ICS 661

Advanced AI

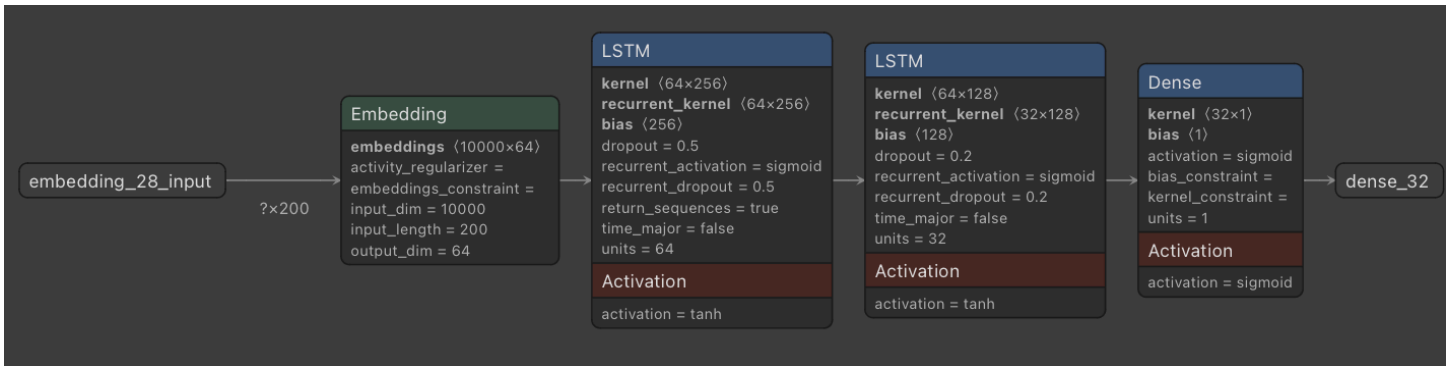
Fall 2024

Assignment 2 Report

Section 1: Task Description

The task of this assignment was to classify movie reviews into a binary sentiment, either positive or negative. This task was to be done using a RNN.

Section 2: Model Description



Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 200, 64)	640000
Lstm (LSTM)	(None, 200, 64)	33024
lstm (LSTM)	(None, 32)	12416
dense (Dense)	(None, 1)	33

Total params: 685,473
Trainable params: 685,473
Non-trainable params: 0

Section 3: Experiment Settings

3.1 Dataset Description

The dataset consists of 50,000 movie reviews split equally between test and train sets. Which is then further split equally into 12,500 positive and 12,500 negative reviews in each data set. I further split the training set in a 80/20 split to be used as a validation set while training. The dataset text was cleaned

and preprocessed removing special characters, changing to lowercase, and removing stop words using the nltk library. The text was then tokenized and used a maxed length of 200 which also acted as a padding length.

3.2 Detailed Experimental Setups

I tested out two models trying to optimize hyper parameters for each. The first one used LSTM layers and the second used BiLSTM layers. I ended up going with the first one as it produced similiar results and was a faster model. I ran short on time so I was not able to fully comit to optimizing the model, however it still scored about the required score for the assignment. I was also seeing over fitting when running for more than 3 epochs so I traind this model only 3 epochs.

Model Architecture:

- Input layer: created an embedding with the vocab size of 10,000, a sequence length of 200 and output with 64 dimensions.
- Hidden Layers: I used two LSTM layers the first was 64 unites and included dropout of .5. The second layers had 32 units with a dropout of .2.
- Output layer: I used a sigmoid activation with a single dense layer.
- Optimizer: I used Adam with a learning rate of .0001 and loss function as binary crossentropy.

3.3 Evaluation Metrics

For this project, the model's performance is evaluated using four key metrics which include accuracy, precision, recall, and F1 Score. These metrics provide a comprehensive understanding of the model's classification performance across different aspects.

Accuracy: This metric represents the proportion of correctly classified instances out of the total instances. It is useful for getting a general sense of how well the model performs, but it can be misleading in cases of imbalanced datasets.

Precision: Precision measures the proportion of true positive predictions out of all instances predicted as positive. It focuses on the accuracy of the positive predictions and is particularly important when the cost of false positives is high.

