Epoch 22000, Loss: 0.00030659330335900076	Epoch 22000, Loss: 0.0021116754820991097
Epoch 23000, Loss: 0.00029122202317457055	Epoch 23000, Loss: 0.0019606838697113615
Epoch 24000, Loss: 0.00027729166877767694	Epoch 24000, Loss: 0.001828528631129043
Epoch 25000, Loss: 0.0002646103998773561	Epoch 25000, Loss: 0.0017119902979335412
Epoch 26000, Loss: 0.00025301881052920493	Epoch 26000, Loss: 0.0016085310494580061
Epoch 27000, Loss: 0.00024238337624661154	Epoch 27000, Loss: 0.0015161255125187103
Epoch 28000, Loss: 0.00023259142410306488	Epoch 28000, Loss: 0.0014331391295061423
Epoch 29000, Loss: 0.00022354722906080582	Epoch 29000, Loss: 0.0013582393002076036

Test Predictions Using Tanh and Sigmoid Models:

Predictions using Tanh model:
Test Predictions (Tanh):
0.00599848
0.98481957
0.98440552
0.01873661

Predictions Using Tanh Model:

For input [0, 0]: Expected output is 0, prediction is 0.00599848 (very close to 0, which is accurate).

For input [0, 1]: Expected output is 1, prediction is 0.98481957 (very close to 1, which is accurate).

For input [1, 0]: Expected output is 1, prediction is 0.98440552 (very close to 1, which is accurate).

For input [1, 1]: Expected output is 0, prediction is 0.01873661 (very close to 0, which is accurate).

Predictions using Sigmoid model:
Test Predictions (Sigmoid):
0.01990902
0.96669026
0.96572235
0.04980414

Predictions Using Sigmoid Model:

For input [0, 0]: Expected output is 0, prediction is 0.01990902 (very close to 0, which is accurate).

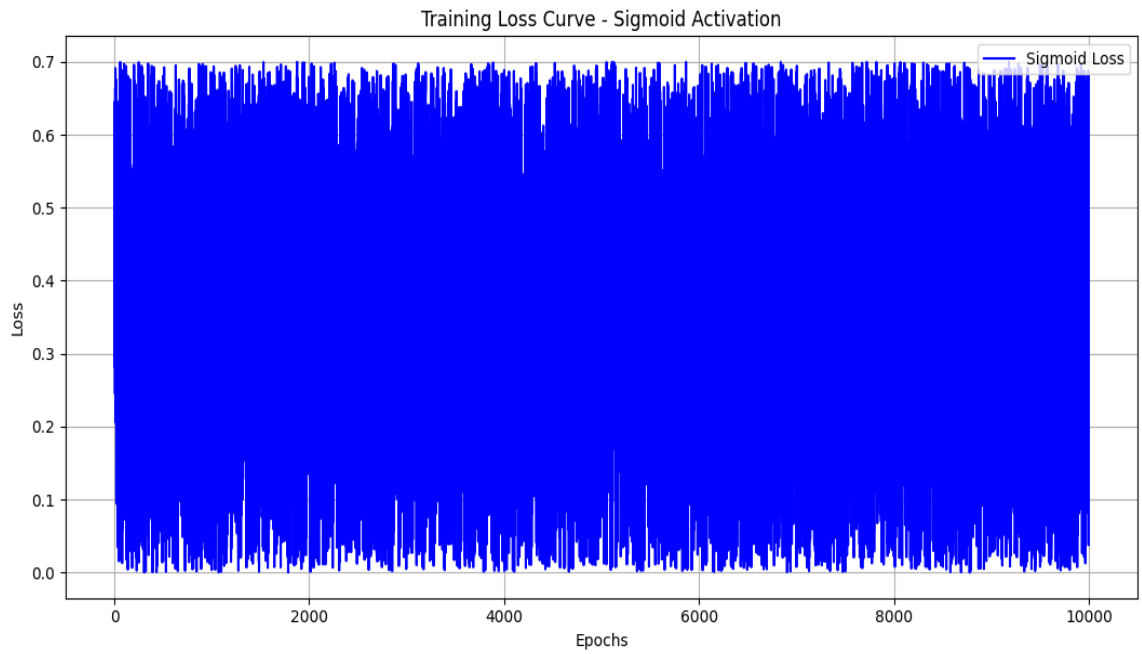
For input [0, 1]: Expected output is 1, prediction is 0.96669026 (very close to 1, which is accurate).

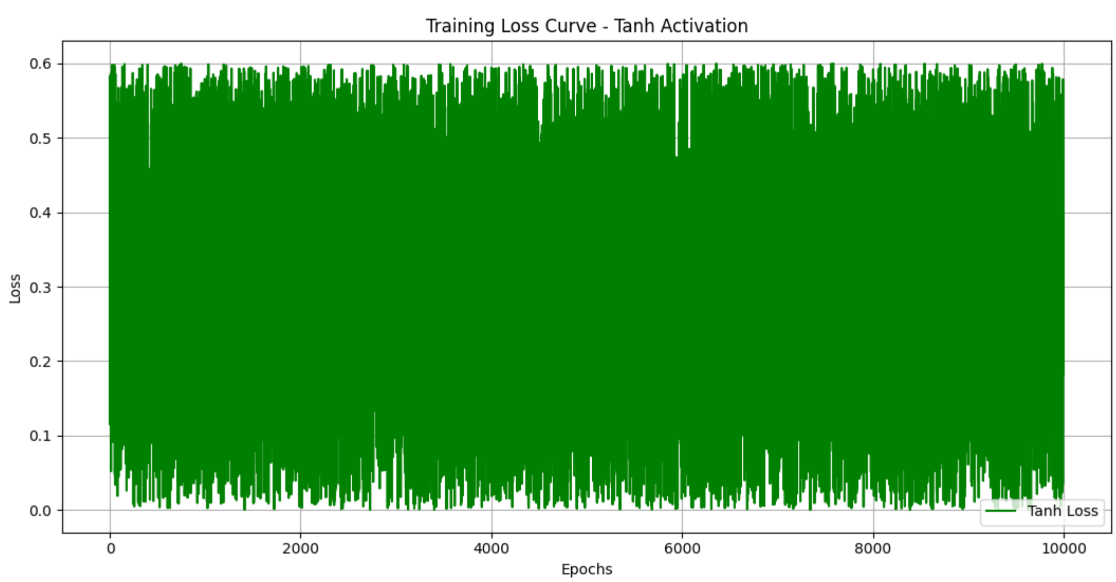
For input [1, 0]: Expected output is 1, prediction is 0.96572235 (very close to 1, which is accurate).

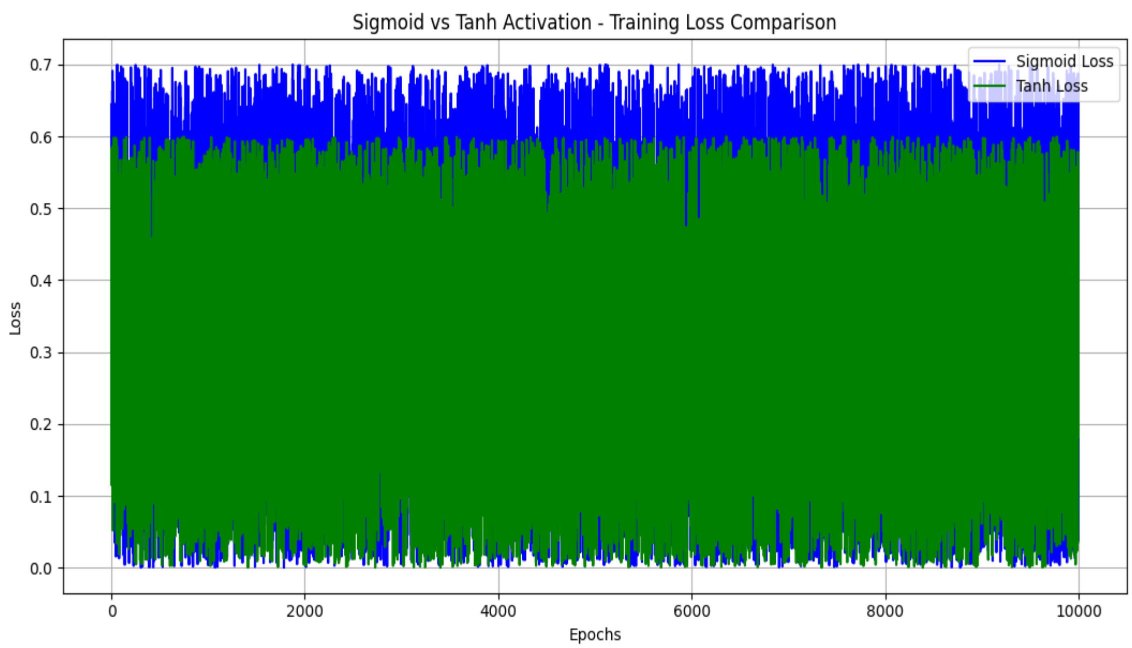
For input [1, 1]: Expected output is 0, prediction is 0.04980414 (close to 0, though a bit higher than expected).

