In [1]:
```
import tensorflow as tf
```

In [2]:
```
from tensorflow.keras.datasets import imdb
```

In [3]:
```
from tensorflow.keras.models import Sequential
```

In [4]:
```
from tensorflow.keras.layers import Embedding, SimpleRNN, Dense
```

In [5]:
```
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

In [6]:
```
vocab_size = 10000
```

In [7]:
```
maxlen = 500
```

In [8]:
```
(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=vocab_size)
```

In [9]:
```
x_train = pad_sequences(x_train, maxlen=maxlen)
```

In [10]:
```
x_test = pad_sequences(x_test, maxlen=maxlen)
```

In [11]:
```
model = Sequential([
    Embedding(input_dim=vocab_size, output_dim=32, input_length=maxlen),
    SimpleRNN(32),
    Dense(1, activation='sigmoid')
])
```

In [12]:
```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accur
```

In [13]:
```
history = model.fit(x_train, y_train, epochs=5, batch_size=64, validation_s
```

```
Epoch 1/5
313/313 [==============================] - 43s 130ms/step - loss: 0.5428 -
accuracy: 0.7163 - val_loss: 0.4685 - val_accuracy: 0.7842
Epoch 2/5
313/313 [==============================] - 41s 130ms/step - loss: 0.3275 -
accuracy: 0.8652 - val_loss: 0.3526 - val_accuracy: 0.8490
Epoch 3/5
313/313 [==============================] - 43s 136ms/step - loss: 0.2313 -
accuracy: 0.9100 - val_loss: 0.4161 - val_accuracy: 0.8304
Epoch 4/5
313/313 [==============================] - 40s 128ms/step - loss: 0.1401 -
accuracy: 0.9509 - val_loss: 0.4713 - val_accuracy: 0.8156
Epoch 5/5
313/313 [==============================] - 44s 141ms/step - loss: 0.0737 -
accuracy: 0.9773 - val_loss: 0.4898 - val_accuracy: 0.8428
```

In [14]:

```
test_loss, test_acc = model.evaluate(x_test, y_test)
print(f'\nTest Accuracy: {test_acc:.4f}')
```

```
782/782 [==============================] - 20s 26ms/step - loss: 0.4846 -
accuracy: 0.8457

Test Accuracy: 0.8457
```

In [ ]: