

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	6	0.1464	94.38%	87.20%	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 1 - Same as the given model	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=1000	6	0.1251	95.26%	86.07%	The number of hidden units is increased to 1000 to make the model overfit. The training accuracy increases a little bit, and the test accuracy is decreased by 1%. The effect on overfitting is not very obvious, but we can still see that as the training accuracy increases, the test accuracy is decreasing. Besides, increasing the number of hidden units increases the training time greatly, from 17 seconds to 27 seconds per epoch.
	Custom 2 - Same as the given model	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=10	6	0.4256	81.44%	83.24%	The number of hidden units is decreased to 10 to make the model underfit. The test accuracy and training accuracy decrease a lot and the training loss is still high, meaning that the model is underfitting.
	Custom 3						
Part 1b	Given Model - Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=100000, HiddenUnits=500	6	0.2984	87.54%	85.72%	The model doesn't seem to suffer from overfitting as the training accuracy and test accuracy are quite close to each other.
	Custom 1 - Same as the given model	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=100000, HiddenUnits=5000	6	0.2887	88.00%	84.52%	The number of hidden units is increased to 5000 to make the model overfit. The training accuracy increases a little bit, and the test accuracy is decreased by 1%. The effect on overfitting is not very obvious. The training time doesn't change much since it's already very small.
	Custom 2 - Same as the given model	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=10	6	0.3694	84.04%	85.34%	The number of hidden units is decreased to 10 to make the model underfit. The training accuracy decreases by 3%, while the test accuracy doesn't change much. Despite its purpose, this change seems to alleviate the overfitting problem of the original model. A possible reason is that the function we want to approximate using our network is not that complex so making the model simpler won't hurt the performance.
	Custom 3						
Part 2a	Given Model - Word Embedding Layer + LSTM Layer + Max Pooling + Fully Connected Layer + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=500, SequenceLengthOfTrain=100, SequenceLengthOfTest=400	20	0.1015	96.37%	87.69%	The model does not perform very well because even though the training accuracy reaches 96%, the test accuracy is just 87%.
	Custom 1 - Same as the given model	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=500, SequenceLengthOfTrain=400, SequenceLengthOfTest=400	20	0.0175	99.43%	79.07%	The sequence length for training is increased from 100 to 400. The model is now overfitting. Training accuracy is approaching 100% while test accuracy is only 79.07%, which is worse than the given model.
	Custom 2 - Same as the given model	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=100, SequenceLengthOfTrain=100, SequenceLengthOfTest=400	20	0.2251	91.00%	84.49%	The number of hidden units is decreased to 100 to make the model underfit. Now the training accuracy is smaller than the given model. Interestingly, the test accuracy doesn't decrease much. A potential explanation is that the given model is overfitting so decreasing the number of hidden units is actually alleviating this problem.
	Custom 3						
Part 2b	Given Model - LSTM Layer + Max Pooling + Fully Connected Layer + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=100000, HiddenUnits=500, SequenceLengthOfTrain=100, SequenceLengthOfTest=400	20	0.2099	91.52%	88.33%	The model doesn't seem to suffer from overfitting as the training accuracy and test accuracy are quite close to each other. However, it might be a little bit underfitting because during the training process, the test accuracy is still increasing.
	Custom 1 - Same as the given model	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=100000, HiddenUnits=500, SequenceLengthOfTrain=400, SequenceLengthOfTest=400	20	0.0707	97.45%	82.48%	The sequence length for training is increased from 100 to 400. The model is now overfitting. Training accuracy is approaching 100% while test accuracy is only 82.48%, which is worse than the given model.
	Custom 2 - Same as the given model	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=100000, HiddenUnits=100, SequenceLengthOfTrain=100, SequenceLengthOfTest=400	20	0.3012	87.01%	86.18%	The number of hidden units is decreased to 100 to make the model underfit. Now the training accuracy is smaller than the given model. Interestingly, the test accuracy doesn't decrease much. A potential explanation is that the given model is overfitting so decreasing the number of hidden units is actually alleviating this problem.
	Custom 3						