

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 наиболее 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 оптимайзеров
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              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)
```

Using TensorFlow backend.

Загрузка данных...

Downloading data from <https://s3.amazonaws.com/text-datasets/imdb.npz>

17465344/17464789 [=====] - 4s 0us/step

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/python/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 shape."

Train on 25000 samples, validate on 25000 samples

Epoch 1/50

25000/25000 [=====] - 260s 10ms/step - loss: 0.4649 - accuracy: 0.7786 - val_loss: 0.3627 - val_accuracy: 0.8434

Epoch 2/50

25000/25000 [=====] - 266s 11ms/step - loss: 0.3030 - accuracy: 0.8782 - val_loss: 0.3941 - val_accuracy: 0.8188

Epoch 3/50

25000/25000 [=====] - 269s 11ms/step - loss: 0.2341 - accuracy: 0.9092 - val_loss: 0.3932 - val_accuracy: 0.8253

Epoch 4/50

25000/25000 [=====] - 266s 11ms/step - loss: 0.1860 - accuracy: 0.9292 - val_loss: 0.5500 - val_accuracy: 0.8181

Epoch 5/50

25000/25000 [=====] - 265s 11ms/step - loss: 0.1455 - accuracy: 0.9477 - val_loss: 0.6443 - val_accuracy: 0.8106

Epoch 6/50

25000/25000 [=====] - 265s 11ms/step - loss: 0.1100 - accuracy: 0.9602 - val_loss: 0.6045 - val_accuracy: 0.8147

Epoch 7/50

25000/25000 [=====] - 265s 11ms/step - loss: 0.0864 - accuracy: 0.9701 - val_loss: 0.7593 - val_accuracy: 0.8117

Epoch 8/50

25000/25000 [=====] - 267s 11ms/step - loss: 0.0712 - accuracy: 0.9757 - val_loss: 0.7251 - val_accuracy: 0.8117

Epoch 9/50

25000/25000 [=====] - 267s 11ms/step - loss: 0.0585 - accuracy: 0.9797 - val_loss: 0.7361 - val_accuracy: 0.8125

Epoch 10/50

25000/25000 [=====] - 267s 11ms/step - loss: 0.0551 - accuracy: 0.9810 - val_loss: 0.8365 - val_accuracy: 0.8126

Epoch 11/50

25000/25000 [=====] - 268s 11ms/step - loss: 0.0454 - accuracy: 0.9851 - val_loss: 0.8115 - val_accuracy: 0.8019

Epoch 12/50

25000/25000 [=====] - 269s 11ms/step - loss: 0.0382 - accuracy: 0.9869 - val_loss: 0.9943 - val_accuracy: 0.8042

Epoch 13/50

25000/25000 [=====] - 269s 11ms/step - loss: 0.0327 - accuracy: 0.9896 - val_loss: 0.9356 - val_accuracy: 0.7990

Epoch 14/50

25000/25000 [=====] - 270s 11ms/step - loss: 0.0295 - accuracy: 0.9905 - val_loss: 1.0331 - val_accuracy: 0.8048

Epoch 15/50

25000/25000 [=====] - 278s 11ms/step - loss: 0.0284 - accuracy: 0.9904 - val_loss: 1.0826 - val_accuracy: 0.7987

```
Epoch 16/50
25000/25000 [=====] - 270s 11ms/step - los
s: 0.0249 - accuracy: 0.9922 - val_loss: 0.9895 - val_accuracy: 0.80
24
Epoch 17/50
25000/25000 [=====] - 269s 11ms/step - los
s: 0.0207 - accuracy: 0.9939 - val_loss: 1.0285 - val_accuracy: 0.79
88
Epoch 18/50
25000/25000 [=====] - 269s 11ms/step - los
s: 0.0166 - accuracy: 0.9947 - val_loss: 1.1477 - val_accuracy: 0.80
18
Epoch 19/50
25000/25000 [=====] - 269s 11ms/step - los
s: 0.0133 - accuracy: 0.9960 - val_loss: 1.1724 - val_accuracy: 0.80
50
Epoch 20/50
25000/25000 [=====] - 269s 11ms/step - los
s: 0.0150 - accuracy: 0.9950 - val_loss: 1.0411 - val_accuracy: 0.80
69
Epoch 21/50
20736/25000 [=====>.....] - ETA: 35s - loss: 0.01
40 - accuracy: 0.9962
```

In []: