Assignment 3 Report

Neural Network Library –

Using numpy package , a neural network library was created by using Layer class as base and Linear, Sigmoid, Tanh, Relu layers extending the Layer class and overriding forward and backword methods. Also BinaryCrossEntropy and MeanSquaredLoss classes are created to calculate loss and corresponding gradient.

Package implementation can be found in NeuralNetoworkodule.py.

Task 1 Sigmoid vs Tanh-

Implementing NeuralNetwork to solve XOR module –

Following architecture with sigmoid activation is implemented-

# Build the neural network

model = Sequential([

    Linear(2, 10),

    Sigmoid(),

    Linear(10, 1),

    Sigmoid()

])

Learning rate of 0.01 and 200 epochs of training was done.

Similarly , Sigmoid in the middle is replaced by Tanh activation function. Sigmoid in the end can’t be replaced as the prediction should be a value between 0 to 1 where as Tanh outputs value between -1 and 1.

Following is the architecture used for Tanh implementation-

# Build the neural network

model = Sequential([

    Linear(2, 10),

    Tanh(),          #replacing sigmoid with tanh

    Linear(10, 1),

    Sigmoid()

])

Binary Cross Entropy is used as loss function for both architectures.

A graph with a line and a red line

Description automatically generated

As we can observe from the plot tanh model converged must faster than sigmoid model. This could be due to very little data used.

Although both the models gave 100% accuracy, Tanh predictions are more confident (high probability)

Model weights for sigmoid model is stored in XOR\_solved.w.npz file.

Implementation can be found in XOR\_Example.ipynb

Task 2 ( NYC Taxi price prediction)-

Data is loaded as per the instructions provided.

Total train rows = 1.3 M and total test rows = 145 K

Columns = 25

Target = Price.

EDA –

A graph with blue dots

Description automatically generated

A clear seasonal pattern can be observed when # of pickups is plotted against the days. This could mean a greater number of pickups on weekends vs fewer on weekdays or vice versa.

A graph with a line and blue dots

Description automatically generated

Similarly, when grouped by hour , we can see a peak in number of pickups during the evening time and lowest during the early morning.

Hence generating datetime related features such as day , week , month , dayofweek etc can help in predicting prices more effectively.

Latitudes and Longitudes coordinated of pickup an drop locations are provided. This can be used to extract features such as distance from pickup location to drop location as a positive correlation can be found between distance and price.

Also when distribution of price is looked at, it is evident that data is skewed with large values. Initially when raw price is used for modelling, it was leading to gradient overflow problem. Hence it is essential to scale the target variable. This can be done by converting the target variable into log as prices are always positive.

Before scaling price.

A graph with a bar graph

Description automatically generated

After scaling Y.

A graph with a bar graph

Description automatically generated

Architecture 1-

Only sigmoid activation units are used.

model = Sequential([

    Linear(X\_train\_scaled.shape[1], 25),

    Sigmoid(),

    Linear(25, 10),

    Sigmoid(),

    Linear(10, 1),

])

Mean squared loss is used as loss function. Learing rate = 0.001 , epochs = 50 and batch size =512.

A graph with a line and a red line

Description automatically generated

RMSLE on test data- 0.24

Architecture 2-

While the number of layers and activation units are kept same, sigmoid is replaced by Relu

model = Sequential([

    Linear(X\_train\_scaled.shape[1], 25),

    ReLU(),

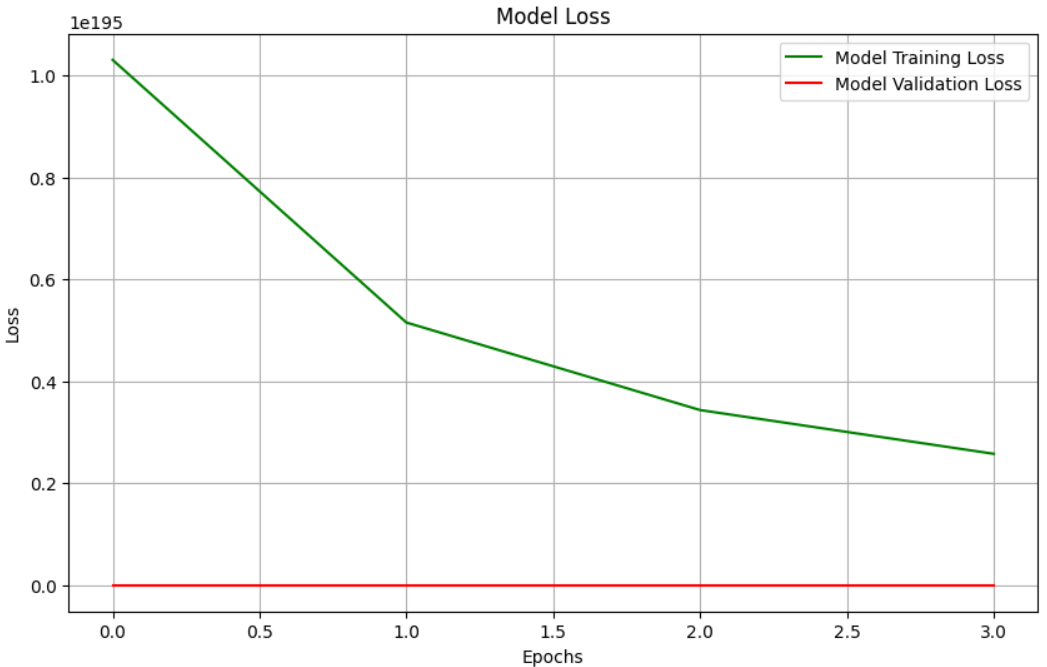
    Linear(25, 10),

    ReLU(),

    Linear(10, 1),

])

Hyperparameters are also kept same as architecture 1



When weights are analysed, exploding gradient problem is observed in the model.

RMSLE on test data - 0.79

Architecture 3-

Only sigmoid activation is used but which much deeper network.

# Build the neural network

model = Sequential([

    Linear(X\_train\_scaled.shape[1], 25),

    Sigmoid(),

    Linear(25, 50),

    Sigmoid(),

    Linear(50, 100),

    Sigmoid(),

    Linear(100, 25),

    Sigmoid(),

    Linear(25, 1),

])

Hyperparameters are kept same as earlier models.

Results-

A graph with a green line and red line

Description automatically generated

RMSLE on test data- 0.468

While it performed much better than second model, it lacked in performance when compared to first model suggesting that more number of layers = better model is not always true and it depends on size of our dataset and distribution of data as well.

Code and plots can be found in nyc\_taxi\_prediction.ipynb notebook.