

## Natural Language Processing - IMDB Movie Review

	Description	Hyperparameters	Epochs	Training	Training	Test Accuracy	Comments
				Loss	Accuracy		
Part 1a (without GloVe Features)	Given model - Word Embedding Layer + Mean Pooling + Fully Connected Layer + Batch normalization + Relu + Dropout Layer + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=500, Dropout rate=0.5	6	0.1403	94.60%	87.23%	Since the number of hidden unit is 500 which is relatively enough and the vocabulary size is 8000, the model can get an good accuracy.
	Overfit model - Word Embedding Layer + Mean Pooling + Fully Connected Layer + Batch normalization + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=5000	6	0.0792	97.16%	85.90%	Since dropout layer has been removed and hidden units in model is increased to 5000(ten times of the Given model), the model has more hidden units and is easier to get overfit in training set. And because the number of hidden units is increased, the training time is much longer.

Underfit model - Word Embedding Layer + Mean Pooling + Fully Connected Layer + Batch normalization + Relu + Dropout Layer + Output Layer		ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=80, HiddenUnits=500, Dropout rate=0.5	6	0.5724	70.70%	68.98%	I have tried the given model except that only 50 hidden units were used. However, The test accuracy can still get to ~80%. Then I decrease the vocabulary size. Now, even with 500 hidden units, the model's underfit situation becomes worse. So, when vocabulary size is large, it seems that slightly change the number of hidden units has little effect on test accuracy. In this case, the relatively small number of hidden units and vocabulary size leads to less training time.
Part 1b (with GloVe Features)	Given model – Fully Connected Layer + Batch normalization + Relu + Dropout Layer + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=10000 0, GloVe dimentions=300, HiddenUnits=500, Dropout rate=0.5	6	0.3057	87.14%	83.84%	With the GloVe features, the test accuracy becomes worse. From my training results, the training accuracy seems reaching an plateau(around 87.1%) after the 4 <sup>th</sup> training epoch. If we look at the training results of 1a, it keeps increasing when epoch goes by(overfitting). But the model in 1b is much simpler than 1a(without the embedding layer) and the test accuracy is affordable. And it also needs a much larger vocabulary to achieve this accuracy with GloVe features.

Part 2a (without GloVe Features)	Overfit model – Fully Connected Layer1 + Batch normalization + Fully Connected Layer2 + Batch normalization + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=10000 0, GloVe dimensions=300, HiddenUnits=5000	6	0.1982	91.84%	73.11%	The dropout layer is removed and a second fully connected layer is added to the model structure. Besides, hidden units is increased to 5000. So, the model overfits the training set and has a low accuracy on test set. In addition, the training time is much longer.
	Underfit model – Fully Connected Layer + Batch normalization + Dropout + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=1000, GloVe dimensions=300, HiddenUnits=50, Dropout rate = 0.5	6	0.4756	77.55%	77.51%	Because of the less number of hidden units(50) and smaller vocabulary size(1000), the model is underfitting.
	Given model - Word Embedding Layer + (LSTM Layer + Batch normalization + Dropout Layer) + Maxpooling1d + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, Sequence Length=100, HiddenUnits=500, Dropout rate=0.5	20	0.0872	96.98%	50-76.50%, 100-84.00%, 150-86.00%, 200-87.50%, 250-84.50%, 300-88.50%, 350-88.00%, 400-90.00%, 450-89.00%, 500-87.00%	The given model also seems overfitting.

Part 2b (with GloVe Features)	Test model - Word Embedding Layer + (LSTM Layer + Batch normalization + Dropout)x2 + Maxpooling1d + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, Sequence Length=100, HiddenUnits=500, Dropout rate1=0.5, Dropout rate2=0.3	20	0.0428	98.68%	50-70.00%, 100-81.50%, 150-81.50%, 200-91.00%, I choose 2 LSTM Layer to see 250-88.00%, whether adding more layer will 300-86.50%, increase the test accuracy. Although 350-85.00%, training accuracy and test accuracy 400-85.50%, on sequence length of 500, 450, 250, 450-91.00%, 200 are increased, test accuracy on 500-88.00% the other group in decreased.
	Underfit model - Word Embedding Layer + (LSTM Layer + Batch normalization + Dropout Layer) + Maxpooling1d + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=80, Sequence Length=100, HiddenUnits=500, Dropout rate=0.5	20	0.4179	80.67%	50-66.50%, 100-67.50%, 150-74.00%, 200-78.00%, 250-75.00%, 300-76.50%, 350-72.00%, 400-76.50%, Because of the less number of 450-77.00%, hidden units and smaller vocabulary 500-73.00% size, the model is underfitting.
	Given model - (LSTM Layer + Batch normalization + Dropout) + Max Pooling1d + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, Vocabulary_Size=10000, sequence_length=100, HiddenUnits=500, Dropout rate=0.5	20	0.0872	96.98%	50-83.00%, 100-82.50%, 150-85.50%, 200-88.00%, 250-89.50%, 300-90.50%, 350-91.50%, 400-93.00%, 450-92.50%, 500-90.00% The given model seems overfitting.

Test model - (LSTM Layer + Batch normalization + Dropout) + Max Pooling1d + Output Layer	ADAM optimizer with LR=0.001, BatchSize=400, Vocabulary_Size=1000 00, sequence_length=400, HiddenUnits=500, Dropout rate=0.5	10	0.1739	93.14%	50-77.00%, 100-84.50%, From the given model, test accuracy 150-92.00%, reach its maximum when sequence 200-88.00%, length is 400. So I set sequence 250-85.50%, length of 400 for this trail. It did 300-92.00%, increase the test accuracy when 350-91.00%, sequence length is 400. But the 400-94.50%, overfit situation seems getting 450-92.00%, worse due to the long sequence 500-92.50% length in training.
Underfit model - (LSTM Layer + Batch normalization + Dropout) + Max Pooling1d + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, Vocabulary_Size=100, sequence_length=100, HiddenUnits=50, Dropout rate=0.5	20	0.5833	69.24%	50-58.50%, 100-66.50%, 150-69.50%, 200-76.00%, 250-68.00%, 300-74.00%, 350-75.50%, 400-72.00%, Because of the less number of 450-75.00%, hidden units and smaller vocabulary 500-72.00% size, the model is underfitting.