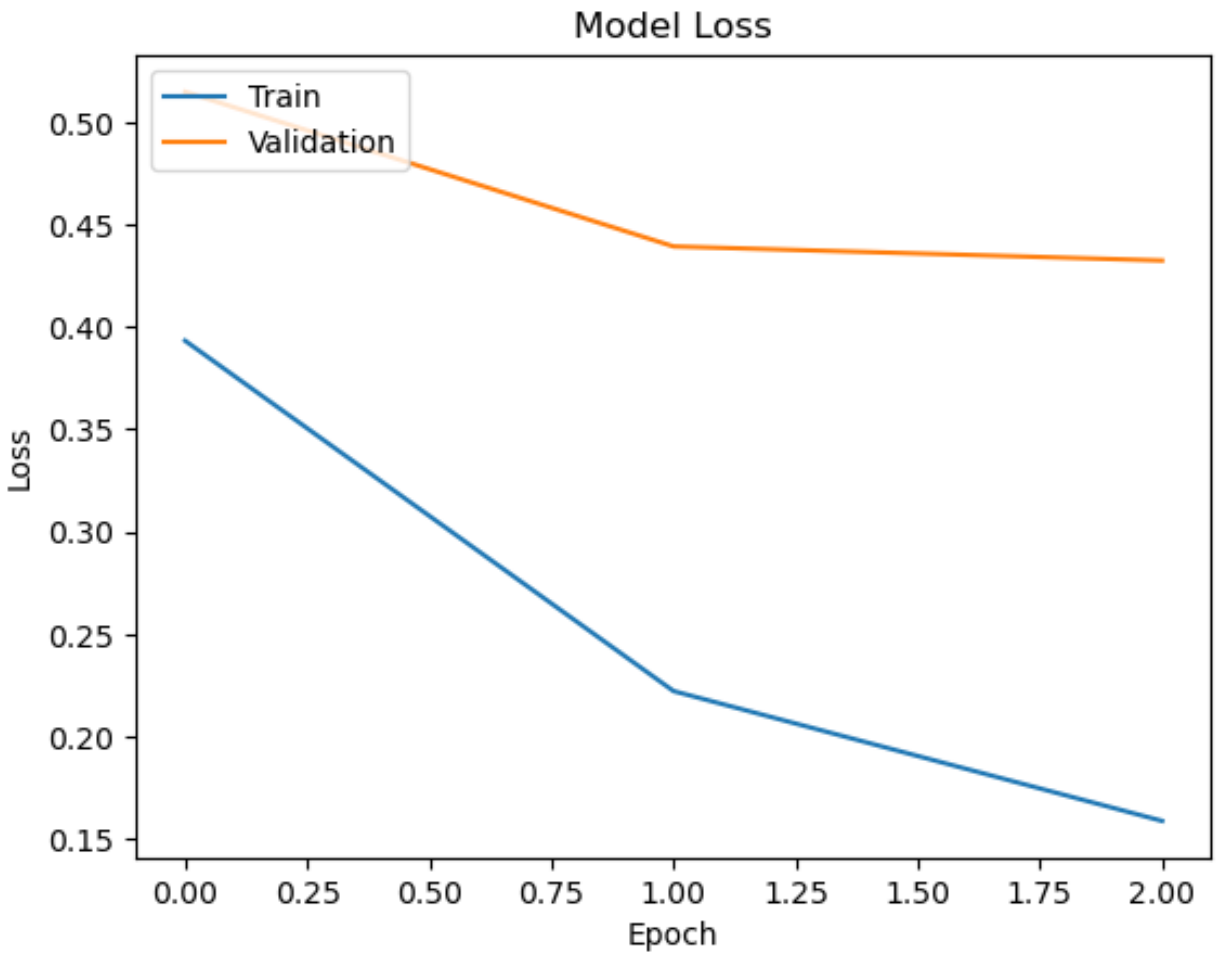
Recall: Also known as sensitivity or true positive rate, recall represents the proportion of true positives out of all actual positive instances. It is crucial in scenarios where missing positive instances (false negatives) is costly.

F1 Score: The F1 Score is the harmonic mean of precision and recall. It provides a balanced measure that considers both precision and recall, making it useful when you need to balance false positives and false negatives.

3.4 Source Code

Source code can be found here on GitHub

3.5 Training Convergence Plot



3.6 Model Performance

	Accuracy	Precision	Recall	F1
Training	0.9486	0.9324	0.9674	0.9496
Test	0.8582	0.8576	0.8589	0.8583

3.7 Ablation Studies

Model 1:

	Accuracy	Precision	Recall	F1
Training	0.9486	0.9324	0.9674	0.9496
Test	0.8582	0.8576	0.8589	0.8583

Model 2:

	Accuracy	Precision	Recall	F1
Training	0.9559	0.9233	0.9944	0.9575
Test	0.8539	0.8272	0.8947	0.8596