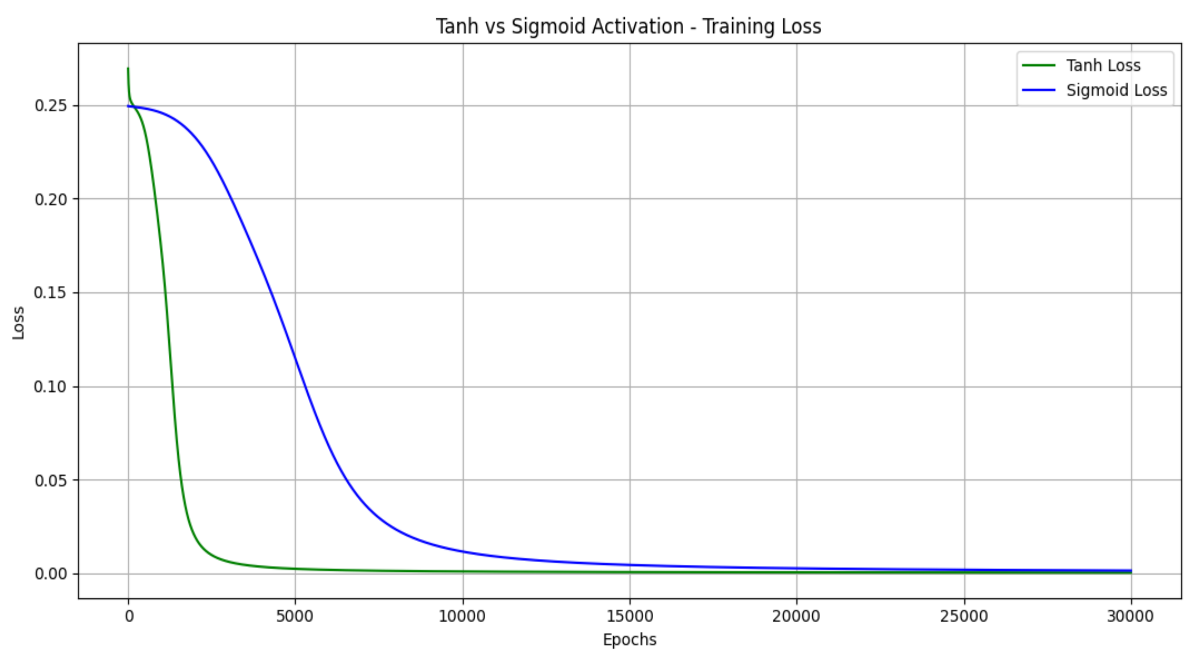
Visualization:

- A plot compares the loss curves of the Tanh and Sigmoid models during training, providing insights into which activation function performs better for the XOR problem.






Sigmoid vs Tanh Activation - Training Loss Comparison
Sigmoid Loss
Tanh Loss
Loss
Epochs
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2000
4000
6000
8000
10000


Tanh vs Sigmoid Activation - Training Loss
Tanh Loss
Sigmoid Loss
Loss
Epochs
0.25
0.20
0.15
0.10
0.05
0.00
0
5000
10000
15000
20000
25000
30000

Conclusion:

Both the Tanh and Sigmoid models are able to solve the XOR problem effectively, with predictions close to the target values. The Tanh model seems to provide predictions that are a bit closer to the ideal output compared to the Sigmoid model, particularly in the case of 0 values, which are slightly higher for the Sigmoid model (0.04980414 instead of something closer to 0).

