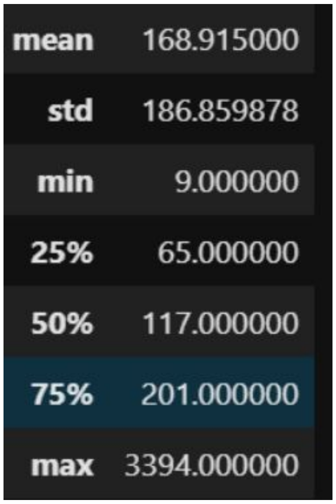
Your Name : AN YOUNG PILL (Epsilon)

Your Email address : kokomong1316@gmail.com

1. Training Set Construction (5 pts)

Construct the training set for the amazon review dataset as instructed and report the following statistics.

Statistics	
the total number of unique words in T	22764
the total number of training examples in T	2000
the ratio of positive examples to negative examples in T	1:1
the average length of document in T	168.915
the max length of document in T	3394



```
mean   168.915000
 std   186.859878
 min     9.000000
 25%    65.000000
 50%   117.000000
 75%   201.000000
 max  3394.000000
```

<fig 1.1 statistics of word counts in document>

2. Performance of deep neural network for classification (20 pts)

Suggested hyperparameters:

1. Data processing:

    a. Word embedding dimension: 100
    b. Word Index: keep the most frequent 10k words
2. CNN
    a. Network: Word embedding lookup layer -> 1D CNN layer -> fully connected layer -> output prediction
    b. Number of filters: 100
    c. Filter length: 3
    d. CNN Activation: Relu
    e. Fully connected layer dimension 100, activation: None (i.e. this layer is linear)
3. RNN:
    a. Network: Word embedding lookup layer -> LSTM layer -> fully connected layer(on the hidden state of the last LSTM cell) -> output prediction
    b. Hidden dimension for LSTM cell: 100
    c. Activation for LSTM cell: tanh
    d. Fully connected layer dimension 100, activation: None (i.e. this layer is linear)

(when the validation loss is the lowest)

	Accuracy	Training time(in seconds)
RNN w/o pre trained embedding	0.759000(epoch:02,val_loss:0.5232)	1.23
RNN w/ pre trained embedding	0.858000(epoch:16,val_loss:0.3707)	0.9558
CNN w/o pre trained embedding	0.533(epoch:3,val_loss:0.687)	0.6975
CNN w/ pre trained embedding	0.5316(epoch:20,val_loss:6826)	0.4283

pre-trained emb +cnn + lstm	0.913500(epoch:40,0.2823)	about 1 second

	mean of training time about 100 epochs (in seconds)
RNN w/o pre trained embedding	1.23
RNN w/ pre trained embedding	0.9810
CNN w/o pre trained embedding	0.6813
CNN w/ pre trained embedding	0.4255

3. Training behavior (10 pts)

Plot the training/testing objective, training/testing accuracy over time for the 4 model combinations (correspond to 4 rows in the above table). In other word, there should be 2*4=8 graphs in total, each of which contains two curves (training and testing).

