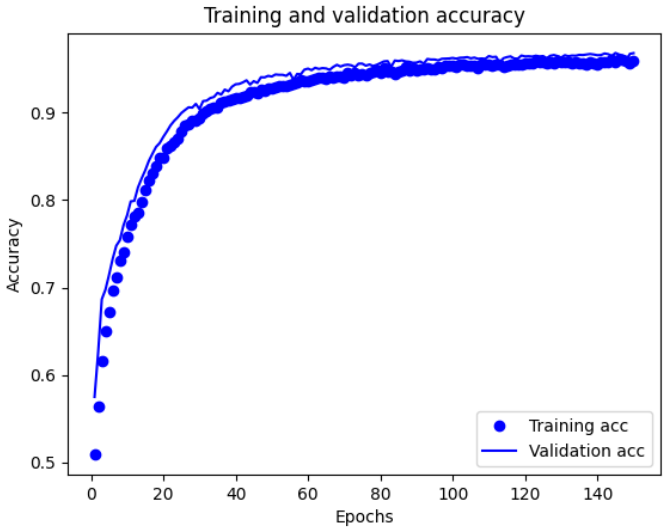
Model testing

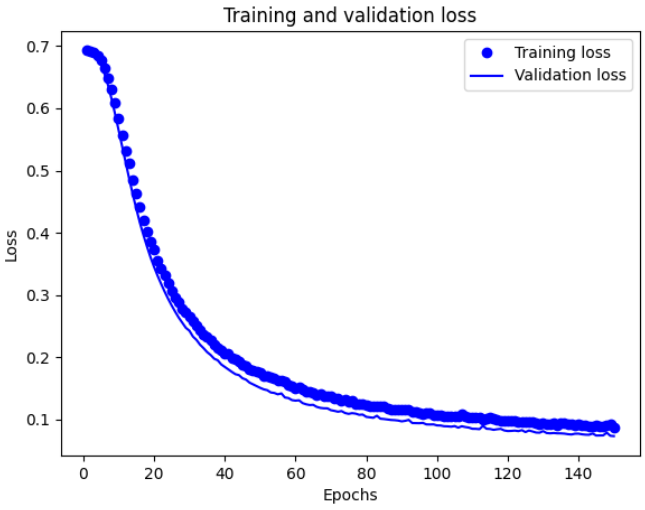
Model used

```
# Create model
vocab_size = 10000
batch = 512
model = keras.Sequential()
model.add(keras.layers.Embedding(vocab_size, 16))
# model.add(keras.layers.Embedding(vocab_size, 32, input_length=25))
model.add(keras.layers.Dropout(0.2))
model.add(keras.layers.GlobalAveragePooling1D())
# model.add(keras.layers.LSTM(64))
model.add(keras.layers.Dense(16, activation=tf.nn.relu))
model.add(keras.layers.Dropout(0.2))
# model.add(keras.layers.Dense(16, activation=tf.nn.tanh))
# model.add(keras.layers.Dropout(0.2))
model.add(keras.layers.Dense(1, activation=tf.nn.sigmoid))
```

Using all training data

This test is using all of the available training data (10k sentences) for training. The testing data is divided in half, one half for testing and the other half for validation. The epoch size is 150.

