Natural Language Processing - IMDB Movie Review

	Description	Hyperparameters	Number of Epochs	Training Loss	Training Accuracy	Test Accuracy	Comments	
Part 1a	Given model - Word Embedding Layer + Mean Pooling + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=500	20	0.025532	0.991	0.9216	The model trained with given hyperparameters returns a rather good result. I trained it with 20 epochs so the accuracy is better than expected.	should be around 86-88%
	Custom 1	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=1000	20	0.021467	0.993	0.9237	From the results, we've seen that a higher vocabulary size will bring a better performance. Theoretically, a larger vocabulary size takes more words into account so that less information in each sentence will be omitted, and hence our prediction is more accurate.	Describe more about the model/results such as why certain hyperparamters were chosen or the effect it had on the accuracy/training time/overfitting/etc.
	Custom 2	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=15000, HiddenUnits=500	20	0.010972	0.997	0.9244	From the results, we've seen that more hidden units will bring a better performance. Theoretically, more hidden units can also help us capture more types of information, hence also increase the accuracy.	
	Custom 3	SGD optimizer with LR=0.1, BatchSize=200, VocabularySize=8000, HiddenUnits=500	20	0.204234	0.921	0.8662	From the results, we've seen that ADAM optimizer works better than SGD if learning rates are properly adjusted.	
Part 1b	Given Model - Word Embedding Layer + Mean Pooling + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=100000, HiddenUnits=500	100	0.083506	0.97	0.9099	The model trained with given hyperparameters returns a rather good result. I trained it with 100 epochs so the accuracy is better than expected.	Should be around ~81-87%
	Custom 1	ADAM optimizer with LR=0.01, BatchSize=200, VocabularySize=100000, HiddenUnits=500	100	0.093555	0.967	0.9084	From the results, we've seen that learning rate 0.001 works a little better than 0.01. Theoretically, the learning rate should be adjusted properly so that it won't be either too slow to converge, or too fast to fail convergence.	
	Custom 2	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=100000, HiddenUnits=1000	100	0.056714	0.982	0.9131	From the results, we've seen that more hidden units will bring a better performance. Theoretically, more hidden units can also help us capture more types of information, hence also increase the accuracy.	
	Custom 3	SGD optimizer with LR=0.01, BatchSize=200, VocabularySize=100000, HiddenUnits=500	100	0.157885	0.939	0.8877	From the results, we've seen that ADAM optimizer works better than SGD if learning rates are properly adjusted, in the case with GloVe.	
Part 2a	Given Model - (write description)	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=500	20	0.082083	0.971	0.8705	The model trained with given hyperparameters returns a rather good result as expected.	~87%
	Custom 1	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=1000	20	0.057841	0.98	0.8627	From the results, we've seen that more hidden units will bring a better performance on training set, but a little worse on testing set. This implies that too many hidden units may cause our model to be a bit overfitting.	
	Custom 2	ADAM optimizer with LR=0.01, BatchSize=200, VocabularySize=8000, HiddenUnits=500	20	0.273012	0.888	0.8107	From the results, we've seen that learning rate 0.001 works much better than 0.01 compared to part 1a and 1b. This maybe because for more complicated models, different learning rates are more effective.	
	Custom 3	SGD optimizer with LR=0.1, BatchSize=200, VocabularySize=8000, HiddenUnits=500	20	0.267985	0.888	0.783	From the results, we've seen that ADAM optimizer works better than SGD if learning rates are properly adjusted.	
Part 2b	Given Model - Word Embedding Layer + Mean Pooling + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=100000, HiddenUnits=500, SequenceLength=100	30	0.152165	0.94	0.8751	The model trained with given hyperparameters returns a rather good result, but not as good as expected. The reason may be that more epochs are needed, or the sequence length should increase.	~91%
	Custom 1	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=100000, HiddenUnits=1000, SequenceLength=100	30	0.118245	0.957	0.874	From the results, we've seen that more hidden units will bring a better performance on training set, but a little worse on testing set. This implies that too many hidden units may cause our model to be a bit overfitting.	
	Custom 2	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=100000, HiddenUnits=500, SequenceLength=200	30	0.06071	0.978	0.8622	From the results, we've seen that a larger sequence length leads to a better performance on training set, but worse on testing set. This implies that a larger sequence length causes our model to be overfitting, because not all parts in one sentence can reflect its true sentiment, but a very large sequence length will mislead the model to "think" in this way and hence overfits on the training set.	
	Custom 3	SGD optimizer with LR=0.01, BatchSize=200, VocabularySize=100000, HiddenUnits=500, SequenceLength=100	30	0.43502	0.798	0.6891	In this case, ADAM optimizer works much better than SGD if learning rates are properly adjusted. This shows the advantage of ADAM over SGD in NLP tasks.	
Part 3a	Given Model							

Part 3b	Generated Review	Temperature=1.0	this is a film that keeps you laughing and cheering for your own reason to	
	Generated Review	Temperature 0.5	i was so excited that the movie was over , i was very disappointed . it was n't	
	Generated Review			
Part 3c	Given Model			~91%+
	Custom 1			
	Custom 2			
	Custom 3			