

## Natural Language Processing - IMDB Movie Review

	Description	Hyperparameters	Number of Epochs	Training Loss	Training Accuracy	Test Accuracy	Comments
Part 1a	<b>Given model</b> - Word Embedding Layer + Mean Pooling + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=500	6	0.146	94.51%	87.23%	This is the default model suggested in the HW document. The training time for 500 hidden units is about 20 minutes for 6 epochs. The model achieved an accuracy of ~94% on training set and ~87% on test set, so clearly, it overfit the training dataset. Maybe by reducing the number of hidden units, will help reduce some over fitting.
	<b>Custom 1</b> - Word Embedding Layer + Mean Pooling + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=100, VocabularySize=8000, HiddenUnits=5000	6	0.45	95.92%	86.51%	I increased the number of hidden units in the network to analyse the behaviour of the model. The training time for 5000 hidden units is about 30 minutes for 6 epochs, which is more than the default model. Hence, increasing the number of hidden units increased the model training time. The model achieved an accuracy of ~96% on training set and ~86.5% on test set, so clearly, it overfit the training dataset even more by increasing the number of hidden units. Hence, we observe that increasing the number of hidden units increase training time and over fitting of the model. I also decrease the batch since I made a bigger model. Next, I will try smaller no. of hidden units to check if it reduces overfitting of the model.
	<b>Custom 2</b> - Word Embedding Layer + Mean Pooling + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=20	10	0.32	87.50%	86.20%	I decreased the number of hidden units in the network to 20 and increased no. of epochs to 10 to analyse the behaviour of the model. The training time for 20 hidden units is about 15 minutes for 10 epochs, which is less than the default model. Hence, decreasing the number of hidden units decreased the model training time. The model achieved an accuracy of ~87.5% on training set and ~86% on test set, so clearly, it performed almost the same on both the sets. Hence, we observe that decreasing the number of hidden units decreases training time and helped reduce the over fitting of the model.
	<b>Custom 3</b> - Word Embedding Layer + Mean Pooling + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=5	15	0.6	68.02%	73.12%	I further decreased the number of hidden units to 5 and increased the number of epochs to 15 in the network to analyse the behaviour of the model. The training time for 5 hidden units is about 15 minutes for 15 epochs, which is less than the default model. Hence, decreasing the number of hidden units decreased the model training time. The model achieved an accuracy of ~68% on training set and ~73% on test set, so clearly, it performs worse on the training dataset than the test set by reducing hidden units to very low. Hence, we observe that decreasing the number of hidden units decrease training time and it introduces underfitting of the model. The model performs worse on the training set than the test set. Hence, we can conclude that too large of hidden units overfit the model to training set and lesser number of units do not learn all features properly.

Part 1b	<b>Given model</b> - Mean GloVe features + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=100000, HiddenUnits=500	6	0.299	87.44%	85.92%	This is the default model suggested in the HW document. The training time for 500 hidden units is about 20 minutes for 6 epochs. The model achieved an accuracy of ~87% on training set and ~86% on test set. Notice that the accuracy with GloVe embedding is lower than the above model in Part 1a without GloVe embeddings. The training accuracy stops improving after 86% with GloVe embeddings.
	<b>Custom 1</b> - Mean GloVe features + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=100, VocabularySize=100000, HiddenUnits=5000	6	0.288	87.84%	84.88%	I increased the number of hidden units in the network to analyse the behaviour of the model. The training time for 5000 hidden units is about 30 minutes for 6 epochs, which is more than the default model. Hence, increasing the number of hidden units increased the model training time. The model achieved an accuracy of ~88% on training set and ~85% on test set, so, it overfit the training dataset a bit by increasing the number of hidden units. Hence, we observe that increasing the number of hidden units increases training time and the potential of over fitting of the model. But the accuracy even with higher no. of units did not exceed the accuracy of Bag of Words model without GloVe embeddings. I also decrease the batch size since I have a bigger model.
	<b>Custom 2</b> - Mean GloVe features + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=100000, HiddenUnits=20	10	0.34	85.56%	85.32%	I decreased the number of hidden units in the network to 20 and increased no. of epochs to 10 to analyse the behaviour of the model. The training time for 20 hidden units is about 15 minutes for 10 epochs, which is less than the default model. Hence, decreasing the number of hidden units decreased the model training time. The model achieved an accuracy of ~85.5% on training set and ~85% on test set, so clearly, it performed almost the same on both the sets. Hence, we observe that decreasing the number of hidden units decreases training time and this model did not overfit the training data.
	<b>Custom 3</b> - Mean GloVe features + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=100000, HiddenUnits=5	15	0.4196	81.11%	85.56%	I further decreased the number of hidden units to 5 and increased the number of epochs to 15 in the network to analyse the behaviour of the model. The training time for 5 hidden units is about 15 minutes for 15 epochs, which is less than the default model. Hence, decreasing the number of hidden units decreased the model training time. The model achieved an accuracy of ~81% on training set and ~85.5% on test set, so clearly, it performs worse on the training dataset than the test set by reducing hidden units to very low. Hence, we observe that decreasing the number of hidden units decreases training time and it introduces underfitting of the model. The model performs worse on the training set than the test set. Hence, we can conclude that too large of hidden units overfit the model to training set and lesser number of units do not learn all features properly.

Part 2a	<b>Given Model</b> - Word embedding layer + LSTM stateful cell + same dropout mask for an entire sequence + Batch normalization + Max pooling + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=8000, HiddenUnits=500, Training sequence Length = 100, Testing sequence Length = 500	30	0.054	98.13%	87.03%	<p>This is the default model suggested in the HW document. The training time for 500 hidden units with training sequence length = 100 is about 1 hour for 30 epochs. The model achieved an accuracy of ~98% on training set and ~87% on test set, so clearly, it overfit the training dataset. This model can get better accuracy but we have more risk of overfitting in this case as well. Also, I tried with various sequence lengths from 50 to 500, with gaps of 50, at the test time and I notice that as I increase the testing time sequence length, the model's accuracy increases. Notice that this model performs worse than the Bag of words implementation without GloVe embedding in Part 1a. The reason is overfitting of model. Hence, the performance of the model can be increased if we try some method to reduce overfitting like dropout, or training on smaller sequences than 100.</p>
	<b>Custom 1</b> - Word embedding layer + LSTM stateful cell + same dropout mask for an entire sequence + Batch normalization + Max pooling + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=50, VocabularySize=8000, HiddenUnits=500, Training sequence Length = 1000, Testing sequence Length = 250	10	0.1158	95.43%	76.65%	<p>I increased the training sequence length to 1000 to observe the behaviour of the model. Also, I reduced the batch size to 50 from 200, since I took a longer sequence length and a larger batch size may not fit into the memory. The training time for 500 hidden units with training sequence length = 1000 is about 8 hours for 10 epochs. The model achieved an accuracy of ~95% on training set and ~76% on test set, so clearly, it overfit the training dataset. Also, I tried with various sequence lengths from 50 to 500, with gaps of 50, at the test time and I report the results for testing length = 250. Clearly, this model overfits the training data set more than the given model where the training sequence length was 100. The reason is overfitting of model, hence increasing the training sequence length a lot. Also, the performance of the model declines when I increased the training sequence length to 1000. Hence, the performance of the model can be increased if we try some method to reduce overfitting like training on smaller sequences than 1000.</p>

	<b>Custom 2</b> - Word embedding layer + LSTM stateful cell + same dropout mask for an entire sequence + Batch normalization + Max pooling + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=300, VocabularySize=8000, HiddenUnits=500, Training sequence Length = 20, Testing sequence Length = 500	30	0.4263	79.76%	81.93%	I decreased the training sequence length to 20 to observe the behaviour of the model. Also, I increased the batch size 300, since I took a shorter sequence length and a bit larger batch size may now be able fit into the memory. The training time for 500 hidden units with training sequence length = 20 is about 1 hour for 30 epochs. The model achieved an accuracy of ~79% on training set and ~82% on test set, so clearly, it underperforms on the training set. Hence, we observe that decreasing the training sequence length introduces underfitting of the model. The model performs worse on the training set than the test set. Also, I tried with various sequence lengths from 50 to 500, with gaps of 50, at the test time and I report the results for testing sequence length = 500. This is an interesting case because the model performs better on test set than it does on the training set. Hence, we can conclude that too large of sequence length overfit the model to training set and too less sequence length in training number of units does not have very high performance of the model.
Part 2b	<b>Given Model</b> - GloVe embedding layer + LSTM stateful cell + same dropout mask for an entire sequence + Batch normalization + Max pooling + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=200, VocabularySize=100000, HiddenUnits=500, Training sequence Length = 100, Testing sequence Length = 500	30	0.1389	93.14%	91.24%	This is the default model suggested in the HW document. The training time for 500 hidden units with training sequence length = 100 is about 1 hour for 30 epochs. The model achieved an accuracy of ~94% on training set and ~91% on test set. This model gets better accuracy than all the other models tried yet. Notice that the GloVe embeddings increase the accuracy. Also, I tried with various sequence lengths from 50 to 500, with gaps of 50, at the test time and I notice that as I increase the testing time sequence length, the model's accuracy increases. The model does not seem to be overfitting very much now, hence we see that GloVe embeddings along with RNN LSTM model reduces the risk of overfitting.
	<b>Custom 1</b> - GloVe embedding layer + LSTM stateful cell + same dropout mask for an entire sequence + Batch normalization + Max pooling + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=50, VocabularySize=100000, HiddenUnits=500, Training sequence Length = 1000, Testing sequence Length = 250	5	0.2183	91.35%	82.77%	I increased the training sequence length to 1000 to observe the behaviour of the model. Also, I reduced the batch size to 50 from 200, since I took a longer sequence length and a larger batch size may not fit into the memory. The training time for 500 hidden units with training sequence length = 1000 is about 8 hours for 5 epochs. The model achieved an accuracy of ~91% on training set and ~82% on test set, so clearly, it overfit the training dataset. Also, I tried with various sequence lengths from 50 to 500, with gaps of 50 and I report the results for testing sequence length = 250. Notice that this model performs worse than the default case when length of training sequence is 100. Hence, increasing the training sequence length is the reason is overfitting of the model and reduction in performance. The performance of the model can be increased if we train on smaller sequences than 1000. This is an interesting case because increasing highly the training sequence length increases overfitting and training time as well.

	<b>Custom 2</b> - GloVe embedding layer + LSTM stateful cell + same droupout mask for an entire sequence + Batch normalization + Max pooling + Fully Connected Layer + Relu + Output Layer	ADAM optimizer with LR=0.001, BatchSize=300, VocabularySize=100000, HiddenUnits=500, Training sequence Length = 20, Testing sequence Length = 200	30	0.5097	74.84%	84.12%	I decreased the training sequence length to 20 to observe the behaviour of the model. Also, I increased the batch size 300, since I took a shorter sequence length and a bit larger batch size may now be able fit into the memory. The training time for 500 hidden units with training sequence length = 20 is about 1 hour for 30 epochs. The model achieved an accuarcy of ~75% on training set and ~84% on test set, so clearly, it underperforms on the training set. Hence, we observe that decreasing the training sequence length introduces underfitting of the model. The model performs worse on the training set than the test set. Also, I tried with various sequence lengths from 50 to 500, with gaps of 50, at the test time and I report the results for testing sequece length = 200. This is an interesting case because the model performs better on test set that it does on the training set. Hence, we can conclude that too large of sequence length overfit the model to training set and too less sequence length in training number of units does not have very high performance of the model.
--	--	--	----	--------	--------	--------	---

Part 3a    Given Model

Part 3b

Generated Review  
Generated Review  
Generated Review

Temperature=1.0  
Temperature 0.5

a hugely influential , very strong , nuanced  
i usually like this movie , but this is one of

Part 3c

Given Model  
Custom 1  
Custom 2  
  
Custom 3

~91%+

Describe more about the model/results such as why certain  
hyperparamters were chosen or the effect it had on the accuracy/training