

Text and Sequence Assignment4

Downloading and Preparing the IMDB Dataset

```
In [1]: !curl -O https://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz
!tar -xf aclImdb_v1.tar.gz
!rm -r aclImdb/train/unsup
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current				
			Dload	Upload	Total	Spent	Left	Speed			
100	80.2M	100	80.2M	0	0	5768k	0	0:00:14	0:00:14	--:--:--	12.7M

Preprocessing the Text Data

```
In [2]: import os, pathlib, shutil, random
from tensorflow import keras
batch_size = 32
base_dir= pathlib.Path("/content/aclImdb")
val_dir = base_dir/ "val"
train_dir = base_dir / "train"
for category in ("neg", "pos"):
    os.makedirs(val_dir / category, exist_ok=True )

    files = os.listdir(train_dir / category)
    random.Random(1337).shuffle(files)
    num_val_samples = int(0.2 * len(files))
    val_files = files[-num_val_samples:]
    for fname in val_files:
        shutil.move(train_dir / category / fname,
                    val_dir / category / fname)

train_ds = keras.utils.text_dataset_from_directory(
    "aclImdb/train", batch_size=batch_size
).take(100)

val_ds = keras.utils.text_dataset_from_directory(
    "/content/aclImdb/val", batch_size=batch_size
).take(10000)
test_ds = keras.utils.text_dataset_from_directory(
    "aclImdb/test", batch_size=batch_size
)
text_only_train_ds = train_ds.map(lambda x, y: x)
```

Found 20000 files belonging to 2 classes.
Found 5000 files belonging to 2 classes.
Found 25000 files belonging to 2 classes.

Vectorizing the Text Sequences

```
In [3]: from tensorflow.keras import layers

max_length = 150
```

```

max_tokens = 10000
text_vectorization = layers.TextVectorization(
    max_tokens=max_tokens,
    output_mode="int",
    output_sequence_length=max_length,
)
text_vectorization.adapt(text_only_train_ds)

int_train_ds = train_ds.map(
    lambda x, y: (text_vectorization(x), y),
    num_parallel_calls=4)
int_val_ds = val_ds.map(
    lambda x, y: (text_vectorization(x), y),
    num_parallel_calls=4)
int_test_ds = test_ds.map(
    lambda x, y: (text_vectorization(x), y),
    num_parallel_calls=4)

```

Baseline Model: One-Hot Encoded Vectors

```

In [4]: import tensorflow as tf
inputs = keras.Input(shape=(None,), dtype="int64")
embedded = tf.one_hot(inputs, depth=max_tokens)
x = layers.Bidirectional(layers.LSTM(32))(embedded)
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs, outputs)
model.compile(optimizer="rmsprop",
              loss="binary_crossentropy",
              metrics=["accuracy"])
model.summary()

```

Model: "model"

Layer (type)	Output Shape	Param #
=====		
input_1 (InputLayer)	[(None, None)]	0
tf.one_hot (TFOpLambda)	(None, None, 10000)	0
bidirectional (Bidirectional)	(None, 64)	2568448
dropout (Dropout)	(None, 64)	0
dense (Dense)	(None, 1)	65
=====		
Total params: 2568513 (9.80 MB)		
Trainable params: 2568513 (9.80 MB)		
Non-trainable params: 0 (0.00 Byte)		

One Hot Model

```
In [5]: callbacks = [
        keras.callbacks.ModelCheckpoint("one_hot_bidir_lstm.x",
                                         save_best_only=True)
    ]

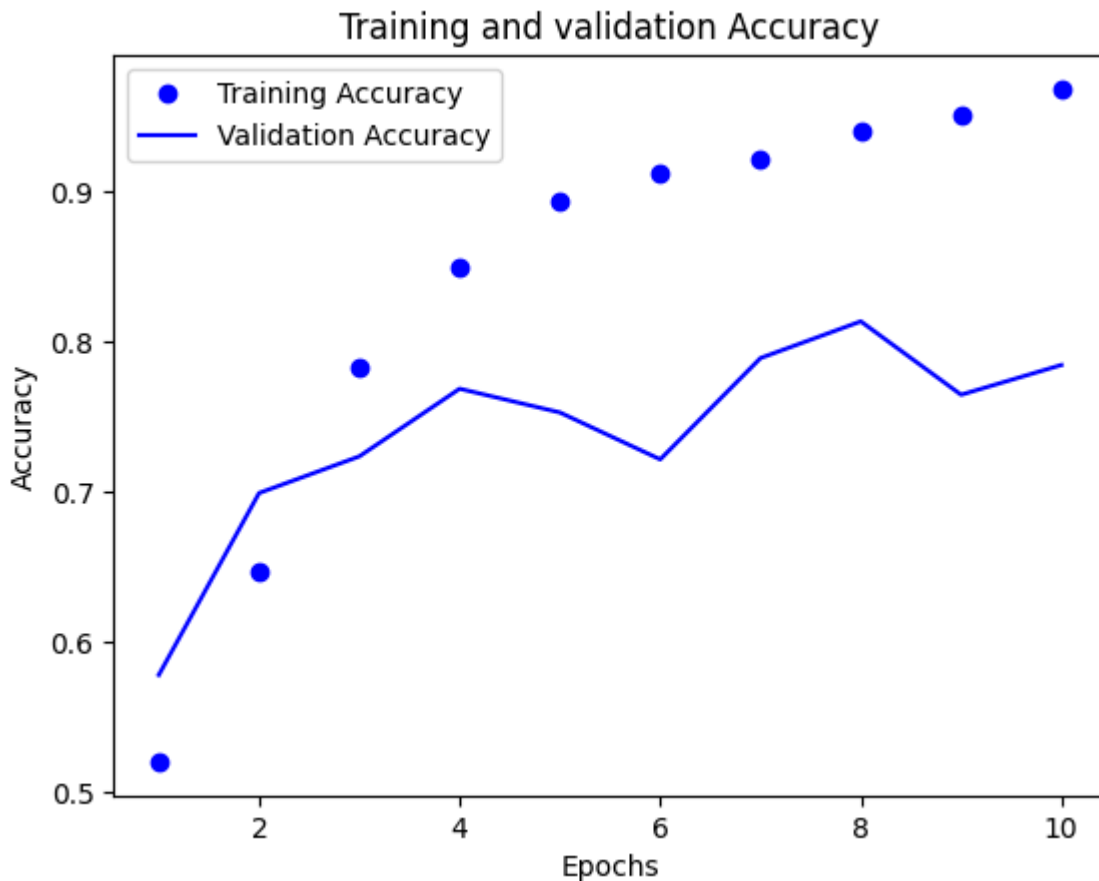
    model.compile(optimizer="rmsprop", loss="binary_crossentropy", metrics=["accuracy"])
    history = model.fit(int_train_ds, validation_data=int_val_ds, epochs=10, callbacks=callbacks)

    model = keras.models.load_model("one_hot_bidir_lstm.x")
    print(f"Test acc: {model.evaluate(int_test_ds)[1]:.3f}")

import matplotlib.pyplot as plt

accuracy = history.history["accuracy"]
val_accuracy = history.history["val_accuracy"]
epochs = range(1, len(accuracy) + 1)
plt.figure()
plt.plot(epochs, accuracy, "bo", label="Training Accuracy")
plt.plot(epochs, val_accuracy, "b", label="Validation Accuracy")
plt.title("Training and validation Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()
```

Epoch 1/10
100/100 [=====] - 25s 201ms/step - loss: 0.6923 - accuracy: 0.5203 - val_loss: 0.6899 - val_accuracy: 0.5780
Epoch 2/10
100/100 [=====] - 16s 166ms/step - loss: 0.6413 - accuracy: 0.6463 - val_loss: 0.5929 - val_accuracy: 0.6990
Epoch 3/10
100/100 [=====] - 6s 65ms/step - loss: 0.5122 - accuracy: 0.7825 - val_loss: 0.6595 - val_accuracy: 0.7234
Epoch 4/10
100/100 [=====] - 17s 166ms/step - loss: 0.3961 - accuracy: 0.8497 - val_loss: 0.4821 - val_accuracy: 0.7684
Epoch 5/10
100/100 [=====] - 7s 69ms/step - loss: 0.3231 - accuracy: 0.8934 - val_loss: 0.4929 - val_accuracy: 0.7526
Epoch 6/10
100/100 [=====] - 6s 62ms/step - loss: 0.2650 - accuracy: 0.9116 - val_loss: 0.6802 - val_accuracy: 0.7214
Epoch 7/10
100/100 [=====] - 16s 166ms/step - loss: 0.2292 - accuracy: 0.9206 - val_loss: 0.4809 - val_accuracy: 0.7888
Epoch 8/10
100/100 [=====] - 16s 158ms/step - loss: 0.1903 - accuracy: 0.9397 - val_loss: 0.4312 - val_accuracy: 0.8132
Epoch 9/10
100/100 [=====] - 6s 64ms/step - loss: 0.1542 - accuracy: 0.9506 - val_loss: 0.7079 - val_accuracy: 0.7644
Epoch 10/10
100/100 [=====] - 9s 89ms/step - loss: 0.1214 - accuracy: 0.9678 - val_loss: 0.5657 - val_accuracy: 0.7842
782/782 [=====] - 16s 20ms/step - loss: 0.4488 - accuracy: 0.8095
Test acc: 0.810



Developing the Embedding Layer

```
In [6]: embedding_layer = layers.Embedding(input_dim=max_tokens, output_dim=256)
```

Model with Trainable Embedding Layer

```
In [7]: inputs = keras.Input(shape=(None,), dtype="int64")
embedded = layers.Embedding(input_dim=max_tokens, output_dim=256)(inputs)
x = layers.Bidirectional(layers.LSTM(32))(embedded)
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs, outputs)
model.compile(optimizer="rmsprop",
              loss="binary_crossentropy",
              metrics=["accuracy"])
model.summary()

callbacks = [
    keras.callbacks.ModelCheckpoint("embeddings_bidir_gru.x",
                                   save_best_only=True)
]
history = model.fit(int_train_ds, validation_data=int_val_ds, epochs=10, callbacks=
model = keras.models.load_model("embeddings_bidir_gru.x")
print(f"Test acc: {model.evaluate(int_test_ds)[1]:.3f}")

accuracy = history.history["accuracy"]
val_accuracy = history.history["val_accuracy"]
```

