Background:

Use Vanilla RNN and LSTM to predict the IMDB dataset's sentiment (negative (0) or positive (1)), and take 0.5 as threshold for final sigmoid function (if it is larger than 0.5, that sentence's sentiment is positive). Meanwhile, I fixed embedding dimension as 100, run each experiment for 5 epochs, and tuned some hyper parameters as below:

(1) Divided original training set (250000) as training set and validation set with ratio: 0.7,0.8,0.9

(2) Hidden dimension:20,50,100,200,500

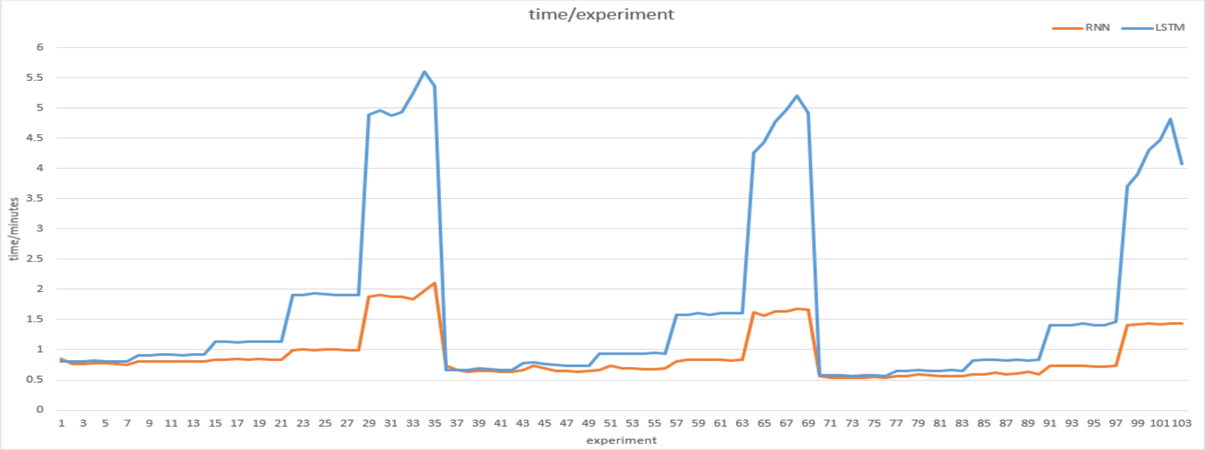
(3) Learning rate:0.001,0.003,0.005,0.008,0.01,0.02,0.03

As a result, 105 experiments for each Vanilla RNN and LSTM were run, spending 284.23 minutes(about 4.7hours). The highest test accuracy is 62% from LSTM, when ratio=0.9, hidden dimension=100 and learning rate=0.005.

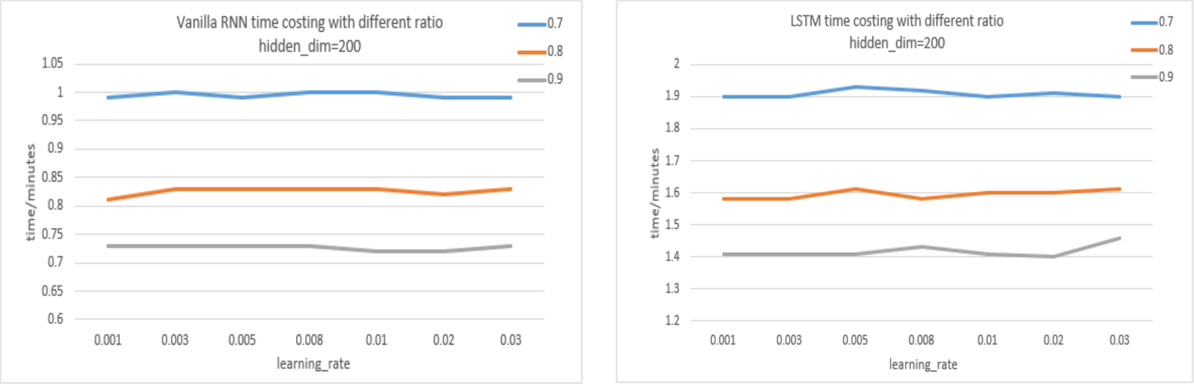
Analysis:

1 About time costing:

(1) Gernerally speaking, LSTM's time for each experiment is longer than Vanilla RNN, the maxiumn time for LSTM is about 5.5 mintues while the maxium time for Vanilla RNN is about 2 minutes. But their differences are not such big all the time, In some case, the time difference between them is very small.

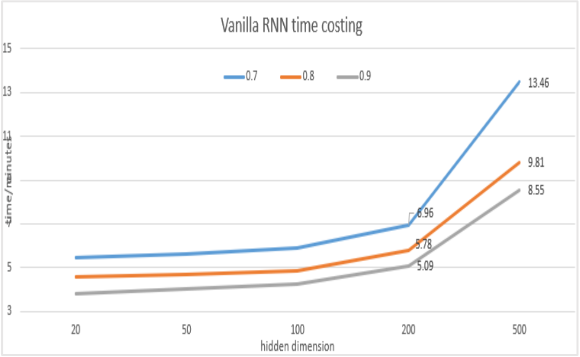


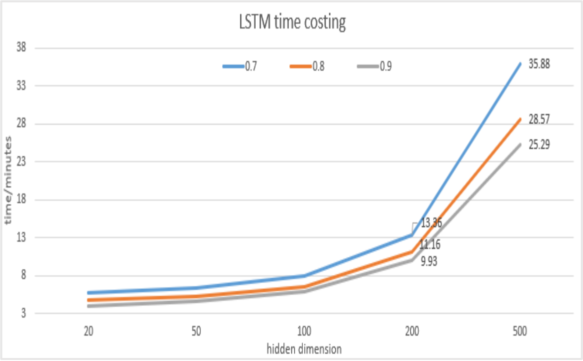
(2) Learning rate did not affect experiment's time much. For different training data size (07,0.8,0.9), their experiment's time do not vary much along with different learning rate. However, there is something interesting, for both Vanilla RNN and LSTM, the experiment's time decrease while the training data size is increased.



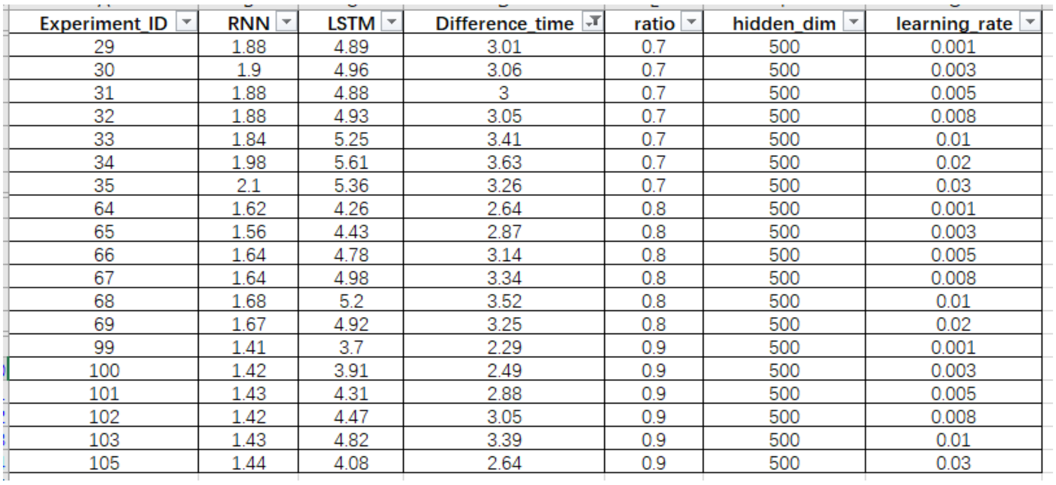
(3) The size of hidden dimension affects the experiment's time much, especially when the dimension

is 500. When the dimension is smaller than 200, time difference for each dimension is not too big.





Furthermore, if we set 1 minutes as threshold (>=1) for the time difference between Vanilla RNN and LSTM, we can see their hidden dimension are all 500.



Experiment_ID	RNN	LSTM	Difference_time	ratio	hidden_dim	learning_rate
29	1.88	4.89	3.01	0.7	500	0.001
30	1.9	4.96	3.06	0.7	500	0.003
31	1.88	4.88	3	0.7	500	0.005
32	1.88	4.93	3.05	0.7	500	0.008
33	1.84	5.25	3.41	0.7	500	0.01
34	1.98	5.61	3.63	0.7	500	0.02
35	2.1	5.36	3.26	0.7	500	0.03
64	1.62	4.26	2.64	0.8	500	0.001
65	1.56	4.43	2.87	0.8	500	0.003
66	1.64	4.78	3.14	0.8	500	0.005
67	1.64	4.98	3.34	0.8	500	0.008
68	1.68	5.2	3.52	0.8	500	0.01
69	1.67	4.92	3.25	0.8	500	0.02
99	1.41	3.7	2.29	0.9	500	0.001
100	1.42	3.91	2.49	0.9	500	0.003
101	1.43	4.31	2.88	0.9	500	0.005
102	1.42	4.47	3.05	0.9	500	0.008
103	1.43	4.82	3.39	0.9	500	0.01
105	1.44	4.08	2.64	0.9	500	0.03

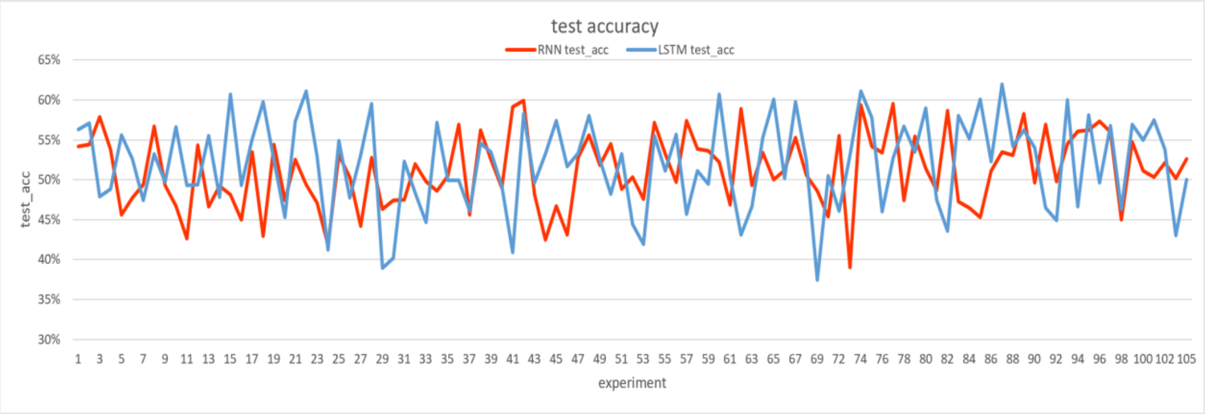
Besides, when hidden dimension is too big. It is easier to cause exploding gradient problem. In my 150 experiments, only two have exploding gradient problems and their hidden dimension size are both 500. Note: I already delete these two outlier experiments when did data analysis.

type	**ratio**	**hidden_dim**	**learning_rate**	**train_loss**
Vanilla RNN	0.8	500	0.03	[nan, nan, nan, nan, nan]
LSTM	0.9	500	0.02	[nan, nan, nan, nan, nan]

As a conclusion, when training set's size increases, the experiment's time decreases; when hidden dimension's size increase, the experiment's time increases; when learning rate increases but is still lower than 0.03, the experience's time almost didn't change.

2 About accuracy:

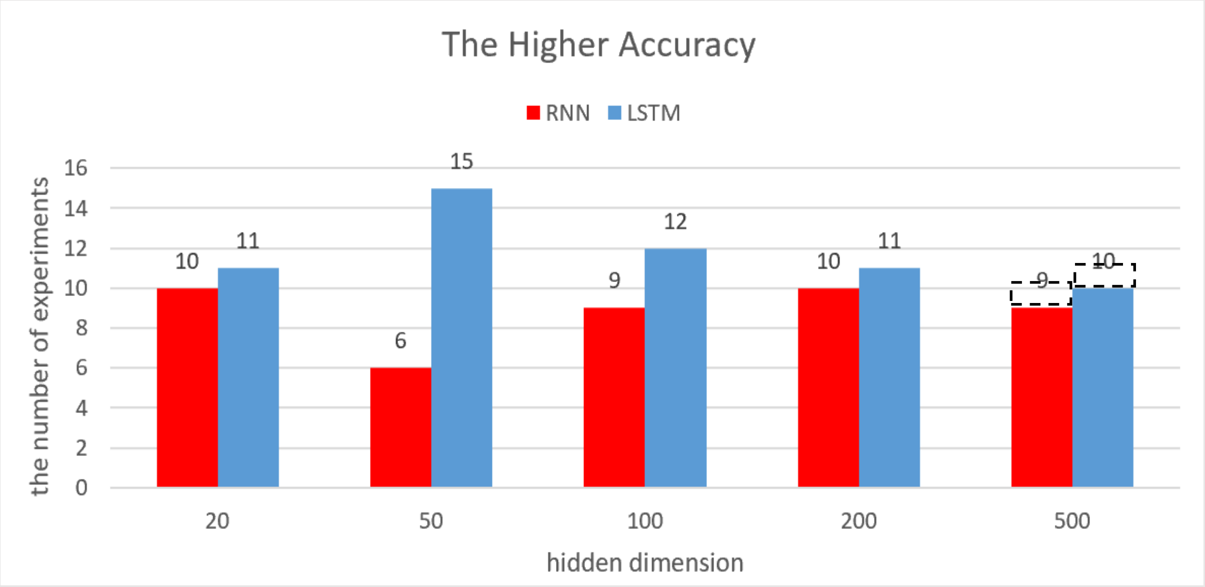
(1) LSTM's test accuracy is not always higher than Vanilla RNN's. In 57.28% experiments, test accuracy from LSTM is larger than that from Vanilla RNN. However, the highest accuracy is 62% from the LSTM while it is 59.9% from Vanilla RNN.



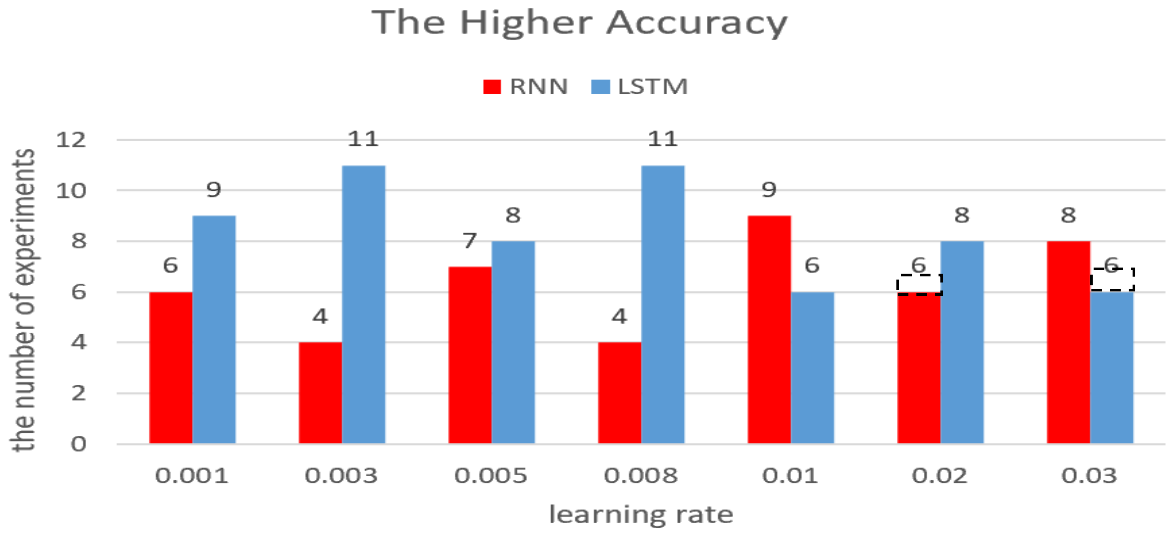
I draw some picture to analysis in which situation the number of experiments with higher accuracy. For example, in the first below picture, when hidden dimension is 20, there are 10 experiments with higher accuracy from Vanilla RNN, and 11 experiments from LSTM.

Note: there are two outliers. Considering when there is gradient explosive problem, the accuracy is incorrect. So, the other one was regard as with higher accuracy. And I recovered them with dotted frame. One dotted frame represents one experiment.

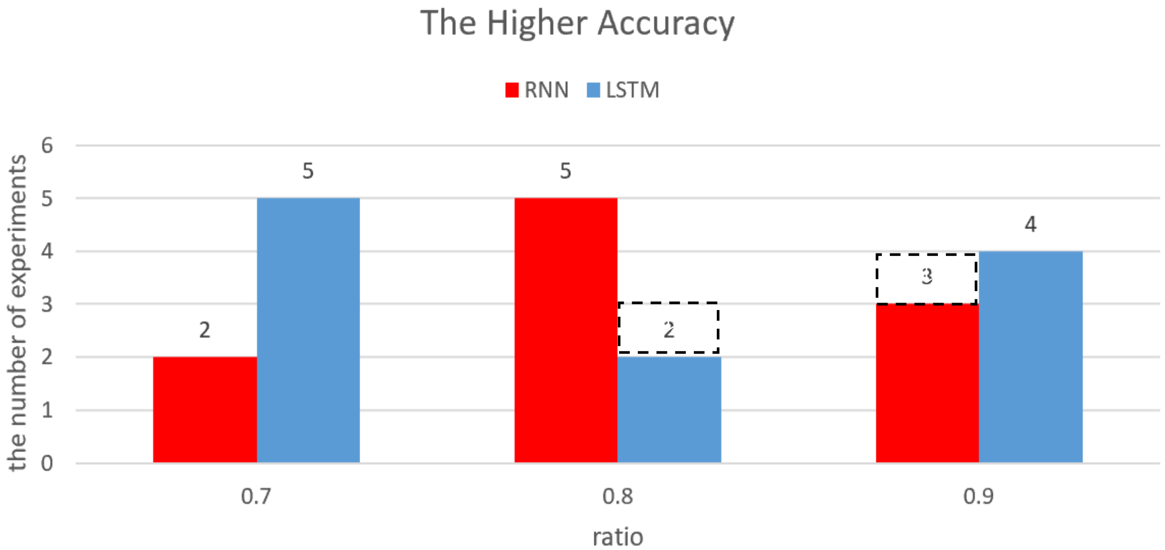
(2) When the hidden dimension is low, it seems the number of experiments with higher accuracy from LSTM is more than that from Vanilla RNN. However, when the hidden dimension is higher, such 200 or 500, the number of experiments with higher accuracy from each method seems similar.



(3) When the learning rate is small, the number of experiments with higher accuracy from LSTM is more than that from Vanilla RNN; when the learning rate is bigger, the number of experiments from both sides seems similar.



(4) Because when the hidden dimension equals to 200, the number of experiments with higher accuracy are similar, so I fixed the dimension as 200 to analysis whether the size of training data set could affect the number of experiment with higher accuracy or not. For the below picture, we can see when the training set's size is smaller, the number of experiments with higher accuracy from LSTM is more than that from Vanilla RNN; when the training set's size is big enough, the number of experiments from both methods seems similar.



As a result, if the size of training data is small, hidden dimension is small than 200 and learning rate is smaller than 0.01, we can get higher accuracy via LSTM with higher probability.

Lesson Learnt:

Before modeling each batch, we should initialize the hidden layers. Otherwise, the calculation's values in hidden layers would be accumulated and the data may be too big for normal computer to calculate. I was required about 12G GPU before initialization hidden layers.