CS550 Assignment 3:
MLP for Regression and Classification

Objectives of the assignment:

a. To provide practice with designing Neural networks for regression and classification
b. Understanding the impact of various hyperparameters:
    i. Different Loss Functions: MSE, MAE, MAPE
    ii. Number of layers, Wide vs. Deep Neural Networks
    iii. Choice of activation functions
    iv. Techniques of Regularization
    v. Optimization Methods, Convergence criterion and rate of convergence
    vi. Metrics for evaluating results

Note: 25 marks bonus for top 3 solutions (Accuracy and F1 score) for Parts A and B

Part 1: Predicting Taxi Fares (75 marks)

Please use the same dataset as Asg 1, Part 1 for this. Use the cleaned and transformed version of the dataset. The goal here is to build a neural network that minimizes loss on the following metrics:

MSE, MAE (Mean Absolute Arror) and MAPE (Mean Absolute Percentage Error).

Reference: https://www.tensorflow.org/api_docs/python/tf/keras/losses/

A. (20 marks) Create a baseline Neural network with the following specifications.

2 hidden layers, each with 16 and 8 neurons respectively. Sigmoid activation, Batch Size=128 for Gradient Descent.

B. (20 marks) Experiment with number of layers and neurons per layer to increase the performance metrics.

Hint: You may want to use the Wide and Deep Neural Architecture from Ch10 of the book.

C. (10 marks) Experiment with activation functions
D. (15 marks) Experiment with regularization techniques: Early stopping, Dropout rate
E. (10 marks) Experiment with at least 2 more Optimizers

Hint: You may want to use the randomized search mentioned in Ch10

Tabulate the 95% confidence intervals of each of the 3 metrics from each of the parts above neatly based on at least 5 experiments on validation.

Report the best architecture and hyper-parameters that you found in your experiments and use it to generate the predictions for the test set.

Part 2: Breaking hcaptcha (75 marks)

Please use the same dataset as Asg 1, Part 1 for this. Use the cleaned and transformed version of the dataset. The goal here is to build a neural network and report their performance on the following metrics: Accuracy, Micro-F1, Macro-F1

Reference: https://www.tensorflow.org/api_docs/python/tf/keras/losses/

(20 marks) Create a baseline Neural network with the following specifications.

4 hidden layers, each with 1024, 256, 64 and 7 neurons respectively followed by Softmax function respectively. Sigmoid activation, Batch Size=128 for Gradient Descent.

Note: The above architecture has a huge number of parameters and requires lot of memory, compute and data to be trained. You may want to scale down the input to 32x32 images. Then use the baseline network with hidden layers having: 256, 128, 64 and 7 neurons respectively followed by Softmax function. Sigmoid activation, Batch Size=128 for Gradient Descent.

(20 marks) Experiment with number of layers and neurons per layer to increase the performance metrics.

(10 marks) Experiment with activation functions

(15 marks) Experiment with regularization techniques: Early stopping, Dropout rate, *L1 for sparse model*

(10 marks) Experiment with at least 2 more Optimizers

Hint: You may want to use a Python library (Ch 10) for hyper-parameter tuning.

Tabulate the 95% confidence intervals of each of the 3 metrics from each of the parts above neatly based on at least 5 experiments on validation.

Report the best architecture and hyper-parameters that you found in your experiments and use it to generate the predictions for the test set.