A2. Understanding and Implementing LSTM from Scratch

Objectives: The goal of this project is to gain a deep understanding of Long Short-Term Memory (LSTM) networks by implementing LSTM manually. ”Manually” means that you cannot use the built-in LSTM libraries like torch.nn.LSTM in Pytorch, tf.keras.layers.LSTM in tensorflow, LSTM in MatLab Deep Learning Toolbox, ect., to call LSTM directly; you need to construct a LSTM network from scratch.

Requirements:
1) Read the original LSTM paper “Long Short-Term Memory” by Hochreiter & Schmidhuber (1997) or other papers that introduce LSTM; understand LSTM, especially the gating mechanism and training mechanism in an LSTM.

2) Manually implementing LSTM from scratch: use only basic matrix operations and activation functions; implements a single LSTM cell that processes one timestep at a time, and then extending LSTM Cell to a full sequence.

3) Train your LSTM on the following task using the attached dataset ”Car_data.zip”:

- Dataset Description: The dataset ”Car data.zip” consists of 18,457 files. Each file contains data for 12 labels, observed over 67 time steps. The data include: i) Coordinates (x, y) of the car; ii) Velocity; iii) Acceleration; iv) Distance from the centerline; v) Indicators if there’s a car on the left, right, or front; vi) Distance from your vehicle to other cars.
- Prediction Task: Your LSTM model should be trained to predict the (x, y) coordinates of the vehicle for the upcoming 5 time steps.
- Error Rate: Evaluate the performance of your model using suitable error metrics like the Root Mean Square Error (RMSE) or any other loss function you find appropriate.
- Data Usage: While the dataset provides a range of data labels, you can decide how many labels you want to use. You might choose to use only the car’s coordinates, or you could incorporate more information to potentially enhance your model’s predictions. Also, determine the portion of training, validation, testing dataset by yourself.
- Data Continuity: Although the data is continuous across many files, it does restart at random points. It is recommended to consider each file (12 labels

x 67 observations) as an individual training unit. For example, use the first 12 labels x 62 observations as the input and the corresponding 2 labels x 5 observations as the output to train the LSTM network.

Note that it is very flexible to process the data and the way of processing them depends on you.

4) Record the prediction performance using the testing dataset you divide in the step 3).

5) Adjust the hyperparameters and train again, and then record, compare, and discuss the corresponding results. Try to improve the prediction accuracy, which will be positively correlated to your final grade.