**Assignment 4: Summary Report**

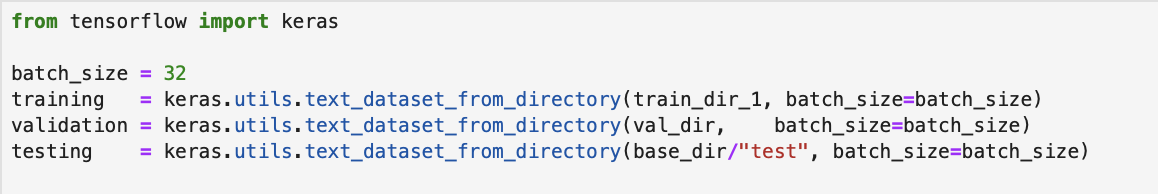
**D Hari Vinayak-811299324**

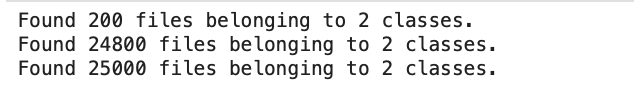
Requirements:

Consider the IMDB example from Chapter 6. Re-run the example modifying the following:  
1. Cutoff reviews after 150 words.  
2. Restrict training samples to 100.  
3. Validate on 10,000 samples.  
4. Consider only the top 10,000 words.  
5. Consider both a embedding layer, and a pretrained word embedding. Which approach  
did better? Now try changing the number of training samples to determine at what  
point the embedding layer gives better performance.

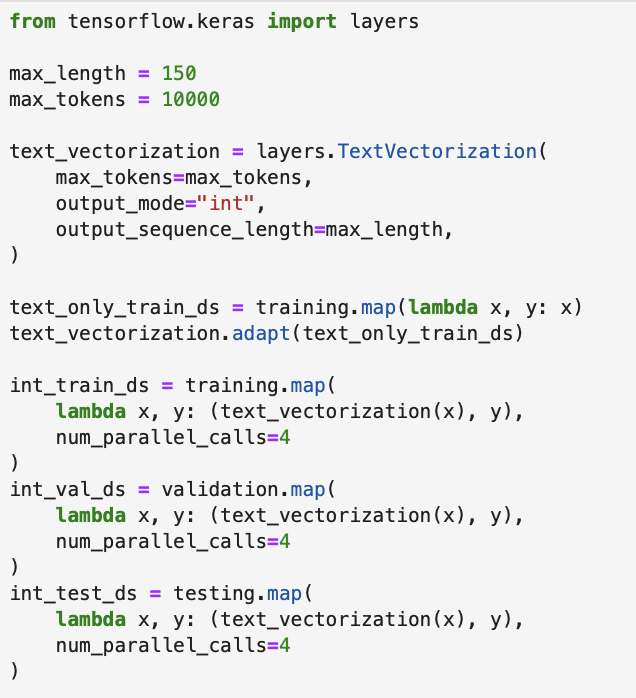
I first start with processing the data set according to the requirements:



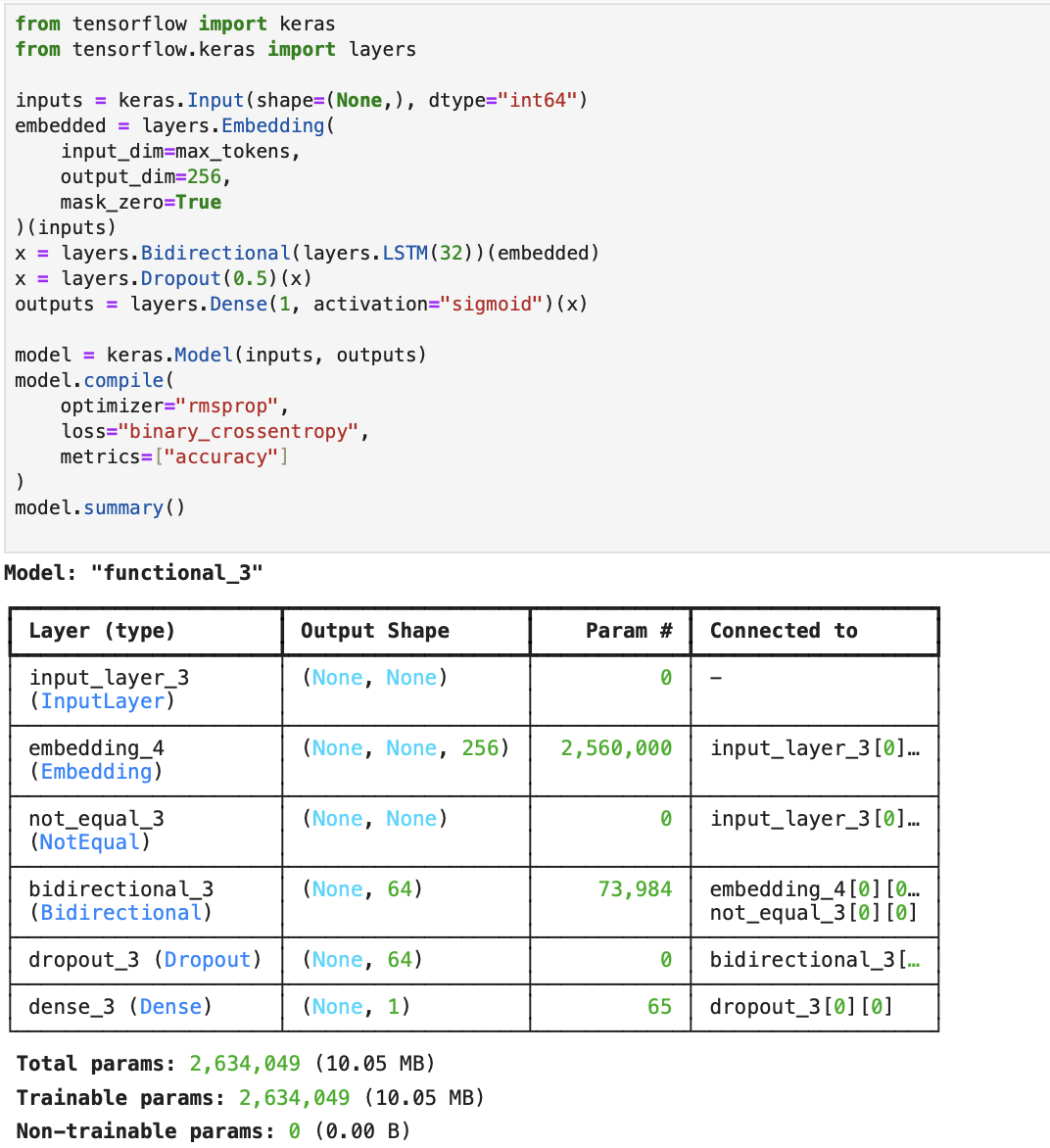




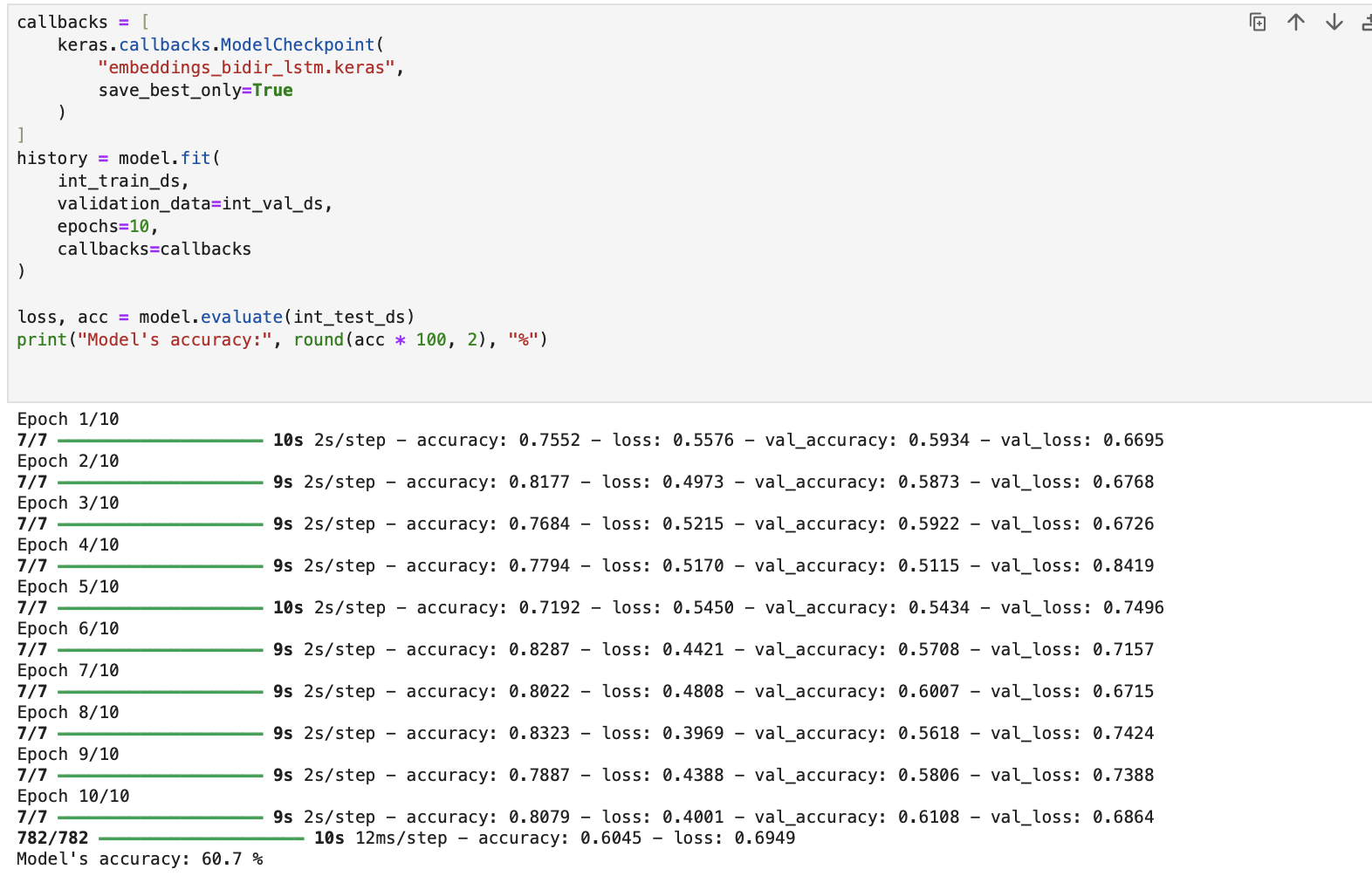
Then I cutoff review after 150 words and consider only 10,000 words:



In the model construction, I tried an embedding layer, here’s the model summary:



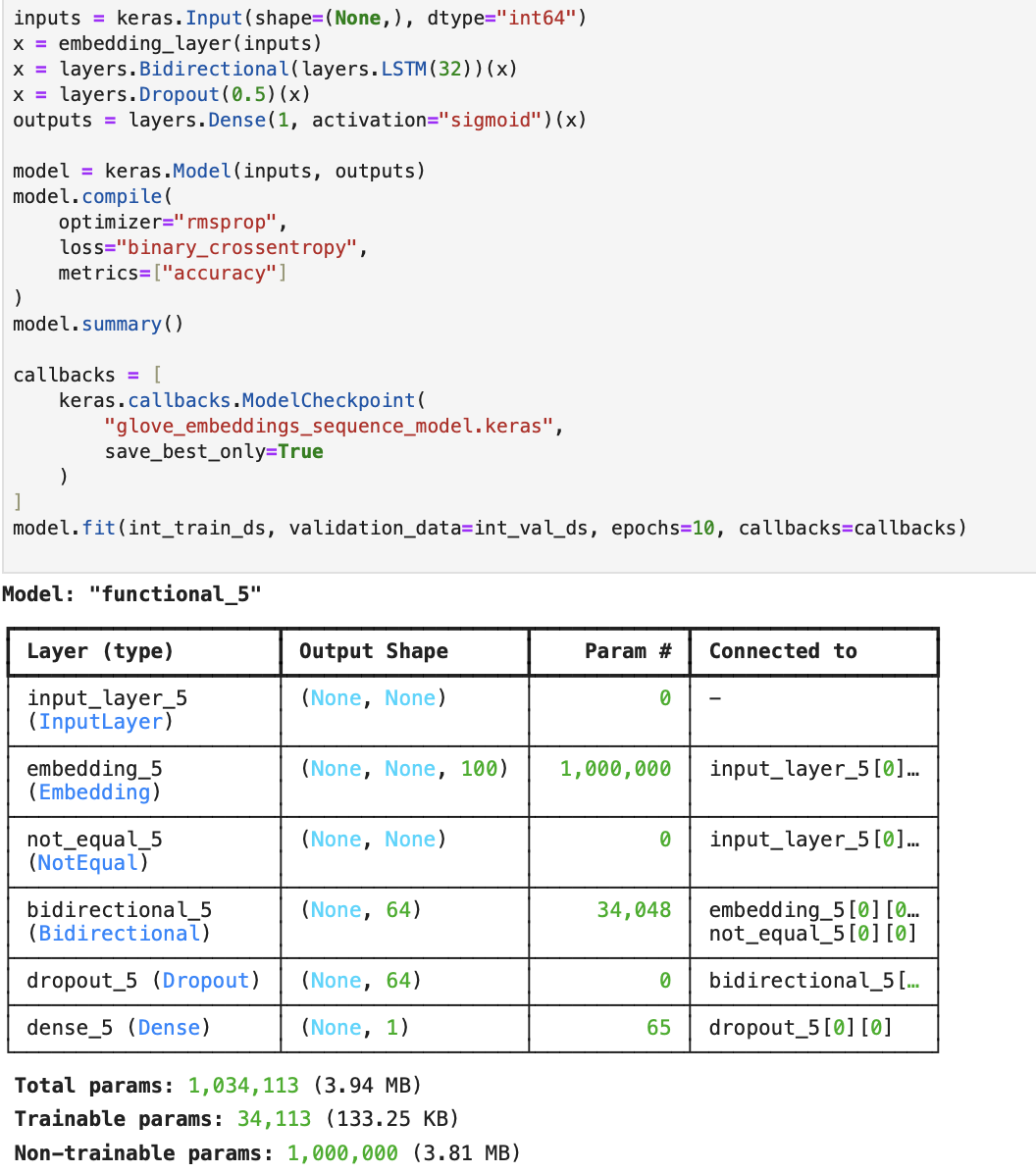
Fitting and testing the model:

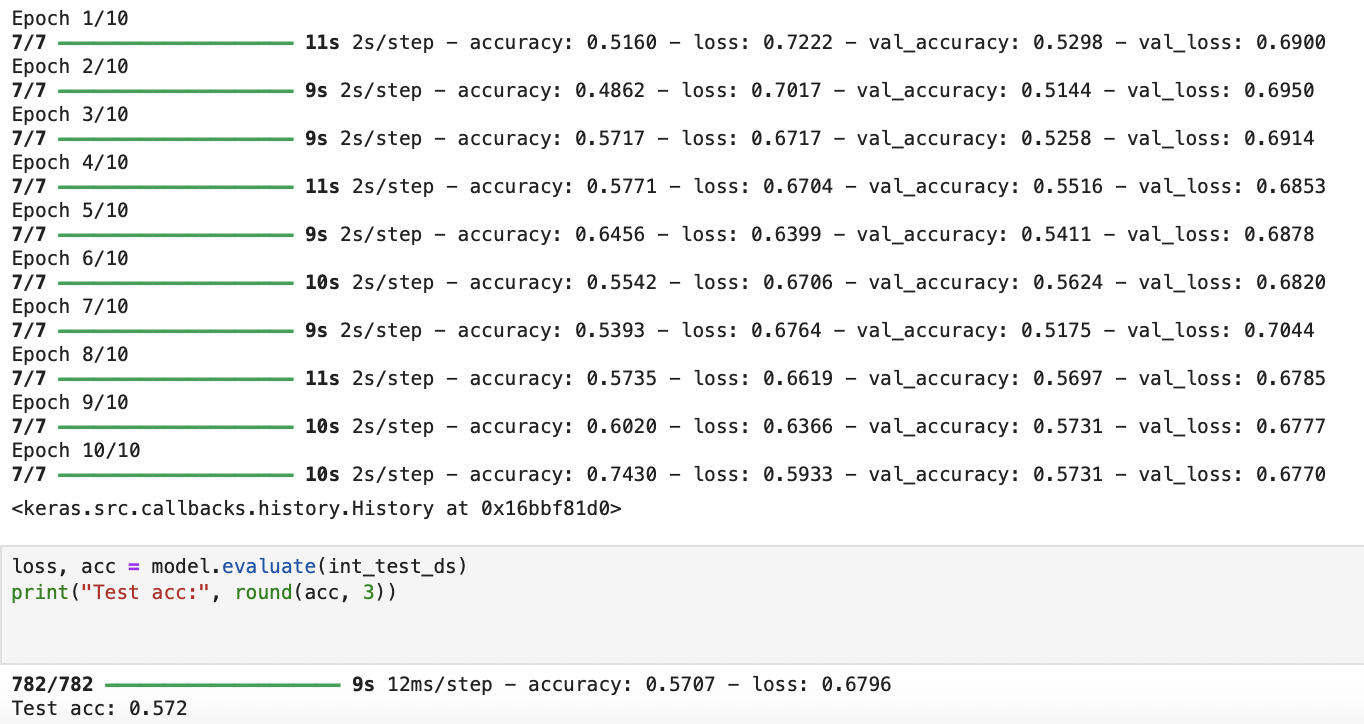


We can see that the testing accuracy is 60.7%, which is too low

Then, we will try the pretrained word embedding on the same size of this data set:





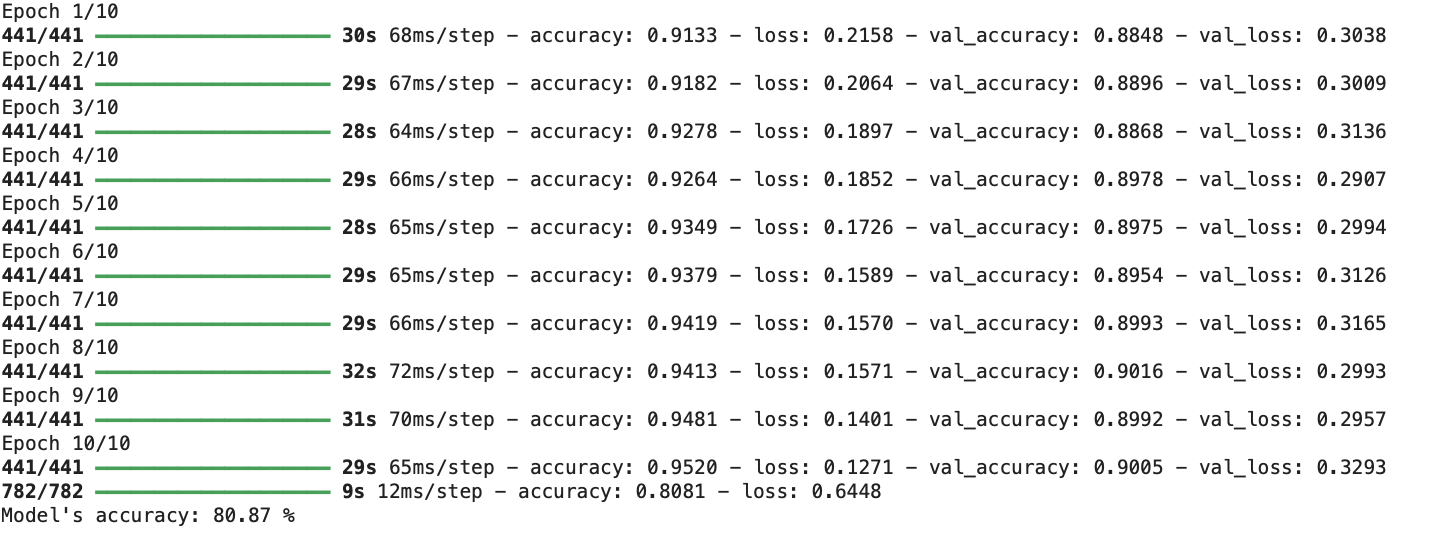


We can see that the testing accuracy using the pretrained model is 57.2%, which is still not high.

Then, I tried increasing the training data size since we only used 100 in the previous, and continue using the pretrained word embedding.

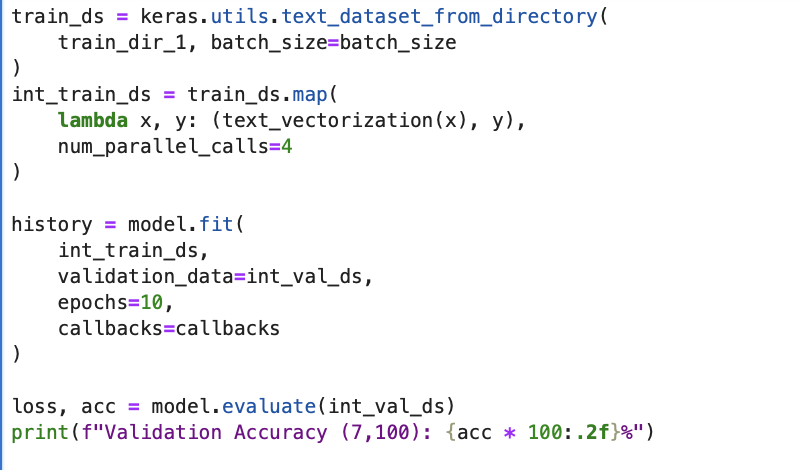


Fitting and testing the model:

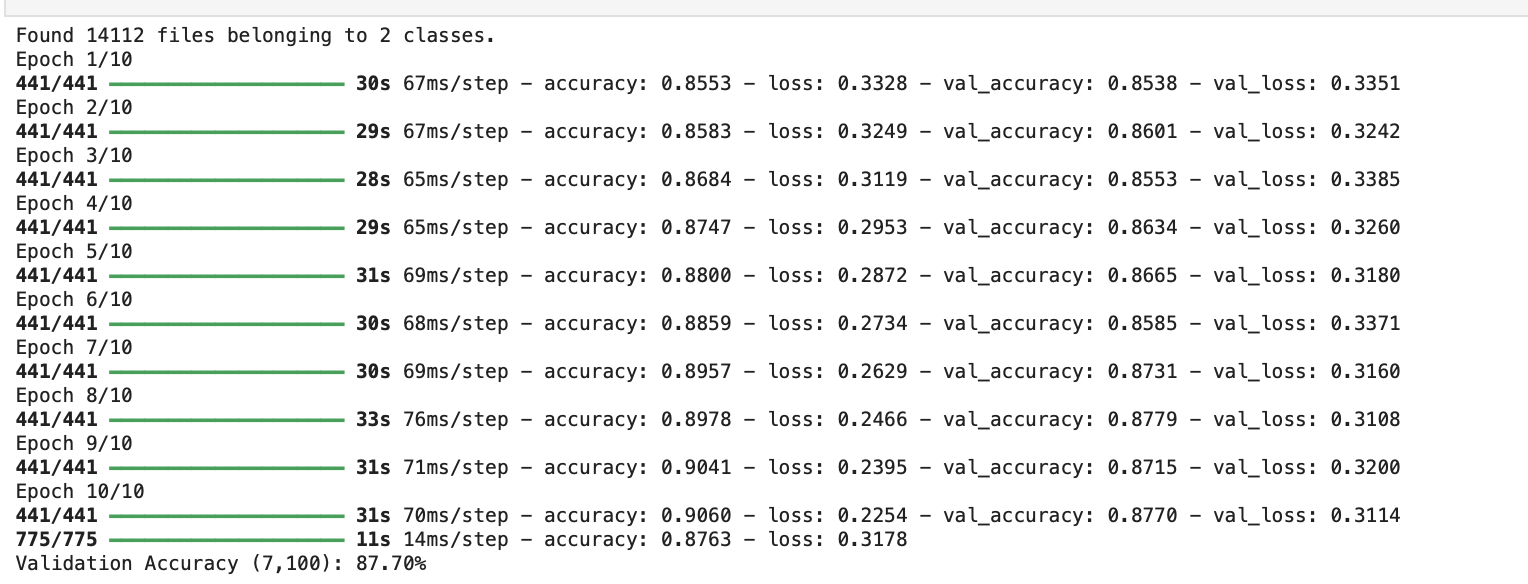


We can see that the accuracy increased to 80.87% after we increased the training size to 7,100

I decided to increase the training size by 7,000 again and see how the model performs:



Fitting and testing the model:



After increasing the training size again, we can see that the accuracy increased to 87.70%

**Conclusion:**

|  |  |  |
| --- | --- | --- |
| Training Size | Model with | Testing Accuracy |
| 100 | embedding layer | 60.7% |
| 100 | Pretrained Word Embeddings | 57.2% |
| 7,100 | Pretrained Word Embeddings | 80.87% |
| 14,100 | Pretrained Word Embeddings | 87.70% |

The table shows that the first model's poor performance was caused by both our training size and the embedding layer's inadequate capacity to capture meaningful word representations. Nevertheless, the performance was not significantly enhanced by the use of pretrained word embedding. The limited training size we had there may be the cause of all of this. Therefore, even while pretrained word embeddings have their uses, they may not offer much of a benefit when the dataset is tiny.  
  
The accuracy rose dramatically to 80.87% when the training data amount was raised to 7,100. Accuracy increased to 87.70% once the training size was increased to 14,100. This indicates that enhancing the quantity of the training data has a significant effect on enhancing the model's performance. Increasing the size more may result in the model being even better.   
Furthermore, taking into account other methods such as data augmentation or experimenting with more intricate model architectures could potentially improve performance.