In [ ]:

```
from __future__ import print_function

from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Embedding
from keras.layers import LSTM
from keras.datasets import imdb

max_features = 20000

# обрезание текстов после данного количества слов (среди top max_features наибол
ее используемые слова)
maxlen = 80
batch_size = 128 # увеличьте значение для ускорения обучения

print('Загрузка данных...')
(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=max_features)
print(len(x_train), 'тренировочные последовательности')
print(len(x_test), 'тестовые последовательности')

print('Pad последовательности (примеров в x единицу времени)')
x_train = sequence.pad_sequences(x_train, maxlen=maxlen)
x_test = sequence.pad_sequences(x_test, maxlen=maxlen)
print('x_train shape:', x_train.shape)
print('x_test shape:', x_test.shape)

print('Построение модели...')
model = Sequential()
model.add(Embedding(max_features, 128))
model.add(LSTM(256, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(1, activation='sigmoid'))

# стоит попробовать использовать другие оптимайзер и другие конфигурации оптимай
зеров
model.compile(loss='binary_crossentropy',
              optimizer='RMSprop',
              metrics=['accuracy'])

print('Процесс обучения...')
model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=50, # увеличьте при необходимости
          validation_data=(x_test, y_test))
score, acc = model.evaluate(x_test, y_test,
                            batch_size=batch_size)
print('Результат при тестировании:', score)
print('Тестовая точность:', acc)
```

```
Загрузка данных...
25000 тренировочные последовательности
25000 тестовые последовательности
Pad последовательности (примеров в x единицу времени)
x_train shape: (25000, 80)
x_test shape: (25000, 80)
Построение модели...
Процесс обучения...
```

```
/home/roman/anaconda3/lib/python3.7/site-packages/tensorflow_core/py
thon/framework/indexed_slices.py:433: UserWarning: Converting sparse
IndexedSlices to a dense Tensor of unknown shape. This may consume a
large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shap
e. "
```

```
Train on 25000 samples, validate on 25000 samples
Epoch 1/50
25000/25000 [==============================] - 260s 10ms/step - los
s: 0.5299 - accuracy: 0.7480 - val_loss: 0.4079 - val_accuracy: 0.82
10
Epoch 2/50
25000/25000 [==============================] - 265s 11ms/step - los
s: 0.3503 - accuracy: 0.8562 - val_loss: 0.4163 - val_accuracy: 0.82
16
Epoch 3/50
25000/25000 [==============================] - 265s 11ms/step - los
s: 0.3009 - accuracy: 0.8805 - val_loss: 0.3610 - val_accuracy: 0.84
53
Epoch 4/50
25000/25000 [==============================] - 266s 11ms/step - los
s: 0.2623 - accuracy: 0.8984 - val_loss: 0.3668 - val_accuracy: 0.83
90
Epoch 5/50
25000/25000 [==============================] - 266s 11ms/step - los
s: 0.2360 - accuracy: 0.9098 - val_loss: 0.4757 - val_accuracy: 0.82
34
Epoch 6/50
25000/25000 [==============================] - 266s 11ms/step - los
s: 0.2120 - accuracy: 0.9192 - val_loss: 0.4321 - val_accuracy: 0.83
29
Epoch 7/50
25000/25000 [==============================] - 272s 11ms/step - los
s: 0.1909 - accuracy: 0.9284 - val_loss: 0.4335 - val_accuracy: 0.83
40
Epoch 8/50
25000/25000 [==============================] - 255s 10ms/step - los
s: 0.1698 - accuracy: 0.9380 - val_loss: 0.4428 - val_accuracy: 0.83
62
Epoch 9/50
25000/25000 [==============================] - 254s 10ms/step - los
s: 0.1513 - accuracy: 0.9442 - val_loss: 0.4080 - val_accuracy: 0.83
09
Epoch 10/50
25000/25000 [==============================] - 254s 10ms/step - los
s: 0.1363 - accuracy: 0.9517 - val_loss: 0.4527 - val_accuracy: 0.82
78
Epoch 11/50
25000/25000 [==============================] - 255s 10ms/step - los
s: 0.1176 - accuracy: 0.9582 - val_loss: 0.5450 - val_accuracy: 0.80
94
Epoch 12/50
25000/25000 [==============================] - 254s 10ms/step - los
s: 0.1038 - accuracy: 0.9637 - val_loss: 0.5176 - val_accuracy: 0.82
00
Epoch 13/50
25000/25000 [==============================] - 254s 10ms/step - los
s: 0.0916 - accuracy: 0.9670 - val_loss: 0.5415 - val_accuracy: 0.82
26
Epoch 14/50
25000/25000 [==============================] - 254s 10ms/step - los
s: 0.0756 - accuracy: 0.9730 - val_loss: 0.6615 - val_accuracy: 0.80
76
Epoch 15/50
25000/25000 [==============================] - 254s 10ms/step - los
s: 0.0671 - accuracy: 0.9769 - val_loss: 0.6226 - val_accuracy: 0.81
95
```

```
Epoch 16/50
25000/25000 [==============================] - 255s 10ms/step - los
s: 0.0557 - accuracy: 0.9813 - val_loss: 0.6502 - val_accuracy: 0.81
44
Epoch 17/50
25000/25000 [==============================] - 254s 10ms/step - los
s: 0.0459 - accuracy: 0.9852 - val_loss: 0.6481 - val_accuracy: 0.80
97
Epoch 18/50
25000/25000 [==============================] - 254s 10ms/step - los
s: 0.0396 - accuracy: 0.9872 - val_loss: 0.7872 - val_accuracy: 0.81
01
Epoch 19/50
25000/25000 [==============================] - 262s 10ms/step - los
s: 0.0335 - accuracy: 0.9887 - val_loss: 0.8980 - val_accuracy: 0.81
26
Epoch 20/50
25000/25000 [==============================] - 266s 11ms/step - los
s: 0.0277 - accuracy: 0.9912 - val_loss: 0.8286 - val_accuracy: 0.81
06
Epoch 21/50
25000/25000 [==============================] - 274s 11ms/step - los
s: 0.0226 - accuracy: 0.9922 - val_loss: 0.9835 - val_accuracy: 0.80
84
Epoch 22/50
12288/25000 [=============>................] - ETA: 1:44 - loss: 0.0
142 - accuracy: 0.9961
```

In [ ]: