AI Homework 5

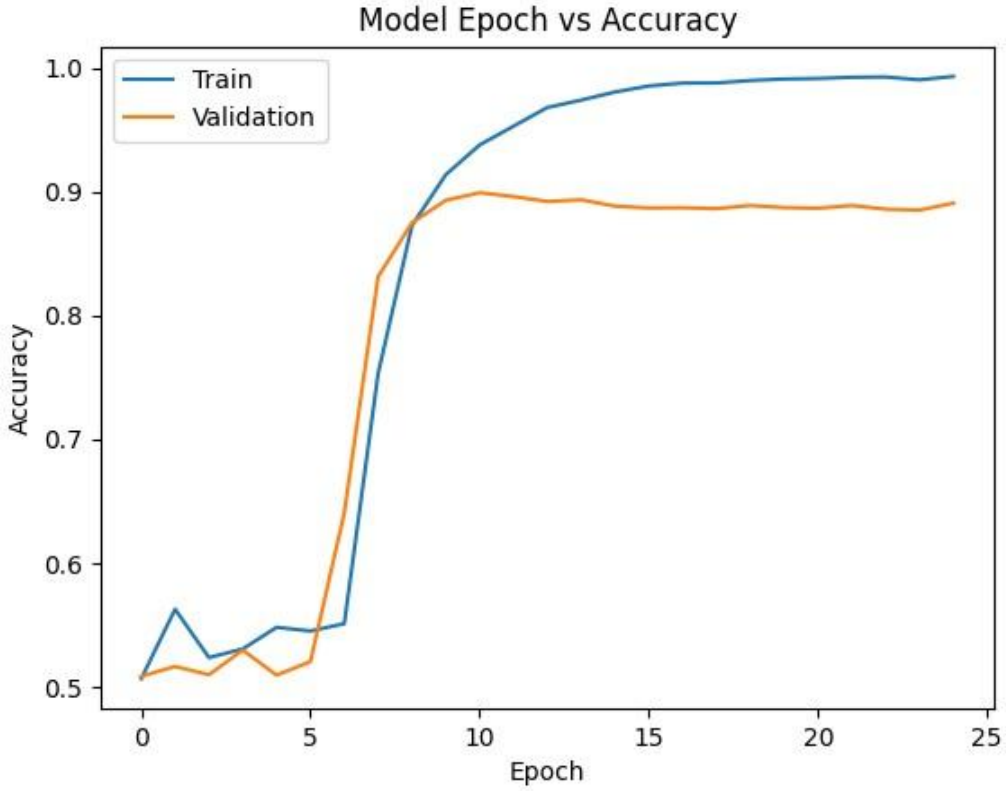
Jaden Patel

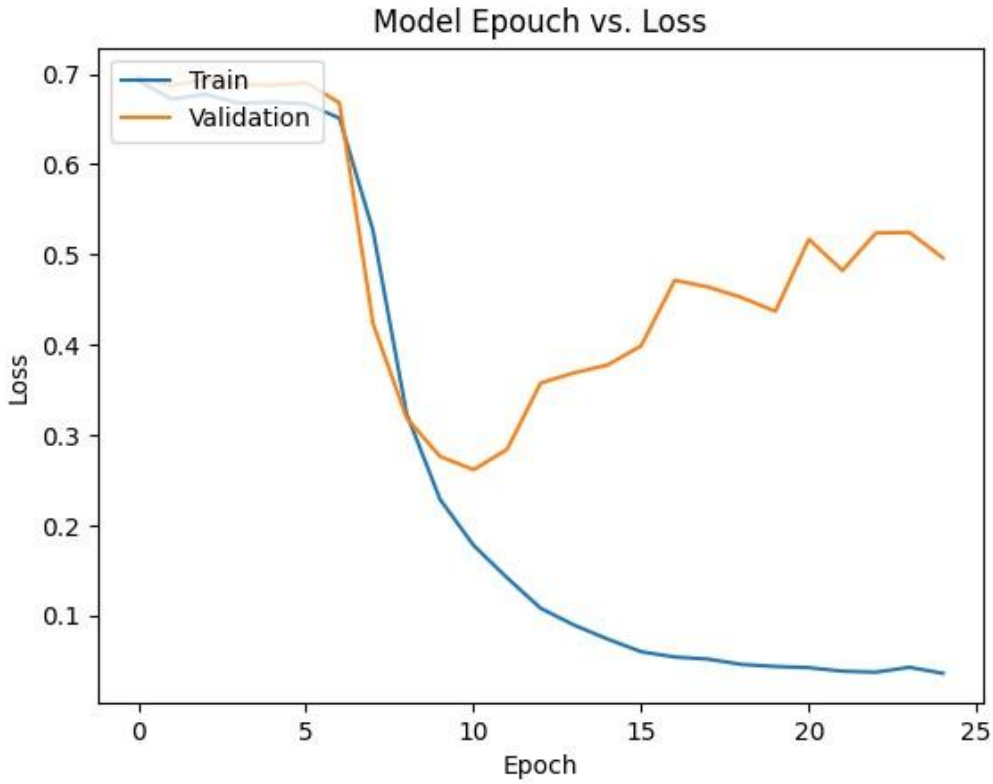
Problem 1:

Steps taken to construct the model:

1. Import necessary libraries (pandas, numpy, tensorflow, matplotlib, keras)
2. Load IMDB dataset using pandas
3. Preprocess data: process the sentiment column to numerical values using negative (0) and positive (1), tokenize the text data using Keras, and pad the sequences so that they are all the same length using Keras.
4. Build the Long Short Term Memory Model consisting of an embedding layer, two LSTM layers, and a dense output layer.
5. Compile the model using Adam as the optimizer, binary cross-entropy as the loss function, and accuracy as the evaluation metric.
6. Train the model using the fit function
7. Evaluate the performance of the model on a test dataset and output the accuracy and loss
8. Plot the epochs vs. accuracy and epochs vs. loss using matplotlib

The difference between the Long Short Term Memory model and models used in previous homework is that LSTM is a recurrent neural network, meaning it is optimized for processing and making predictions on sequential data such as text. For example, a previous model we used in homework 4, ResNet50, is a convolution neural network, meaning it is optimized for grid-like data and extracting spatial features from data. ResNet50 uses convolutional layers, pooling layers, and fully connected layers to extract features from an input image. LSTM uses an input layer, an LSTM layer, and an output layer to allow the model selectively save or forget information over time in memory cells and gates to capture long-term dependencies in an input sequence. This is useful for tasks where the context of the data is critical for an accurate prediction, such as language modeling and speech recognition.


Model Epoch vs Accuracy
Train
Validation
Accuracy
Epoch
1.0
0.9
0.8
0.7
0.6
0.5
0
5
10
15
20
25


Model Epouch vs. Loss
Train
Validation
Loss
Epoch
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25