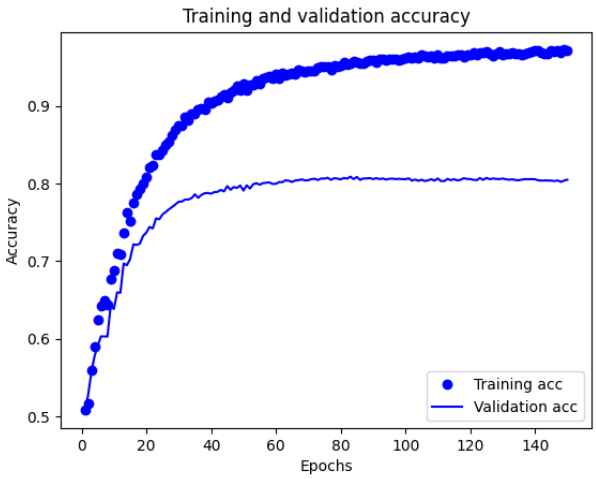


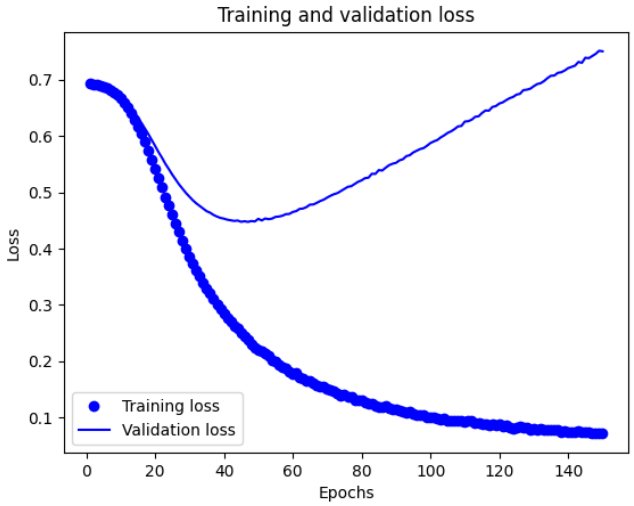


The model accuracy reaches, approximately, 90% accuracy after 40 epochs. The loss, on the other hand, requires more epochs to reach an acceptable level. To reach a loss of 10% we need to train the model using more than 80 epochs. This test is using 2:1 ratio for training data to validation data (10k for training 5k for validation).

Using half of the training data

Everything is the same as the previous test, except, in this test I used half of the training data (5k sentences, 1:1 ratio for training data to validation data).

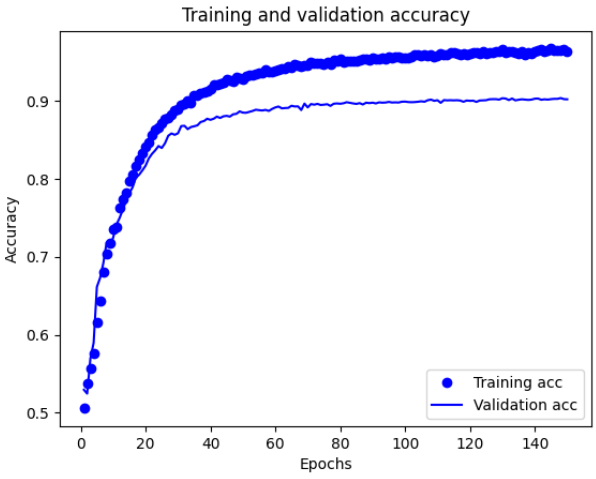


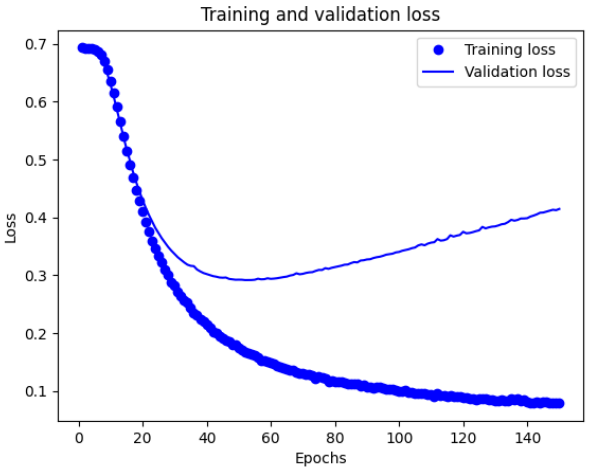


The accuracy stops improving after 30-40 epochs and the loss stops going down after 40 epochs. Using half of the training data would yield a model with 80% accuracy and 50% loss.

