CS577 Assignment 5: Report

Theo Guidroz
Department of Computer Science
Illinois Institute of Technology

May 8, 2021

Abstract

This is a report for the first programming question of Assignment 5. It serves as an introduction sentiment and topic classification.

Problem Statement

Build multiple models to perform sentiment and topic classification. Evaluate and compare their performances. The models will include embedding and LTSM layers.

Implementation details

For question 1:

1. Load the IMDB data and split it into training, testing, and validation.
2. Build a first neural network with a trainable embedding layer and 2 fully connected layers.

```
Model: "sequential_16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_15 (Embedding)     (None, 100, 100)          1000000
_________________________________________________________________
flatten_15 (Flatten)         (None, 10000)             0
_________________________________________________________________
dense_30 (Dense)             (None, 32)                320032
_________________________________________________________________
dense_31 (Dense)             (None, 1)                 33
=================================================================
Total params: 1,320,065
Trainable params: 1,320,065
Non-trainable params: 0
```

3. Download the GloVe word embedding file and load it into an embedding matrix.
4. Train a model with the embedding layer weights frozen.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_16 (Embedding)     (None, 100, 100)          1000000
_________________________________________________________________
flatten_16 (Flatten)         (None, 10000)             0
_________________________________________________________________
dense_32 (Dense)             (None, 32)                320032
_________________________________________________________________
dense_33 (Dense)             (None, 1)                 33
=================================================================
Total params: 1,320,065
Trainable params: 320,065
Non-trainable params: 1,000,000
```

5. Repeat step 4 but with the embedding layer weights trainable.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_17 (Embedding)     (None, 100, 100)          1000000
_________________________________________________________________
flatten_17 (Flatten)         (None, 10000)             0
_________________________________________________________________
dense_34 (Dense)             (None, 32)                320032
_________________________________________________________________
dense_35 (Dense)             (None, 1)                 33
=================================================================
Total params: 1,320,065
Trainable params: 1,320,065
Non-trainable params: 0
```

6. Replace the fully connected layer with an LTSM layer

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_19 (Embedding)     (None, 100, 100)          1000000
_________________________________________________________________
lstm_1 (LSTM)                (None, 100, 32)           17024
_________________________________________________________________
flatten_19 (Flatten)         (None, 3200)              0
_________________________________________________________________
dense_37 (Dense)             (None, 1)                 3201
=================================================================
Total params: 1,020,225
Trainable params: 1,020,225
Non-trainable params: 0
```

7. Compare all the results.

For question 2:

1. Load the Reuters data and split it into training, testing, and validation.
2. Build a first neural network with a trainable embedding layer and 2 fully connected layers.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 100, 100)          1000000
_________________________________________________________________
flatten_1 (Flatten)          (None, 10000)             0
_________________________________________________________________
dense_2 (Dense)              (None, 32)                320032
_________________________________________________________________
dense_3 (Dense)              (None, 46)                1518
=================================================================
Total params: 1,321,550
Trainable params: 1,321,550
Non-trainable params: 0
```

3. Download the GloVe word embedding file and load it into an embedding matrix.
4. Train with another model with the embedding layer weights frozen.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (None, 100, 100)          1000000
_________________________________________________________________
flatten_3 (Flatten)          (None, 10000)             0
_________________________________________________________________
dense_6 (Dense)              (None, 32)                320032
_________________________________________________________________
dense_7 (Dense)              (None, 46)                1518
=================================================================
Total params: 1,321,550
Trainable params: 321,550
Non-trainable params: 1,000,000
```

5. Repeat step 4 but with the embedding layer weights trainable.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_4 (Embedding)      (None, 100, 100)          1000000
_________________________________________________________________
flatten_4 (Flatten)          (None, 10000)             0
_________________________________________________________________
dense_8 (Dense)              (None, 32)                320032
_________________________________________________________________
dense_9 (Dense)              (None, 46)                1518
=================================================================
Total params: 1,321,550
Trainable params: 1,321,550
Non-trainable params: 0
_________________________________________________________________
```

6. Replace the fully connected layer with an LTSM layer.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_5 (Embedding)      (None, 100, 100)          1000000
_________________________________________________________________
lstm (LSTM)                  (None, 100, 32)           17024
_________________________________________________________________
flatten_5 (Flatten)          (None, 3200)              0
_________________________________________________________________
dense_10 (Dense)             (None, 46)                147246
=================================================================
Total params: 1,164,270
Trainable params: 1,164,270
Non-trainable params: 0
```

7. Add a second LTSM layer.

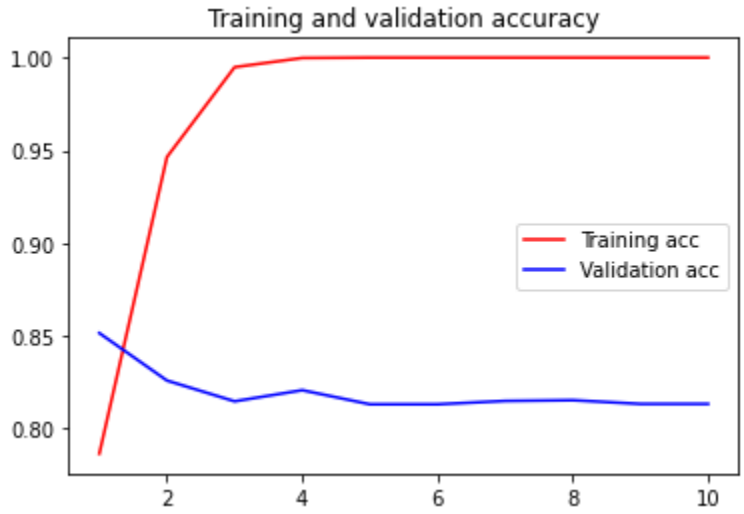
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_6 (Embedding)      (None, 100, 100)          1000000
_________________________________________________________________
lstm_1 (LSTM)                (None, 100, 32)           17024
_________________________________________________________________
lstm_2 (LSTM)                (None, 100, 32)           8320
_________________________________________________________________
flatten_6 (Flatten)          (None, 3200)              0
_________________________________________________________________
dense_11 (Dense)             (None, 46)                147246
=================================================================
Total params: 1,172,590
Trainable params: 1,172,590
Non-trainable params: 0
```

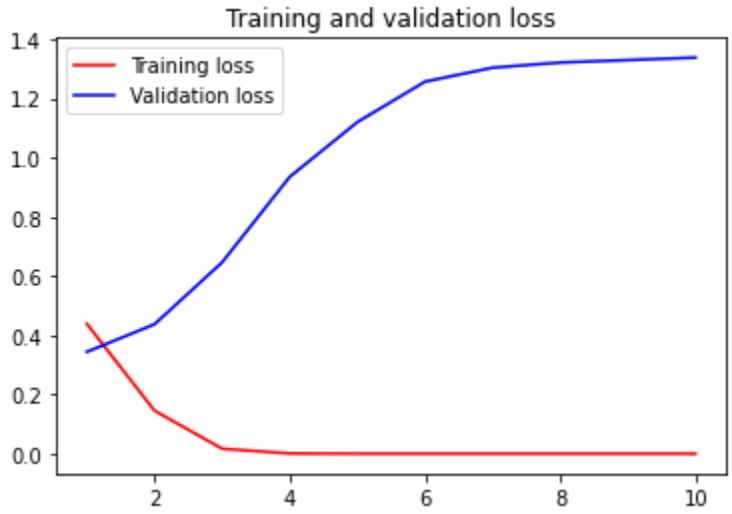
8. Compare all the results.

