CSI 5340 Homework 3

Name: Jayshil Patel

Student Number: 300312550

Dataset

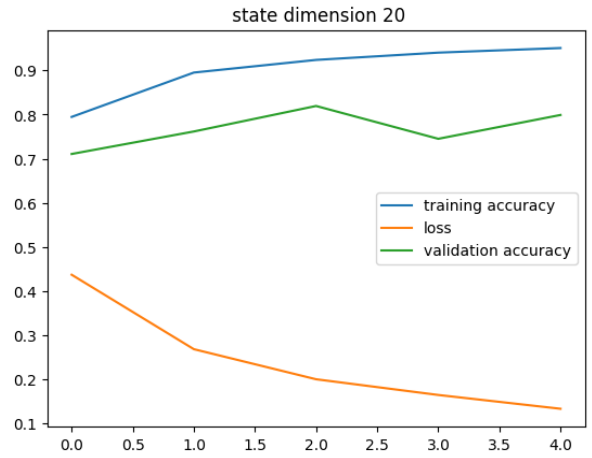
The dataset was downloaded from Stanford's datasets for sentiment analysis. The dataset contained 2 folders for the train and test containing 2 more folders in each for each negative and positive review containing 12500 .txt files for reviews. To make the textual data machine-readable, a tokenization process was applied. The 'Tokenizer' class from the tokenization module in Keras was employed. This class facilitated converting raw text into sequences of integers. Subsequently, padding was applied to ensure uniform sequence lengths.

Vanilla RNN

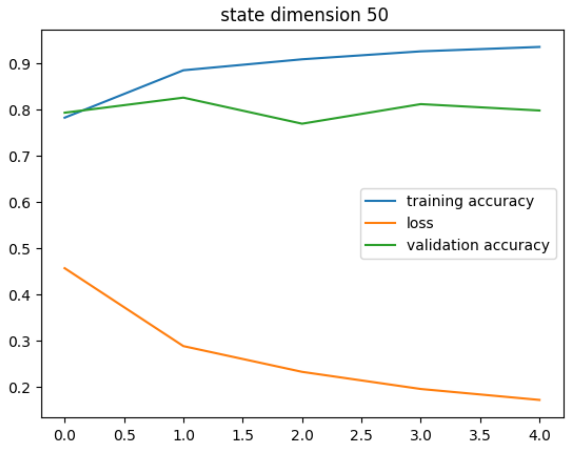
Parameter Settings

Dropout choice = [0.0, 0.4, 0.5, 0.6]
Learning rate choice = [1e-2, 1e-3, 1e-4]
State dimensions choice = [20, 50, 100, 200, 500]

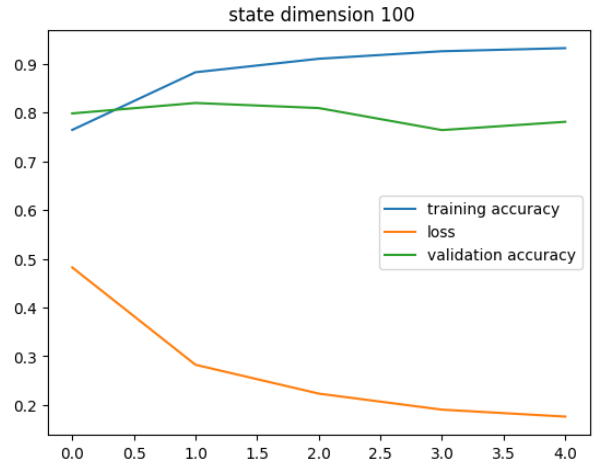
For state dimensions 20:-
Best model's hyperparameters:-
Dropout = 0.4
Learning rate = 0.001



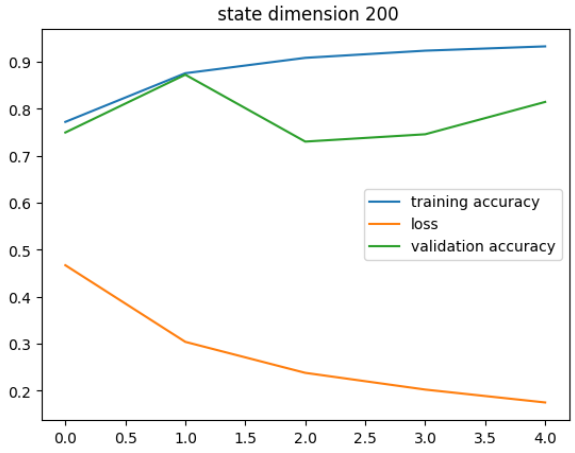
For state dimensions 50:-
Best model's hyperparameters:-
Dropout = 0.6
Learning rate = 0.001



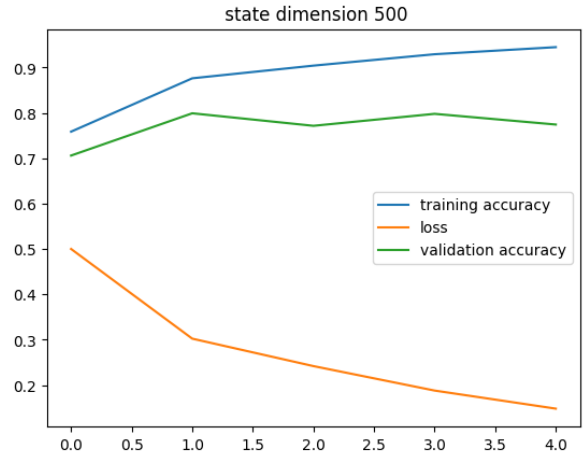
For state dimensions 100:-
Best model's hyperparameters:-
Dropout = 0.6
Learning rate = 0.01



For state dimensions = 200:-
Best model's hyperparameters:-
Dropout = 0.6
Learning rate = 0.001



For state dimensions 500:-
Best model's hyperparameters:-
Dropout = 0.4
Learning rate = 0.001



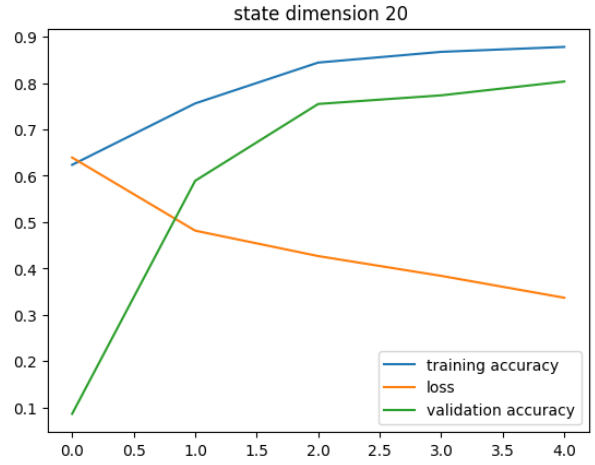
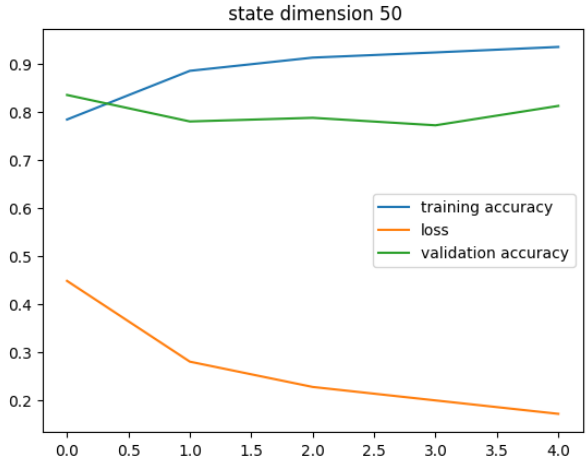
Test Accuracy is observed to be the best when we have state Dimension = 50 and the test loss for the same state dimensions is second lowest among all the test losses. Here we can also observe that the train accuracy for each epoch was better than validation accuracy for all the combinations of state dimensions.

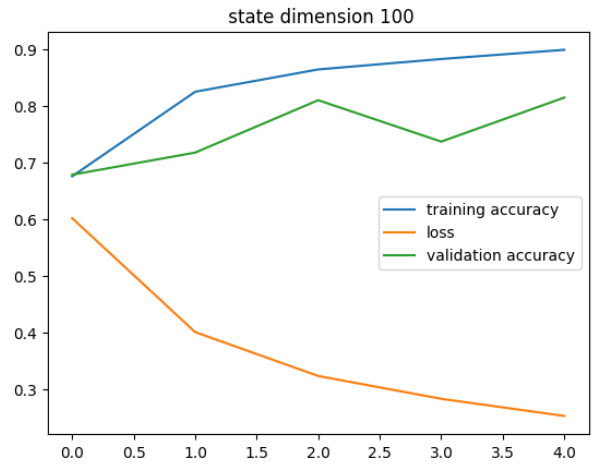
State Dimensions	Test Accuracy	Test Loss
20	0.8303999900817871	0.46684908866882324
50	**0.8388800024986267**	0.4198630452156067
100	0.8366400003433228	0.5038560032844543
200	0.8370000123977661	**0.41630539298057556**
500	0.8248000144958496	0.5762128829956055

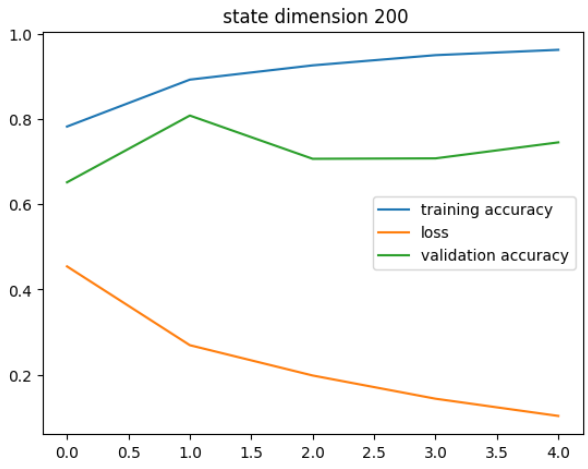
LSTM model

Parameter setting

Dropout choice = [0.0, 0.4, 0.5, 0.6]
Learning rate choice = [1e-2, 1e-3, 1e-4]
State dimensions choice = [20, 50, 100, 200, 500]

<table>
<tr>
<td>For state dimensions 20:-
Best model's hyperparameters:-
Dropout = 0.5
Learning rate = 0.0001
  </td>
<td>For state dimensions 50:-
Best model's hyperparameters:-
Dropout = 0.6
Learning rate = 0.001
  </td>
</tr>
<tr>
<td>For state dimensions 100:-
Best model's hyperparameters:-
Dropout = 0.5
Learning rate = 0.0001</td>
<td>For state dimensions 200:-
Best model's hyperparameters:-
Dropout = 0.0
Learning rate = 0.001</td>
</tr>
</table>



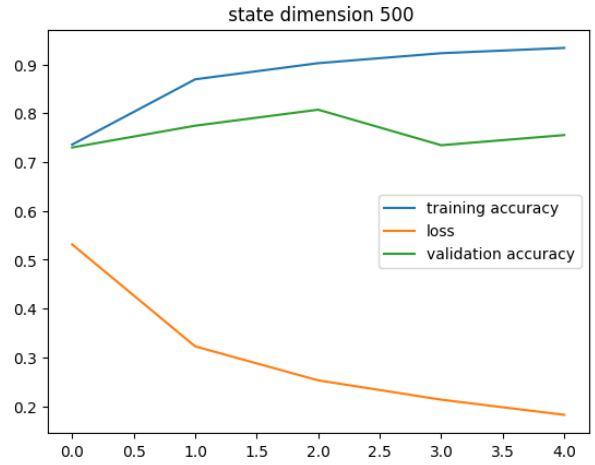


For state dimensions 500:-

Best model's hyperparameters:-

Dropout = 0.0

Learning rate = 0.0001



Here, same as the vanilla RNN model, the LSTM also has better training accuracy when compared to validation accuracy for each epoch. Here best stats are for state dimensions = 100. Both test Accuracy and test loss are the highest and lowest respectively.

State Dimensions	Test Accuracy	Test Loss
20	0.8328400254249573	0.43734198808670044
50	0.8384000062942505	0.4352225065231323
100	**0.8453199863433838**	**0.34723758697509766**
200	0.8171600103378296	0.5396192073822021
500	0.832360029220581	0.40082207322120667

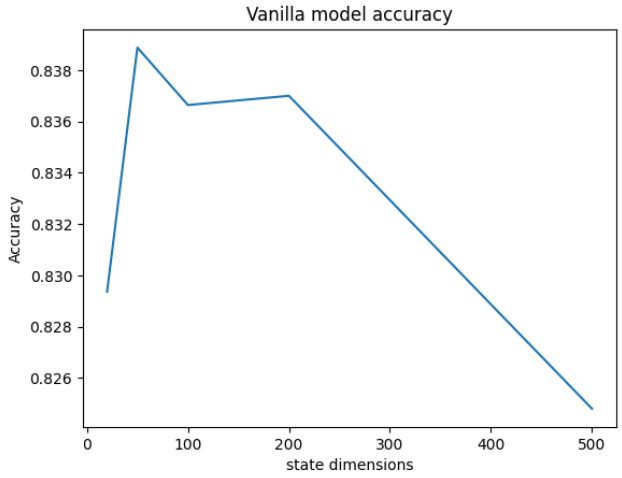


Figure - Vanilla model Accruacy for different state dimensions.

Comparing the LSTM model and the Vanilla RNN model. It is obvious to conclude that the LSTM model is better than the Vanilla RNN model because of its higher accuracy and lower loss. However, the training time for the LSTM model is more when compared to Vanilla for the same parameter settings. The sudden decrease in accuracy as the state dimension increases from 200 to 500 in Vanilla RNN can be attributed to the overfitting of the model or the gradient vanishing.