ECS 189G-001

Deep Learning

Winter 2023

Course Project: Stage 4 Report

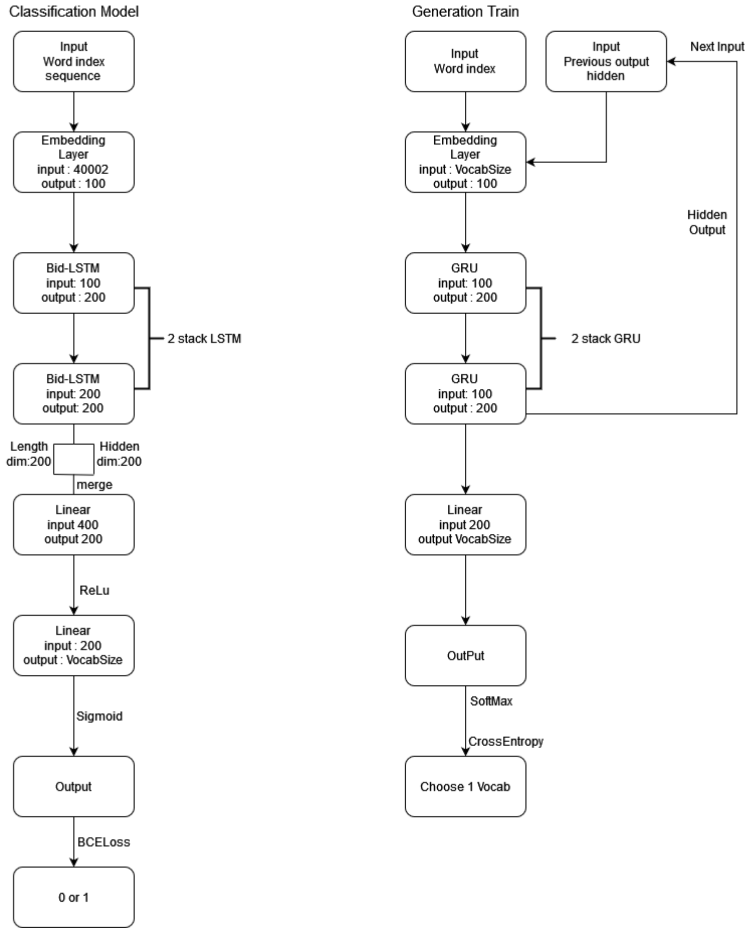
Team Information

Enter Your Team Name Here **(delete the extra rows if your team has less than 4 students)**		
Name : Chung Ying Hsu	ID: 920918764	Email: cyhsu@ucdavis.edu

Section 1: Task Description

In this project, I have two datasets, one based on IMDB ratings which will be used for text classification, and another dataset from the jokes subreddit which will be used for text generation. The first task is to train the RNN model to classify IMDB ratings into positive or negative based on the text content. The second task involves training an RNN model to generate text using the second dataset.

Section 2: Model Description


Classification Model
Input Word index sequence
Embedding Layer input : 40002 output : 100
Bid-LSTM input: 100 output : 200
2 stack LSTM
Bid-LSTM input: 200 output: 200
Length dim:200
Hidden dim:200
merge
Linear input 400 output 200
ReLu
Linear input : 200 output : VocabSize
Sigmoid
Output
BCELoss
0 or 1
Generation Train
Input Word index
Input Previous output hidden
Next Input
Embedding Layer input : VocabSize output : 100
Hidden Output
GRU input: 100 output : 200
2 stack GRU
GRU input: 100 output : 200
Linear input 200 output VocabSize
OutPut
SoftMax
CrossEntropy
Choose 1 Vocab

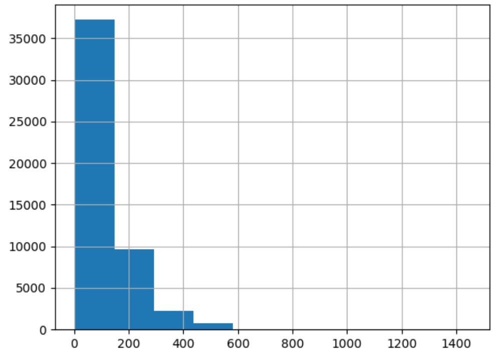
Click the picture to get the source image

Section 3: Experiment Settings

3.1 Dataset Description

IMDB Dataset:

This IMDB dataset was collected from IMDB, a movie review website. It is divided into test set and training set, with a total of 50,000 data. The training set and the data set each have 25,000, and the positive and negative sets each have 12,500. So there are 12500*4=50000 data in total.
Here is a plot of the length distribution of each review in the dataset. I set a maximum of 500 characters as a limit. More than 500 words will be deleted, and less than 500 words will be filled with <pad>.



Joke Dataset:

This dataset consists of 1622 jokes collected from the Jokes subreddit on Reddit. The jokes are typically presented in text form. The topics of the jokes are highly diverse, including puns, one-liners.

3.2 Detailed Experimental Setups

IMDB:

PreTrain: No pretrain
Depth: 1Emebdding, 2LSTM(bidirectional, stackLSTM), 2Fc
Layer dimensions: 40002 -> 100(Glove dimensions) -> 400(bid LSTM output 200, merge lstm output and hidden) -> 200(FC layer) -> 1(positive or negative)
Learning rate: 1e-2
Test input max length: 500
VocabSize: 40000 + 2
Epoch: 10
Batch: Mini-Batch(100)
Loss function: BCELoss (Here is the binary classification, so I chose to use BCELoss as my loss function.)
Opimizer: Adam

Jokes:

Depth: 1Emebdding, 2GRU (stackGRU), 1Fc
Layer dimensions: VocabSize -> 100 -> 200(GRU output) -> VocabSize (To choose what vocab)
Learning rate: 1e-3
Epoch: 50

Batch: Mini-Batch(1) My training method here is to input each word into the neural network. Pass the hidden output of the previous word into the neural network as the input of the next word. Add the Loss of each input, and after processing a whole sentence, backword the loss. So I thought, my batch should be 1. Because every time I finish processing 1 sentence, I perform backword.
Loss function: CrossEntropy (Here is multiclass classification, to make the neural network choose the next word, and the number of words exceeds two or more. So choose cross entropy loss as my loss function.)
Opimizer: Adam

3.3 Evaluation Metrics

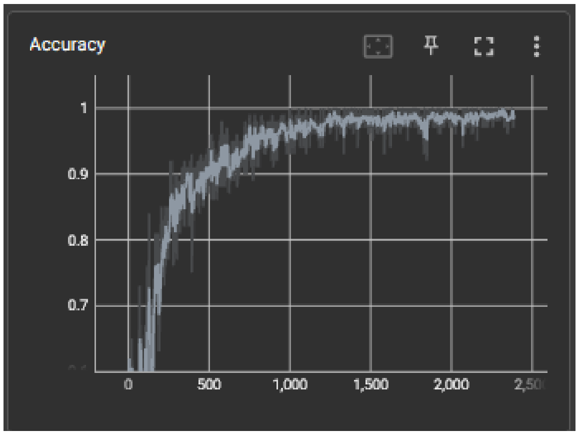
For first task, I will use Accuracy, Precision, Recall, F1 score as our evaluation metrics.
For second task, I'll offer two metrics.
1. Some generated cases
2. During training, the accuracy score of the model.

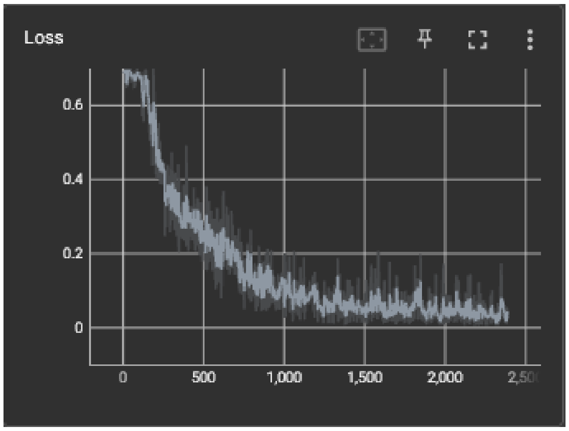
3.4 Source Code

Execute “script/LoadModel_Generate_Sentence.py” to load the model.
https://drive.google.com/file/d/16EChFBqIRkftAoQE2ycwjJPrEz4VE-_H/view?usp=share_link

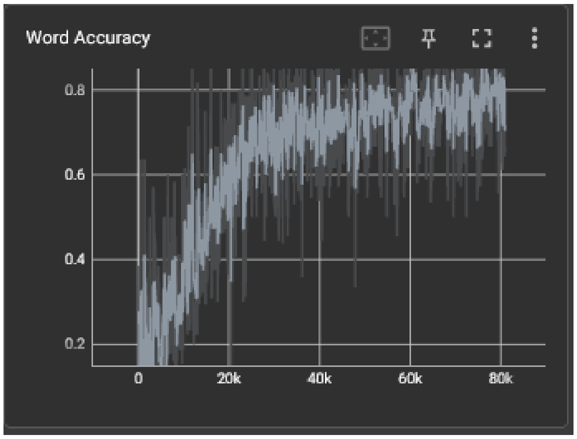
3.5 Training Convergence Plot

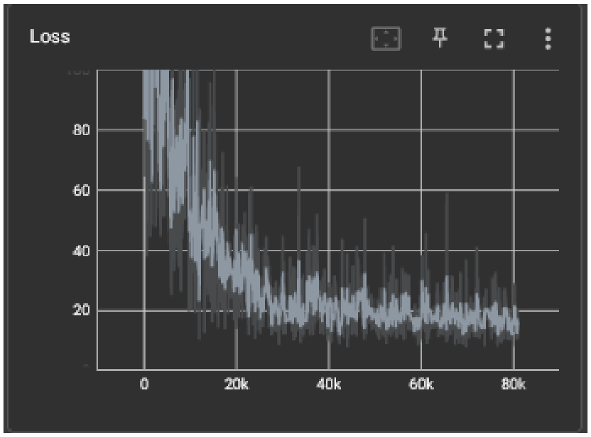
IMDB:





Jokes:





It can be clearly seen here that the accuracy rate and loss are gradually converging.

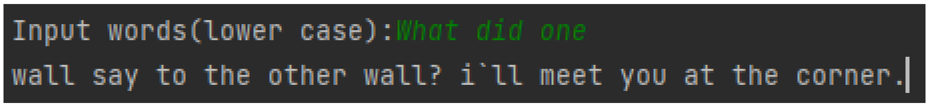
3.6 Model Performance

IMDB:

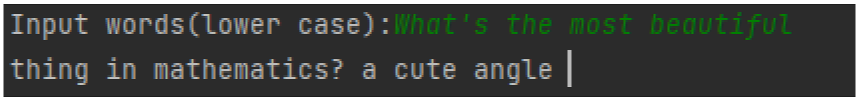
TrainSet:
Accuracy-Score: 0.98792
Precision-Score: 0.9879200499630132
Recall-Score: 0.98792
F1-Score: 0.9879199996907521
TestSet:
Accuracy-Score: 0.83792
Precision-Score: 0.8380425477252694
Recall-Score: 0.83792
F1-Score: 0.8379053092933433

Jokes:

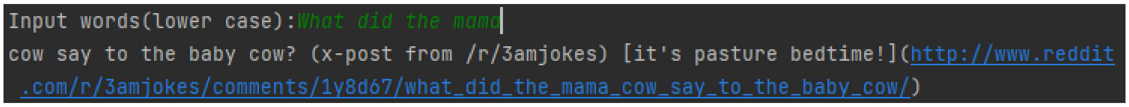
TrainSet:
Accuracy-Score: 0.93628



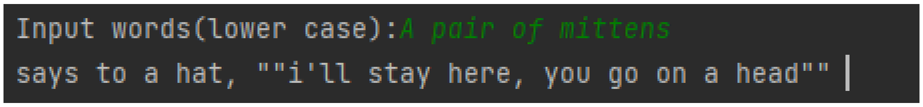
```
Input words(lower case):What did one
wall say to the other wall? i`ll meet you at the corner.
```



```
Input words(lower case):What's the most beautiful
thing in mathematics? a cute angle
```

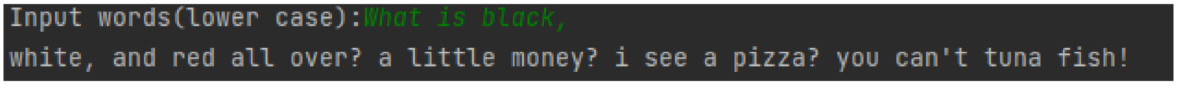


```
Input words(lower case):What did the mama
cow say to the baby cow? (x-post from /r/3amjokes) [it's pasture bedtime!](http://www.reddit
 .com/r/3amjokes/comments/1y8d67/what_did_the_mama_cow_say_to_the_baby_cow/)
```



```
Input words(lower case):A pair of mittens
says to a hat, ""i'll stay here, you go on a head""
```

But sometimes, it is still impossible to write a logical sentence.

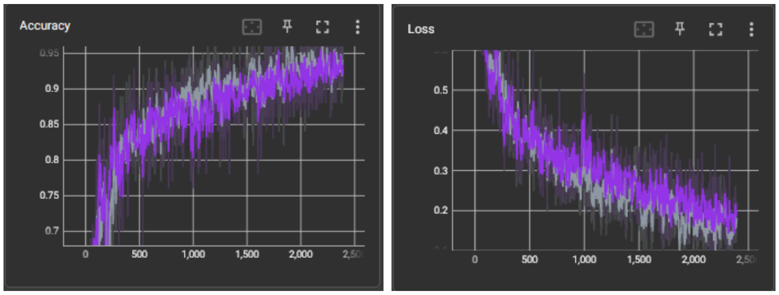


```
Input words(lower case):What is black,
white, and red all over? a little money? i see a pizza? you can't tuna fish!
```

3.7 Ablation Studies

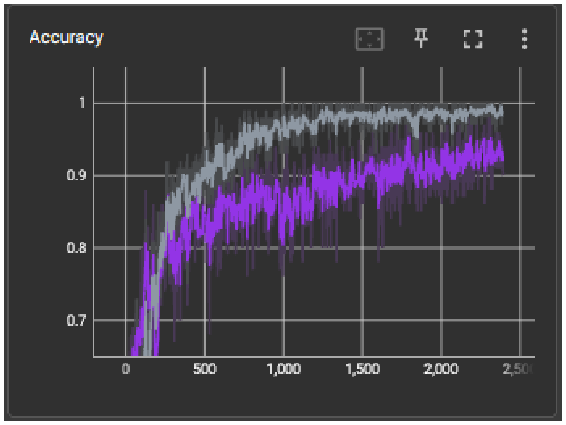
IMDB:

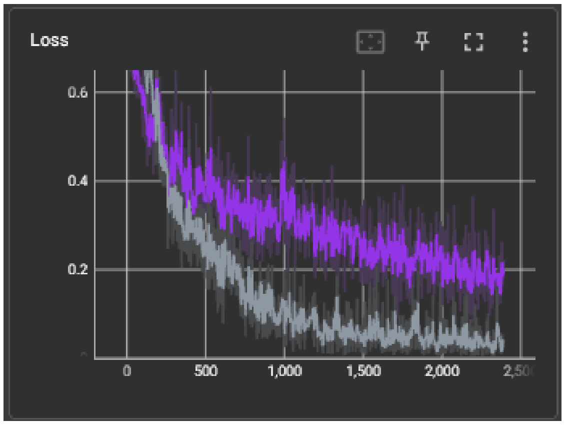
In the first task, I tried many different VocabSize and the impact of maxLen. When VocabSize =20000 maxLen=300, no obvious difference was found. Even better performance on the training set.
Gray is Vocab=20000 maxLen=300
Purple is Vocab=40000 maxLen=500



I tried to use the Glove to help me train the imdb dataset. But Glove didn't give me better results under the same conditions.

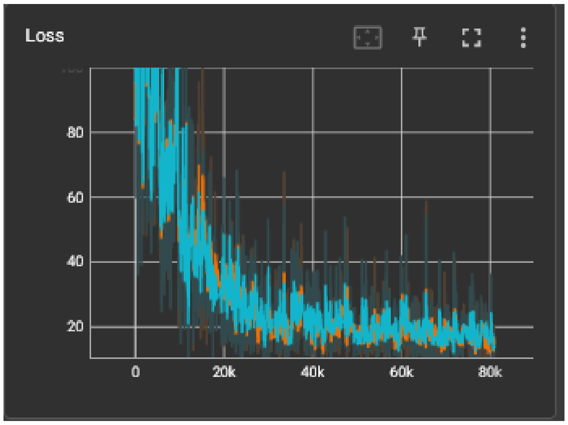
TrainSet: Accuracy-Score: 0.93628 **TestSet:**Accuracy-Score: 0.82508
Purple is Glove one.

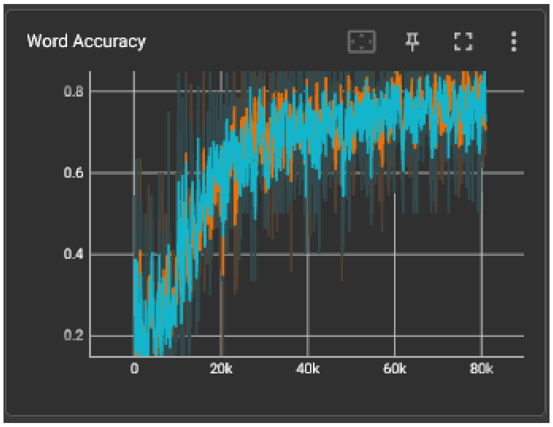


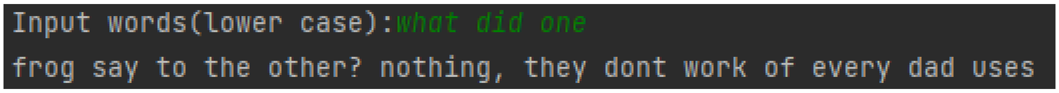


Jokes

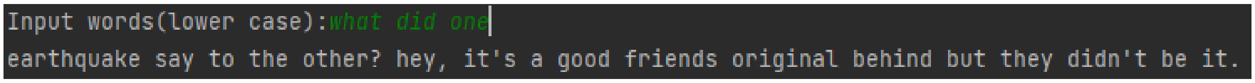
There doesn't seem to be much difference between the 1-layer GRU and the 2-layer GRU. Maybe my training method is wrong.



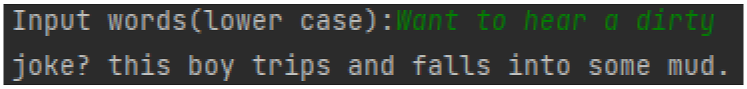




```
Input words(lower case):what did one
frog say to the other? nothing, they dont work of every dad uses
```



```
Input words(lower case):what did one
earthquake say to the other? hey, it's a good friends original behind but they didn't be it.
```



```
Input words(lower case):Want to hear a dirty
joke? this boy trips and falls into some mud.
```

Sometimes good sentences can be given, and sometimes there is still no logic.