95% accuracy hunt

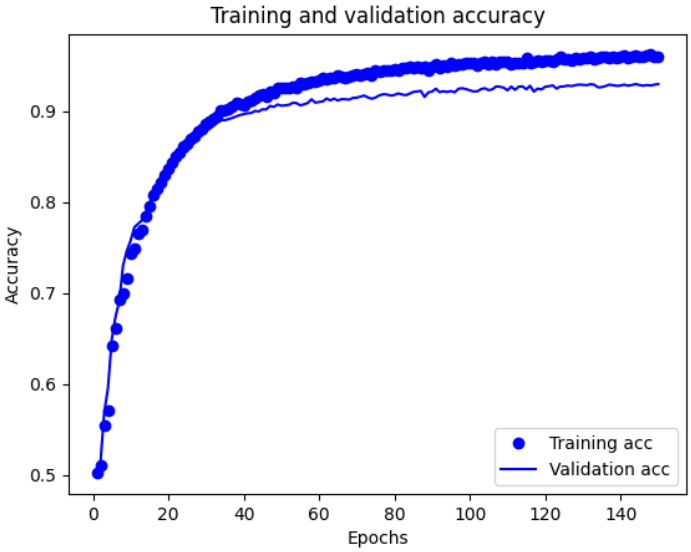
8k sentences

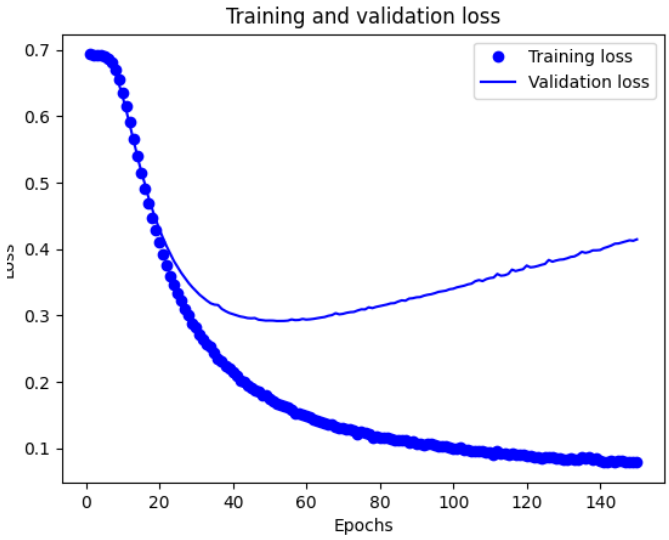




Almost 90% accuracy with a high loss of 30% after 40 epochs. Model is still unusable; accuracy is too low and the loss too high.

9k sentences

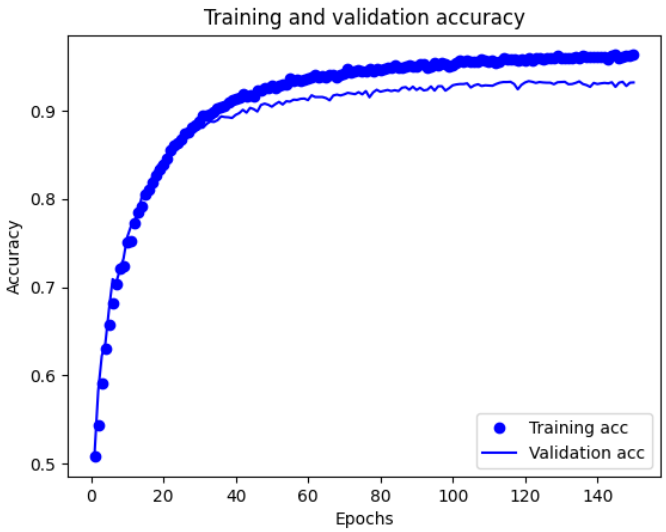


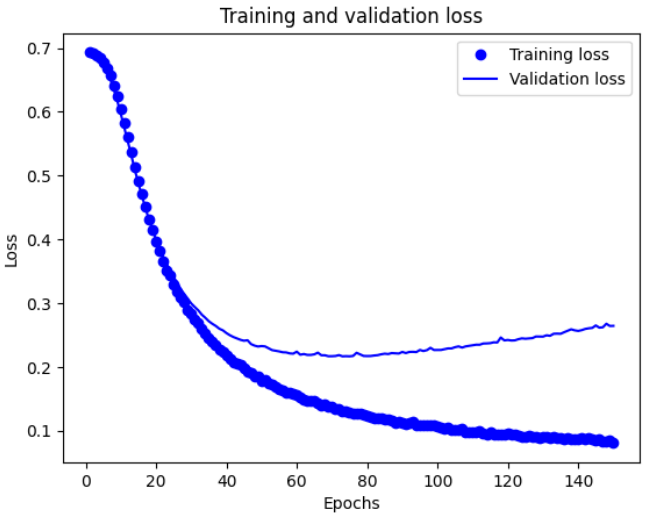


The accuracy is better, 90% after 40 epochs but the loss is still very high at approximately 30% after 40 epochs.

Investigating training/validation ratio

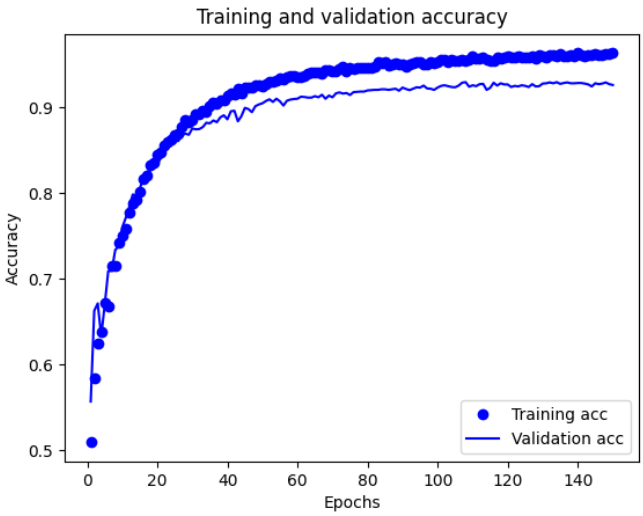
9k training with 4k validation

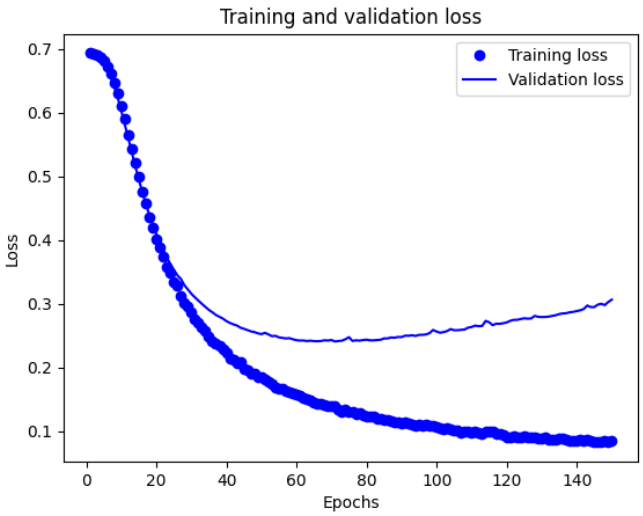




The accuracy is the same as the “9k sentences” test but the loss is slightly lower.

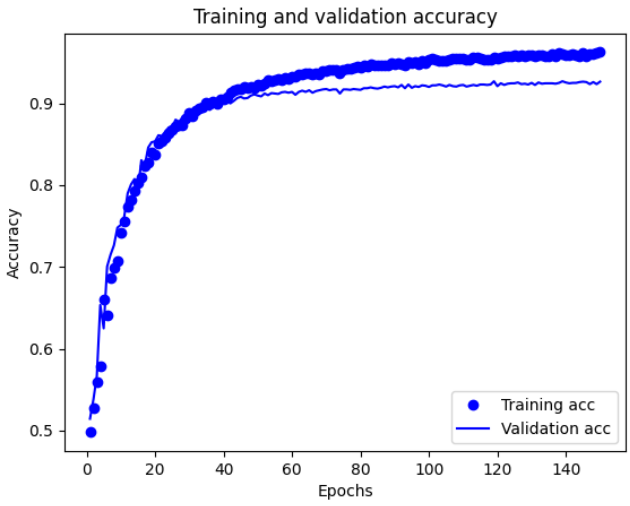
9k training with 3k validation

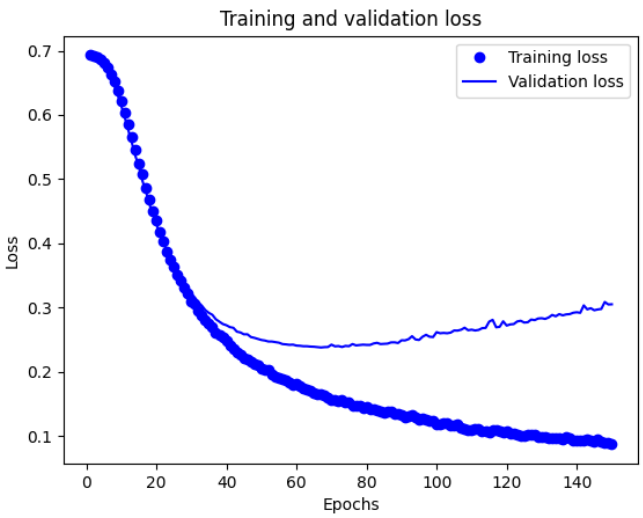




The accuracy and loss are basically the same as previous test.

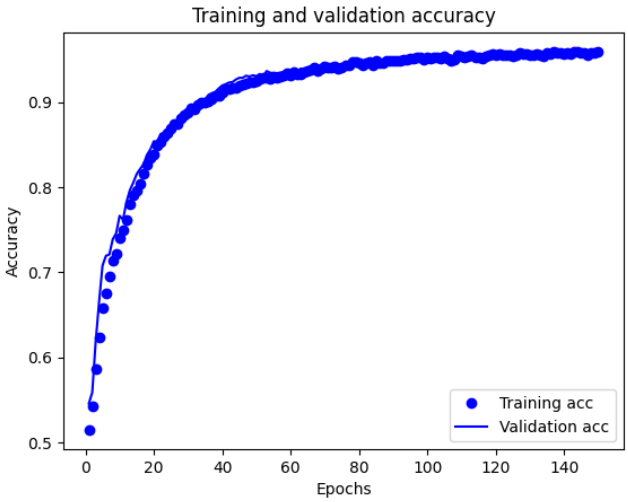
9k training with 2k validation

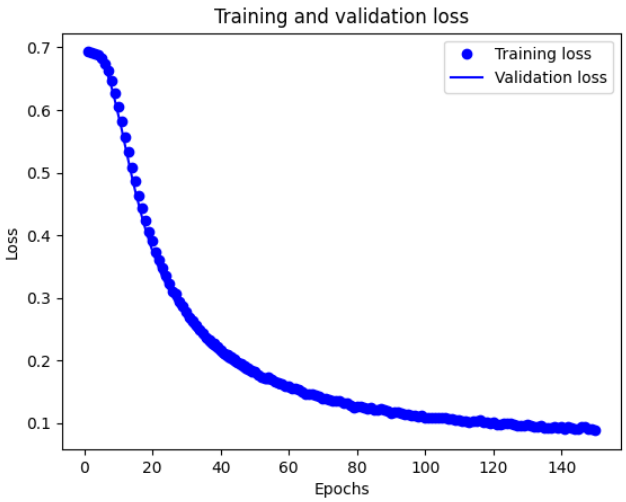




Like previous tests, scaling back the amount of data used for validation does not seem to affect the loss.

10k training 2k validation





This test shows that the loss and accuracy is not significantly affected by reducing the amount of data used for validation.

Conclusion/discussion

To achieve a 95% accuracy, we would have to use all the available training data, or at least 9k if 93% is acceptable. Decreasing the amount training data used affects both the accuracy and loss. Through testing we can show that the amount of data used for validation is not that important, it is more important to use all available training data (compare the “10k training and 2k validation” test with the first test “using all the data”). This means that the total amount of data required can be reduced since we can use more of it for training and much less of it for validation.