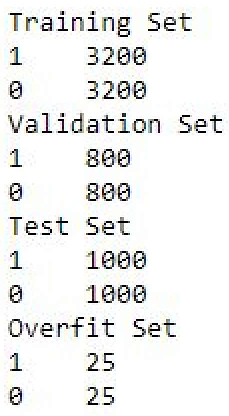
Esmat Sahak
ECE324

Assignment 5: Subjective/Objective Sentence Classification Using Word Vectors and NLP

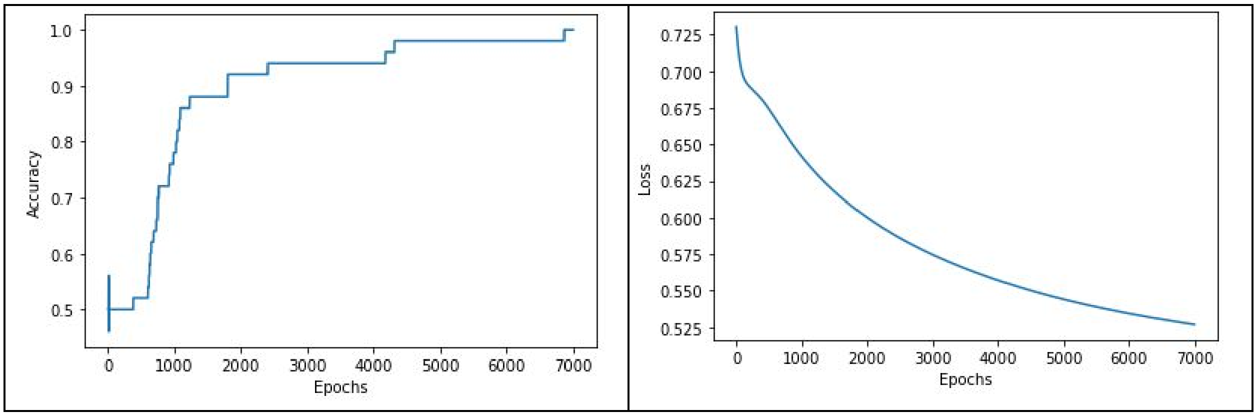
3.1: Create train/validation/test splits



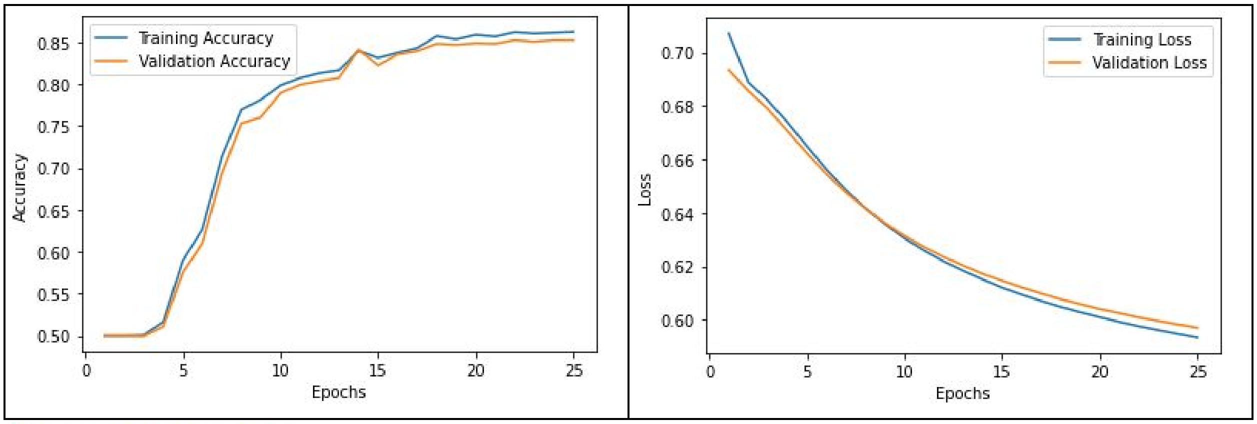
```
Training Set
1    3200
0    3200
Validation Set
1    800
0    800
Test Set
1    1000
0    1000
Overfit Set
1    25
0    25
```

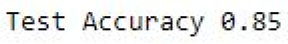
4. Baseline Model and Training

Overfit



Training and Validation



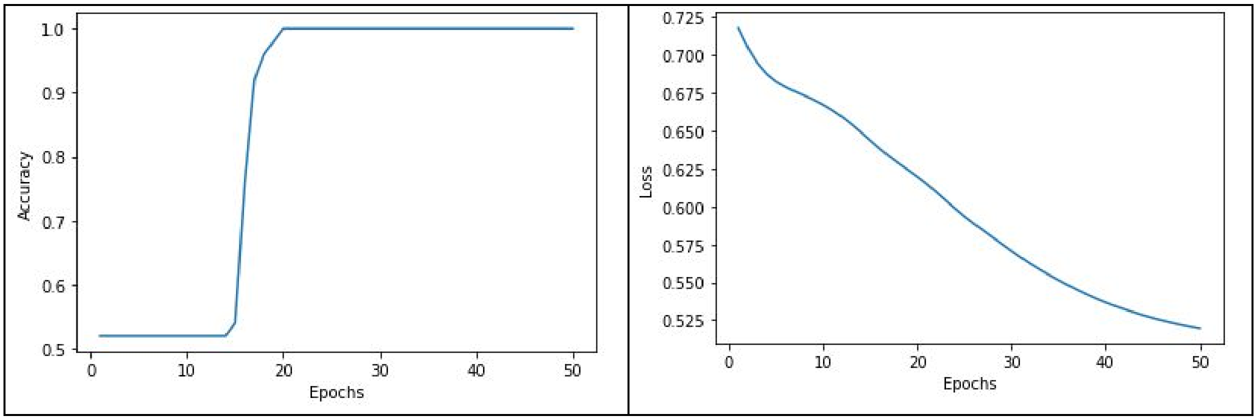


```
Test Accuracy 0.85
```

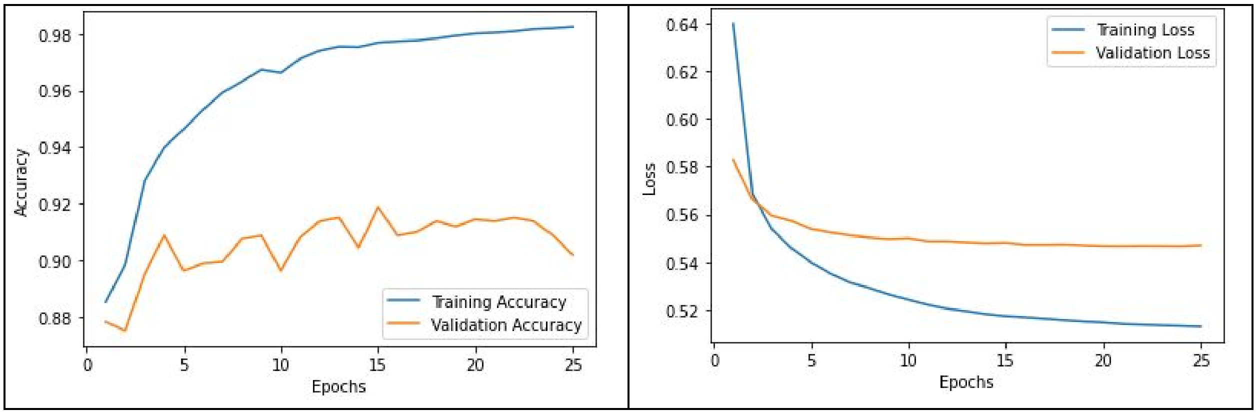
Esmat Sahak
ECE324

5. Convolutional Neural Network (CNN)

Overfit



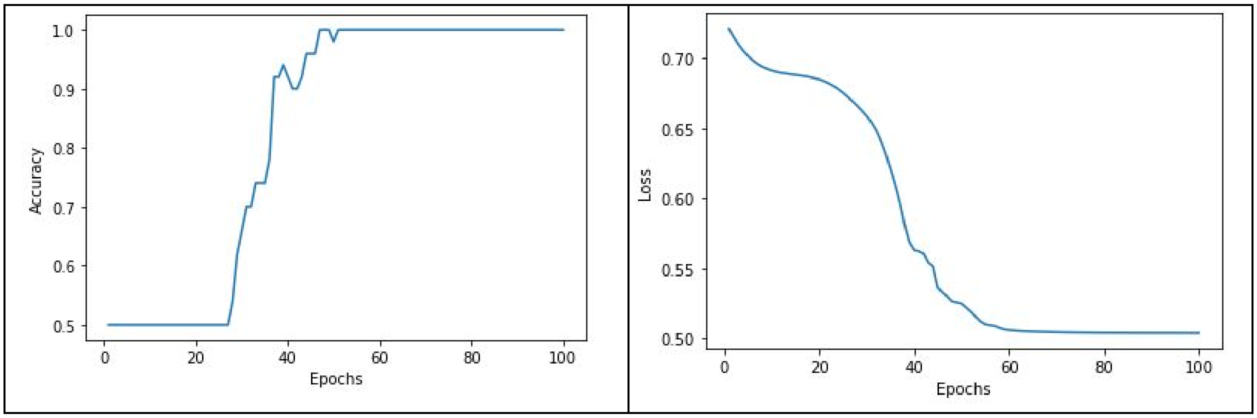
Training and Validation



```
Test Accuracy 0.92
```

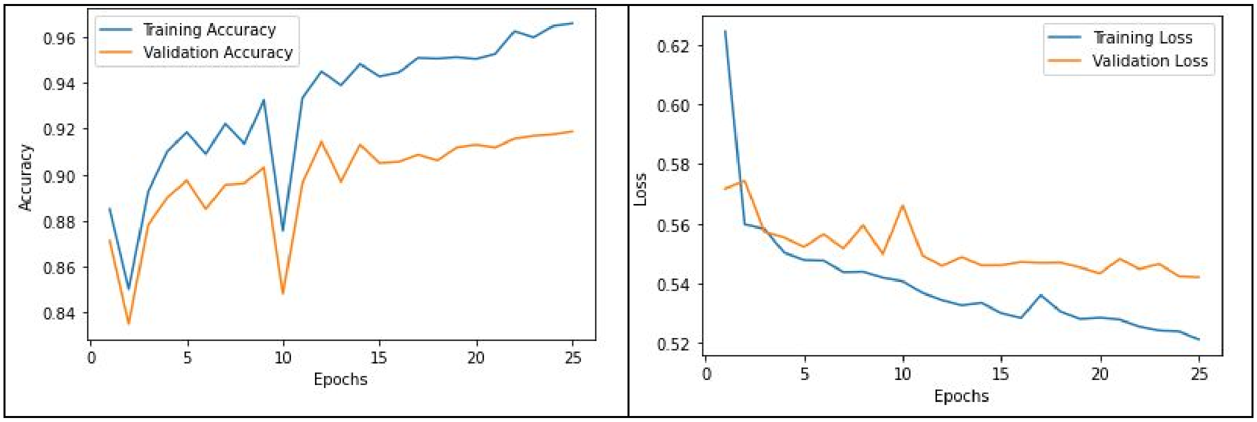
6. Recurrent Neural Network (RNN)

Overfit



Esmat Sahak
ECE324

Training and Validation Accuracy



```
Test Accuracy 0.91
```

8. Experimental and Conceptual Questions

After training on the three models, report the loss and accuracy on the train/validation/test in a total. There should be a total of 18 numbers. Which model performed the best? Is there a significant difference between the validation and test accuracy? Provide a reason for your answer.

Model	Train		Validation		Test	
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy
Baseline	0.59	0.86	0.60	0.85	0.60	0.85
CNN	0.51	0.98	0.55	0.91	0.55	0.92
RNN	0.52	0.97	0.54	0.92	0.55	0.91

The CNN performs slightly better than the RNN given its higher test accuracy. There is no significant difference between the validation and test accuracy as these sets have no impact on the training process (i.e. do not contribute to the process of modifying model parameters). Given that they are of similar sizes (16% and 20%), accuracies are expected to be similar.

