DS-GA 1011 NLP Assignment 1
Zihao Zhao (zz913)

Bag of N-Grams Document Classification, Hyperparameter Exploration and Analysis

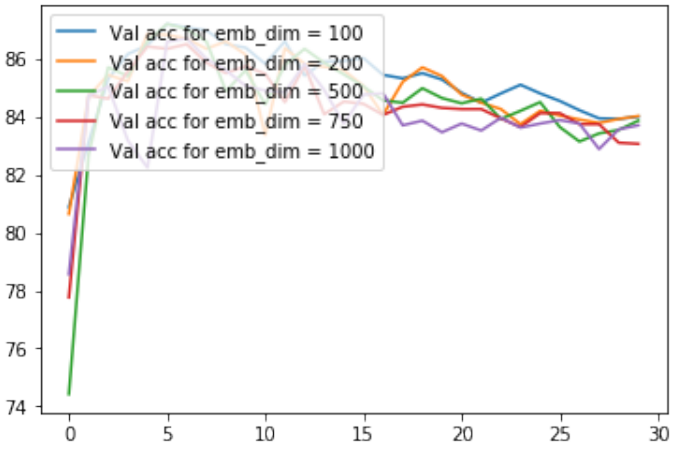


Figure 1: Validation curve for different embedding dimensions

From the curve we see that the validation accuracy decrease as the embedding dimension increases, this is probably because longer embedding vectors don't add enough information and too many parameters may lead to overfitting.

So we choose embedding size = 100.

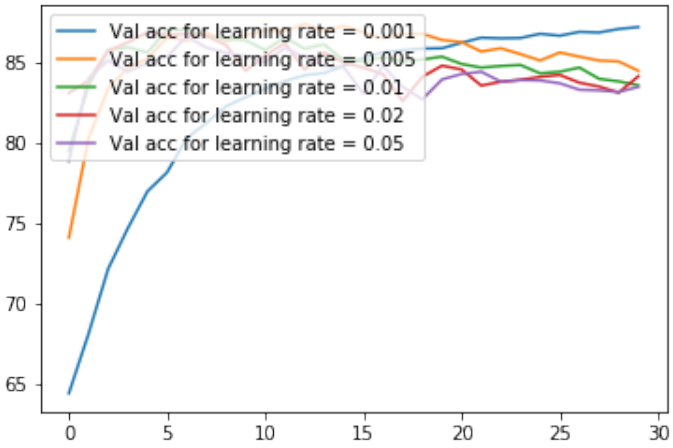


Figure 2: Validation curve for different learning rates

From the curve we see that although higher learning rate performs better at the beginning, but the lowest learning rate works the best when the epoch increases. Lower learning rate also makes a more stable curve with increasing validation accuracy. This is probably because when learning rate is large, the algorithm is likely to frequently miss the optimized solution.

So we choose learning rate = 0.001.

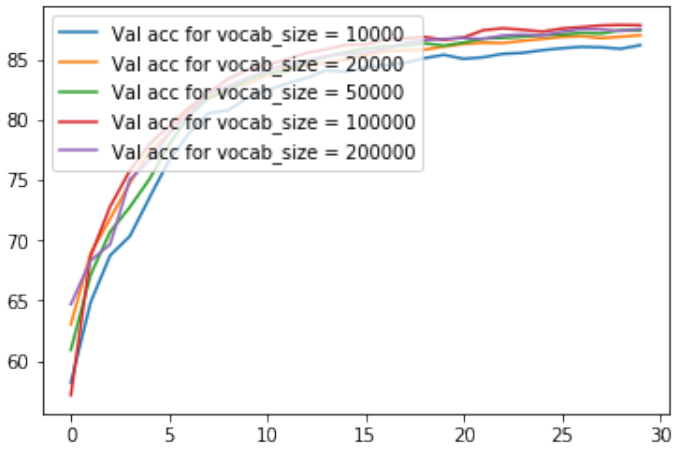
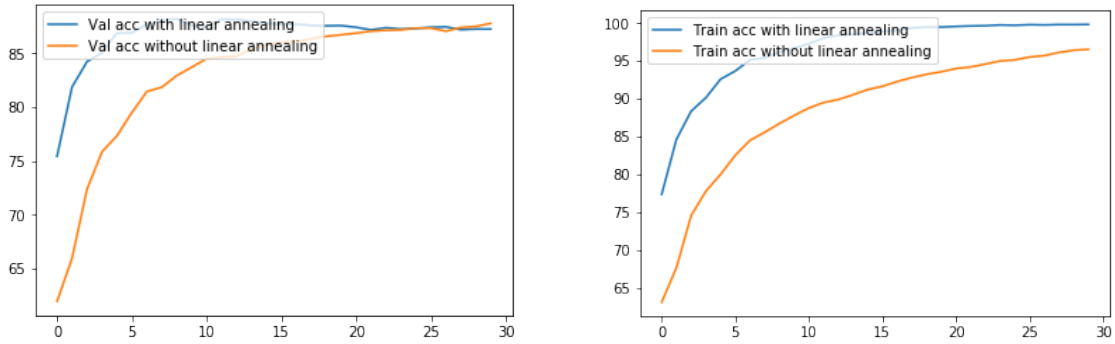


Figure 3: Validation curve for different vocabulary sizes

From the curve we see that the validation accuracy increases as the vocabulary size increases from 10,000 to 100,000, but with vocabulary size = 200,000, the accuracy starts to decrease. This shows that we need the vocabulary large enough to contain enough key words, but adding too much words does not help, because those are not so frequent, and has little meaning to the model.

So we choose vocabulary size = 100,000



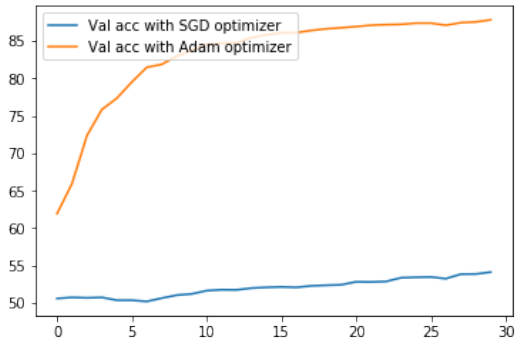
(a) Validation curve for linear annealing

(b) Training curve for linear annealing

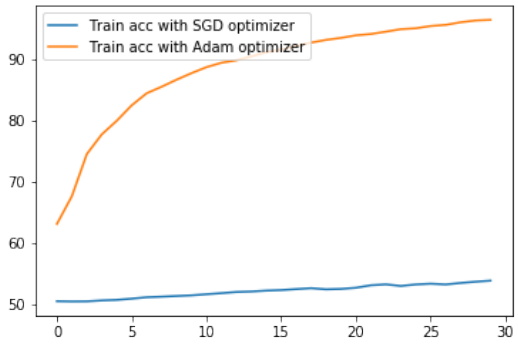
Figure 4: Test for linear annealing

From the curve we see that linear annealing has much better performance at the beginning of training, both in validation curve and the training curve. However, as the epoch increases, the optimizer without linear annealing could get the same and even higher validation accuracy. This shows that linear annealing has better training accuracy but not the validation accuracy. Probably it has the risk of overfitting.

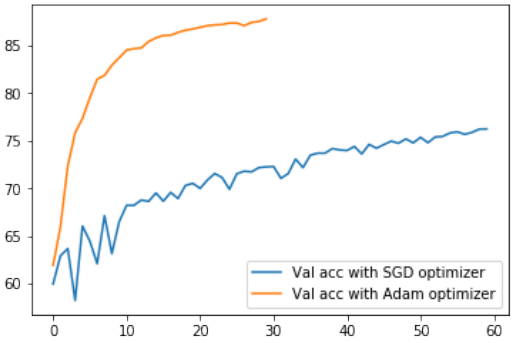
So we choose not to use linear annealing.



(a) Validation curve for different optimizer



(b) Training curve for different optimizer

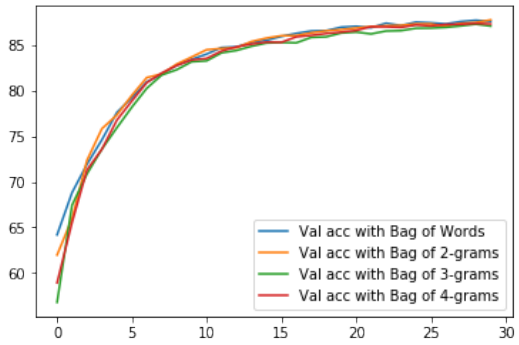


(c) Validation curve for different optimizer

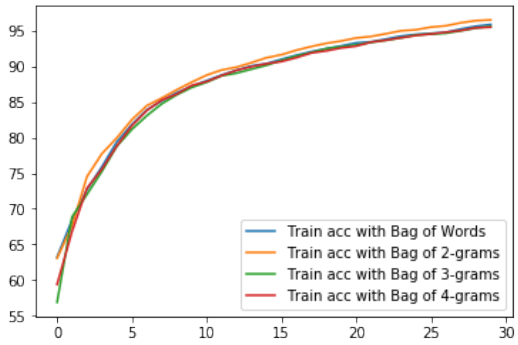
Figure 5: Test for different optimizer

From the curve above, we clearly see that SGD works much worse than Adam in this sample. Considering the effect of learning rate and time, we try a larger learning rate and use more epochs. In graph (c), although the training curve is better with these parameters, the SGD still does not work as well as Adam.

So we choose Adam optimizer.



(a) Validation curve for n-grams

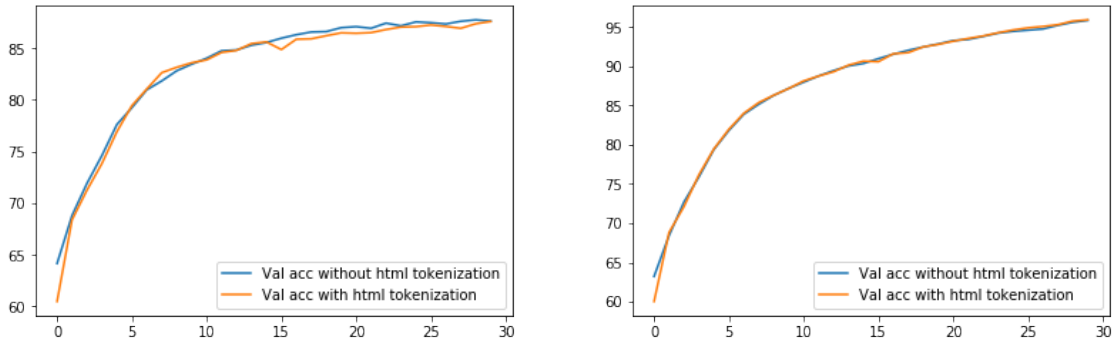


(b) Training curve for n-grams

Figure 6: Test for different n in n-grams

From the validation curve above, we cannot tell which of n-grams (n=1,2,3,4) is better, since all of them have almost the same performance. From the training curve, we can see that 2-grams has a slightly higher training accuracy. Probably this is because we have fixed the maximum length of sentence, and it is too short that the tokenized data does not include much more information when n increases.

So far, we just keep n=2.



(a) Validation curve

(b) Training curve

Figure 7: Test for different tokenization method

From the curve, we can see that when we drop the html tag in tokenization the validation accuracy will become a little higher. This means that the tag does not give more information about the sentiment of a comment, and it does not matter a lot whether we contain the tags in training.

Table 1: Parameter Tuning Results

emb_ dim	learning_ rate	vocab_ size	linear annealing	optimizer	n (n-gram)	html	Val acc
200	0.01	20,000	No	Adam	2	No	83.76
100	0.01	20,000	No	Adam	2	No	83.86
100	0.001	20,000	No	Adam	2	No	87.16
100	0.001	100,000	No	Adam	2	No	87.97
100	0.001	100,000	Yes	Adam	2	No	87.18
100	0.001	100,000	No	SGD	2	No	76.24
100	0.001	100,000	No	Adam	1	No	87.76
100	0.001	100,000	No	Adam	2	Yes	87.6

From the table, we can conclude that the best configuration for this model is the circled one:

embedding size = 100, learning rate = 0.001, vocabulary size = 100,000, no linear annealing, Adam optimizer, using 2-grams, drop the html tag

With the best configuration, we could report:

Test Accuracy: 85.76

3 correct predictions:

"i do n't know what it is i find so endearing about this film but the first time i saw it i wanted to see how it ended ..."

"this documentary now available free on video google com is a fantastic demonstration of the power of ordinary people to overcome injustice ..."

"and i really mean that i caught it last night on vh1 and i was not expecting it to be so good this is now one of my favorites ..."

3 incorrect predictions:

"there is great detail in a bug 's life everything is covered the film looks great and the animation is sometimes jaw dropping the film is n't too terribly ..."

"i 've probably been spoilt by having firstly seen the 1973 version with michael ¡unk¿ and ¡unk¿ cusack so the 1983 adaptation is such a disappointment ..."

"so neighbor was killing neighbor reminds me of iraq as i watched the american flag 50 stars in ¡unk¿ being dragged behind the horse ..."

Due to the page limit, the complete paragraph of predictions can be found at the end of trainer.ipynb file. Link to the github page: `https://github.com/cdsherry/DS-GA1011-NLP-hw1.git`