

```
import tensorflow as tf
from tensorflow.keras.datasets import imdb
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, Dense, GlobalAveragePooling1D
```



```
# Load IMDB dataset
(X_train, y_train), (X_test, y_test) = imdb.load_data(num_words=10000)
X_train = pad_sequences(X_train, maxlen=200)
X_test = pad_sequences(X_test, maxlen=200)
```

```
# Define model
model = Sequential([
    Embedding(10000, 32),
    GlobalAveragePooling1D(),
    Dense(1, activation='sigmoid')
])
```

```
# Compile and train model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=5, batch_size=64, validation_data=(X_test, y_test))
```

```
↗ Epoch 1/5
391/391 ————— 7s 13ms/step - accuracy: 0.6322 - loss: 0.6669 - val_accuracy: 0.7994 - val_loss: 0.5436
Epoch 2/5
391/391 ————— 4s 11ms/step - accuracy: 0.8163 - loss: 0.4976 - val_accuracy: 0.8252 - val_loss: 0.4230
Epoch 3/5
391/391 ————— 7s 17ms/step - accuracy: 0.8576 - loss: 0.3801 - val_accuracy: 0.8570 - val_loss: 0.3601
Epoch 4/5
391/391 ————— 4s 10ms/step - accuracy: 0.8807 - loss: 0.3216 - val_accuracy: 0.8685 - val_loss: 0.3280
Epoch 5/5
391/391 ————— 7s 14ms/step - accuracy: 0.8894 - loss: 0.2887 - val_accuracy: 0.8703 - val_loss: 0.3136
<keras.src.callbacks.history.History at 0x7ca67abda210>
```

```
# Evaluate model
test_loss, test_acc = model.evaluate(X_test, y_test)
print(f"Test Accuracy: {test_acc}")
```

 **782/782**  **2s** 3ms/step - accuracy: 0.8723 - loss: 0.3118
Test Accuracy: 0.8703200221061707

Start coding or generate with AI.