```
epochs = range(1, len(accuracy) + 1)
plt.figure()
plt.plot(epochs, accuracy, "bo", label="Training Accuracy")
plt.plot(epochs, val_accuracy, "b", label="Validation Accuracy")
plt.title("Training and validation Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()
```

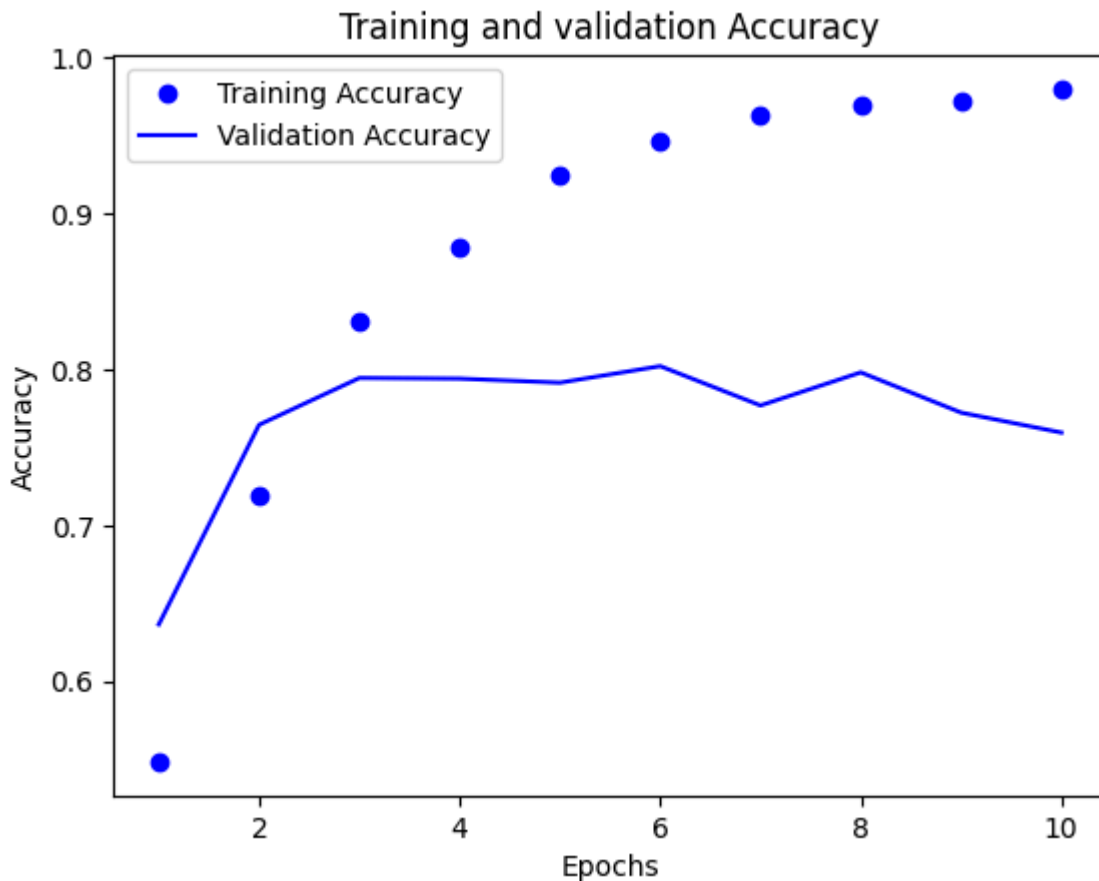
Model: "model_1"

Layer (type)	Output Shape	Param #
=====		
input_2 (InputLayer)	[(None, None)]	0
embedding_1 (Embedding)	(None, None, 256)	2560000
bidirectional_1 (Bidirectional)	(None, 64)	73984
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

=====

Total params: 2634049 (10.05 MB)
Trainable params: 2634049 (10.05 MB)
Non-trainable params: 0 (0.00 Byte)

Epoch 1/10
100/100 [=====] - 22s 187ms/step - loss: 0.6856 - accuracy: 0.5484 - val_loss: 0.6397 - val_accuracy: 0.6364
Epoch 2/10
100/100 [=====] - 15s 151ms/step - loss: 0.5710 - accuracy: 0.7194 - val_loss: 0.5037 - val_accuracy: 0.7644
Epoch 3/10
100/100 [=====] - 14s 144ms/step - loss: 0.4213 - accuracy: 0.8306 - val_loss: 0.4592 - val_accuracy: 0.7944
Epoch 4/10
100/100 [=====] - 5s 49ms/step - loss: 0.3317 - accuracy: 0.8788 - val_loss: 0.4721 - val_accuracy: 0.7940
Epoch 5/10
100/100 [=====] - 3s 34ms/step - loss: 0.2223 - accuracy: 0.9250 - val_loss: 0.4796 - val_accuracy: 0.7914
Epoch 6/10
100/100 [=====] - 5s 49ms/step - loss: 0.1773 - accuracy: 0.9463 - val_loss: 0.5590 - val_accuracy: 0.8020
Epoch 7/10
100/100 [=====] - 3s 33ms/step - loss: 0.1189 - accuracy: 0.9631 - val_loss: 0.6186 - val_accuracy: 0.7768
Epoch 8/10
100/100 [=====] - 3s 26ms/step - loss: 0.1060 - accuracy: 0.9688 - val_loss: 0.6379 - val_accuracy: 0.7980
Epoch 9/10
100/100 [=====] - 5s 53ms/step - loss: 0.0818 - accuracy: 0.9722 - val_loss: 0.7108 - val_accuracy: 0.7722
Epoch 10/10
100/100 [=====] - 3s 31ms/step - loss: 0.0666 - accuracy: 0.9797 - val_loss: 0.8444 - val_accuracy: 0.7594
782/782 [=====] - 8s 9ms/step - loss: 0.4607 - accuracy: 0.7928
Test acc: 0.793



Masking Padded Sequences in the Embedding Layer

```
In [8]: inputs = keras.Input(shape=(None,), dtype="int64")
embedded = layers.Embedding(
    input_dim=max_tokens, output_dim=256, mask_zero=True)(inputs)
x = layers.Bidirectional(layers.LSTM(32))(embedded)
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs, outputs)
model.compile(optimizer="rmsprop",
              loss="binary_crossentropy",
              metrics=["accuracy"])
model.summary()

callbacks = [
    keras.callbacks.ModelCheckpoint("embeddings_bidir_gru_with_masking.x",
                                   save_best_only=True)
]
history = model.fit(int_train_ds, validation_data=int_val_ds, epochs=10, callbacks=
model = keras.models.load_model("embeddings_bidir_gru_with_masking.x")
print(f"Test acc: {model.evaluate(int_test_ds)[1]:.3f}")

accuracy = history.history["accuracy"]
val_accuracy = history.history["val_accuracy"]
epochs = range(1, len(accuracy) + 1)
plt.figure()
plt.plot(epochs, accuracy, "bo", label="Training Accuracy")
plt.plot(epochs, val_accuracy, "b", label="Validation Accuracy")
```



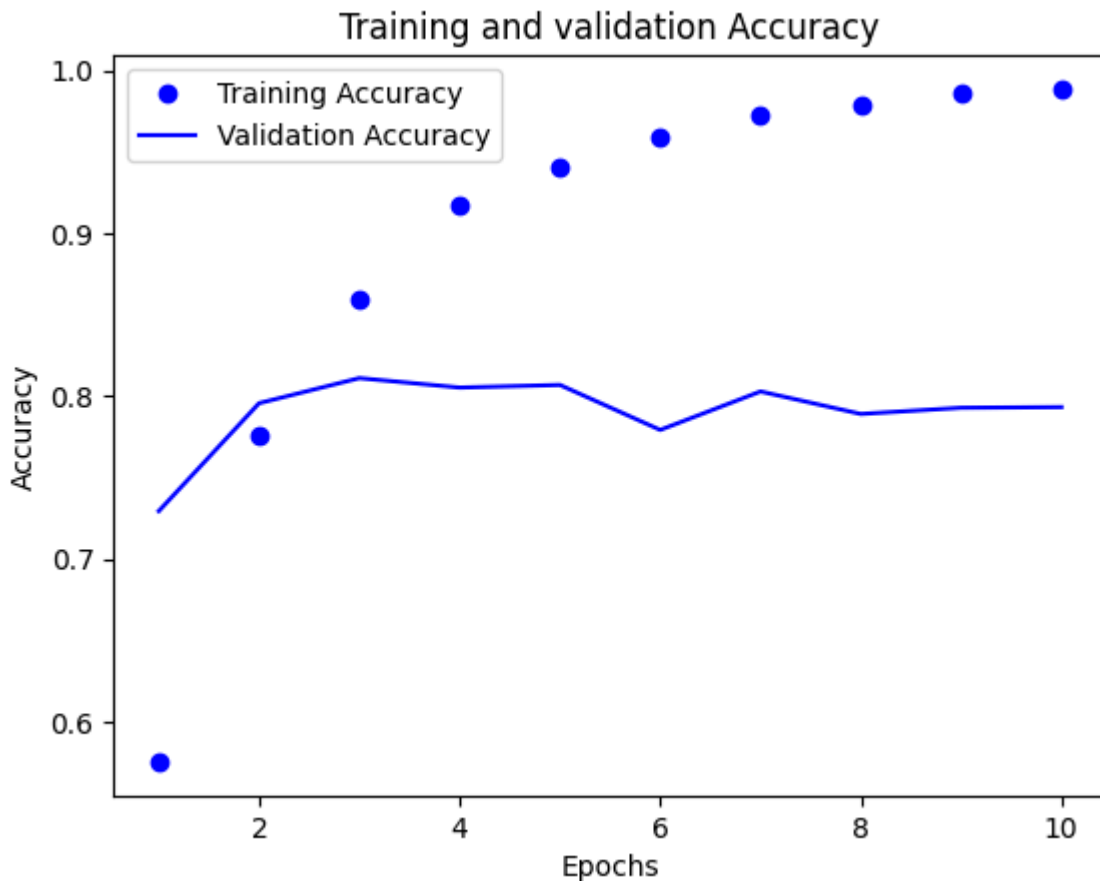
```
plt.title("Training and validation Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()
```

Model: "model_2"

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, None)]	0
embedding_2 (Embedding)	(None, None, 256)	2560000
bidirectional_2 (Bidirectional)	(None, 64)	73984
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65

Total params: 2634049 (10.05 MB)
Trainable params: 2634049 (10.05 MB)
Non-trainable params: 0 (0.00 Byte)

Epoch 1/10
100/100 [=====] - 37s 288ms/step - loss: 0.6712 - accuracy: 0.5756 - val_loss: 0.5780 - val_accuracy: 0.7294
Epoch 2/10
100/100 [=====] - 25s 249ms/step - loss: 0.4958 - accuracy: 0.7759 - val_loss: 0.4467 - val_accuracy: 0.7956
Epoch 3/10
100/100 [=====] - 23s 231ms/step - loss: 0.3352 - accuracy: 0.8594 - val_loss: 0.4232 - val_accuracy: 0.8110
Epoch 4/10
100/100 [=====] - 5s 51ms/step - loss: 0.2369 - accuracy: 0.9172 - val_loss: 0.4464 - val_accuracy: 0.8052
Epoch 5/10
100/100 [=====] - 5s 49ms/step - loss: 0.1736 - accuracy: 0.9400 - val_loss: 0.4379 - val_accuracy: 0.8068
Epoch 6/10
100/100 [=====] - 5s 49ms/step - loss: 0.1235 - accuracy: 0.9588 - val_loss: 0.5905 - val_accuracy: 0.7792
Epoch 7/10
100/100 [=====] - 4s 39ms/step - loss: 0.0874 - accuracy: 0.9728 - val_loss: 0.5418 - val_accuracy: 0.8028
Epoch 8/10
100/100 [=====] - 4s 38ms/step - loss: 0.0642 - accuracy: 0.9784 - val_loss: 0.5129 - val_accuracy: 0.7890
Epoch 9/10
100/100 [=====] - 4s 38ms/step - loss: 0.0463 - accuracy: 0.9856 - val_loss: 0.5664 - val_accuracy: 0.7928
Epoch 10/10
100/100 [=====] - 4s 45ms/step - loss: 0.0408 - accuracy: 0.9884 - val_loss: 0.6331 - val_accuracy: 0.7932
782/782 [=====] - 12s 10ms/step - loss: 0.4350 - accuracy: 0.8052
Test acc: 0.805



Using Pre-trained Word Embeddings (GloVe)

```
In [9]: !wget http://nlp.stanford.edu/data/glove.6B.zip
!unzip -q glove.6B.zip

--2024-05-05 20:24:02-- http://nlp.stanford.edu/data/glove.6B.zip
Resolving nlp.stanford.edu (nlp.stanford.edu)... 171.64.67.140
Connecting to nlp.stanford.edu (nlp.stanford.edu)|171.64.67.140|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://nlp.stanford.edu/data/glove.6B.zip [following]
--2024-05-05 20:24:03-- https://nlp.stanford.edu/data/glove.6B.zip
Connecting to nlp.stanford.edu (nlp.stanford.edu)|171.64.67.140|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://downloads.cs.stanford.edu/nlp/data/glove.6B.zip [following]
--2024-05-05 20:24:03-- https://downloads.cs.stanford.edu/nlp/data/glove.6B.zip
Resolving downloads.cs.stanford.edu (downloads.cs.stanford.edu)... 171.64.64.22
Connecting to downloads.cs.stanford.edu (downloads.cs.stanford.edu)|171.64.64.22|:
443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 862182613 (822M) [application/zip]
Saving to: 'glove.6B.zip'

glove.6B.zip          100%[=====>] 822.24M  5.16MB/s   in 2m 41s

2024-05-05 20:26:45 (5.11 MB/s) - 'glove.6B.zip' saved [862182613/862182613]
```



```
]
history = model.fit(int_train_ds, validation_data=int_val_ds, epochs=10, callbacks=
model = keras.models.load_model("glove_embeddings_sequence_model.x")
print(f"Test acc: {model.evaluate(int_test_ds)[1]:.3f}")

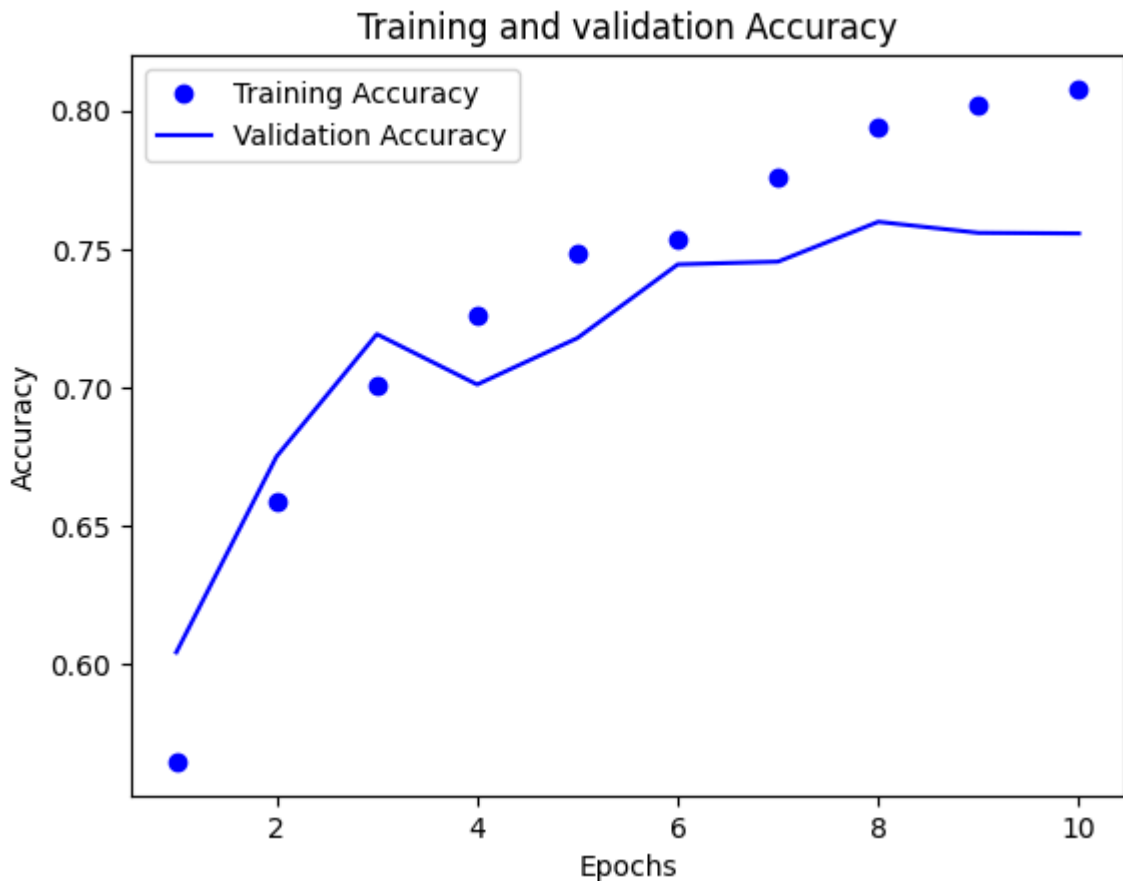
accuracy = history.history["accuracy"]
val_accuracy = history.history["val_accuracy"]
epochs = range(1, len(accuracy) + 1)
plt.figure()
plt.plot(epochs, accuracy, "bo", label="Training Accuracy")
plt.plot(epochs, val_accuracy, "b", label="Validation Accuracy")
plt.title("Training and validation Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()
```

Model: "model_3"

Layer (type)	Output Shape	Param #
input_4 (InputLayer)	[(None, None)]	0
embedding_3 (Embedding)	(None, None, 100)	1000000
bidirectional_3 (Bidirectional)	(None, 64)	34048
dropout_3 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 1)	65

Total params: 1034113 (3.94 MB)
Trainable params: 34113 (133.25 KB)
Non-trainable params: 1000000 (3.81 MB)

Epoch 1/10
100/100 [=====] - 46s 390ms/step - loss: 0.6839 - accuracy: 0.5647 - val_loss: 0.6587 - val_accuracy: 0.6042
Epoch 2/10
100/100 [=====] - 32s 322ms/step - loss: 0.6262 - accuracy: 0.6587 - val_loss: 0.6063 - val_accuracy: 0.6750
Epoch 3/10
100/100 [=====] - 35s 350ms/step - loss: 0.5851 - accuracy: 0.7003 - val_loss: 0.5563 - val_accuracy: 0.7192
Epoch 4/10
100/100 [=====] - 6s 60ms/step - loss: 0.5416 - accuracy: 0.7262 - val_loss: 0.5842 - val_accuracy: 0.7010
Epoch 5/10
100/100 [=====] - 35s 354ms/step - loss: 0.5129 - accuracy: 0.7487 - val_loss: 0.5558 - val_accuracy: 0.7178
Epoch 6/10
100/100 [=====] - 34s 344ms/step - loss: 0.4943 - accuracy: 0.7538 - val_loss: 0.5172 - val_accuracy: 0.7444
Epoch 7/10
100/100 [=====] - 5s 51ms/step - loss: 0.4772 - accuracy: 0.7759 - val_loss: 0.5214 - val_accuracy: 0.7454
Epoch 8/10
100/100 [=====] - 31s 316ms/step - loss: 0.4451 - accuracy: 0.7937 - val_loss: 0.4873 - val_accuracy: 0.7598
Epoch 9/10
100/100 [=====] - 3s 33ms/step - loss: 0.4324 - accuracy: 0.8022 - val_loss: 0.5017 - val_accuracy: 0.7558
Epoch 10/10
100/100 [=====] - 4s 42ms/step - loss: 0.4318 - accuracy: 0.8078 - val_loss: 0.4971 - val_accuracy: 0.7556
782/782 [=====] - 11s 12ms/step - loss: 0.4913 - accuracy: 0.7618
Test acc: 0.762



Comparing Model Performance with Different Training Set Sizes

```
In [15]: train_sample_sizes = [100, 500, 1000, 5000, 10000, 20000]
for train_size in train_sample_sizes:
    train_ds = keras.utils.text_dataset_from_directory(
        "aclImdb/train", batch_size=batch_size
    ).take(train_size)

    int_train_ds = train_ds.map(
        lambda x, y: (text_vectorization(x), y),
        num_parallel_calls=4
    )
    int_val_ds = val_ds.map(
        lambda x, y: (text_vectorization(x), y),
        num_parallel_calls=4
    )
    int_test_ds = test_ds.map(
        lambda x, y: (text_vectorization(x), y),
        num_parallel_calls=4
    )

    # Train and evaluate the model with the embedding layer
    embedding_layer = layers.Embedding(max_tokens, embedding_dim)

    inputs = keras.Input(shape=(None,), dtype="int64")
    embedded = embedding_layer(inputs)
    x = layers.Bidirectional(layers.LSTM(32))(embedded)
    x = layers.Dropout(0.5)(x)
```

```

outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs, outputs)
model.compile(optimizer="rmsprop",
              loss="binary_crossentropy",
              metrics=["accuracy"])

callbacks = [
    keras.callbacks.ModelCheckpoint("embeddings_model.x",
                                   save_best_only=True)
]
history = model.fit(int_train_ds, validation_data=int_val_ds, epochs=10, callba
model = keras.models.load_model("embeddings_model.x")
embedding_layer_test_acc = model.evaluate(int_test_ds)[1]

accuracy = history.history["accuracy"]
val_accuracy = history.history["val_accuracy"]
epochs = range(1, len(accuracy) + 1)
plt.figure()
plt.plot(epochs, accuracy, "bo", label="Training Accuracy")
plt.plot(epochs, val_accuracy, "b", label="Validation Accuracy")
plt.title("Training and validation Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()

# Train and evaluate the model with the pretrained word embeddings
embedding_layer = layers.Embedding(
    max_tokens,
    embedding_dim,
    embeddings_initializer=keras.initializers.Constant(embedding_matrix),
    trainable=False,
    mask_zero=True,
)

inputs = keras.Input(shape=(None,), dtype="int64")
embedded = embedding_layer(inputs)
x = layers.Bidirectional(layers.LSTM(32))(embedded)
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs, outputs)
model.compile(optimizer="rmsprop",
              loss="binary_crossentropy",
              metrics=["accuracy"])

callbacks = [
    keras.callbacks.ModelCheckpoint("pretrained_embeddings_model.x",
                                   save_best_only=True)
]
history = model.fit(int_train_ds, validation_data=int_val_ds, epochs=10, callba
model = keras.models.load_model("pretrained_embeddings_model.x")
pretrained_embeddings_test_acc = model.evaluate(int_test_ds)[1]

accuracy = history.history["accuracy"]
val_accuracy = history.history["val_accuracy"]
epochs = range(1, len(accuracy) + 1)

```



```

plt.figure()
plt.plot(epochs, accuracy, "bo", label="Training Accuracy")
plt.plot(epochs, val_accuracy, "b", label="Validation Accuracy")
plt.title("Training and validation Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()

# Compare the performance and store the results
print(f"Training samples: {train_size}")
print(f"Embedding layer test accuracy: {embedding_layer_test_acc:.3f}")
print(f"Pretrained embeddings test accuracy: {pretrained_embeddings_test_acc:.3f}")
print("-" * 50)

```

Found 20000 files belonging to 2 classes.

Epoch 1/10

100/100 [=====] - 32s 249ms/step - loss: 0.6874 - accuracy: 0.5253 - val_loss: 0.6515 - val_accuracy: 0.6452

Epoch 2/10

100/100 [=====] - 22s 221ms/step - loss: 0.5658 - accuracy: 0.7347 - val_loss: 0.5902 - val_accuracy: 0.7132

Epoch 3/10

100/100 [=====] - 15s 147ms/step - loss: 0.4248 - accuracy: 0.8363 - val_loss: 0.5746 - val_accuracy: 0.7664

Epoch 4/10

100/100 [=====] - 18s 186ms/step - loss: 0.3156 - accuracy: 0.8850 - val_loss: 0.4597 - val_accuracy: 0.7944

Epoch 5/10

100/100 [=====] - 6s 56ms/step - loss: 0.2063 - accuracy: 0.9287 - val_loss: 0.6980 - val_accuracy: 0.7626

Epoch 6/10

100/100 [=====] - 7s 69ms/step - loss: 0.1534 - accuracy: 0.9488 - val_loss: 0.4957 - val_accuracy: 0.8034

Epoch 7/10

100/100 [=====] - 5s 50ms/step - loss: 0.1161 - accuracy: 0.9603 - val_loss: 0.5231 - val_accuracy: 0.8032

Epoch 8/10

100/100 [=====] - 3s 31ms/step - loss: 0.0867 - accuracy: 0.9750 - val_loss: 0.7628 - val_accuracy: 0.7404

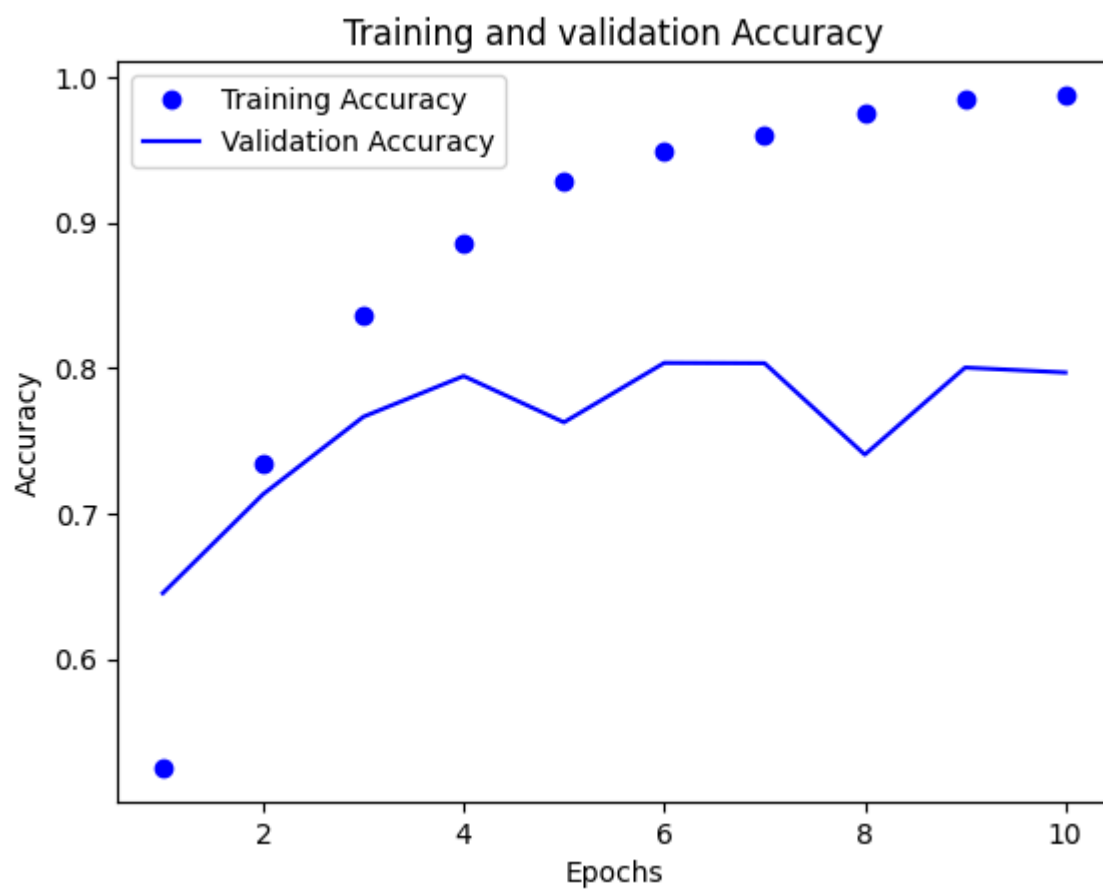
Epoch 9/10

100/100 [=====] - 4s 36ms/step - loss: 0.0609 - accuracy: 0.9841 - val_loss: 0.6994 - val_accuracy: 0.8002

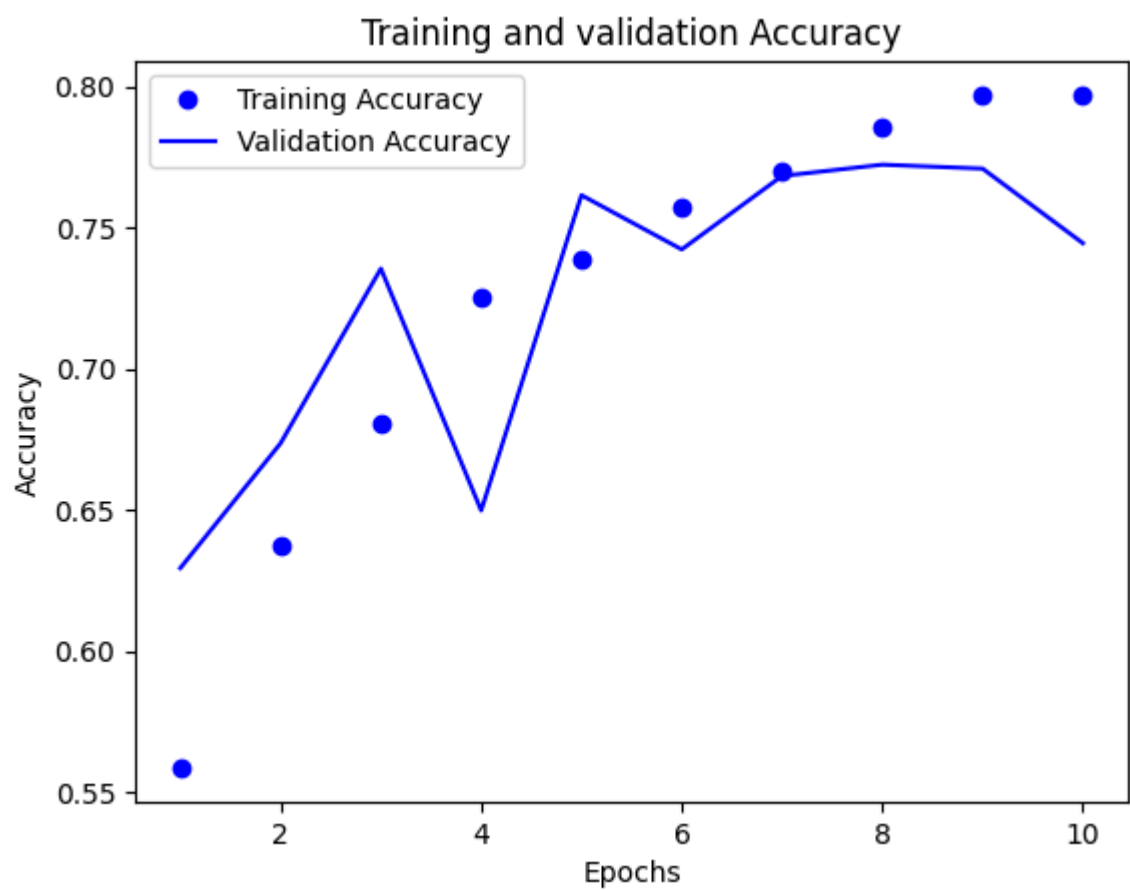
Epoch 10/10

100/100 [=====] - 6s 58ms/step - loss: 0.0465 - accuracy: 0.9875 - val_loss: 0.6768 - val_accuracy: 0.7968

782/782 [=====] - 12s 13ms/step - loss: 0.4723 - accuracy: 0.7889



Epoch 1/10
100/100 [=====] - 44s 332ms/step - loss: 0.6845 - accuracy: 0.5587 - val_loss: 0.6520 - val_accuracy: 0.6294
Epoch 2/10
100/100 [=====] - 35s 349ms/step - loss: 0.6348 - accuracy: 0.6375 - val_loss: 0.6060 - val_accuracy: 0.6736
Epoch 3/10
100/100 [=====] - 33s 330ms/step - loss: 0.5910 - accuracy: 0.6809 - val_loss: 0.5536 - val_accuracy: 0.7354
Epoch 4/10
100/100 [=====] - 10s 98ms/step - loss: 0.5549 - accuracy: 0.7250 - val_loss: 0.6530 - val_accuracy: 0.6498
Epoch 5/10
100/100 [=====] - 42s 418ms/step - loss: 0.5213 - accuracy: 0.7387 - val_loss: 0.5009 - val_accuracy: 0.7614
Epoch 6/10
100/100 [=====] - 5s 48ms/step - loss: 0.5002 - accuracy: 0.7572 - val_loss: 0.5209 - val_accuracy: 0.7422
Epoch 7/10
100/100 [=====] - 32s 318ms/step - loss: 0.4753 - accuracy: 0.7697 - val_loss: 0.4848 - val_accuracy: 0.7682
Epoch 8/10
100/100 [=====] - 34s 344ms/step - loss: 0.4590 - accuracy: 0.7856 - val_loss: 0.4776 - val_accuracy: 0.7722
Epoch 9/10
100/100 [=====] - 36s 366ms/step - loss: 0.4448 - accuracy: 0.7969 - val_loss: 0.4758 - val_accuracy: 0.7708
Epoch 10/10
100/100 [=====] - 6s 61ms/step - loss: 0.4337 - accuracy: 0.7969 - val_loss: 0.5043 - val_accuracy: 0.7444
782/782 [=====] - 11s 9ms/step - loss: 0.4732 - accuracy: 0.7753



Training samples: 100

Embedding layer test accuracy: 0.789

Pretrained embeddings test accuracy: 0.775

Found 20000 files belonging to 2 classes.

Epoch 1/10

500/500 [=====] - 50s 89ms/step - loss: 0.5362 - accuracy: 0.7259 - val_loss: 0.4049 - val_accuracy: 0.8230

Epoch 2/10

500/500 [=====] - 28s 56ms/step - loss: 0.3627 - accuracy: 0.8549 - val_loss: 0.3829 - val_accuracy: 0.8330

Epoch 3/10

500/500 [=====] - 27s 53ms/step - loss: 0.2977 - accuracy: 0.8863 - val_loss: 0.3720 - val_accuracy: 0.8382

Epoch 4/10

500/500 [=====] - 13s 27ms/step - loss: 0.2517 - accuracy: 0.9043 - val_loss: 0.3731 - val_accuracy: 0.8398

Epoch 5/10

500/500 [=====] - 18s 35ms/step - loss: 0.2274 - accuracy: 0.9159 - val_loss: 0.4701 - val_accuracy: 0.7808

Epoch 6/10

500/500 [=====] - 13s 25ms/step - loss: 0.2053 - accuracy: 0.9276 - val_loss: 0.3955 - val_accuracy: 0.8450

Epoch 7/10

500/500 [=====] - 15s 31ms/step - loss: 0.1842 - accuracy: 0.9346 - val_loss: 0.4021 - val_accuracy: 0.8382

Epoch 8/10

500/500 [=====] - 10s 21ms/step - loss: 0.1622 - accuracy: 0.9441 - val_loss: 0.4111 - val_accuracy: 0.8394

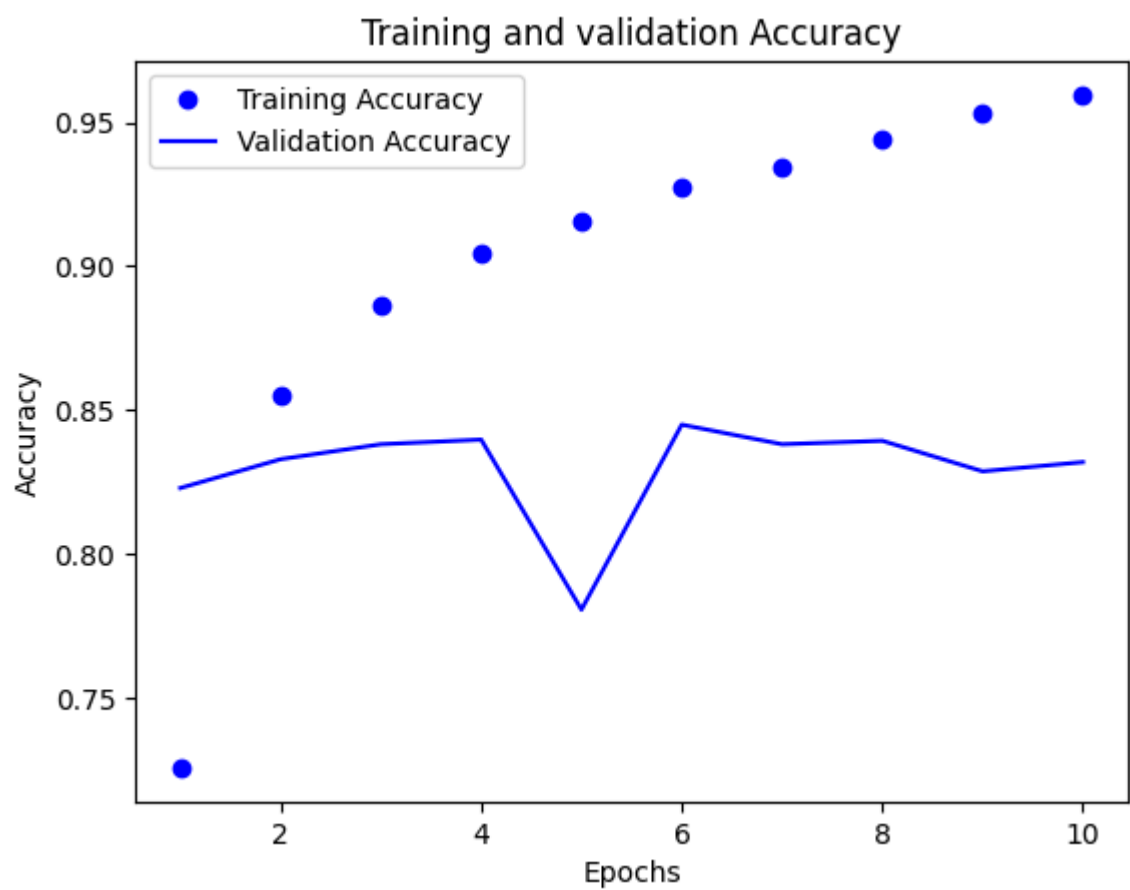
Epoch 9/10

500/500 [=====] - 11s 23ms/step - loss: 0.1354 - accuracy: 0.9532 - val_loss: 0.4602 - val_accuracy: 0.8288

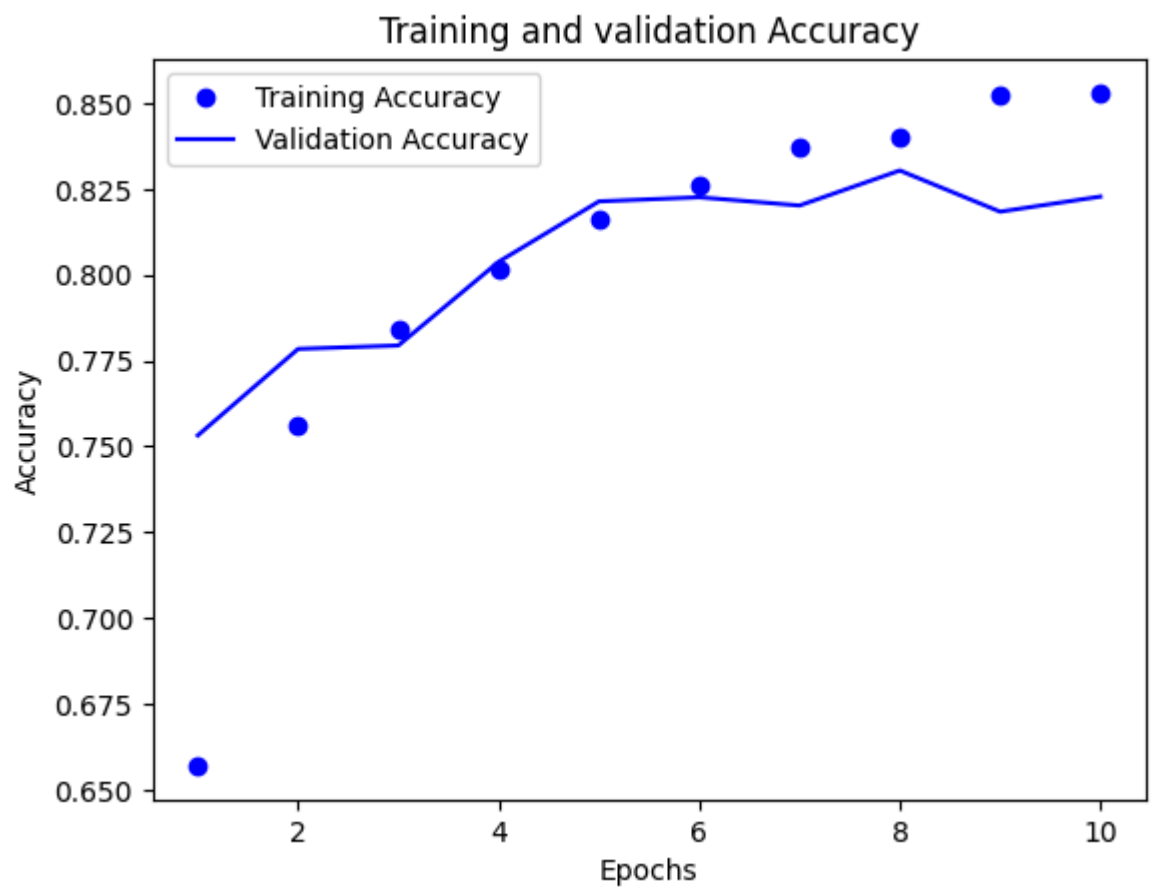
Epoch 10/10

500/500 [=====] - 14s 28ms/step - loss: 0.1198 - accuracy: 0.9594 - val_loss: 0.5685 - val_accuracy: 0.8320

782/782 [=====] - 8s 9ms/step - loss: 0.3903 - accuracy: 0.8285



Epoch 1/10
500/500 [=====] - 65s 110ms/step - loss: 0.6142 - accuracy: 0.6569 - val_loss: 0.5141 - val_accuracy: 0.7532
Epoch 2/10
500/500 [=====] - 46s 92ms/step - loss: 0.5095 - accuracy: 0.7564 - val_loss: 0.4610 - val_accuracy: 0.7784
Epoch 3/10
500/500 [=====] - 47s 94ms/step - loss: 0.4614 - accuracy: 0.7842 - val_loss: 0.4580 - val_accuracy: 0.7794
Epoch 4/10
500/500 [=====] - 46s 93ms/step - loss: 0.4317 - accuracy: 0.8017 - val_loss: 0.4253 - val_accuracy: 0.8038
Epoch 5/10
500/500 [=====] - 46s 93ms/step - loss: 0.4098 - accuracy: 0.8162 - val_loss: 0.3957 - val_accuracy: 0.8214
Epoch 6/10
500/500 [=====] - 48s 96ms/step - loss: 0.3915 - accuracy: 0.8260 - val_loss: 0.3908 - val_accuracy: 0.8226
Epoch 7/10
500/500 [=====] - 15s 29ms/step - loss: 0.3742 - accuracy: 0.8372 - val_loss: 0.4012 - val_accuracy: 0.8202
Epoch 8/10
500/500 [=====] - 41s 81ms/step - loss: 0.3634 - accuracy: 0.8399 - val_loss: 0.3836 - val_accuracy: 0.8304
Epoch 9/10
500/500 [=====] - 19s 37ms/step - loss: 0.3484 - accuracy: 0.8523 - val_loss: 0.4024 - val_accuracy: 0.8184
Epoch 10/10
500/500 [=====] - 12s 24ms/step - loss: 0.3350 - accuracy: 0.8530 - val_loss: 0.3922 - val_accuracy: 0.8228
782/782 [=====] - 10s 10ms/step - loss: 0.3864 - accuracy: 0.8246



Training samples: 500
Embedding layer test accuracy: 0.828
Pretrained embeddings test accuracy: 0.825

Found 20000 files belonging to 2 classes.

Epoch 1/10

625/625 [=====] - 57s 83ms/step - loss: 0.5332 - accuracy: 0.7164 - val_loss: 0.3814 - val_accuracy: 0.8302

Epoch 2/10

625/625 [=====] - 32s 51ms/step - loss: 0.3554 - accuracy: 0.8525 - val_loss: 0.3560 - val_accuracy: 0.8446

Epoch 3/10

625/625 [=====] - 14s 22ms/step - loss: 0.3024 - accuracy: 0.8802 - val_loss: 0.3681 - val_accuracy: 0.8530

Epoch 4/10

625/625 [=====] - 18s 28ms/step - loss: 0.2638 - accuracy: 0.8969 - val_loss: 0.3847 - val_accuracy: 0.8544

Epoch 5/10

625/625 [=====] - 16s 26ms/step - loss: 0.2396 - accuracy: 0.9119 - val_loss: 0.3718 - val_accuracy: 0.8576

Epoch 6/10

625/625 [=====] - 20s 32ms/step - loss: 0.2055 - accuracy: 0.9247 - val_loss: 0.4726 - val_accuracy: 0.8356

Epoch 7/10

625/625 [=====] - 13s 21ms/step - loss: 0.1854 - accuracy: 0.9331 - val_loss: 0.4748 - val_accuracy: 0.8446

Epoch 8/10

625/625 [=====] - 17s 28ms/step - loss: 0.1640 - accuracy: 0.9415 - val_loss: 0.4202 - val_accuracy: 0.8390

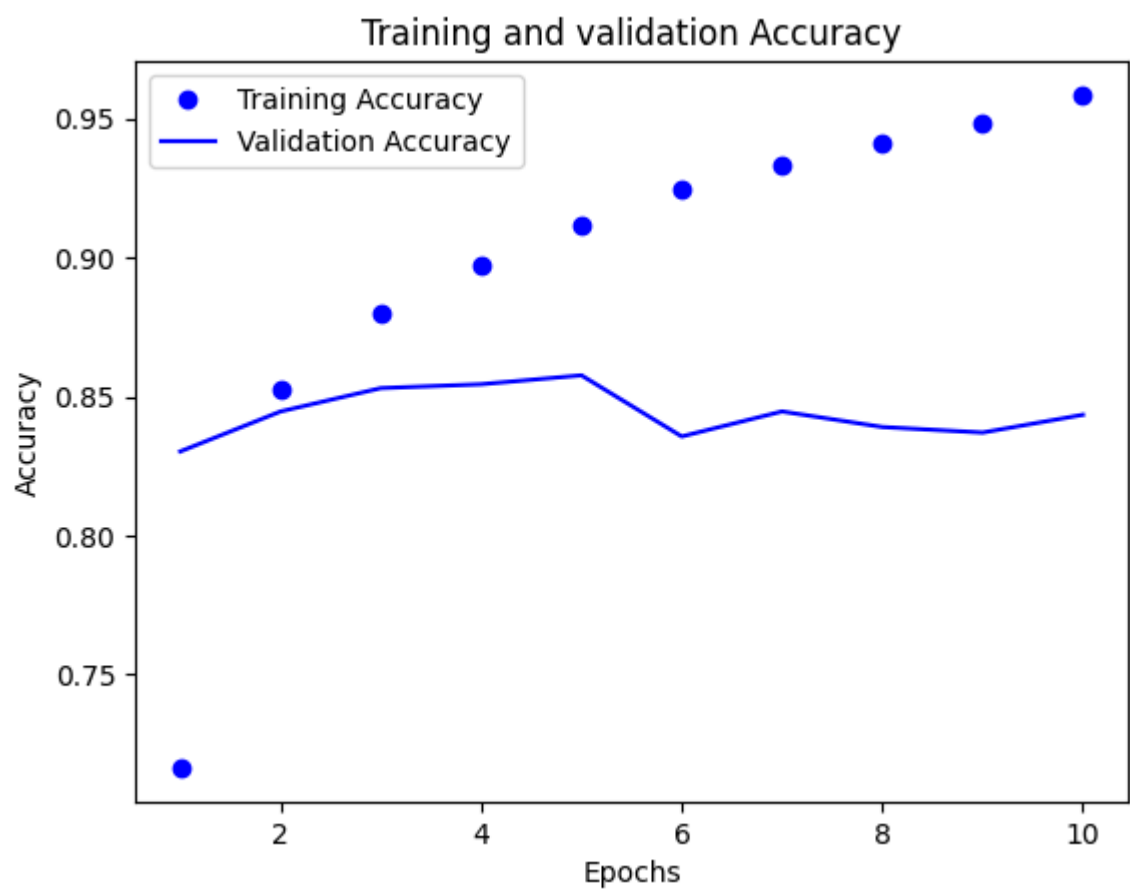
Epoch 9/10

625/625 [=====] - 13s 21ms/step - loss: 0.1417 - accuracy: 0.9485 - val_loss: 0.4955 - val_accuracy: 0.8370

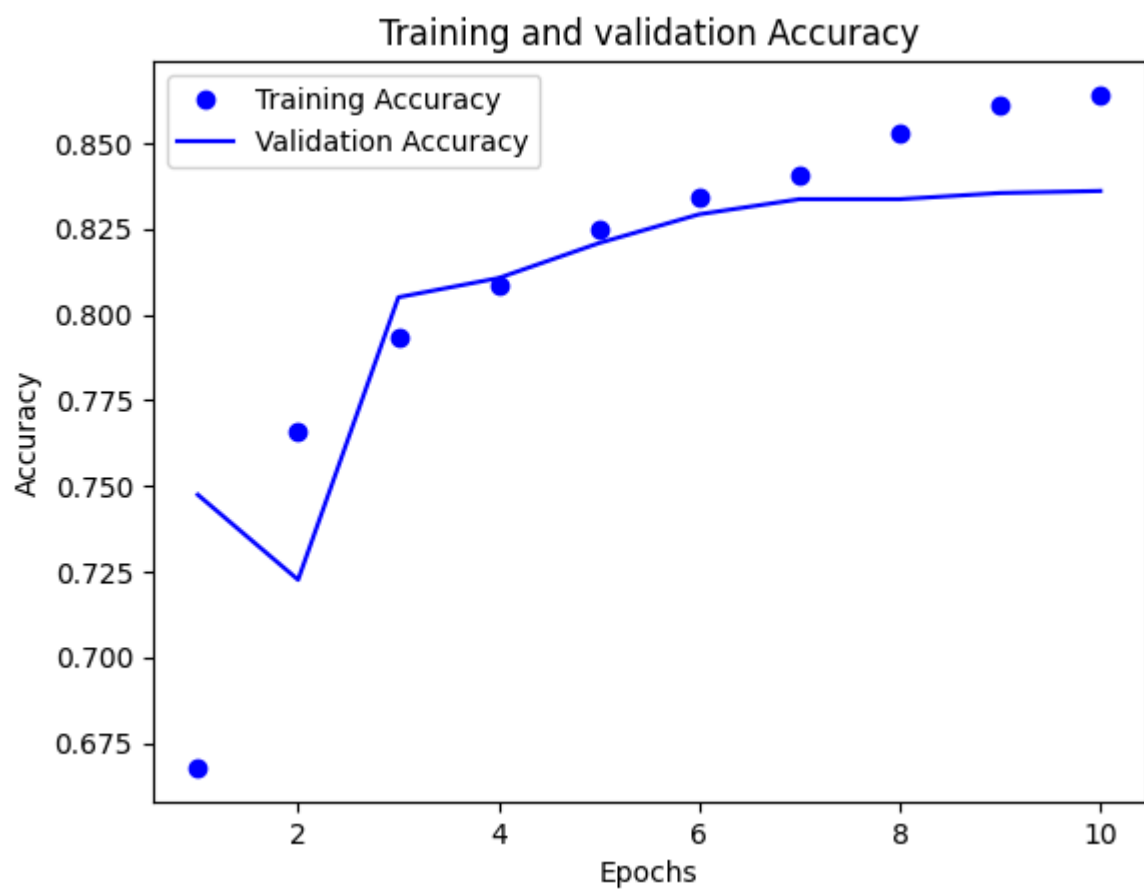
Epoch 10/10

625/625 [=====] - 13s 21ms/step - loss: 0.1201 - accuracy: 0.9585 - val_loss: 0.5371 - val_accuracy: 0.8434

782/782 [=====] - 13s 15ms/step - loss: 0.3765 - accuracy: 0.8372



Epoch 1/10
625/625 [=====] - 66s 94ms/step - loss: 0.6005 - accuracy: 0.6678 - val_loss: 0.5177 - val_accuracy: 0.7474
Epoch 2/10
625/625 [=====] - 18s 28ms/step - loss: 0.4907 - accuracy: 0.7661 - val_loss: 0.5948 - val_accuracy: 0.7226
Epoch 3/10
625/625 [=====] - 50s 80ms/step - loss: 0.4474 - accuracy: 0.7932 - val_loss: 0.4179 - val_accuracy: 0.8050
Epoch 4/10
625/625 [=====] - 60s 96ms/step - loss: 0.4198 - accuracy: 0.8083 - val_loss: 0.4026 - val_accuracy: 0.8106
Epoch 5/10
625/625 [=====] - 47s 74ms/step - loss: 0.3957 - accuracy: 0.8246 - val_loss: 0.3892 - val_accuracy: 0.8208
Epoch 6/10
625/625 [=====] - 48s 77ms/step - loss: 0.3775 - accuracy: 0.8339 - val_loss: 0.3741 - val_accuracy: 0.8292
Epoch 7/10
625/625 [=====] - 47s 76ms/step - loss: 0.3616 - accuracy: 0.8407 - val_loss: 0.3740 - val_accuracy: 0.8336
Epoch 8/10
625/625 [=====] - 18s 28ms/step - loss: 0.3437 - accuracy: 0.8532 - val_loss: 0.3799 - val_accuracy: 0.8336
Epoch 9/10
625/625 [=====] - 22s 35ms/step - loss: 0.3260 - accuracy: 0.8612 - val_loss: 0.3806 - val_accuracy: 0.8354
Epoch 10/10
625/625 [=====] - 54s 87ms/step - loss: 0.3170 - accuracy: 0.8640 - val_loss: 0.3646 - val_accuracy: 0.8360
782/782 [=====] - 11s 11ms/step - loss: 0.3655 - accuracy: 0.8378



Training samples: 1000

Embedding layer test accuracy: 0.837

Pretrained embeddings test accuracy: 0.838

Found 20000 files belonging to 2 classes.

Epoch 1/10

625/625 [=====] - 48s 69ms/step - loss: 0.5195 - accuracy: 0.7322 - val_loss: 0.3742 - val_accuracy: 0.8352

Epoch 2/10

625/625 [=====] - 38s 60ms/step - loss: 0.3474 - accuracy: 0.8587 - val_loss: 0.3535 - val_accuracy: 0.8460

Epoch 3/10

625/625 [=====] - 16s 25ms/step - loss: 0.2960 - accuracy: 0.8857 - val_loss: 0.3630 - val_accuracy: 0.8500

Epoch 4/10

625/625 [=====] - 13s 21ms/step - loss: 0.2563 - accuracy: 0.9017 - val_loss: 0.4089 - val_accuracy: 0.8210

Epoch 5/10

625/625 [=====] - 18s 29ms/step - loss: 0.2319 - accuracy: 0.9135 - val_loss: 0.4178 - val_accuracy: 0.8384

Epoch 6/10

625/625 [=====] - 13s 21ms/step - loss: 0.2073 - accuracy: 0.9232 - val_loss: 0.4147 - val_accuracy: 0.8478

Epoch 7/10

625/625 [=====] - 20s 31ms/step - loss: 0.1840 - accuracy: 0.9341 - val_loss: 0.5501 - val_accuracy: 0.8332

Epoch 8/10

625/625 [=====] - 12s 19ms/step - loss: 0.1585 - accuracy: 0.9437 - val_loss: 0.4653 - val_accuracy: 0.8498

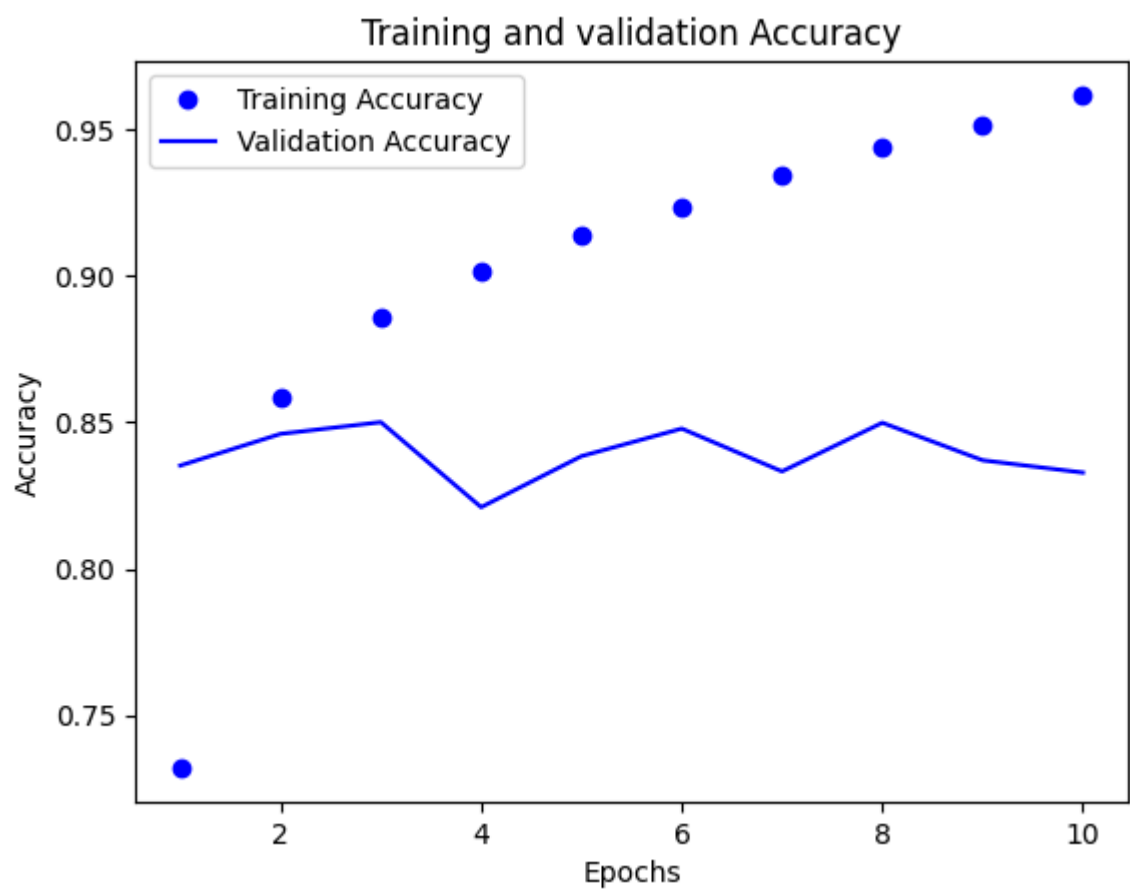
Epoch 9/10

625/625 [=====] - 21s 33ms/step - loss: 0.1363 - accuracy: 0.9514 - val_loss: 0.5286 - val_accuracy: 0.8370

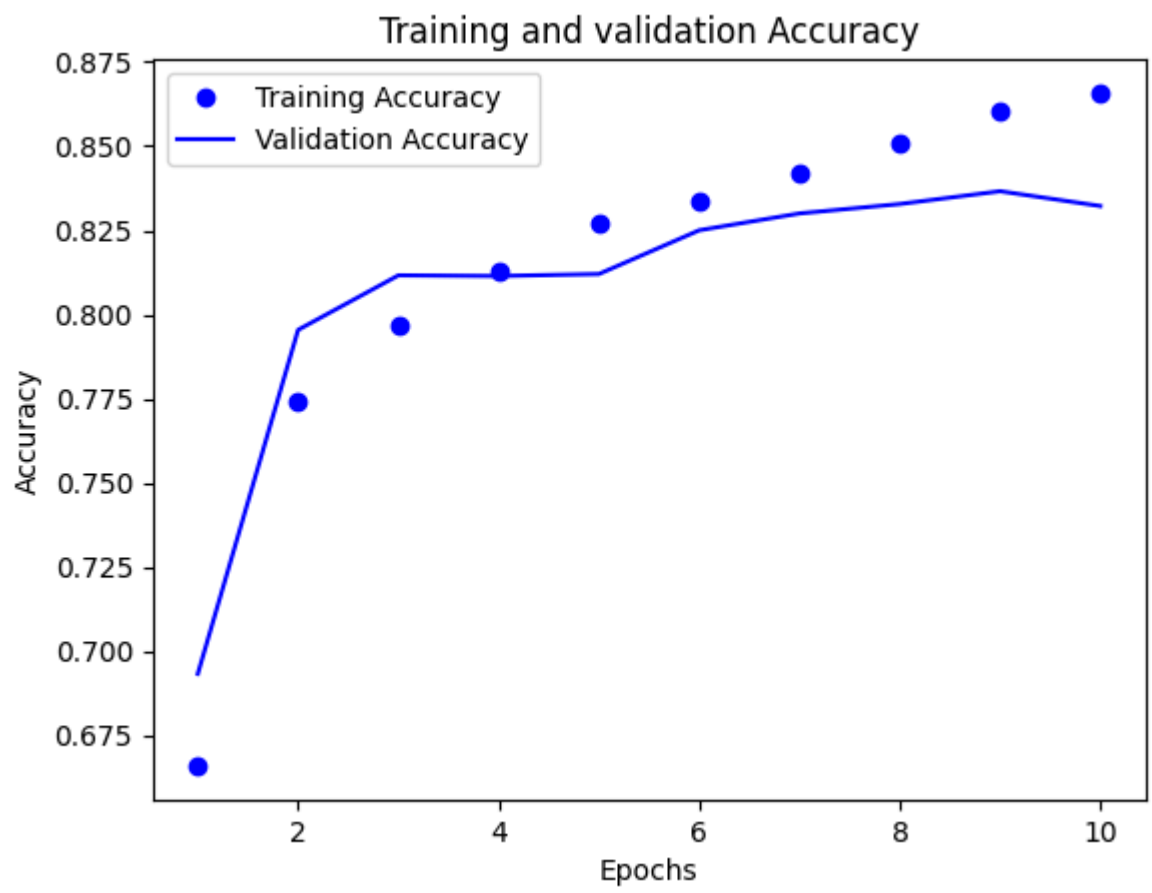
Epoch 10/10

625/625 [=====] - 20s 33ms/step - loss: 0.1109 - accuracy: 0.9616 - val_loss: 0.6006 - val_accuracy: 0.8328

782/782 [=====] - 12s 14ms/step - loss: 0.3715 - accuracy: 0.8376



Epoch 1/10
625/625 [=====] - 63s 84ms/step - loss: 0.6013 - accuracy: 0.6659 - val_loss: 0.6257 - val_accuracy: 0.6932
Epoch 2/10
625/625 [=====] - 58s 93ms/step - loss: 0.4827 - accuracy: 0.7739 - val_loss: 0.4402 - val_accuracy: 0.7954
Epoch 3/10
625/625 [=====] - 49s 79ms/step - loss: 0.4426 - accuracy: 0.7965 - val_loss: 0.4129 - val_accuracy: 0.8116
Epoch 4/10
625/625 [=====] - 53s 85ms/step - loss: 0.4147 - accuracy: 0.8126 - val_loss: 0.4109 - val_accuracy: 0.8114
Epoch 5/10
625/625 [=====] - 17s 26ms/step - loss: 0.3921 - accuracy: 0.8274 - val_loss: 0.4202 - val_accuracy: 0.8120
Epoch 6/10
625/625 [=====] - 48s 76ms/step - loss: 0.3753 - accuracy: 0.8335 - val_loss: 0.3949 - val_accuracy: 0.8250
Epoch 7/10
625/625 [=====] - 54s 86ms/step - loss: 0.3591 - accuracy: 0.8421 - val_loss: 0.3904 - val_accuracy: 0.8300
Epoch 8/10
625/625 [=====] - 18s 28ms/step - loss: 0.3452 - accuracy: 0.8506 - val_loss: 0.4032 - val_accuracy: 0.8328
Epoch 9/10
625/625 [=====] - 49s 79ms/step - loss: 0.3308 - accuracy: 0.8602 - val_loss: 0.3776 - val_accuracy: 0.8366
Epoch 10/10
625/625 [=====] - 22s 36ms/step - loss: 0.3183 - accuracy: 0.8658 - val_loss: 0.4097 - val_accuracy: 0.8322
782/782 [=====] - 18s 18ms/step - loss: 0.3740 - accuracy: 0.8359



Training samples: 5000
Embedding layer test accuracy: 0.838
Pretrained embeddings test accuracy: 0.836

Found 20000 files belonging to 2 classes.

Epoch 1/10

625/625 [=====] - 48s 68ms/step - loss: 0.5121 - accuracy: 0.7430 - val_loss: 0.3835 - val_accuracy: 0.8320

Epoch 2/10

625/625 [=====] - 21s 33ms/step - loss: 0.3461 - accuracy: 0.8588 - val_loss: 0.4187 - val_accuracy: 0.8222

Epoch 3/10

625/625 [=====] - 30s 48ms/step - loss: 0.2941 - accuracy: 0.8831 - val_loss: 0.3717 - val_accuracy: 0.8412

Epoch 4/10

625/625 [=====] - 36s 57ms/step - loss: 0.2623 - accuracy: 0.8985 - val_loss: 0.3656 - val_accuracy: 0.8488

Epoch 5/10

625/625 [=====] - 19s 31ms/step - loss: 0.2323 - accuracy: 0.9119 - val_loss: 0.4102 - val_accuracy: 0.8332

Epoch 6/10

625/625 [=====] - 13s 21ms/step - loss: 0.2074 - accuracy: 0.9235 - val_loss: 0.4333 - val_accuracy: 0.8502

Epoch 7/10

625/625 [=====] - 19s 30ms/step - loss: 0.1858 - accuracy: 0.9309 - val_loss: 0.4120 - val_accuracy: 0.8410

Epoch 8/10

625/625 [=====] - 13s 20ms/step - loss: 0.1643 - accuracy: 0.9403 - val_loss: 0.4663 - val_accuracy: 0.8388

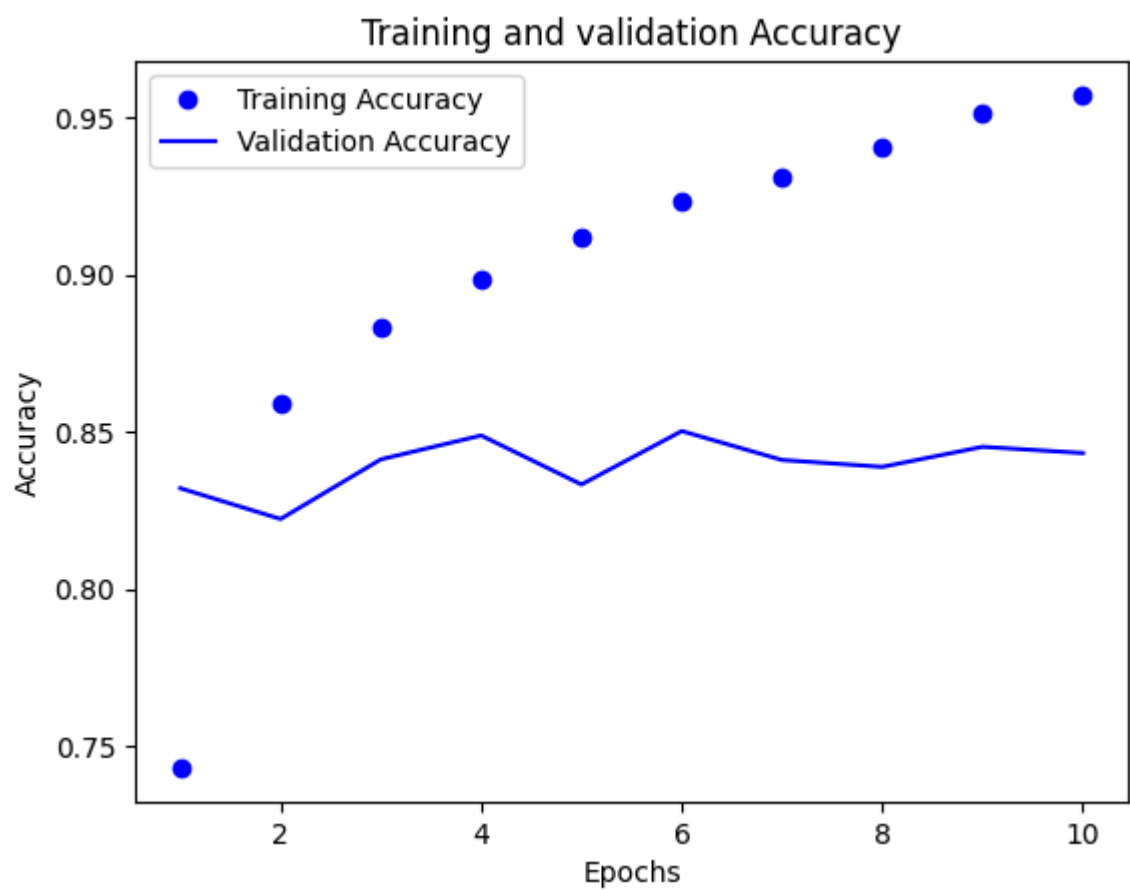
Epoch 9/10

625/625 [=====] - 14s 22ms/step - loss: 0.1436 - accuracy: 0.9513 - val_loss: 0.4868 - val_accuracy: 0.8452

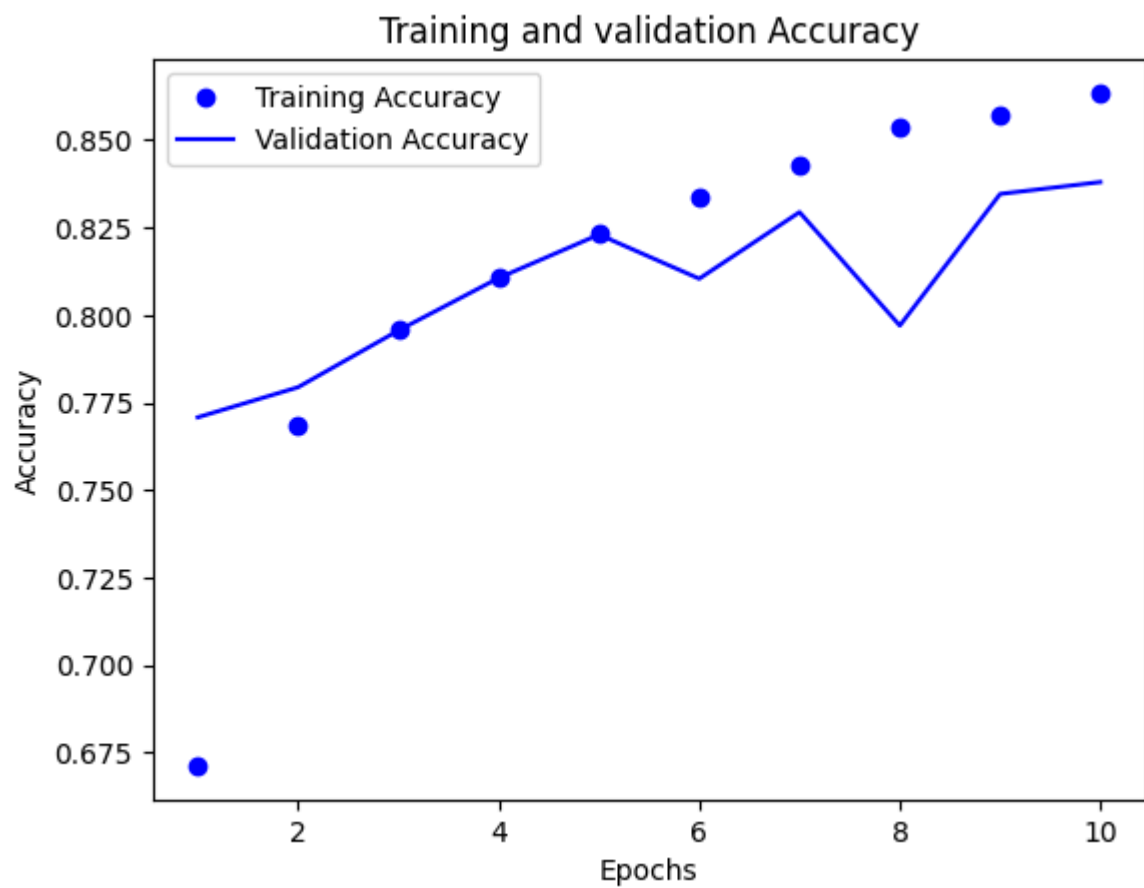
Epoch 10/10

625/625 [=====] - 18s 29ms/step - loss: 0.1223 - accuracy: 0.9572 - val_loss: 0.5394 - val_accuracy: 0.8432

782/782 [=====] - 8s 9ms/step - loss: 0.4068 - accuracy: 0.8364



Epoch 1/10
625/625 [=====] - 67s 89ms/step - loss: 0.5964 - accuracy: 0.6713 - val_loss: 0.4875 - val_accuracy: 0.7708
Epoch 2/10
625/625 [=====] - 51s 82ms/step - loss: 0.4917 - accuracy: 0.7688 - val_loss: 0.4587 - val_accuracy: 0.7794
Epoch 3/10
625/625 [=====] - 47s 76ms/step - loss: 0.4413 - accuracy: 0.7957 - val_loss: 0.4436 - val_accuracy: 0.7956
Epoch 4/10
625/625 [=====] - 46s 74ms/step - loss: 0.4169 - accuracy: 0.8106 - val_loss: 0.4149 - val_accuracy: 0.8106
Epoch 5/10
625/625 [=====] - 56s 89ms/step - loss: 0.3904 - accuracy: 0.8236 - val_loss: 0.3888 - val_accuracy: 0.8230
Epoch 6/10
625/625 [=====] - 18s 28ms/step - loss: 0.3740 - accuracy: 0.8338 - val_loss: 0.4113 - val_accuracy: 0.8104
Epoch 7/10
625/625 [=====] - 51s 81ms/step - loss: 0.3600 - accuracy: 0.8431 - val_loss: 0.3859 - val_accuracy: 0.8294
Epoch 8/10
625/625 [=====] - 29s 46ms/step - loss: 0.3437 - accuracy: 0.8535 - val_loss: 0.4620 - val_accuracy: 0.7970
Epoch 9/10
625/625 [=====] - 51s 82ms/step - loss: 0.3324 - accuracy: 0.8572 - val_loss: 0.3661 - val_accuracy: 0.8346
Epoch 10/10
625/625 [=====] - 23s 37ms/step - loss: 0.3179 - accuracy: 0.8634 - val_loss: 0.3730 - val_accuracy: 0.8380
782/782 [=====] - 13s 14ms/step - loss: 0.3718 - accuracy: 0.8334



Training samples: 10000

Embedding layer test accuracy: 0.836

Pretrained embeddings test accuracy: 0.833

Found 20000 files belonging to 2 classes.

Epoch 1/10

625/625 [=====] - 48s 68ms/step - loss: 0.5249 - accuracy: 0.7293 - val_loss: 0.4052 - val_accuracy: 0.8182

Epoch 2/10

625/625 [=====] - 35s 55ms/step - loss: 0.3497 - accuracy: 0.8547 - val_loss: 0.3709 - val_accuracy: 0.8412

Epoch 3/10

625/625 [=====] - 34s 54ms/step - loss: 0.2936 - accuracy: 0.8820 - val_loss: 0.3611 - val_accuracy: 0.8420

Epoch 4/10

625/625 [=====] - 15s 23ms/step - loss: 0.2601 - accuracy: 0.9017 - val_loss: 0.3912 - val_accuracy: 0.8336

Epoch 5/10

625/625 [=====] - 18s 28ms/step - loss: 0.2327 - accuracy: 0.9121 - val_loss: 0.4071 - val_accuracy: 0.8380

Epoch 6/10

625/625 [=====] - 15s 23ms/step - loss: 0.2064 - accuracy: 0.9237 - val_loss: 0.4005 - val_accuracy: 0.8302

Epoch 7/10

625/625 [=====] - 14s 23ms/step - loss: 0.1829 - accuracy: 0.9330 - val_loss: 0.4021 - val_accuracy: 0.8364

Epoch 8/10

625/625 [=====] - 19s 30ms/step - loss: 0.1580 - accuracy: 0.9439 - val_loss: 0.5228 - val_accuracy: 0.8098

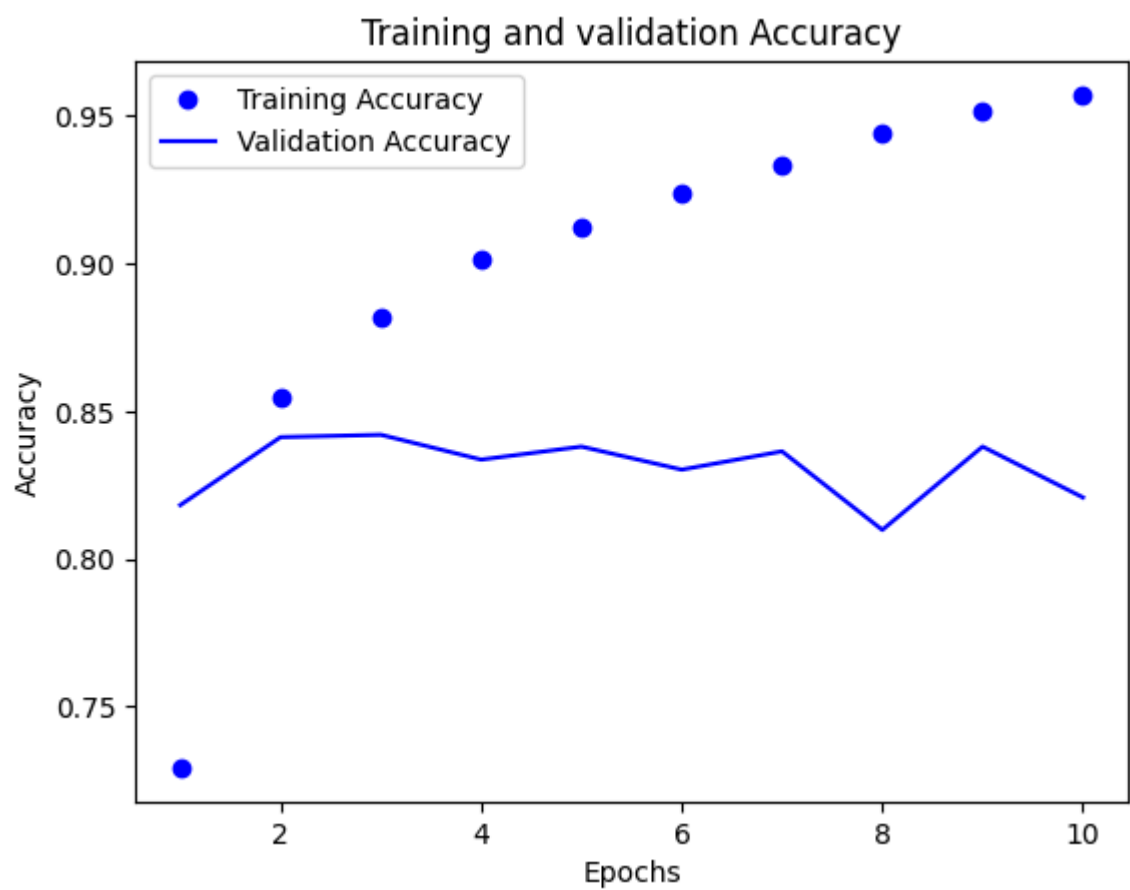
Epoch 9/10

625/625 [=====] - 13s 21ms/step - loss: 0.1386 - accuracy: 0.9513 - val_loss: 0.4990 - val_accuracy: 0.8380

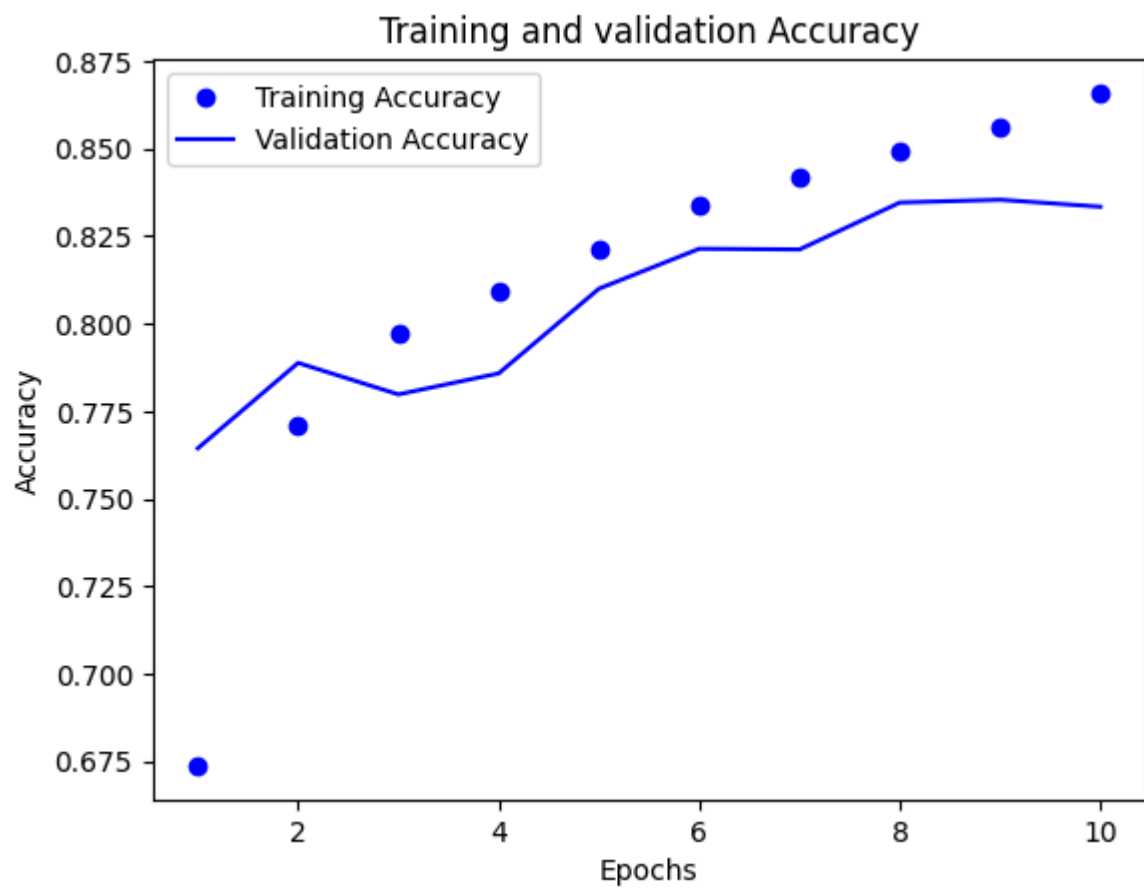
Epoch 10/10

625/625 [=====] - 23s 37ms/step - loss: 0.1222 - accuracy: 0.9571 - val_loss: 0.4844 - val_accuracy: 0.8208

782/782 [=====] - 9s 10ms/step - loss: 0.3827 - accuracy: 0.8335



Epoch 1/10
625/625 [=====] - 61s 78ms/step - loss: 0.5938 - accuracy: 0.6737 - val_loss: 0.4875 - val_accuracy: 0.7644
Epoch 2/10
625/625 [=====] - 56s 89ms/step - loss: 0.4834 - accuracy: 0.7711 - val_loss: 0.4448 - val_accuracy: 0.7888
Epoch 3/10
625/625 [=====] - 19s 30ms/step - loss: 0.4434 - accuracy: 0.7973 - val_loss: 0.4606 - val_accuracy: 0.7798
Epoch 4/10
625/625 [=====] - 50s 80ms/step - loss: 0.4176 - accuracy: 0.8093 - val_loss: 0.4436 - val_accuracy: 0.7858
Epoch 5/10
625/625 [=====] - 51s 82ms/step - loss: 0.3950 - accuracy: 0.8215 - val_loss: 0.4152 - val_accuracy: 0.8100
Epoch 6/10
625/625 [=====] - 47s 75ms/step - loss: 0.3785 - accuracy: 0.8337 - val_loss: 0.3919 - val_accuracy: 0.8214
Epoch 7/10
625/625 [=====] - 48s 76ms/step - loss: 0.3599 - accuracy: 0.8420 - val_loss: 0.3823 - val_accuracy: 0.8212
Epoch 8/10
625/625 [=====] - 48s 77ms/step - loss: 0.3454 - accuracy: 0.8492 - val_loss: 0.3693 - val_accuracy: 0.8346
Epoch 9/10
625/625 [=====] - 46s 74ms/step - loss: 0.3335 - accuracy: 0.8563 - val_loss: 0.3636 - val_accuracy: 0.8354
Epoch 10/10
625/625 [=====] - 32s 51ms/step - loss: 0.3167 - accuracy: 0.8659 - val_loss: 0.3743 - val_accuracy: 0.8334
782/782 [=====] - 12s 10ms/step - loss: 0.3591 - accuracy: 0.8388



Training samples: 20000

Embedding layer test accuracy: 0.833

Pretrained embeddings test accuracy: 0.839
