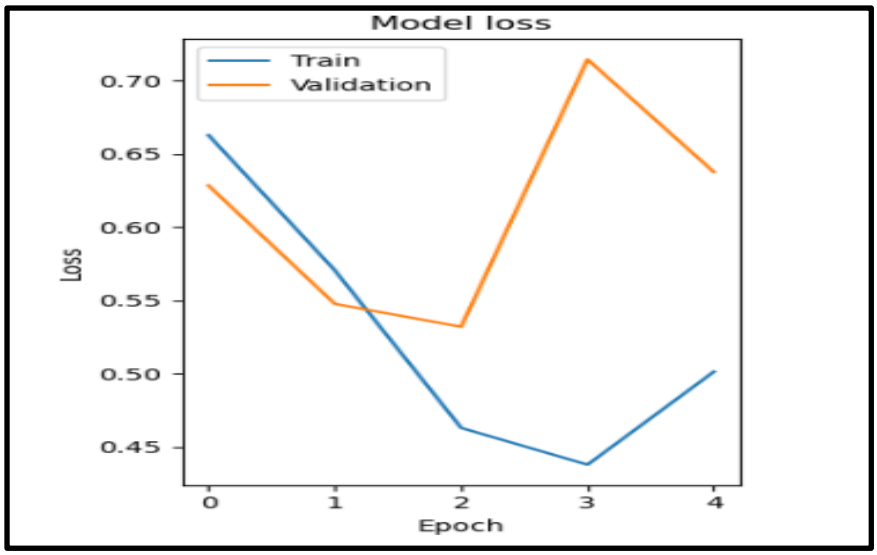


Training the Model

[7]:
```
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    epochs=10,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // validation_generator.batch_size
)
```

```
Epoch 1/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 14s 3s/step - accuracy: 0.9271 - loss: 0.5018 - val_accuracy: 0.9062 - val_loss: 0.4647
Epoch 2/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 204ms/step - accuracy: 0.8500 - loss: 0.4357 - val_accuracy: 1.0000 - val_loss: 0.3846
Epoch 3/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 12s 2s/step - accuracy: 0.9497 - loss: 0.4174 - val_accuracy: 0.9141 - val_loss: 0.4023
Epoch 4/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 255ms/step - accuracy: 0.9375 - loss: 0.4139 - val_accuracy: 0.9500 - val_loss: 0.3556
Epoch 5/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 11s 2s/step - accuracy: 0.8655 - loss: 0.3878 - val_accuracy: 0.8984 - val_loss: 0.3469
Epoch 6/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 267ms/step - accuracy: 0.8750 - loss: 0.3784 - val_accuracy: 0.9500 - val_loss: 0.3229
Epoch 7/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 11s 2s/step - accuracy: 0.9613 - loss: 0.3346 - val_accuracy: 0.9453 - val_loss: 0.3016
Epoch 8/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 236ms/step - accuracy: 0.9375 - loss: 0.3084 - val_accuracy: 0.8500 - val_loss: 0.3117
Epoch 9/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 11s 2s/step - accuracy: 0.9204 - loss: 0.2683 - val_accuracy: 0.8984 - val_loss: 0.2803
Epoch 10/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 176ms/step - accuracy: 0.9062 - loss: 0.3338 - val_accuracy: 0.9500 - val_loss: 0.2235
```

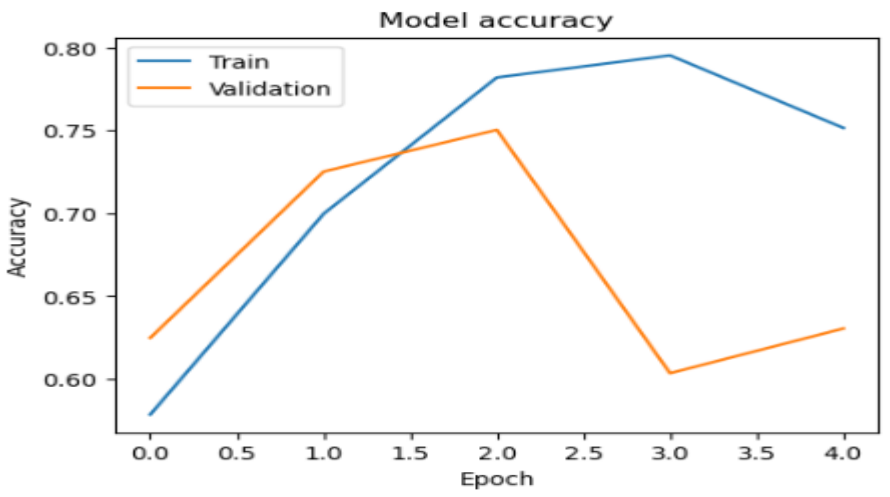
```
# Load the data
print('Loading data...')
(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=max_features)
print(len(x_train), 'train sequences')
print(len(x_test), 'test sequences')

print('Pad sequences (samples x time)')
x_train = sequence.pad_sequences(x_train, maxlen=maxlen)
x_test = sequence.pad_sequences(x_test, maxlen=maxlen)
print('x_train shape:', x_train.shape)
print('x_test shape:', x_test.shape)
```

```
Loading data...
25000 train sequences
25000 test sequences
Pad sequences (samples x time)
x_train shape: (25000, 500)
x_test shape: (25000, 500)
```

```
print('Evaluate model...')
score = model.evaluate(x_test, y_test,batch_size=batch_size)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Evaluate model...
782/782 ━━━━━━━━━━━━━━━━━━━━ 39s 49ms/step - accuracy: 0.6295 - loss: 0.6440
Test loss: 0.6364988684654236
Test accuracy: 0.6304799914360046
```

```
<matplotlib.legend.Legend at 0x1aead3d7590>
```

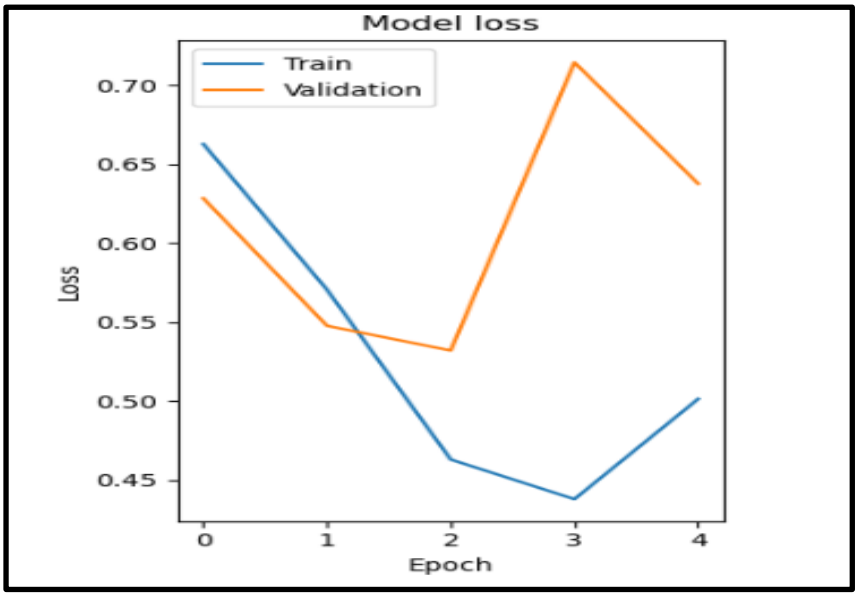



```
model = Sequential()
model.add(Embedding(max_features, 128,))
model.add(SimpleRNN(128))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

print(model.summary())
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	?	0 (unbuilt)
simple_rnn (SimpleRNN)	?	0 (unbuilt)
dense (Dense)	?	0 (unbuilt)

```
Total params: 0 (0.00 B)
Trainable params: 0 (0.00 B)
Non-trainable params: 0 (0.00 B)
None
```




```
print('Training model...')
history = model.fit(x_train, y_train,batch_size=batch_size,epochs=5,validation_split=0.2)
```

```
Training model...
Epoch 1/5
625/625 ━━━━━━━━━━━━━━━━━━━━ 117s 183ms/step - accuracy: 0.5459 - loss: 0.6818 - val_accuracy: 0.6248 - val_loss: 0.6284
Epoch 2/5
625/625 ━━━━━━━━━━━━━━━━━━━━ 109s 174ms/step - accuracy: 0.6967 - loss: 0.5766 - val_accuracy: 0.7252 - val_loss: 0.5477
Epoch 3/5
625/625 ━━━━━━━━━━━━━━━━━━━━ 104s 166ms/step - accuracy: 0.7774 - loss: 0.4691 - val_accuracy: 0.7504 - val_loss: 0.5320
Epoch 4/5
625/625 ━━━━━━━━━━━━━━━━━━━━ 107s 171ms/step - accuracy: 0.8341 - loss: 0.3765 - val_accuracy: 0.6034 - val_loss: 0.7144
Epoch 5/5
625/625 ━━━━━━━━━━━━━━━━━━━━ 113s 181ms/step - accuracy: 0.7272 - loss: 0.5314 - val_accuracy: 0.6304 - val_loss: 0.6376
```