In the baseline model, what information contained in the original sentence is being ignored? How will the performance of the baseline model inform you about the importance of that information?

The baseline model computes the average of a sentence, in this process meaning of phrases is lost and the order of words is ignored. This is problematic as two sentences with different meanings (or in some cases no meaning) but close averages will be classified similarly. For example, sentences such as “Donald Trump is a sore loser” and “Loser sore Trump is Donald” would give the same result.

For the RNN architecture, examine the effect of using pack padded sequence to ensure that we did indeed get the correct last hidden state. Train the RNN and report the loss and accuracy on the train/validation/test under these 3 scenarios:

(a) ***Default scenario, with using pack padded sequence and using the BucketIterator***

Esmat Sahak
ECE324

(b) Without calling pack padded sequence, and using the BucketIterator
(c) Without calling pack padded sequence, and using the Iterator.

What do you notice about the lengths of the sentences in the batch when using Iterator class instead? Given the results of the experiment, explain how you think these two factors affect the performance and why.

Scenario	Train		Validation		Test	
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy
(a)	0.52	0.97	0.54	0.92	0.55	0.91
(b)	0.53	0.95	0.55	0.91	0.55	0.91
(c)	0.58	0.88	0.57	0.87	0.57	0.87

When the Iterator class is called, each batch consists of a wide range of lengths; this contrasts with the BucketIterator class, which groups sentences of similar lengths together. Grouping sentences of similar lengths reduces padding leading to higher accuracies.

The default model results in the higher accuracies and lower losses implying that the correct last hidden state was obtained. Removing the pack padding sequence results in passing the incorrect hidden state as shorter sentences are padded with zero vectors. Using an iterator instead of a bucket iterator (in addition to removing the pack padding sequence) further amplifies the above error as sentences are of inconsistent lengths leading to more padding.

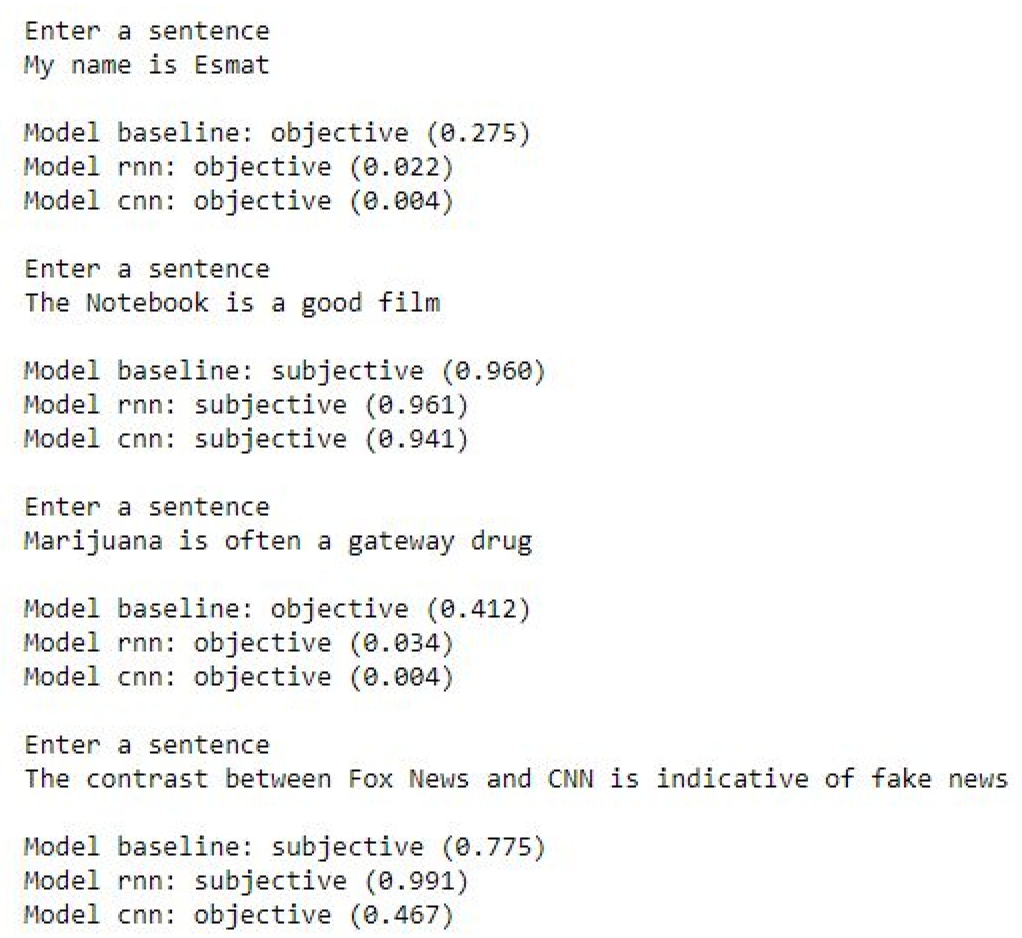
In the CNN architecture, what do you think the kernels are learning to detect? When performing max-pooling on the output of the convolutions, what kind of information is the model discarding? Compare how this is different or similar to the baseline model's discarding of information.

The kernels are learning to detect 2 or 4 length phrases contained within a sentence that is suggestive of subjectivity or objectivity based on various concepts that are learned. Max pooling extracts the largest value of each feature map and discards the other elements that aren't as indicative or relevant. In other words, groups of words that do not contribute to the overall subjectivity or objectivity of the sentence (based on the concept the kernel maps to) as much are discarded. This performs better than the baseline as by taking the average of a sentence it loses more information as subjectivity or objectivity of a sentence in most cases can be deduced from a localized set of words.

Try running the subjective bot.py script on 4 sentences that you come up with yourself, where 2 are definitely objective/subjective, while 2 are borderline subjective/objective, according to your opinion. Include your console output in the write up. Comment on how the three models performed and whether they are behaving as you expected. Do they agree with each other? Does the majority vote of the models lead to the correct answer for the 4 cases? Which model seems to be performing the best?

Sentence order in output: objective, subjective, borderline objective, borderline subjective

Esmat Sahak
ECE324



```
Enter a sentence
My name is Esmat

Model baseline: objective (0.275)
Model rnn: objective (0.022)
Model cnn: objective (0.004)

Enter a sentence
The Notebook is a good film

Model baseline: subjective (0.960)
Model rnn: subjective (0.961)
Model cnn: subjective (0.941)

Enter a sentence
Marijuana is often a gateway drug

Model baseline: objective (0.412)
Model rnn: objective (0.034)
Model cnn: objective (0.004)

Enter a sentence
The contrast between Fox News and CNN is indicative of fake news

Model baseline: subjective (0.775)
Model rnn: subjective (0.991)
Model cnn: objective (0.467)
```

The models generally performed well. The RNN and CNN returned more firm probabilities (i.e. these networks were more certain of the sentence's classification) than the baseline. This is expected as the baseline loses more information through its averaging method. In most cases, the models are in agreement with each other except for the last case, where the CNN returns a more reasonable probability than the RNN (around 0.5) but arrives at the "wrong" conclusion (according to my interpretation). Hence, the RNN performs the best in terms of accuracy. For these four cases, a majority vote does lead to the correct answer but is not recommended as the methods for each model is very distinct and the baseline is not a very reliable method.

Describe your experience with Assignment 4:

(a) **How much time did you spend on Assignment 4?**
8 hours.

(b) **What did you find challenging?**
Tracking the dimensions of inputs and outputs (i.e. knowing where squeeze() is necessary).

(c) **What did you enjoy?**
I enjoyed the extension Professor Rose granted and testing each model using the bot.

(d) **What did you find confusing?**
I was initially unsure as to why permutations were required for the CNN inputs.

(e) **What was helpful?**

Esmat Sahak
ECE324

The illustrations for each model were very helpful in terms of understanding how the model operates. Additionally, providing the code that wasn't very useful in terms of understanding RNNs and CNNs was helpful and saved time.