

```
import tensorflow
from tensorflow import keras
from keras.datasets import imdb
import numpy as np
from keras.preprocessing.sequence import pad_sequences
from keras import Sequential
from keras.layers import Embedding, LSTM, Dense, Dropout,GRU,Bidirectional,SimpleRNN,Flatten

vocabulary_size =5000
max_words = 2697
batch_size = 64
num_epochs = 3

(X_train, y_train), (X_test, y_test) = imdb.load_data(num_words=vocabulary_size)
# print(np.shape(X_train))
# print(np.shape(X_test))
print('Loaded dataset with {} training samples, {} test samples'.format(len(X_train), len(X_test)))
print('---review---')
print(X_train[6])
print('---label---')
print(y_train[6])
word2id = imdb.get_word_index()
id2word = {i: word for word, i in word2id.items()}
print('---review with words---')
print([id2word.get(i, ' ') for i in X_train[6]])
print('---label---')
print(y_train[6])
print('Maximum review length: {}'.format(len(max((X_train + X_test), key=len))))
print('Minimum review length: {}'.format(len(min((X_test + X_test), key=len))))
X_train = pad_sequences(X_train, maxlen=max_words)
X_test = pad_sequences(X_test, maxlen=max_words)
embedding_size = 32
model = Sequential()
model.add(Embedding(vocabulary_size, embedding_size, input_length=max_words))
model.add(SimpleRNN(100))
model.add(Dense(64))
model.add(Dense(1, activation='sigmoid'))
print(model.summary())
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
X_valid, y_valid = X_train[:batch_size], y_train[:batch_size]
X_train2, y_train2 = X_train[batch_size:], y_train[batch_size:]
model.fit(X_train2, y_train2, validation_data=(X_valid, y_valid), batch_size=batch_size, epochs=num_epochs, verbose=1)
scores = model.evaluate(X_test, y_test, verbose=1)
print('Test accuracy:', scores[1])
```



