

```

1 import numpy as np
2 from tensorflow import keras
3 from tensorflow.keras.datasets import imdb
4 from tensorflow.keras.preprocessing.sequence import pad_sequences
5
6 # Set the hyperparameters
7 max_features = 10000 # Number of words to consider as features
8 max_len = 200 # Maximum length of each sequence (in words)
9 batch_size = 32
10 epochs = 10
11
12 # Load the IMDB dataset
13 (X_train, y_train), (X_test, y_test) = imdb.load_data(num_words=max_features)
14
15 # Pad the sequences to have a consistent length
16 X_train = pad_sequences(X_train, maxlen=max_len)
17 X_test = pad_sequences(X_test, maxlen=max_len)

```

```

1 # Create a sequential model with an embedding layer and two dense layers
2 model = keras.Sequential([
3     keras.layers.Embedding(input_dim=max_features, output_dim=128, input_length=max_len),
4     keras.layers.Flatten(),
5     keras.layers.Dense(64, activation='relu'),
6     keras.layers.Dense(1, activation='sigmoid')
7 ])
8
9 # Compile the model with binary crossentropy loss and adam optimizer
10 model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
11
12 # Train the model on the training data
13 model.fit(X_train, y_train, batch_size=batch_size, epochs=epochs, validation_data=(X_test, y_test))

```

```

📄 Epoch 1/10
782/782 [=====] - 38s 48ms/step - loss: 0.3833 - accuracy: 0.8155 - val_loss: 0.3243 - val_accuracy: 0.857
Epoch 2/10
782/782 [=====] - 37s 47ms/step - loss: 0.0623 - accuracy: 0.9788 - val_loss: 0.4993 - val_accuracy: 0.840
Epoch 3/10
782/782 [=====] - 38s 49ms/step - loss: 0.0081 - accuracy: 0.9978 - val_loss: 0.6708 - val_accuracy: 0.849
Epoch 4/10
782/782 [=====] - 36s 46ms/step - loss: 8.5647e-04 - accuracy: 1.0000 - val_loss: 0.7097 - val_accuracy: 0
Epoch 5/10
782/782 [=====] - 36s 46ms/step - loss: 8.2243e-05 - accuracy: 1.0000 - val_loss: 0.7343 - val_accuracy: 0
Epoch 6/10
782/782 [=====] - 35s 45ms/step - loss: 4.1825e-05 - accuracy: 1.0000 - val_loss: 0.7535 - val_accuracy: 0
Epoch 7/10
782/782 [=====] - 37s 47ms/step - loss: 2.5442e-05 - accuracy: 1.0000 - val_loss: 0.7720 - val_accuracy: 0
Epoch 8/10
782/782 [=====] - 38s 48ms/step - loss: 1.6625e-05 - accuracy: 1.0000 - val_loss: 0.7908 - val_accuracy: 0
Epoch 9/10
782/782 [=====] - 39s 50ms/step - loss: 1.0947e-05 - accuracy: 1.0000 - val_loss: 0.8093 - val_accuracy: 0
Epoch 10/10
782/782 [=====] - 38s 48ms/step - loss: 7.3412e-06 - accuracy: 1.0000 - val_loss: 0.8285 - val_accuracy: 0
<keras.callbacks.History at 0x7fa46000ee30>

```

```

1 # Evaluate the model on the testing data
2 test_loss, test_acc = model.evaluate(X_test, y_test)
3
4 # Print the test accuracy
5 print('Test accuracy:', test_acc)
6

```

```

782/782 [=====] - 4s 5ms/step - loss: 0.8285 - accuracy: 0.8578
Test accuracy: 0.8578400015830994

```