

```

import numpy as np
from tensorflow.keras.datasets import imdb
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Embedding, Flatten
from tensorflow.keras.preprocessing import sequence

max_features = 5000 # Number of words to consider as features
maxlen = 400 # Cut texts after this number of words
batch_size = 32
embedding_dims = 50
epochs = 5

# Load 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')

➞ 25000 train sequences
   25000 test sequences

# Pad sequences to make them all the same length
x_train = sequence.pad_sequences(x_train, maxlen=maxlen)
x_test = sequence.pad_sequences(x_test, maxlen=maxlen)

# Build the model
model = Sequential()

# Embedding layer
model.add(Embedding(max_features, embedding_dims, input_length=maxlen))

# Flatten the 3D tensor to 2D for dense layer
model.add(Flatten())

# Dense layer
model.add(Dense(256, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

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

# Train the model
history = model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs, validation_da

Epoch 1/5
782/782 [=====] - 69s 88ms/step - loss: 0.3874 - accuracy: 0
Epoch 2/5
782/782 [=====] - 66s 84ms/step - loss: 0.1050 - accuracy: 0
Epoch 3/5
782/782 [=====] - 69s 89ms/step - loss: 0.0235 - accuracy: 0

```

Epoch 4/5

782/782 [=====] - 67s 86ms/step - loss: 0.0146 - accuracy: 0

Epoch 5/5

782/782 [=====] - 64s 82ms/step - loss: 0.0236 - accuracy: 0

```
results = model.evaluate(x_test, y_test)
```

782/782 [=====] - 8s 10ms/step - loss: 0.7933 - accuracy: 0.



results

[0.7932878732681274, 0.8463199734687805]

Start coding or [generate](#) with AI.