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
from keras.layers import Conv1D
from keras.layers import MaxPooling1D
from keras.layers import Dropout
from keras.losses import mean_squared_error

max_features = 20000

# обрезание текстов после данного количества слов (среди top max_features на
иболее 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_feature
s)
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(Conv1D(filters=32, kernel_size=3, padding='same', activation='rel
u'))
#model.add(MaxPooling1D(pool_size=2))
# model.add(keras.layers.Dropout(0.3))

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 опт
имайзеров
model.compile(loss='mean_squared_logarithmic_error', # loss='binary_crossent
ropy' прогоны 1-4
              optimizer='adam', # optimizer='adam' прогон 4
              metrics=['accuracy']) # 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/python/fram
ework/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 [==============================] - 216s 9ms/step - loss: 0.0784 -
accuracy: 0.7419 - val_loss: 0.0614 - val_accuracy: 0.8130
Epoch 2/50
25000/25000 [==============================] - 224s 9ms/step - loss: 0.0467 -
accuracy: 0.8705 - val_loss: 0.0620 - val_accuracy: 0.8128
Epoch 3/50
25000/25000 [==============================] - 212s 8ms/step - loss: 0.0376 -
accuracy: 0.8966 - val_loss: 0.0602 - val_accuracy: 0.8316
Epoch 4/50
25000/25000 [==============================] - 206s 8ms/step - loss: 0.0297 -
accuracy: 0.9203 - val_loss: 0.0622 - val_accuracy: 0.8246
Epoch 5/50
25000/25000 [==============================] - 206s 8ms/step - loss: 0.0242 -
accuracy: 0.9362 - val_loss: 0.0706 - val_accuracy: 0.8080
Epoch 6/50
25000/25000 [==============================] - 204s 8ms/step - loss: 0.0200 -
accuracy: 0.9481 - val_loss: 0.0650 - val_accuracy: 0.8206
Epoch 7/50
25000/25000 [==============================] - 206s 8ms/step - loss: 0.0180 -
accuracy: 0.9536 - val_loss: 0.0685 - val_accuracy: 0.8200
Epoch 8/50
25000/25000 [==============================] - 221s 9ms/step - loss: 0.0153 -
accuracy: 0.9594 - val_loss: 0.0717 - val_accuracy: 0.8206
Epoch 9/50
25000/25000 [==============================] - 219s 9ms/step - loss: 0.0122 -
accuracy: 0.9696 - val_loss: 0.0716 - val_accuracy: 0.8200
Epoch 10/50
25000/25000 [==============================] - 205s 8ms/step - loss: 0.0104 -
accuracy: 0.9744 - val_loss: 0.0766 - val_accuracy: 0.8020
Epoch 11/50
25000/25000 [==============================] - 214s 9ms/step - loss: 0.0102 -
accuracy: 0.9743 - val_loss: 0.0750 - val_accuracy: 0.8128
Epoch 12/50
25000/25000 [==============================] - 210s 8ms/step - loss: 0.0092 -
accuracy: 0.9776 - val_loss: 0.0762 - val_accuracy: 0.8128
Epoch 13/50
25000/25000 [==============================] - 213s 9ms/step - loss: 0.0082 -
accuracy: 0.9804 - val_loss: 0.0775 - val_accuracy: 0.8168
Epoch 14/50
19968/25000 [======================>.......] - ETA: 33s - loss: 0.0071 - accu
racy: 0.9835
```

In [ ]: