CSI 5138 Assignment3

Yu Sun(8472921)

October 2020

1 Data Analysing

In this assignment, I used IMDB Movie Review Dataset, which contains 25000 positive reviews and 25000 negative reviews. Firstly, I did some statistic analysis for the dataset, which could be benefit for building the model and setting the hyper-parameter. I calculated the word length of each review, and plotted in Figure 1. We can see that in general the word length is less than 500, and the minimum length is 10 while the maximum length is 2470.

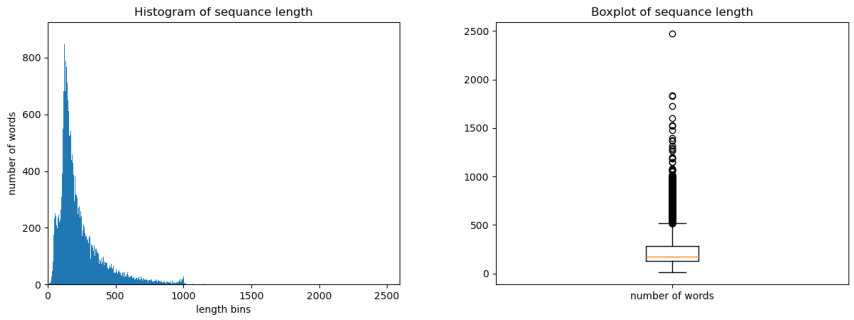

Figure 1: Word Length of Each Review

Meanwhile, like the standard NLP work, I did the preprocessing for the text, including removing all punctuations and numbers, turning the texts into space-separated sequences of words and removing the html tages and multiple spaces.

2 Modeling

The main models I used are Vanilla RNN and LSTM, then for each model I used two different methods to deal with the hidden state. One is only keep the final hidden layer's output, then use the fully connection layer with sigmod activation to predict. While the other one use all output of the hidden layers,

then use the mean pooling layer before the fully connection layer. The network structures plot in Figure 2 and Figure 3 respectively.

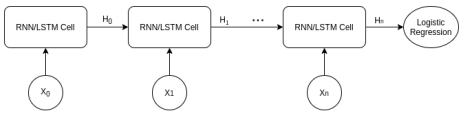

Figure 2: Model without Mean Pooling

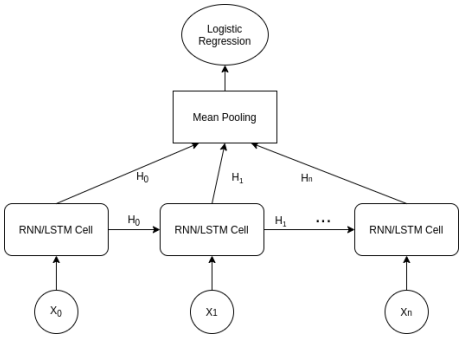

Figure 3: Model with Mean Pooling

Besides, considering to compare the different state dimension in RNN and LSTM, I build 20 models in total, which are summarised in Table 1. First, about the other hyper-parameters, I tuned also several values and choose a relative best one for all models, which also showed in Table 1.

For the batch size, in general too large of a batch size will lead to poor generalization, while small batch size may lose the opportunity to converge to the global optima.

For the optimizer, I used Adam, which combine the benefit of Adadelta and RMSprop and can usually get a relative better result. Besides, I also used learning rate with the exponential decaying. In the beginning of training, it used a larger learning rate(maximum learning rate in Adam) and reduced the rate in the process of training, which can help both optimization and generalization in general.

For the word length of each review(after preprocessing for the text), I choose the 300, which can cover 75% reviews. Then I did the padding with zeroes for the review less than 300 words. For the word embedding vectors, I used the Glove with 50 dimension for each words.

3 Evaluation

The epoch I used is 30, and the result is summarised in Table 2 and plot the test accuracy of each training epoch in Figure 4, training loss in Figure 5. In

	Mean Pooling	State Dimension	Batch Size	Optimizer	Learning Rate
Vanilla RNN-1-1	No	20	64	Adam	0.001
Vanilla RNN-1-2	No	50	64	Adam	0.001
Vanilla RNN-1-3	No	100	64	Adam	0.001
Vanilla RNN-1-4	No	200	64	Adam	0.001
Vanilla RNN-1-5	No	500	64	Adam	0.001
Vanilla RNN-2-1	Yes	20	64	Adam	0.001
Vanilla RNN-2-2	Yes	50	64	Adam	0.001
Vanilla RNN-2-3	Yes	100	64	Adam	0.001
Vanilla RNN-2-4	Yes	200	64	Adam	0.001
Vanilla RNN-2-5	Yes	500	64	Adam	0.001
LSTM-1-1	No	20	64	Adam	0.001
LSTM-1-2	No	50	64	Adam	0.001
LSTM-1-3	No	100	64	Adam	0.001
LSTM-1-4	No	200	64	Adam	0.001
LSTM-1-5	No	500	64	Adam	0.001
LSTM-2-1	Yes	20	64	Adam	0.001
LSTM-2-2	Yes	50	64	Adam	0.001
LSTM-2-3	Yes	100	64	Adam	0.001
LSTM-2-4	Yes	200	64	Adam	0.001
LSTM-2-5	Yes	500	64	Adam	0.001

Table 1: All Models with Hyper-Parameters

my experiment, among all four models the best state dimension is 100. Besides, along with the increasing of state dimension, the ability of model generalization(test accuracy) increased first then began to drop. Smaller state dimension means the model is simple to learn the pattern of the data, while larger one means it is complicated may have the problem of gradient vanishing. Then for the training time, the larger state dimension definitely need more time.

Besides, I also compared the four models with state dimension of 100. We can see the LSTM is better than Vanilla RNN, which can retain long-term dependencies and connect information from the past to the present through the gate operation. The mean pooling can have a relative better result in Vanilla RNN, because it used more information from the RNN cell operation. The training time of LSTM is slower than Vanilla RNN due to more parameters.

4 Discussion

From the experiment, I realized that the state dimension, which is a key hyper-parameter of RNNs, have much more influence on model generalization. We need to do more test and choose a reasonable one based on the dataset. For dif-

Model	Test Accuracy	Model	Test Accuracy
Vanilla RNN-1-1	0.7265	LSTM-1-1	0.8450
Vanilla RNN-1-2	0.7386	LSTM-1-2	0.8586
Vanilla RNN-1-3	**0.7414**	LSTM-1-3	**0.8682**
Vanilla RNN-1-4	0.7350	LSTM-1-4	0.8585
Vanilla RNN-1-5	0.6904	LSTM-1-5	0.8367
Vanilla RNN-2-1	0.7954	LSTM-2-1	0.8552
Vanilla RNN-2-2	0.8221	LSTM-2-2	0.8615
Vanilla RNN-2-3	0.**8345**	LSTM-2-3	**0.8621**
Vanilla RNN-2-4	0.7952	LSTM-2-4	0.8464
Vanilla RNN-2-5	0.7815	LSTM-2-5	0.8508

Table 2: The Result of All Models

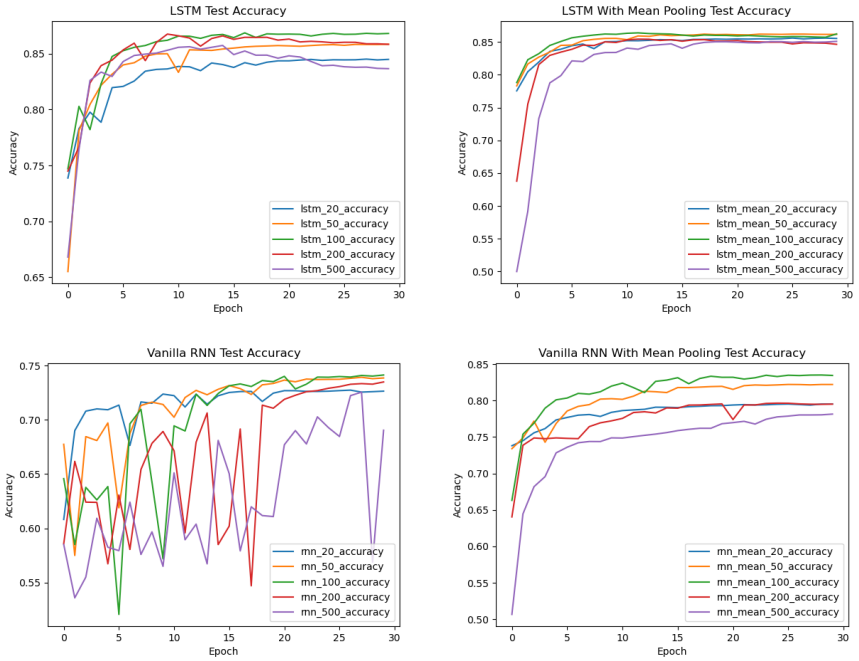

Figure 4: Test Accuracy of Each Model

ferent models, LSTM is usually better than Vanilla RNN and the mean pooling can have a positive effect on the model.

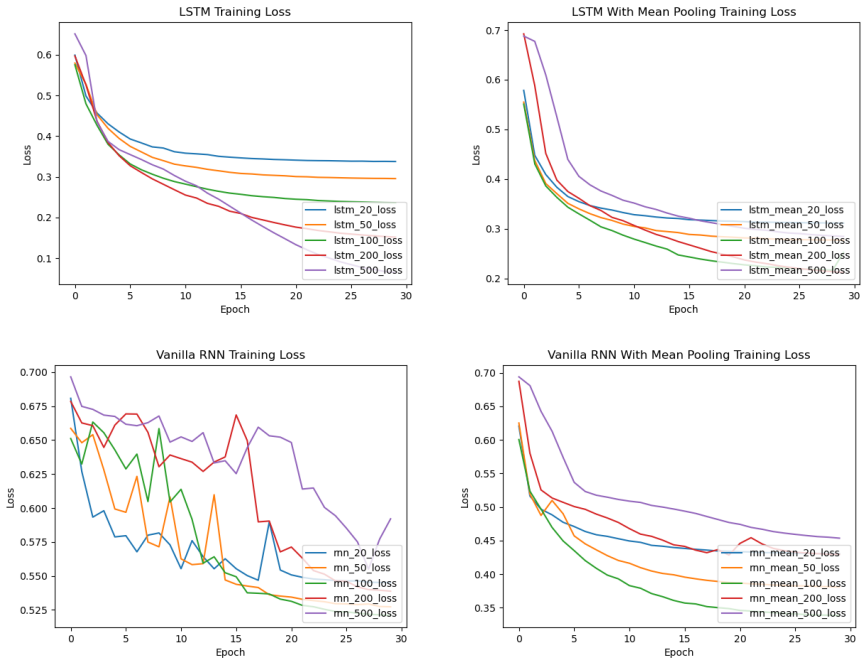

Figure 5: Training Loss of Each Model

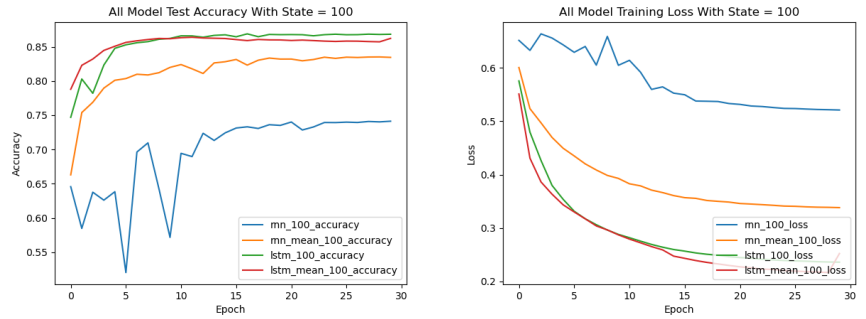

Figure 6: Test Accuracy and Training Loss of Each Model