Results and discussion

Sentiment Classification

Model 1: 2 Dense layers

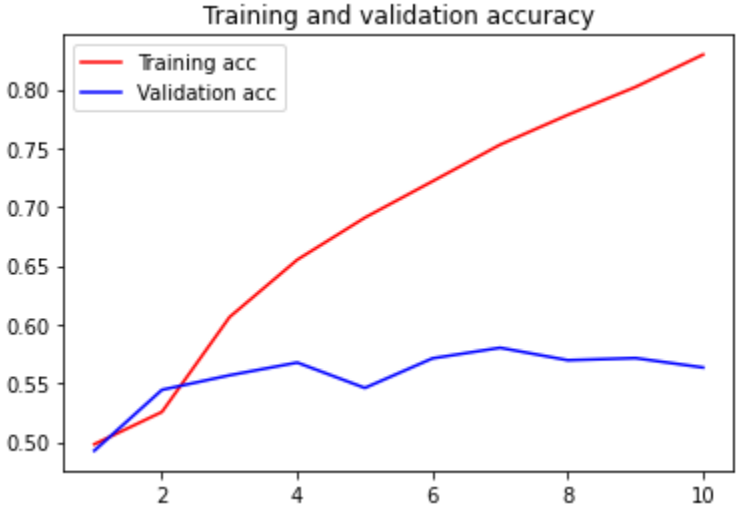


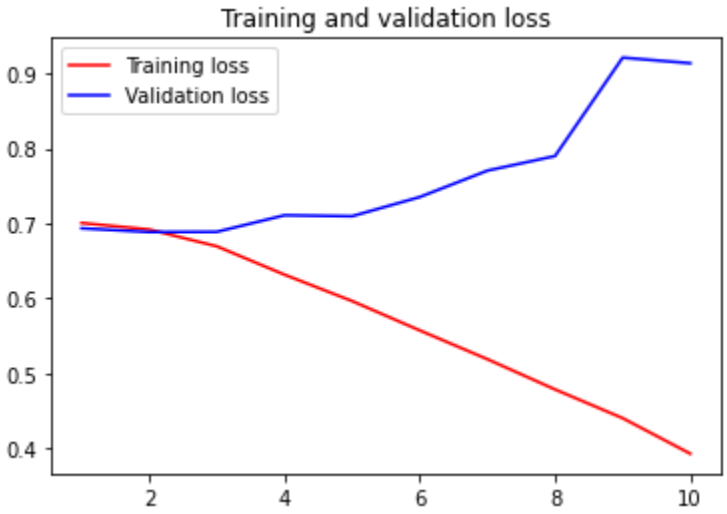


The graph above shows that the model overfits very rapidly.

```
Model has an accuracy of 0.8179600238800049
```