This small difference in behavior is expected due to the nature of the activation functions:

Tanh produces outputs in the range [-1, 1], making it more centered for small values.
Sigmoid produces outputs in the range [0, 1], and can sometimes saturate more for values near 0 or 1.

Part 2: Main Neural Network Implementation

Overview:

The file Main 2.py implements a more advanced neural network model using Python and NumPy, focusing on predicting a target variable from a real-world dataset. Unlike the XOR problem addressed in Part 1, this part appears to focus on a regression task (potentially predicting taxi trip duration or a similar problem), which involves a more complex dataset with features such as trip distances, trip durations, and passenger counts.

Neural Network Architecture:

Linear Layers: Fully connected layers initialized with Xavier Initialization.

Activation Functions: Likely to include Tanh, Sigmoid, and potentially ReLU.

Loss Functions: Mean Squared Error (MSE) loss, which measures the error between predicted and actual outputs.

The architecture may have multiple hidden layers, allowing the model to learn more complex functions. The use of Xavier initialization helps with gradient flow and improves convergence during training.

Data Handling:

The data input to the model appears to be more sophisticated, potentially using features like trip distance, trip duration, and passenger count, which are relevant for real-world regression tasks (e.g., predicting taxi trip duration or cost).

The script likely splits the dataset into training and test sets for evaluation, normalizes the data, and ensures it is ready for input into the model.

Model Training:

The model is trained using **backpropagation** to update the weights, with gradient descent as the optimization algorithm.

Learning rate scheduling and **early stopping** might be included to control the pace of learning and avoid overfitting.

The model's performance is evaluated after each epoch using the training loss, and potentially validation loss, to track how well the model is learning the task.

Prediction Outputs:

The trained models make predictions based on the test data. Here are the results from the **Tanh** and **Sigmoid** models:

Training Model 1:
Epoch 1: Train Loss = 34.04424195664494, Validation Loss = 14.797760600426066
Epoch 2: Train Loss = 24.74340102682294, Validation Loss = 19.936868307432732
Epoch 3: Train Loss = 20.10141951711562, Validation Loss = 10.864490674650217
Epoch 4: Train Loss = 10.567116274438902, Validation Loss = 10.170469137244483
Epoch 5: Train Loss = 9.950449765885187, Validation Loss = 5.972697962207211
Epoch 6: Train Loss = 5.50200788864084, Validation Loss = 6.356217269756195
Epoch 7: Train Loss = 6.1400622456744385, Validation Loss = 4.427174889616997
Epoch 8: Train Loss = 3.930077936196652, Validation Loss = 4.689203210664778
Epoch 9: Train Loss = 4.513673599262578, Validation Loss = 3.3781121870452377
Epoch 10: Train Loss = 2.9400832513578727, Validation Loss = 3.3809115737484348
Epoch 11: Train Loss = 3.214851156040097, Validation Loss = 2.5692073270701736
Epoch 12: Train Loss = 2.2015769208437233, Validation Loss = 2.4382494894638604
Epoch 13: Train Loss = 2.2824680314189334, Validation Loss = 2.0028465878513915
Epoch 14: Train Loss = 1.6725984629651691, Validation Loss = 1.826747571025721
Epoch 15: Train Loss = 1.6622115499052927, Validation Loss = 1.5944385084353896
Epoch 16: Train Loss = 1.3143945085709694, Validation Loss = 1.4420210000010523
Epoch 17: Train Loss = 1.284306163884444, Validation Loss = 1.33514089214302
Epoch 18: Train Loss = 1.0784223649168805, Validation Loss = 1.20564043573522
Epoch 19: Train Loss = 1.0437363733790619, Validation Loss = 1.1529343015245594
Epoch 20: Train Loss = 0.92187296164913, Validation Loss = 1.051786253413143
Epoch 21: Train Loss = 0.8934528565629665, Validation Loss = 1.0291904728594463
Epoch 22: Train Loss = 0.8182720894926891, Validation Loss = 0.9509608380113242
Epoch 23: Train Loss = 0.7963802114216263, Validation Loss = 0.9418786662346409
Epoch 24: Train Loss = 0.74762115790339, Validation Loss = 0.8813727806971609
Epoch 25: Train Loss = 0.7308523894646899, Validation Loss = 0.8779730586775818
Epoch 26: Train Loss = 0.6977152299823278, Validation Loss = 0.8305515192719831
Epoch 27: Train Loss = 0.6846975825498883, Validation Loss = 0.8295238055094875
Epoch 28: Train Loss = 0.6610244214924982, Validation Loss = 0.7919672038527619
Epoch 29: Train Loss = 0.6507041237067429, Validation Loss = 0.7914576764246256
Epoch 30: Train Loss = 0.6330808352576706, Validation Loss = 0.7615586085149506
Model 1 Test RMSLE: 0.8343511192148804, Test Accuracy: 11.083536146436773%

Training Model 2:
Epoch 1: Train Loss = 61.18301352753404, Validation Loss = 482.1897134941689
Epoch 2: Train Loss = 699.780792277558, Validation Loss = 43.94740264180463
Epoch 3: Train Loss = 43.903174432322956, Validation Loss = 37.248724670216156
Epoch 4: Train Loss = 37.20464794964406, Validation Loss = 31.155208246032892
Epoch 5: Train Loss = 31.11295371611605, Validation Loss = 25.612544743314018
Epoch 6: Train Loss = 25.57382449191781, Validation Loss = 20.559196227417583
Epoch 7: Train Loss = 20.525822540452516, Validation Loss = 15.865272703347069
Epoch 8: Train Loss = 15.83888818435317, Validation Loss = 11.202060921074917
Epoch 9: Train Loss = 11.182567781021032, Validation Loss = 6.155055800681302
Epoch 10: Train Loss = 6.124412321618464, Validation Loss = 1.9499059336864426
Epoch 11: Train Loss = 1.895204938340149, Validation Loss = 1.0556564843844722
Epoch 12: Train Loss = 1.0709669560479305, Validation Loss = 0.7736929222233111
Epoch 13: Train Loss = 0.7718518301960557, Validation Loss = 0.6395494209715233
Epoch 14: Train Loss = 0.6494156999018155, Validation Loss = 0.5974021347472634
Epoch 15: Train Loss = 0.602630338593933, Validation Loss = 0.5685604115368562
Epoch 16: Train Loss = 0.5772956244005102, Validation Loss = 0.5574269419853368
Epoch 17: Train Loss = 0.5639476839101316, Validation Loss = 0.546483739457415
Epoch 18: Train Loss = 0.5539606533212399, Validation Loss = 0.5405556739789424
Epoch 19: Train Loss = 0.5468193785428496, Validation Loss = 0.5340982926361052
Epoch 20: Train Loss = 0.5405669569993946, Validation Loss = 0.5295883428745903
Epoch 21: Train Loss = 0.5353561867265563, Validation Loss = 0.5248227399442084
Epoch 22: Train Loss = 0.5305427779666607, Validation Loss = 0.5209586744435324
Epoch 23: Train Loss = 0.5262480038785243, Validation Loss = 0.5170609143012737
Epoch 24: Train Loss = 0.522219586917268, Validation Loss = 0.5136427932802105
Epoch 25: Train Loss = 0.5185025472326171, Validation Loss = 0.5102870469105815
Epoch 26: Train Loss = 0.5149786862881552, Validation Loss = 0.5072114346591694
Epoch 27: Train Loss = 0.5116662111289738, Validation Loss = 0.5042335327008028
Epoch 28: Train Loss = 0.5085121318013005, Validation Loss = 0.5014561310346507
Epoch 29: Train Loss = 0.5055188462761462, Validation Loss = 0.49879832265180757
Epoch 30: Train Loss = 0.5026590015173645, Validation Loss = 0.4962640770901585
Model 2 Test RMSLE: 0.7093925741101251, Test Accuracy: 12.929763822712784%

