

```
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

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(LSTM(100))
# model.add(GRU(100))
# model.add(Bidirectional(LSTM(100)))
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])
```



```
Using TensorFlow backend.
Downloading data from https://s3.amazonaws.com/text-datasets/imdb.npz
17465344/17464789 [=====] - 1s 0us/step
Loaded dataset with 25000 training samples, 25000 test samples
---review---
[1, 2, 365, 1234, 5, 1156, 354, 11, 14, 2, 2, 7, 1016, 2, 2, 356, 44, 4, 1349, 500, 746, 5, 200, 4, 4132, 11, 2, 2, 111]
---label---
1
Downloading data from https://s3.amazonaws.com/text-datasets/imdb_word_index.json
1646592/1641221 [=====] - 0s 0us/step
---review with words---
['the', 'and', 'full', 'involving', 'to', 'impressive', 'boring', 'this', 'as', 'and', 'and', 'br', 'villain', 'and', '']
---label---
1
Maximum review length: 2697
Minimum review length: 14
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 2697, 32)	160000
lstm_1 (LSTM)	(None, 100)	53200
dense_1 (Dense)	(None, 1)	101
Total params: 213,301		
Trainable params: 213,301		
Non-trainable params: 0		

```
None
/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/indexed_slices.py:434: UserWarning: Converting sparse IndexedSlices to a dense Tensor of unknown shape.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
Train on 24936 samples, validate on 64 samples
Epoch 1/3
24936/24936 [=====] - 1747s 70ms/step - loss: 0.4489 - accuracy: 0.7790 - val_loss: 0.2556 - val_accuracy: 0.8745
Epoch 2/3
24936/24936 [=====] - 1683s 67ms/step - loss: 0.2848 - accuracy: 0.8871 - val_loss: 0.2360 - val_accuracy: 0.8745
Epoch 3/3
24936/24936 [=====] - 1683s 67ms/step - loss: 0.2455 - accuracy: 0.9062 - val_loss: 0.2153 - val_accuracy: 0.8745
25000/25000 [=====] - 253s 10ms/step
Test accuracy: 0.8745599985122681
```