Model 2: Embedding layer frozen

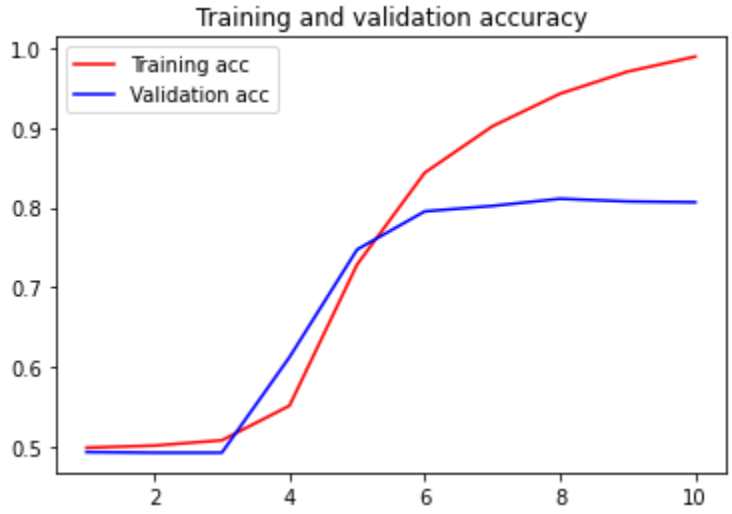


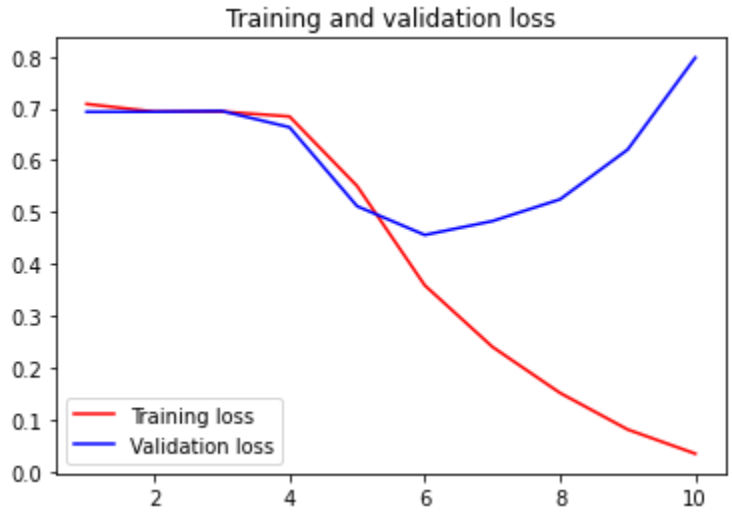


Using the frozen layer, the model takes more time to overfit but does not perform as well.

```
Accuracy with frozen weights: 0.5626800060272217
```

Model 3: Embedding layer trainable

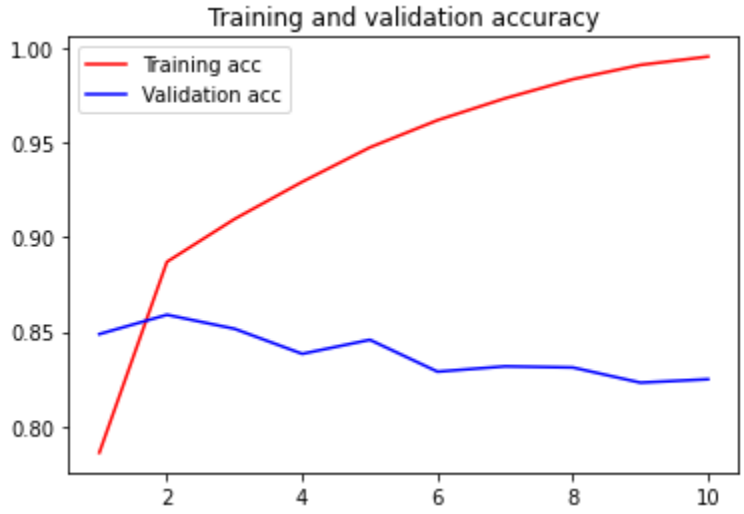


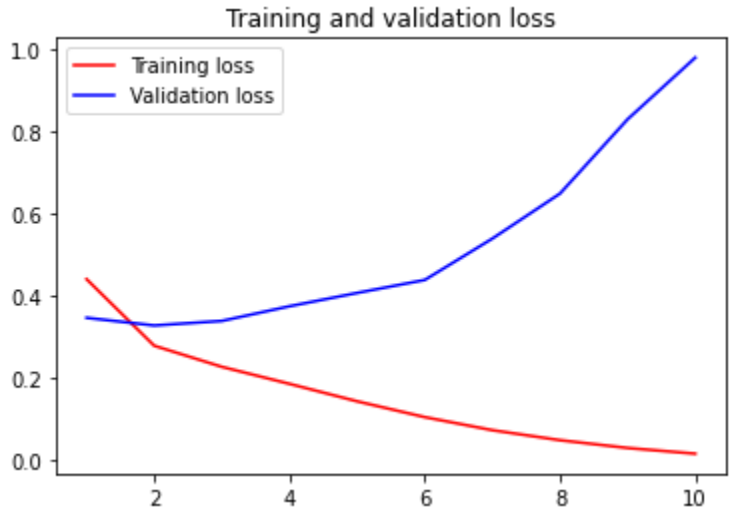


Using the trainable layer, the model takes more time to overfit than both previous models. It also out outperforms the same model but with the frozen layer by 30%.

```
Accuracy without frozen weights: 0.8041599988937378
```

Model 4: LTSM layer



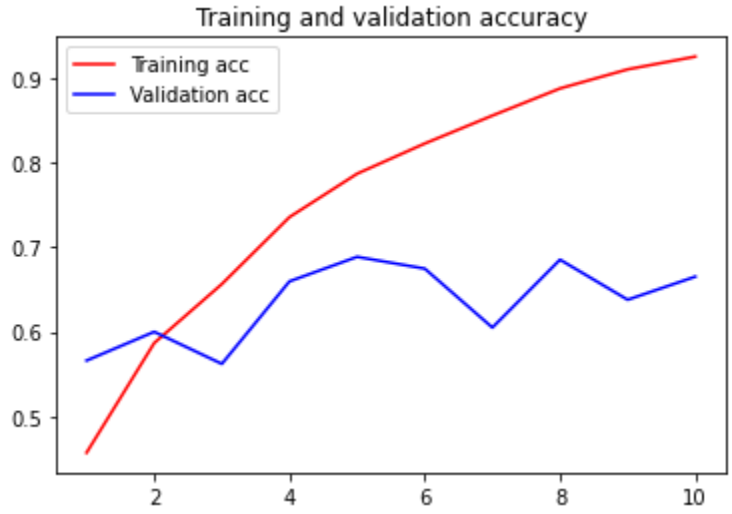


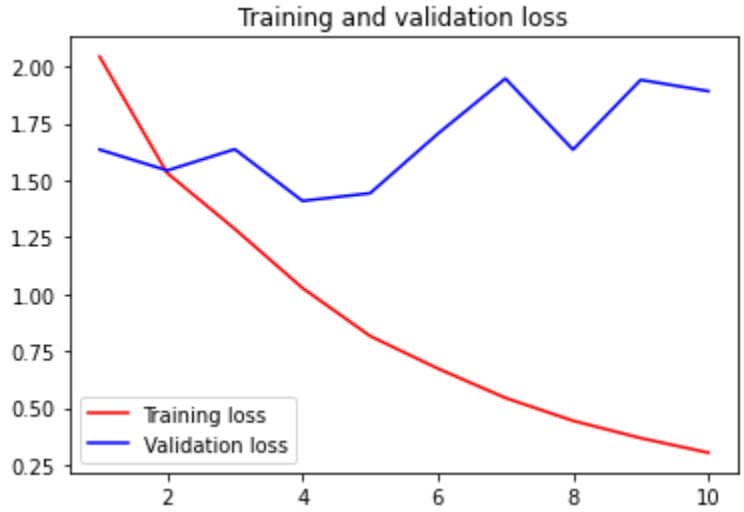
The model overfits rapidly but replacing the dense layer with an LSTM layer slightly improves the accuracy.

```
Accuracy with LSTM: 0.8162800073623657
```

Topic Classification

Model 1: 2 Dense layers

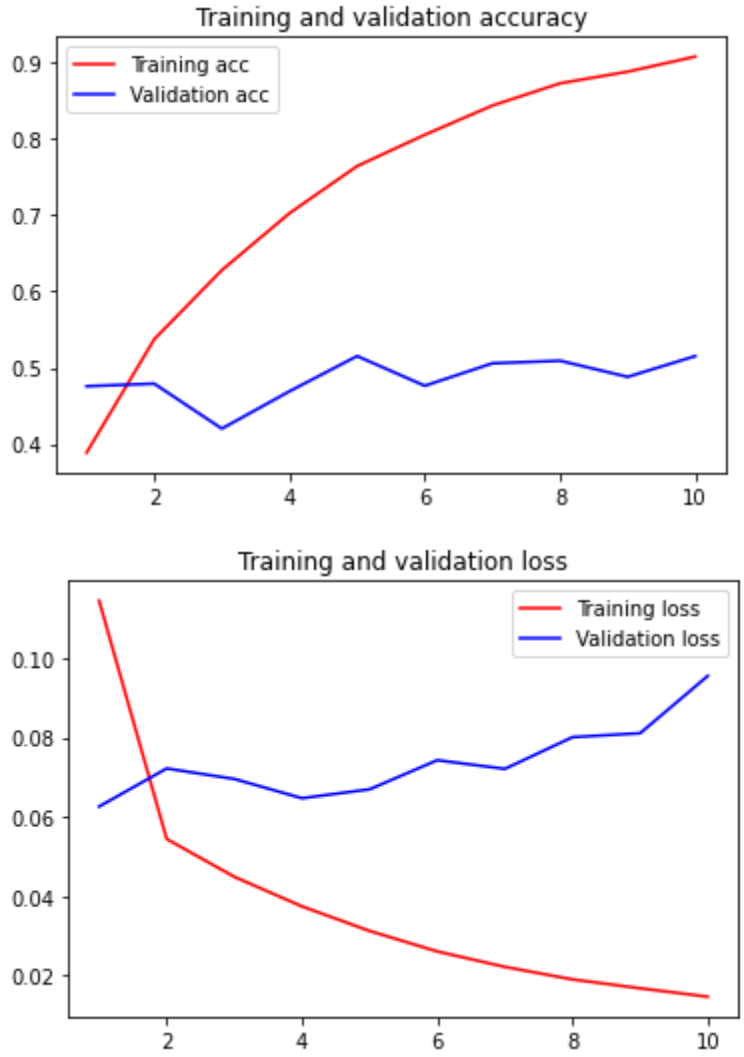




The graph above shows that the model overfits after 4 epochs.

```
Model has an accuracy of 0.6918966770172119
```

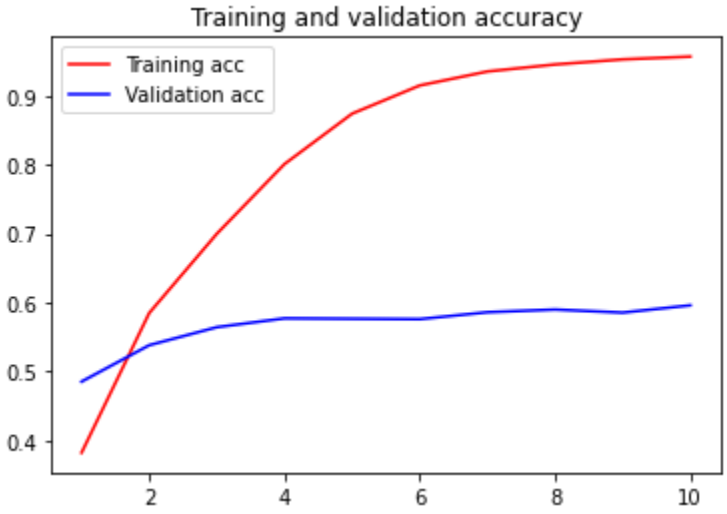
Model 2: Embedding layer frozen

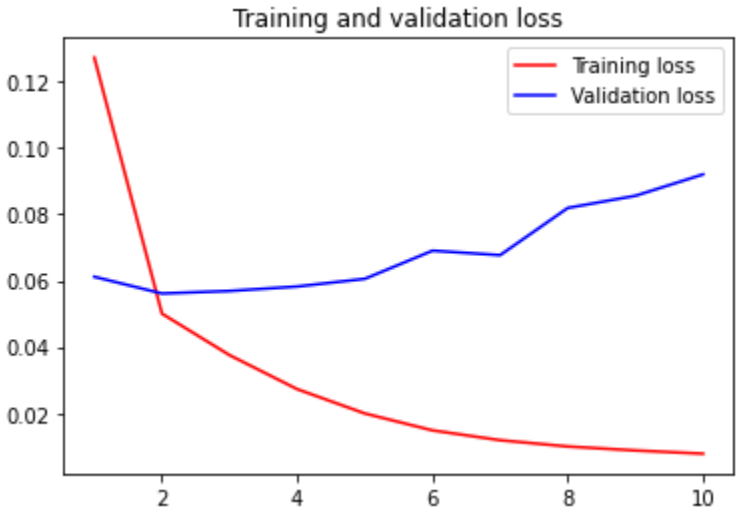


With the frozen layer, the model overfits rapidly and the accuracy decreases.

```
Accuracy with frozen weights: 0.5053428411483765
```

Model 3: Embedding layer trainable

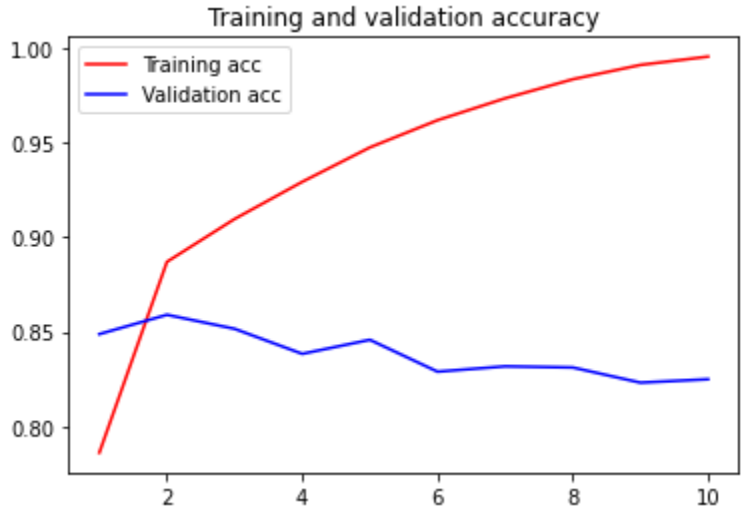


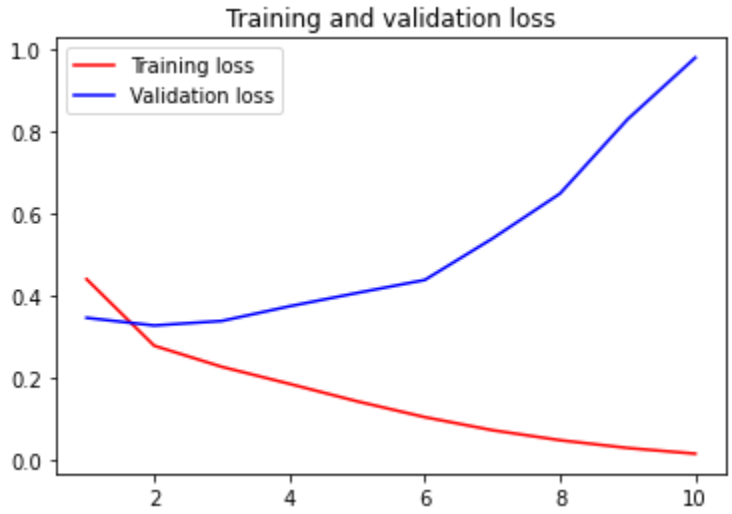


Using the trainable layer, the model takes more time to overfit than both previous models. It also out outperforms the same model but with the frozen layer.

`Accuracy without frozen weights: 0.589937686920166`

Model 4: LTSM layer





Using the LTSM layer overfits rapidly but increases the accuracy.

```
Accuracy with LSTM: 0.6727515459060669
```

Model 4: LTSM layer

Using 2 LTSM layers helps generalizing but the accuracy decreases.

```
Accuracy with 2 LSTM: 0.642920732498169
```

Conclusion

In both cases, using one LTSM provides the best accuracy. The models tend to overfit very rapidly so there is no need to train on many epochs. This assignment was a great introduction to embedding and LTSM layers.

References

Professor’s slides