In [ ]:

```
# Thanks to François Chollet  (Deep Learning with Python)
```

In [1]:

```
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
```

```
Using TensorFlow backend.
```

In [2]:

```
max_features = 20000
# cut texts after this number of words (among top max_features most common words)
maxlen = 80
batch_size = 32

print('Loading data...')
(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=max_features)
print(len(x_train), 'train sequences')
print(len(x_test), 'test sequences')
```

```
Loading data...
25000 train sequences
25000 test sequences
```

In [3]:

```
print('Pad sequences (samples x time)')
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)
```

```
Pad sequences (samples x time)
x_train shape: (25000, 80)
x_test shape: (25000, 80)
```

In [4]:

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

```
Build model...
```

In [5]:

```
# try using different optimizers and different optimizer configs
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

print('Train...')
model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=15,
          validation_data=(x_test, y_test))
score, acc = model.evaluate(x_test, y_test,
                            batch_size=batch_size)
print('Test score:', score)
print('Test accuracy:', acc)
```

```
Train...
Train on 25000 samples, validate on 25000 samples
Epoch 1/15
25000/25000 [==============================] - 114s 5ms/step - loss: 0
.4622 - acc: 0.7812 - val_loss: 0.4078 - val_acc: 0.8194
Epoch 2/15
25000/25000 [==============================] - 109s 4ms/step - loss: 0
```

```
.3022 - acc: 0.8758 - val_loss: 0.3732 - val_acc: 0.8356
Epoch 3/15
25000/25000 [==============================] - 109s 4ms/step - loss: 0
.2164 - acc: 0.9161 - val_loss: 0.4160 - val_acc: 0.8204
Epoch 4/15
25000/25000 [==============================] - 109s 4ms/step - loss: 0
.1519 - acc: 0.9427 - val_loss: 0.4447 - val_acc: 0.8262
Epoch 5/15
25000/25000 [==============================] - 109s 4ms/step - loss: 0
.1068 - acc: 0.9608 - val_loss: 0.6525 - val_acc: 0.8225
Epoch 6/15
25000/25000 [==============================] - 109s 4ms/step - loss: 0
.0771 - acc: 0.9736 - val_loss: 0.6617 - val_acc: 0.8235
Epoch 7/15
25000/25000 [==============================] - 109s 4ms/step - loss: 0
.0546 - acc: 0.9806 - val_loss: 0.7177 - val_acc: 0.8214
Epoch 8/15
25000/25000 [==============================] - 109s 4ms/step - loss: 0
.0449 - acc: 0.9852 - val_loss: 0.7366 - val_acc: 0.8195
Epoch 9/15
25000/25000 [==============================] - 109s 4ms/step - loss: 0
.0293 - acc: 0.9910 - val_loss: 0.8055 - val_acc: 0.8117
Epoch 10/15
25000/25000 [==============================] - 110s 4ms/step - loss: 0
.0224 - acc: 0.9930 - val_loss: 0.9841 - val_acc: 0.8200
Epoch 11/15
25000/25000 [==============================] - 132s 5ms/step - loss: 0
.0183 - acc: 0.9944 - val_loss: 0.9466 - val_acc: 0.8128
Epoch 12/15
25000/25000 [==============================] - 153s 6ms/step - loss: 0
.0177 - acc: 0.9947 - val_loss: 0.9773 - val_acc: 0.8144
Epoch 13/15
25000/25000 [==============================] - 153s 6ms/step - loss: 0
.0107 - acc: 0.9970 - val_loss: 1.1562 - val_acc: 0.8118
Epoch 14/15
25000/25000 [==============================] - 153s 6ms/step - loss: 0
.0099 - acc: 0.9973 - val_loss: 1.0293 - val_acc: 0.8119
Epoch 15/15
25000/25000 [==============================] - 153s 6ms/step - loss: 0
.0119 - acc: 0.9964 - val_loss: 1.0617 - val_acc: 0.8099
25000/25000 [==============================] - 24s 945us/step
Test score: 1.0616907079005242
Test accuracy: 0.80988
```

In [ ]:

In [ ]:

In [1]:

```
Using TensorFlow backend.
```

In [2]:

```
Loading data...
25000 train sequences
25000 test sequences
```

In [3]:

```
Pad sequences (samples x time)
x_train shape: (25000, 80)
x_test shape: (25000, 80)
```

In [4]:

```
Build model...
```

In [5]:

```
Train...
Train on 25000 samples, validate on 25000 samples
Epoch 1/15
25000/25000 [==============================] - 114s 5ms/step - loss: 0
.4622 - acc: 0.7812 - val_loss: 0.4078 - val_acc: 0.8194
Epoch 2/15
25000/25000 [==============================] - 109s 4ms/step - loss: 0
.3022 - acc: 0.8758 - val_loss: 0.3732 - val_acc: 0.8356
Epoch 3/15
25000/25000 [==============================] - 109s 4ms/step - loss: 0
.2164 - acc: 0.9161 - val_loss: 0.4160 - val_acc: 0.8204
Epoch 4/15
25000/25000 [==============================] - 109s 4ms/step - loss: 0
.1519 - acc: 0.9427 - val_loss: 0.4447 - val_acc: 0.8262
Epoch 5/15
25000/25000 [==============================] - 109s 4ms/step - loss: 0
.1068 - acc: 0.9608 - val_loss: 0.6525 - val_acc: 0.8225
Epoch 6/15
25000/25000 [==============================] - 109s 4ms/step - loss: 0
```

In [ ]: