ASSIGNMENT 4

Student's name: THUY THIEN TRAM NGUYEN

Student's ID: 811293917

I) MODEL'S PERFORMANCE

A. Using one-hot encoded approach

Model's parameter:

- max_length = 150
- max_tokens = 10000
- Training samples: 100
- Validation samples: 10000

Model's performance:

- The model encounters an error when attempting to convert input data in the form of KerasTensors into one-hot encoded vectors. KerasTensors are placeholders that contain information about the shape and data type, but they do not hold the actual values. However, tf.one_hot requires real data to perform the one-hot encoding process. Since Keras doesn't know how to handle the KerasTensor (this symbolic object), it leads to an error. Additionally, using the one-hot encoding method is not very efficient because models that use one-hot encoding require a long max_length. This means that after converting characters into numbers, the resulting vector will have a length equal to max_length, which consumes more resources. Therefore, it's better to use Embedding layers as a replacement.

B. Using Embedding layers approach from scratch

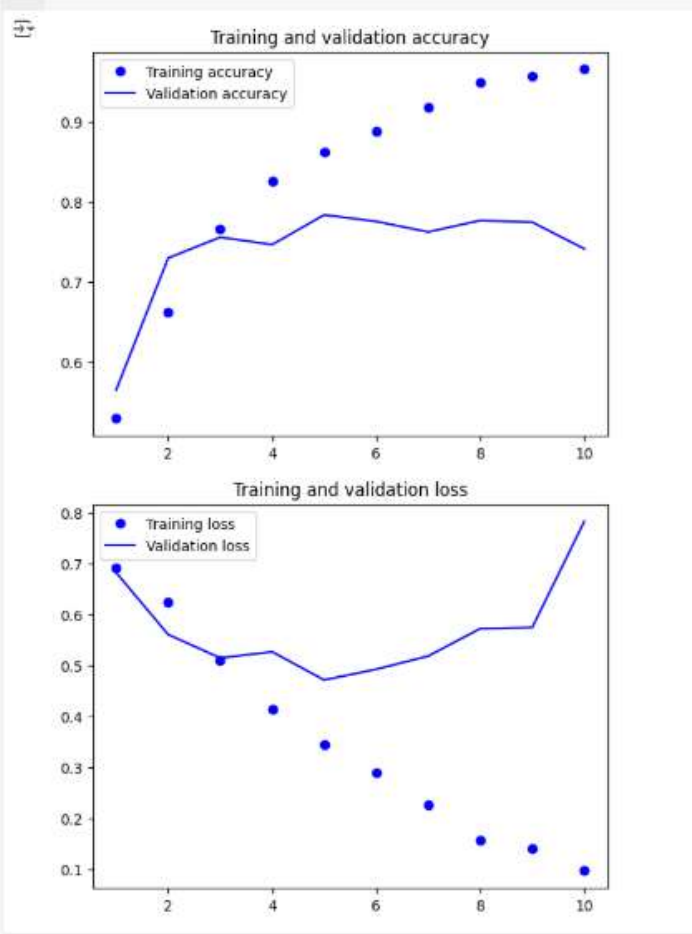
1. SCENARIO 1:

1.1. Model's parameter:

- output_dim=256
- droupout value = 0.5

1.2. Model's performance:

- The results of Accuracy and Loss function depict through:
  + Training set – Accuracy: 0.9672
  + Training set – Loss: 0.0915
  + Validation set – Accuracy: 0.7628
  + Validation set – Loss: 0.4952
  + Test set – Accuracy: 0.769



- Based on the graph, achieving over 96% accuracy on the training data is quite good for a binary classification problem with two labels: positive and negative. The accuracy on the validation and test sets does not show significant differences, so it cannot yet be concluded whether the model is overfitting. However, with these results, the model could be further improved by adding variables to the sample set, adjusting the BiDirectional LSTM units from 32 to 64 or 128 to capture more complex patterns, and reducing the embedding dimension to 128 to lower data dimensionality, save resources, and prevent overfitting.

2. SCENARIO 2:

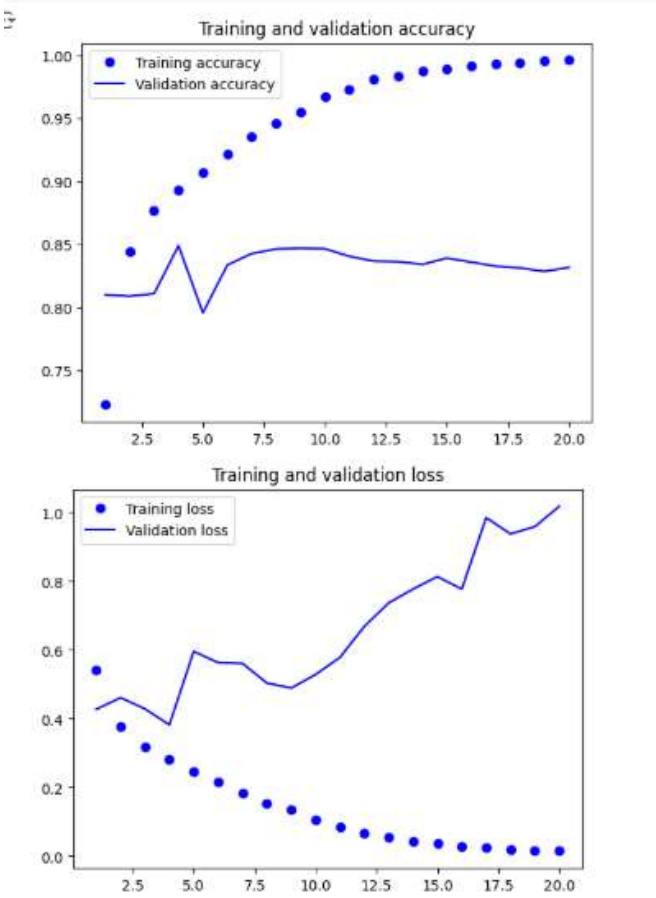
2.1. Model's parameter:

- Training sample = 20000
- output_dim=128
- droupout value = 0.5
- epoch = 20

**Others parameters are kept remain.

2.2. Model's performance:

```
 Total params: 1,321,281 (5.04 MB)
 Trainable params: 1,321,281 (5.04 MB)
 Non-trainable params: 0 (0.00 B)
Epoch 1/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 106s 165ms/step - accuracy: 0.6369 - loss: 0.6206 - val_accuracy: 0.8100 - val_loss: 0.4274
Epoch 2/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 138s 159ms/step - accuracy: 0.8297 - loss: 0.4033 - val_accuracy: 0.8090 - val_loss: 0.4611
Epoch 3/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 143s 162ms/step - accuracy: 0.8671 - loss: 0.3305 - val_accuracy: 0.8110 - val_loss: 0.4285
Epoch 4/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 145s 166ms/step - accuracy: 0.8862 - loss: 0.2910 - val_accuracy: 0.8490 - val_loss: 0.3823
Epoch 5/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 99s 159ms/step - accuracy: 0.9022 - loss: 0.2532 - val_accuracy: 0.7958 - val_loss: 0.5957
Epoch 6/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 142s 160ms/step - accuracy: 0.9181 - loss: 0.2222 - val_accuracy: 0.8336 - val_loss: 0.5631
Epoch 7/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 142s 159ms/step - accuracy: 0.9313 - loss: 0.1867 - val_accuracy: 0.8428 - val_loss: 0.5610
Epoch 8/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 142s 160ms/step - accuracy: 0.9433 - loss: 0.1584 - val_accuracy: 0.8464 - val_loss: 0.5039
Epoch 9/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 141s 159ms/step - accuracy: 0.9541 - loss: 0.1342 - val_accuracy: 0.8470 - val_loss: 0.4891
Epoch 10/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 142s 159ms/step - accuracy: 0.9666 - loss: 0.1056 - val_accuracy: 0.8466 - val_loss: 0.5286
Epoch 11/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 142s 160ms/step - accuracy: 0.9733 - loss: 0.0818 - val_accuracy: 0.8406 - val_loss: 0.5783
Epoch 12/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 141s 158ms/step - accuracy: 0.9806 - loss: 0.0661 - val_accuracy: 0.8368 - val_loss: 0.6685
Epoch 13/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 99s 158ms/step - accuracy: 0.9832 - loss: 0.0558 - val_accuracy: 0.8364 - val_loss: 0.7368
Epoch 14/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 143s 160ms/step - accuracy: 0.9861 - loss: 0.0436 - val_accuracy: 0.8342 - val_loss: 0.7772
Epoch 15/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 140s 158ms/step - accuracy: 0.9886 - loss: 0.0388 - val_accuracy: 0.8392 - val_loss: 0.8134
Epoch 16/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 101s 162ms/step - accuracy: 0.9923 - loss: 0.0290 - val_accuracy: 0.8360 - val_loss: 0.7771
Epoch 17/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 100s 159ms/step - accuracy: 0.9920 - loss: 0.0259 - val_accuracy: 0.8328 - val_loss: 0.9850
Epoch 18/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 143s 161ms/step - accuracy: 0.9940 - loss: 0.0205 - val_accuracy: 0.8314 - val_loss: 0.9377
Epoch 19/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 100s 160ms/step - accuracy: 0.9958 - loss: 0.0163 - val_accuracy: 0.8286 - val_loss: 0.9590
Epoch 20/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 99s 158ms/step - accuracy: 0.9957 - loss: 0.0151 - val_accuracy: 0.8318 - val_loss: 1.0178
782/782 ━━━━━━━━━━━━━━━━━━━━ 31s 39ms/step - accuracy: 0.8459 - loss: 0.3868
Test acc: 0.843
```



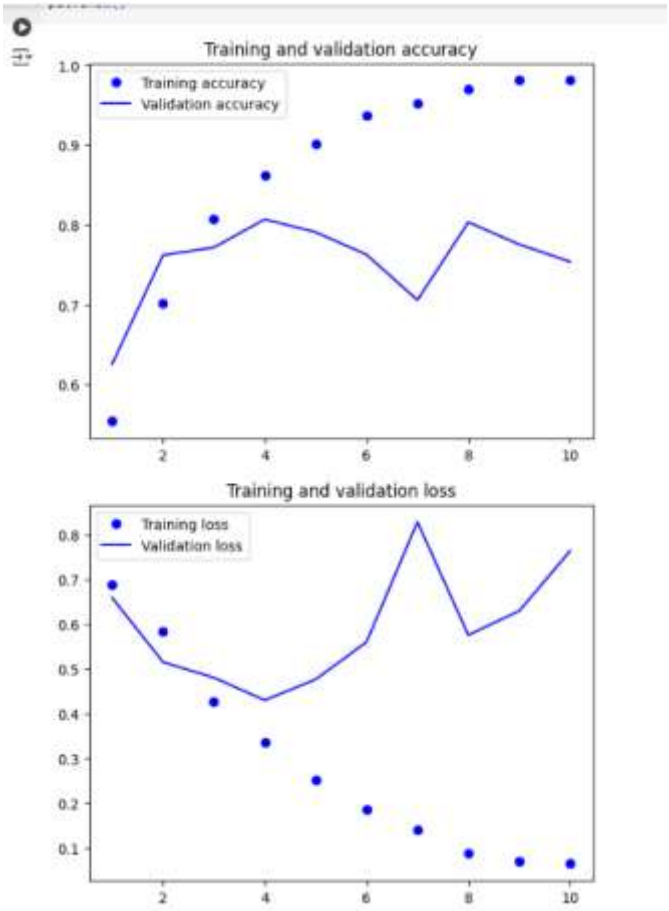
- After increasing the number of variables in the training set, the accuracy of the Training set reached 99.57%, which shows that the model learns very well on the training set. However, high accuracy also leads to signs of Overfitting when the model learns too many details instead of general features. This is shown by the low accuracy (about 83.13%) on the Validation set, indicating that the generalization ability of the model is limited. Test Accuracy at 84.59% also shows that the model is able to classify positive or negative comments. However, the large gap in the accuracy of the Training and test sets is a sign of Overfitting. In addition, the loss function increases gradually in the Validation and test sets, indicating a high possibility of Overfitting.

C. Using an Embedding layer with masking enabled

1. SCENARIO 1:

1.1. Model's performance:

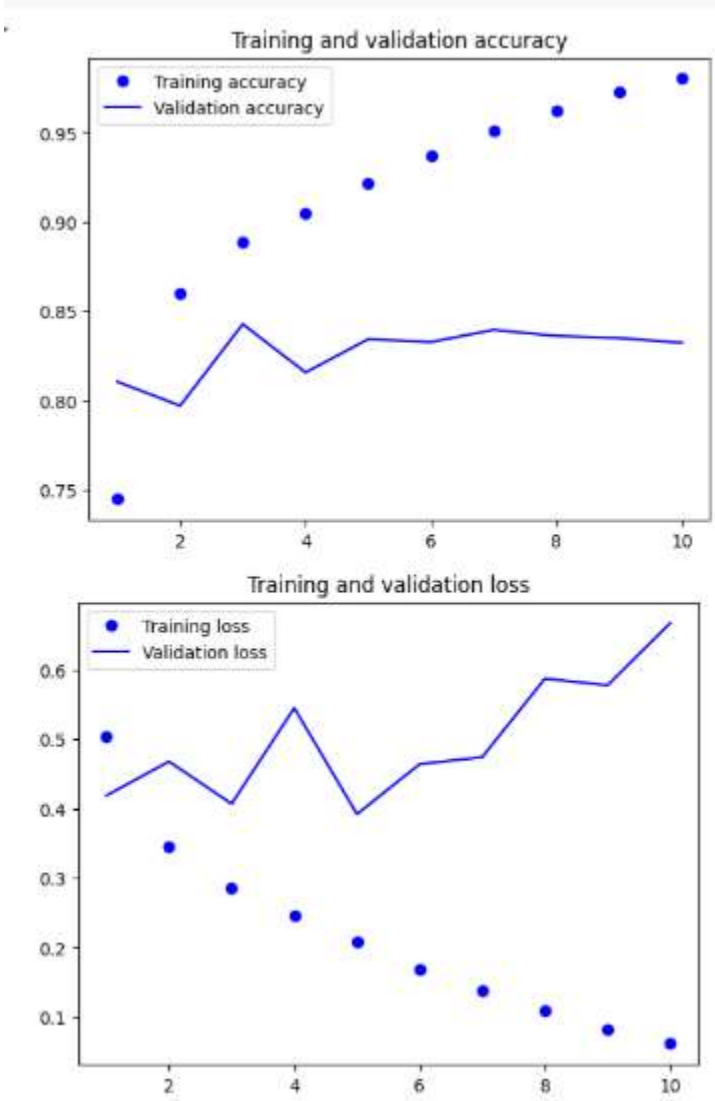
```
Total params: 2,634,849 (10.05 MB)
Trainable params: 2,634,849 (10.05 MB)
Non-trainable params: 0 (0.00 B)
Epoch 1/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 39s 340ms/step - accuracy: 0.5229 - loss: 0.6916 - val_accuracy: 0.6258 - val_loss: 0.6590
Epoch 2/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 42s 349ms/step - accuracy: 0.6625 - loss: 0.6188 - val_accuracy: 0.7622 - val_loss: 0.5153
Epoch 3/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 35s 351ms/step - accuracy: 0.7918 - loss: 0.4562 - val_accuracy: 0.7718 - val_loss: 0.4806
Epoch 4/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 42s 366ms/step - accuracy: 0.8673 - loss: 0.3390 - val_accuracy: 0.8070 - val_loss: 0.4303
Epoch 5/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 39s 344ms/step - accuracy: 0.8926 - loss: 0.2720 - val_accuracy: 0.7910 - val_loss: 0.4766
Epoch 6/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 34s 341ms/step - accuracy: 0.9541 - loss: 0.1858 - val_accuracy: 0.7628 - val_loss: 0.5601
Epoch 7/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 41s 341ms/step - accuracy: 0.9567 - loss: 0.1341 - val_accuracy: 0.7060 - val_loss: 0.8279
Epoch 8/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 40s 335ms/step - accuracy: 0.9730 - loss: 0.0853 - val_accuracy: 0.8034 - val_loss: 0.5757
Epoch 9/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 34s 342ms/step - accuracy: 0.9858 - loss: 0.0576 - val_accuracy: 0.7756 - val_loss: 0.6294
Epoch 10/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 44s 360ms/step - accuracy: 0.9887 - loss: 0.0687 - val_accuracy: 0.7540 - val_loss: 0.7632
782/782 ━━━━━━━━━━━━━━━━━━━━ 48s 61ms/step - accuracy: 0.7933 - loss: 0.4467
Test acc: 0.798
```



D. Enabling mask_zero=True in the Embedding layer allows the model to ignore padding values (elements with a value of 0) used to balance sentence lengths. However, adding many zero values to each sentence can bias the model toward negative results. With this embedding method, the training set's accuracy increased to 98%, but the gap between the Training set and Test set accuracy is too large (98.07% vs. 79.8%). Additionally, the model shows signs of overfitting at epochs 4 and 5, as training accuracy suddenly rises from 79.18% to 86.73%, while validation accuracy drops from 80.7% to 79%, and validation loss steadily increases. The model could be improved by applying L2 regularization in the Dense and LSTM layers, reducing the embedding_dim from 256 to 128, increasing the sample size, and adjusting the dropout value to a range of 0.3–0.5.

2. **SCENARIO 2:**

```
Total params: 1,321,281 (5.04 MB)
Trainable params: 1,321,281 (5.04 MB)
Non-trainable params: 0 (0.00 B)
Epoch 1/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 141s 219ms/step - accuracy: 0.6586 - loss: 0.5924 - val_accuracy: 0.8106 - val_loss: 0.4187
Epoch 2/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 144s 222ms/step - accuracy: 0.8529 - loss: 0.3607 - val_accuracy: 0.7972 - val_loss: 0.4679
Epoch 3/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 137s 214ms/step - accuracy: 0.8849 - loss: 0.2944 - val_accuracy: 0.8428 - val_loss: 0.4069
Epoch 4/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 135s 215ms/step - accuracy: 0.9003 - loss: 0.2553 - val_accuracy: 0.8158 - val_loss: 0.5449
Epoch 5/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 142s 216ms/step - accuracy: 0.9136 - loss: 0.2238 - val_accuracy: 0.8344 - val_loss: 0.3921
Epoch 6/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 140s 213ms/step - accuracy: 0.9319 - loss: 0.1796 - val_accuracy: 0.8328 - val_loss: 0.4639
Epoch 7/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 131s 210ms/step - accuracy: 0.9467 - loss: 0.1516 - val_accuracy: 0.8396 - val_loss: 0.4742
Epoch 8/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 145s 216ms/step - accuracy: 0.9592 - loss: 0.1169 - val_accuracy: 0.8362 - val_loss: 0.5872
Epoch 9/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 134s 215ms/step - accuracy: 0.9733 - loss: 0.0866 - val_accuracy: 0.8350 - val_loss: 0.5776
Epoch 10/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 144s 218ms/step - accuracy: 0.9775 - loss: 0.0651 - val_accuracy: 0.8324 - val_loss: 0.6671
782/782 ━━━━━━━━━━━━━━━━━━━━ 79s 47ms/step - accuracy: 0.8222 - loss: 0.4208
Test acc: 0.820
```



2.1. Model's performace:

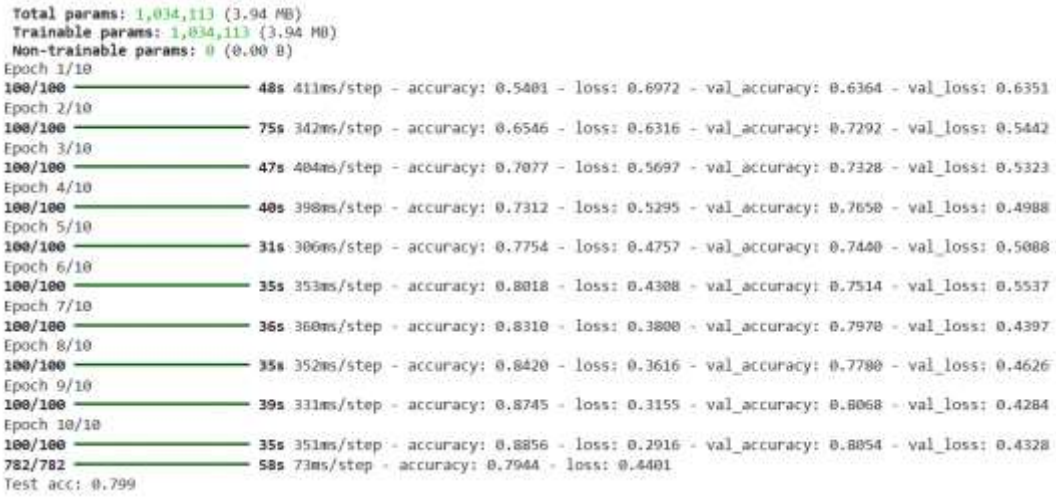
E. The accuracy in the testset has been improved but the model still shows Overfitting.

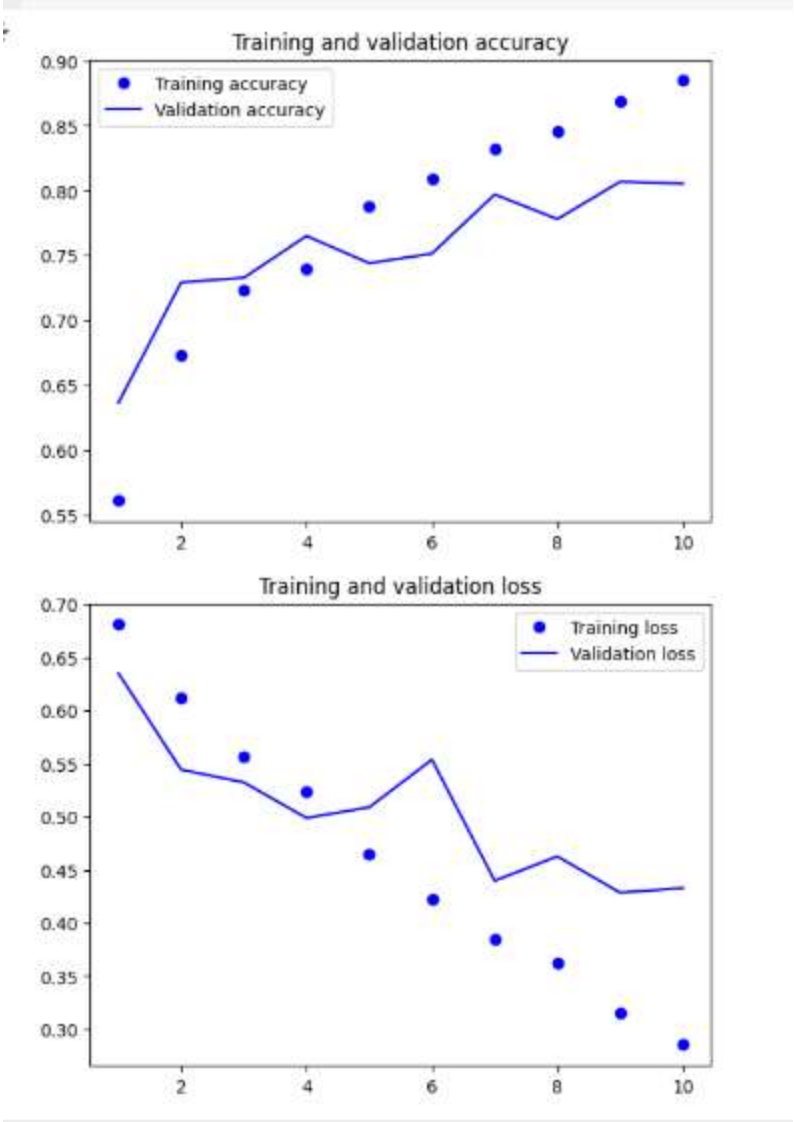
F. Using pretrained word embeddings

1. **SCENARIO 1:**

1.1. Model's performance:

G. The final method involves using pretrained word embeddings obtained from sources such as Wikipedia or Google, which helps diversify the input vocabulary. Additionally, when the model uses GloVe data, related words will have vectors closer to each other, reducing training time and improving accuracy.



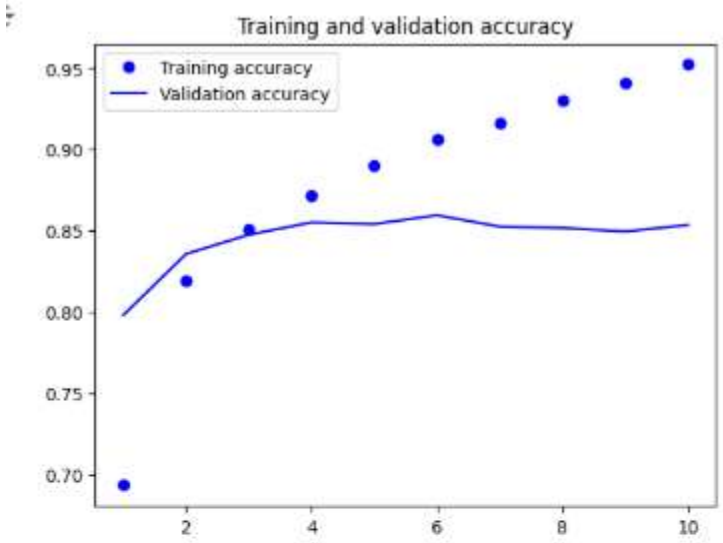


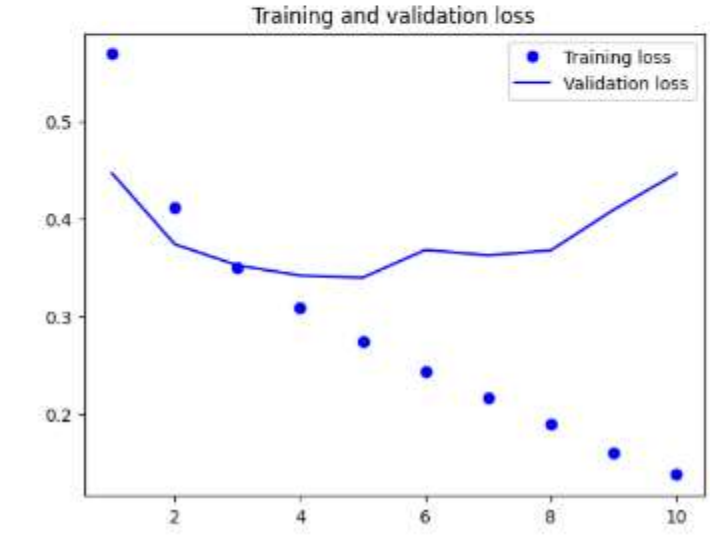
Although the accuracy on the training set has decreased, it can be observed that there is no significant difference in accuracy across the three sets: training, validation, and test.

2. **SCENARIO 2:**

2.1. Model's performance:

```
Total params: 1,034,113 (3.94 MB)
Trainable params: 1,034,113 (3.94 MB)
Non-trainable params: 0 (0.00 B)
Epoch 1/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 179s 275ms/step - accuracy: 0.6191 - loss: 0.6338 - val_accuracy: 0.7982 - val_loss: 0.4462
Epoch 2/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 201s 275ms/step - accuracy: 0.8064 - loss: 0.4360 - val_accuracy: 0.8358 - val_loss: 0.3737
Epoch 3/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 206s 281ms/step - accuracy: 0.8426 - loss: 0.3637 - val_accuracy: 0.8476 - val_loss: 0.3521
Epoch 4/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 193s 266ms/step - accuracy: 0.8679 - loss: 0.3184 - val_accuracy: 0.8552 - val_loss: 0.3416
Epoch 5/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 170s 273ms/step - accuracy: 0.8856 - loss: 0.2828 - val_accuracy: 0.8542 - val_loss: 0.3396
Epoch 6/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 168s 269ms/step - accuracy: 0.9018 - loss: 0.2508 - val_accuracy: 0.8596 - val_loss: 0.3680
Epoch 7/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 198s 263ms/step - accuracy: 0.9123 - loss: 0.2256 - val_accuracy: 0.8526 - val_loss: 0.3623
Epoch 8/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 178s 285ms/step - accuracy: 0.9264 - loss: 0.1934 - val_accuracy: 0.8518 - val_loss: 0.3674
Epoch 9/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 190s 266ms/step - accuracy: 0.9375 - loss: 0.1660 - val_accuracy: 0.8496 - val_loss: 0.4089
Epoch 10/10
625/625 ━━━━━━━━━━━━━━━━━━━━ 202s 267ms/step - accuracy: 0.9500 - loss: 0.1428 - val_accuracy: 0.8536 - val_loss: 0.4460
782/782 ━━━━━━━━━━━━━━━━━━━━ 58s 71ms/step - accuracy: 0.8448 - loss: 0.3453
Test acc: 0.846
```





H. The results show a clear improvement in accuracy for all three datasets. The loss function value is also reduced and the model shows good predictive ability with unseen data in Validation and testset.