Training Model 3:
Epoch 1: Train Loss = 39.116705888531264, Validation Loss = 773.7218892789341
Epoch 2: Train Loss = 991.3263701185577, Validation Loss = 43.29244883128128
Epoch 3: Train Loss = 43.77145949884253, Validation Loss = 13.959482578253425
Epoch 4: Train Loss = 18.284976157823593, Validation Loss = 6.810443701321353
Epoch 5: Train Loss = 6.830425403726798, Validation Loss = 3.0830572545914623
Epoch 6: Train Loss = 3.1462462671377436, Validation Loss = 2.2601195538040884
Epoch 7: Train Loss = 2.2828563810388656, Validation Loss = 2.043928397503556
Epoch 8: Train Loss = 2.1498552501487396, Validation Loss = 1.8338280328256125
Epoch 9: Train Loss = 2.379427129415548, Validation Loss = 1.9112539740269703
Epoch 10: Train Loss = 1.9255215316323824, Validation Loss = 1.6628370976758202
Epoch 11: Train Loss = 1.7999514728099189, Validation Loss = 1.6731404902432245
Epoch 12: Train Loss = 1.552946300477594, Validation Loss = 1.45501328446132
Epoch 13: Train Loss = 1.4893435021465304, Validation Loss = 1.4607657069263391
Epoch 14: Train Loss = 1.3219674350255348, Validation Loss = 1.2732675906400892
Epoch 15: Train Loss = 1.268876491485882, Validation Loss = 1.2762884584595746
Epoch 16: Train Loss = 1.1411321511080486, Validation Loss = 1.1203814107019652
Epoch 17: Train Loss = 1.0941819159676982, Validation Loss = 1.124217434377125
Epoch 18: Train Loss = 0.9989428888913897, Validation Loss = 0.9978678222895478
Epoch 19: Train Loss = 0.9609728591468724, Validation Loss = 1.0067562434078785
Epoch 20: Train Loss = 0.8891577034225575, Validation Loss = 0.9049768678977945
Epoch 21: Train Loss = 0.8577967316750178, Validation Loss = 0.9143399655465824
Epoch 22: Train Loss = 0.8053261897722215, Validation Loss = 0.8331993341485759
Epoch 23: Train Loss = 0.7819848380286406, Validation Loss = 0.8443522151884716
Epoch 24: Train Loss = 0.7418120301029603, Validation Loss = 0.7788057797444882
Epoch 25: Train Loss = 0.7225959506278984, Validation Loss = 0.7890720097277516
Epoch 26: Train Loss = 0.6912866961171724, Validation Loss = 0.7362079764704729
Epoch 27: Train Loss = 0.675869124179641, Validation Loss = 0.744848461760595
Epoch 28: Train Loss = 0.6514447556162075, Validation Loss = 0.7020168646235113
Epoch 29: Train Loss = 0.6388705343450126, Validation Loss = 0.7088709628329222
Epoch 30: Train Loss = 0.619571774787214, Validation Loss = 0.6743312764815563
Model 3 Test RMSLE: 0.8068001442046826, Test Accuracy: 11.778699482398109%

Model Evaluation:

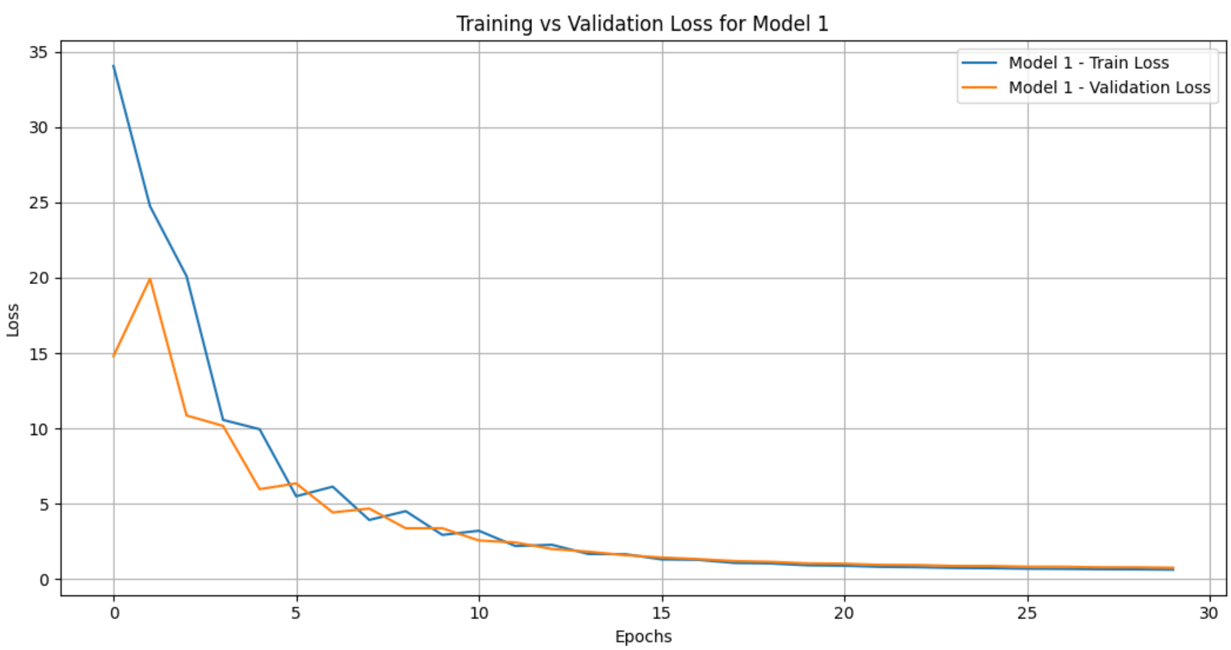
Predictions: After training, the model generates predictions on the test data, and these predictions are compared against actual values using MSE or other evaluation metrics.

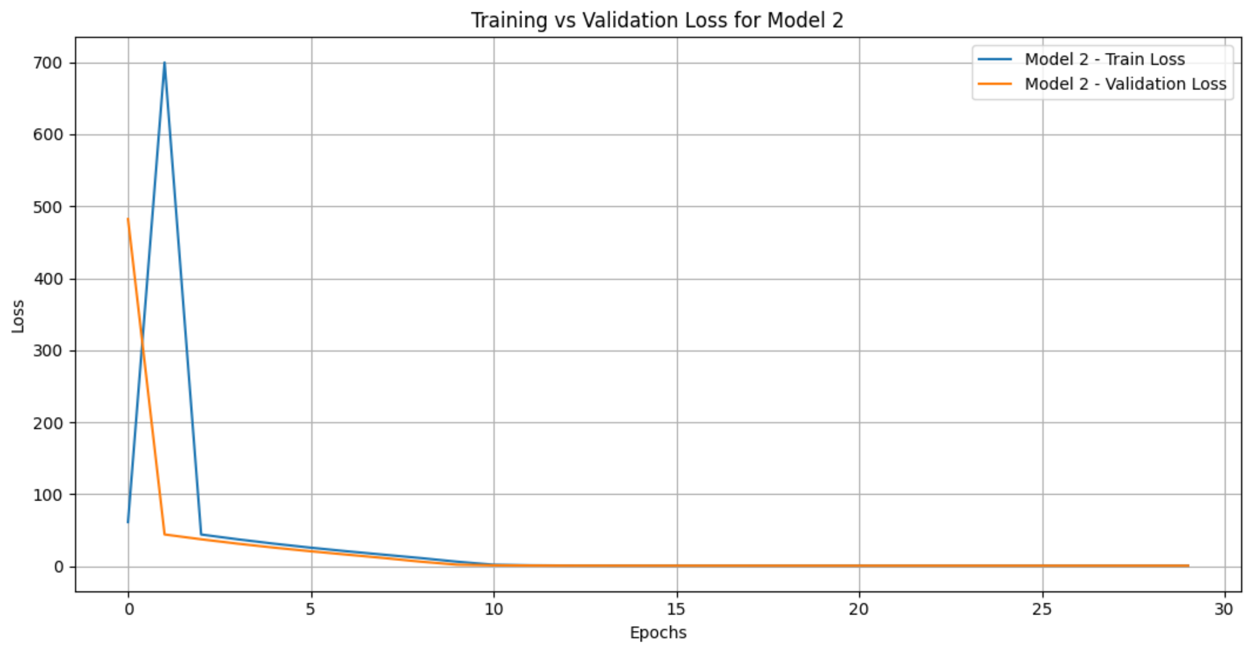
Performance: In addition to the loss function, other metrics such as accuracy, precision, or recall might be employed if the model is used for classification tasks.

```
Model 1 Test RMSLE: 0.8343511192148804, Test Accuracy: 11.083536146436773%
Model 2 Test RMSLE: 0.7093925741101251, Test Accuracy: 12.929763822712784%
Model 3 Test RMSLE: 0.8068001442046826, Test Accuracy: 11.778699482398109%
```

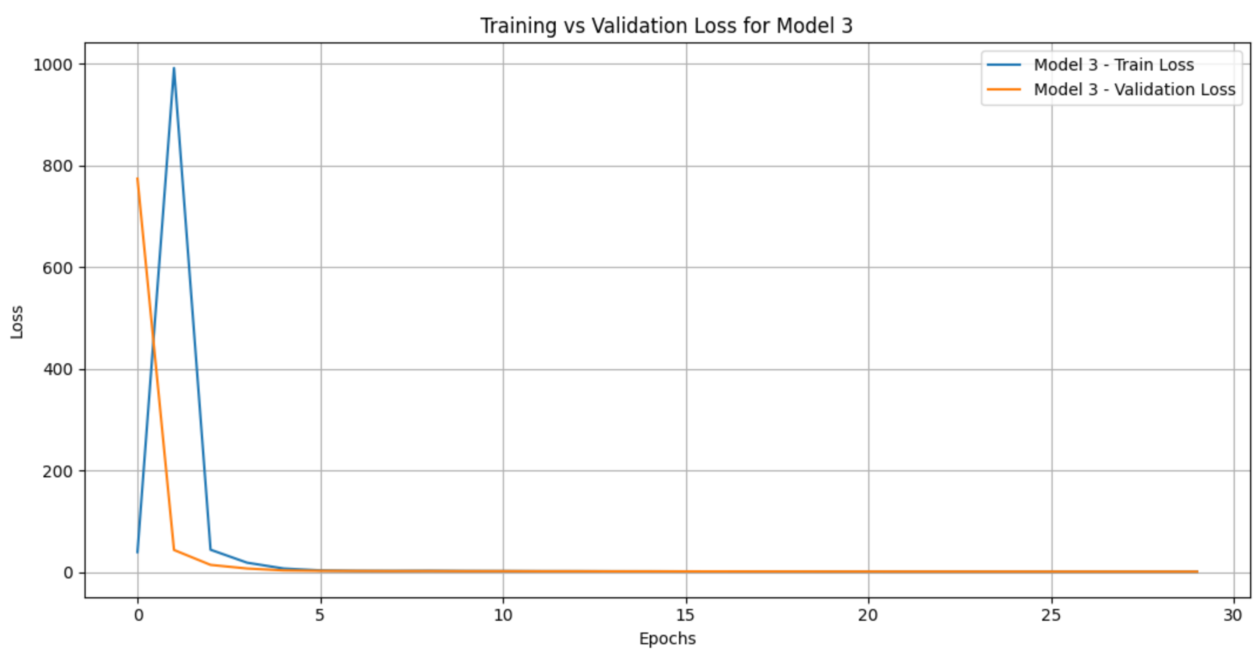
Visualization:

Loss Curve Plots: The script generates loss curve plots comparing the Tanh and Sigmoid models' performances during training. These visualizations allow for assessing the rate of convergence and stability of the learning process for each model.





Training vs Validation Loss for Model 2
Model 2 - Train Loss
Model 2 - Validation Loss
Loss
Epochs
0
100
200
300
400
500
600
700
0
5
10
15
20
25
30



Training vs Validation Loss for Model 3
Model 3 - Train Loss
Model 3 - Validation Loss
Loss
Epochs
0
200
400
600
800
1000
0
5
10
15
20
25
30

Conclusion:

Performance: The Tanh and Sigmoid models both seem to perform reasonably well, but further analysis of the loss curves and evaluation metrics (such as RMSLE or MSE) would be needed to compare the performance thoroughly.

Model Complexity: While the current model may handle the task, adding more layers or adjusting the activation functions (such as using **ReLU** in hidden layers) might improve performance, especially for regression task