CSI 5340 Assignment 3

Name: Alim Manjiyani

Student Number: 300229095

18th October 2021

This exercise is focused on developing Vanilla RNN and LSTM text classification models on the given IMDB Movie Review Dataset. We are to develop these models based on the given set of state dimensions and need to tune the hyper-parameter for each model so that we can compare the best results out of each state dimension and model setting. This exercise enabled me to learn new techniques, and helped me to visualise the theory learned in the classroom.

1 Dataset acquisition and Model setup

1.1 Dataset

Although the dataset can be acquired from Kaggle and pre-processed manually; I decided to use the tensorflow's keras dataset library, which provides the pre-processed version of this dataset, where each review is encoded as a list of word indices. The indexing of words in the reviews is done based on their appearance frequency.

1.2 Model Setup

I decided to use a simple setup and the same number of layers for each model for better accuracy comparison. Each model begins with an Embedding layer with embedding size of 128, followed by the respective model layer (SimpleRNN or LSTM) with 128 units and finally a dense layer with sigmoid activation. Below is the list of Hyper-parameters which are common to both the models.

Table 1 Common Hyper-parameters

Parameter	*Value*
Activation	Sigmoid
Loss function	Binary cross-entropy
Optimizer	Adam
Embedding size	128

1.3 Hyper-parameter Tuning

Both the models are tuned based on the list of hyper-parameters shown in Table 2. The best model is chosen based on it's accuracy score. The hyper-parameter tuning is performed using the RandomSearch method of the keras-tuner library.

Table 2 Hyper-parameter tuning configuration

Parameter	*Value*
Batch sizes	[32, 64, 96]
Dropout	[0.0, 0.4, 0.5, 0.6]
Learning rate	[0.01, 0.001, 0.0001]
Executions per trial	2
Max_trials	3

2 Hyper-parameter Tuning Results

2.1 Vanilla RNN

The best setting for each state-dimension was chosen based on its testing accuracy score, the results of the same are presented in the table below.

Table 3 Tunning Results for Vanilla RNN

State Dimensions	*Parameters*			
	Batch size	*Learning Rate*	*Dropout*	*Accuracy Obtained*
20	64	0.001	0.5	75.21
50	96	0.001	0.5	80.14
100	32	0.0001	0.6	84.13
200	32	0.0001	0.6	86.22
500	32	0.0001	0.0	84.92

2.2 LSTM

The best setting for each state-dimension was chosen based on its testing accuracy score, the results of the same are presented in the table below.

Table 4 Tunning results for LSTM

State Dimensions	*Parameters*			
	Batch size	*Learning Rate*	*Dropout*	*Accuracy Obtained*
20	96	0.001	0.4	76.38
50	64	0.01	0.4	80.10
100	32	0.0001	0.6	84.93
200	32	0.01	0.4	86.04
500	32	0.0001	0.4	88.66

3 Model Comparison and Analysis

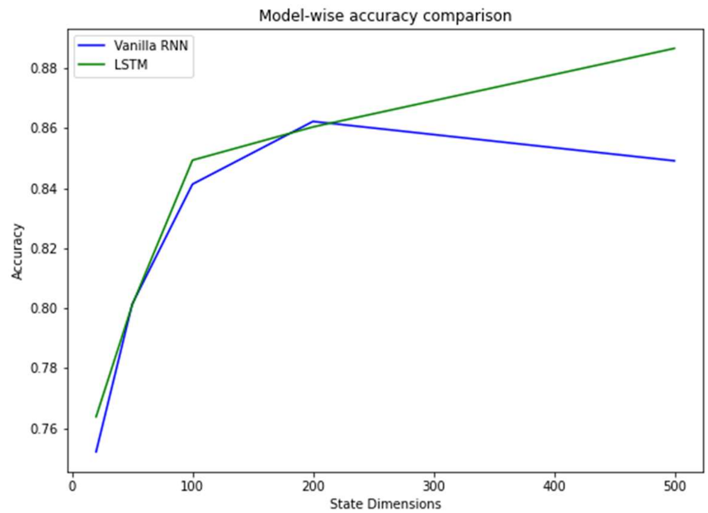

Figure 1 Model-wise accuracy comparison

Figure 1. shows the performance of both models w.r.t their state-dimensions in the best hyper-parameter settings. The following information can be inferred and verified from the results obtained above.

- It can be noticed that LSTM outperforms Vanilla RNN as expected because of the "gradient vanishing problem" in the Vanilla RNN or the better compatibility of LSTM with the given dataset.
- The sudden decrease in accuracy as the state-dimension increases from 200 to 500 in Vanilla RNN can be attributed to overfitting of the model or the gradient vanishing.
- Since each state-dimension has a different tuned hyper-parameter setting, it can be inferred that keeping a fixed set of hyper parameters and increasing the state-dimension will lead to a poor performance, which further tells us that: for a given hyper parameter setting there exists a peak (state-dimension) which will give us the best result on that setting, and hence it is not always true that increasing the state-dimension will lead to a better result.
- Although LSTM performs better than Vanilla RNN in this case, It can be seen in Figure 1, that Vanilla RNN outperforms LSTM when the state-dimension is set to 200, which tells us that RNN is not fundamentally inferior to LSTM and relates to what we have studied in the class.