RNN w/o pretrained embedding

- training/testing objective over time
- training/testing accuracy over time

RNN w/ pretrained embedding

- training/testing objective over time
- training/testing accuracy over time

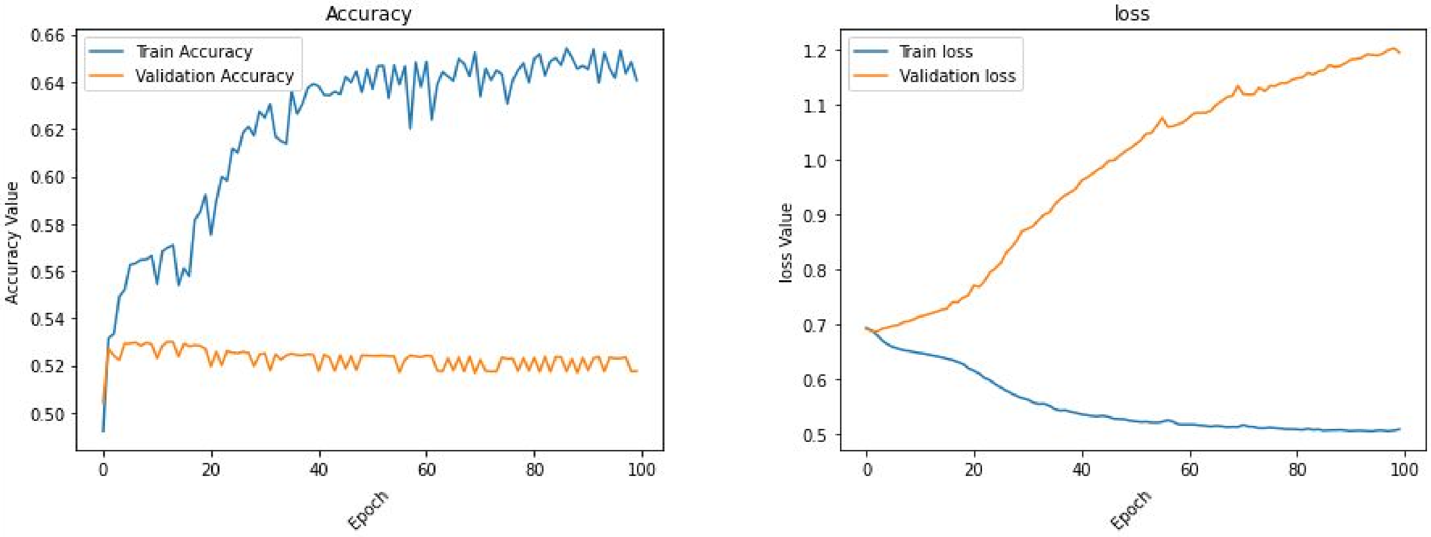
CNN w/o pretrained embedding

- training/testing objective over time
- training/testing accuracy over time

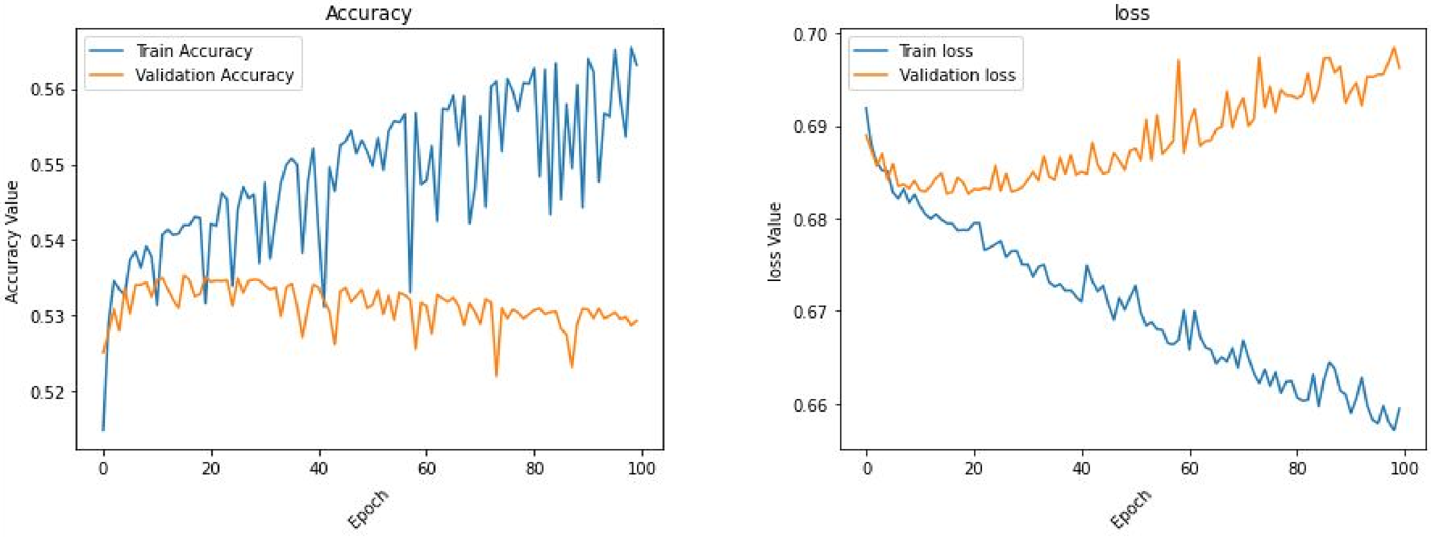
CNN w/ pretrained embedding

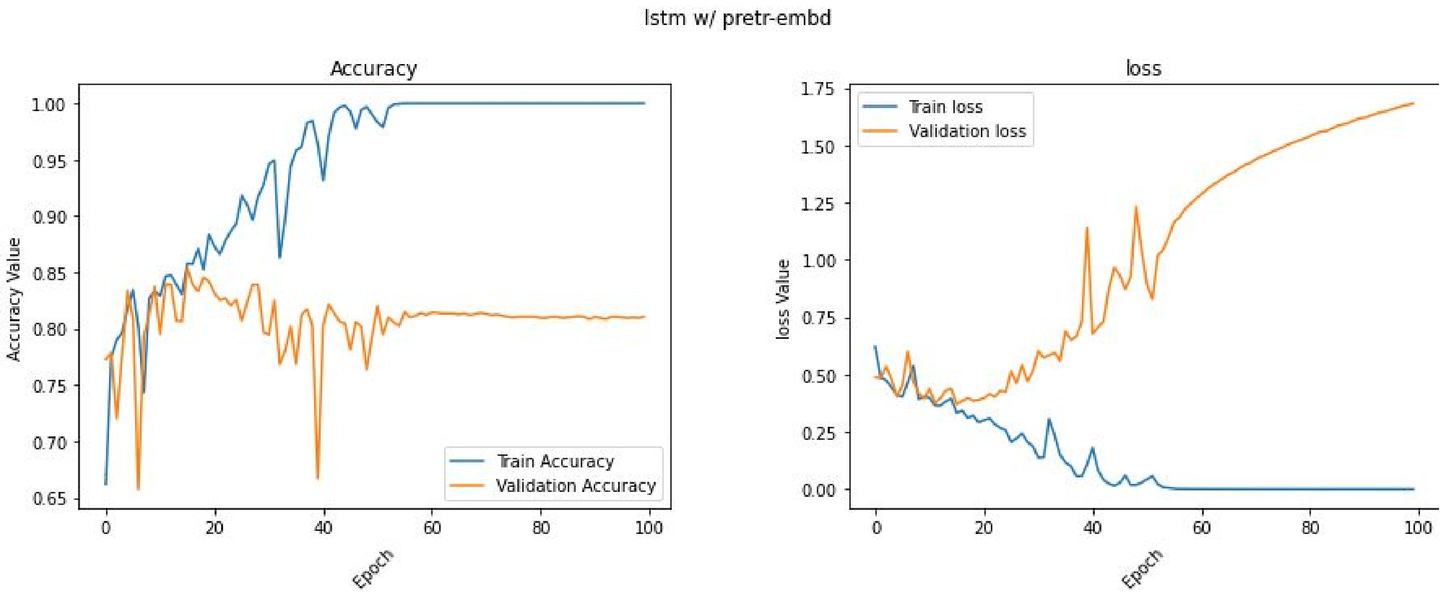
- training/testing objective over time
- training/testing accuracy over time

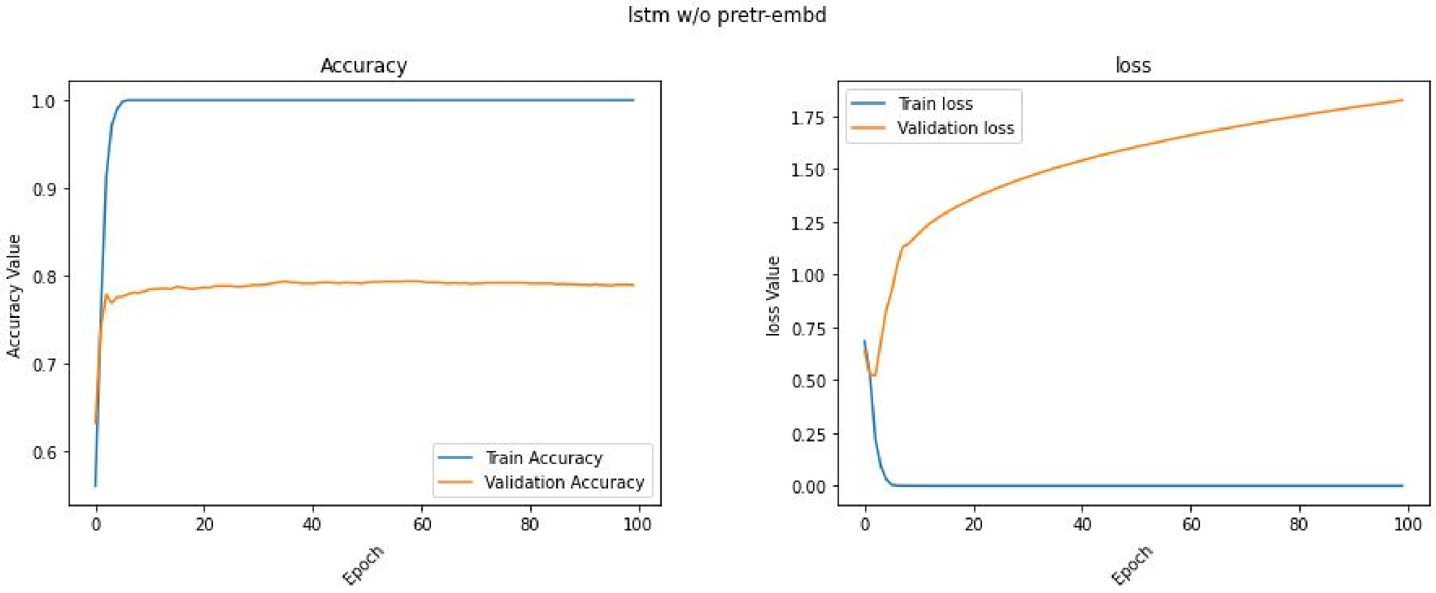
cnn w/o pretr-embd



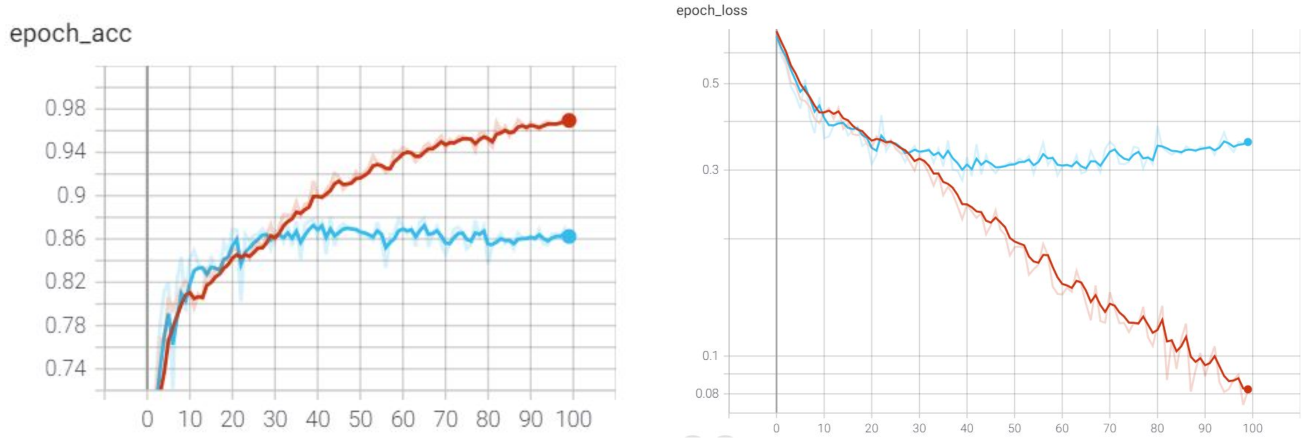
cnn w/ pretr-embd



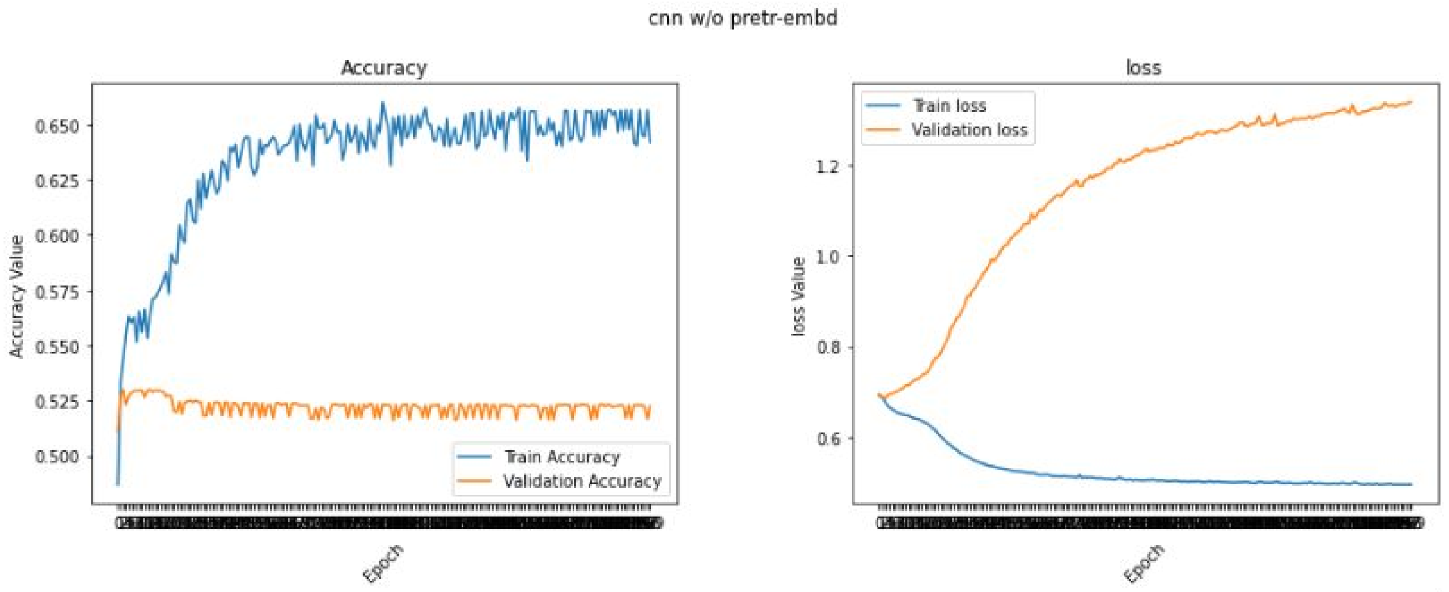


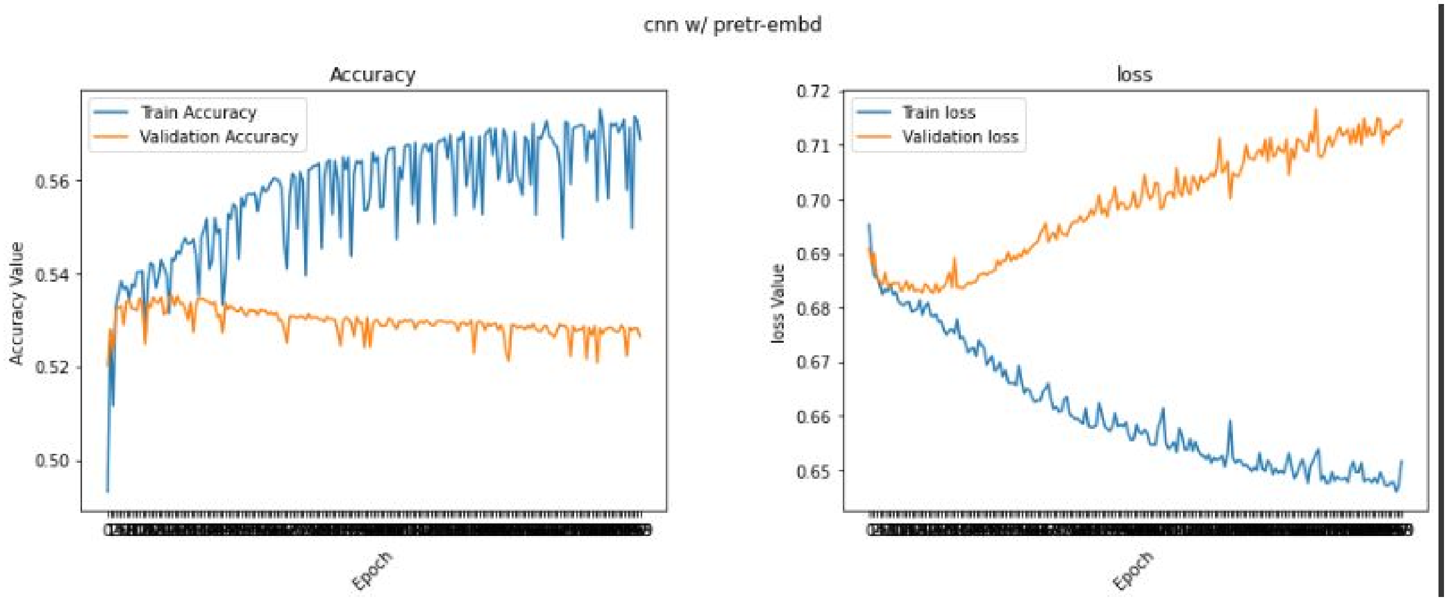


embedding+cnn+lstm



<fig 2.0 epoch 100>

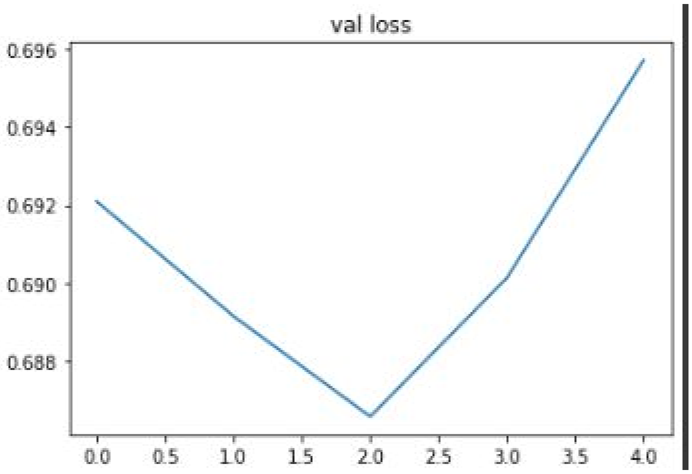




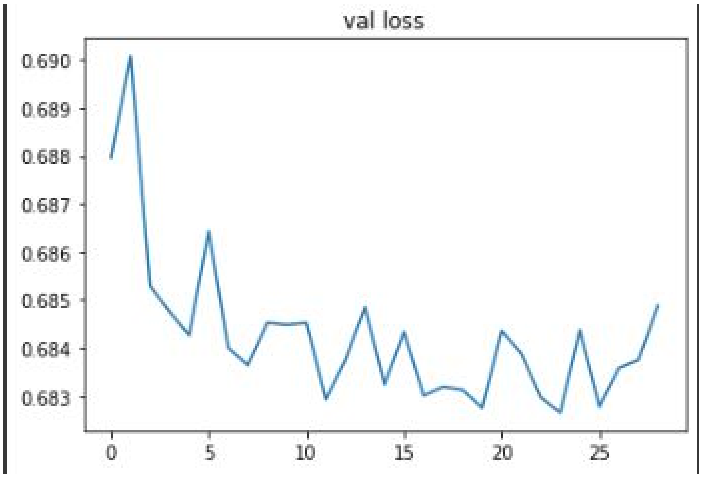
<fig 2.1 epoch 200>

<figs 2.2 each validation_loss zoom in> ⬇

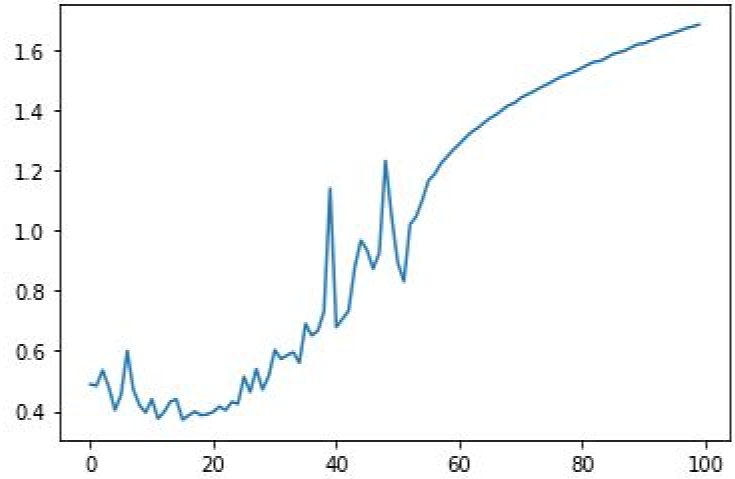
cnn w/o pre trained embedding



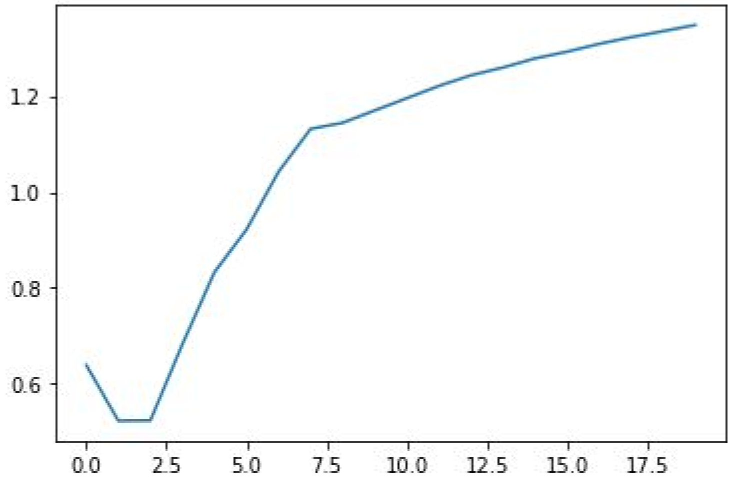
cnn w/ pre trained embedding



lstm w/ pre trained embedding



lstm w/o pre trained embedding



4. Analysis of results (10 pts)

Discuss the complete set of experimental results, comparing the algorithms to each other. Discuss your observations about the various algorithms, i.e., differences in how they performed, different parameters, what worked well and didn't, patterns/trends you observed across the set of experiments, etc. Try to explain why certain algorithms or approaches behaved the way they did.

- In this dataset, Lstm model performance is better than CNNs. but , training time is longer a bit

  - Q : In performance, why Lstm is better than Cnn .

  - A: Lstm is the advanced version of RNN .RNN is appropriate to sequential data.

Lstm = RNN +(solving Long-term dependency)

On the other hand, CNN layer (like convolution layer , max pooling) is for extracting important features in positional data like Image data.

Considering these, RNN family models are better in solving NLP problems like this.

- In this case , Using pre-trained embedding is better than not.

  - Q : which one is better [training Embedding vs Pre-trained Embedding]

  - A : It's depending on what the problem is.

In this article , says there might not be a universally correct answer to that question that works in every scenario

- In performance, a mixed model of lstm and convolution layer is better than both.

5. The software implementation (5 pts)

Add detailed descriptions about software implementation & data preprocessing, including:

1. A description of what you did to preprocess the dataset to make your implementations easier or more efficient.

2. A description of major data structures (if any); any programming tools or libraries that you used;

programming : python ,data preprocessing & Deep learning library : pandas,tensorflow, keras

3. Strengths and weaknesses of your design, and any problems that your system encountered;

In point of programming design pattern (like software engineering), my program is bad.

I didn’t capsuled my code to class. I just focused on building a deep learning model.

If I have enough time and specific programming requirement to provide someone, I can make better.

=========

My thought of this task

Thank you for giving me this task.

This makes me interested and motivated to learn more.

- How to do better NLP preprocessing analysis
- How to do AutoML(DL) and its principle (making that following model , i really felt i have to do some task automatically in designing models.
- To improve performance ,what kind of methods there are in general.