In [1]:

```python
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]:

```python
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')

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)

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'))

# 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)
```

```
Loading data...
25000 train sequences
25000 test sequences
Pad sequences (samples x time)
x_train shape: (25000, 80)
x_test shape: (25000, 80)
Build model...
Train...
Train on 25000 samples, validate on 25000 samples
Epoch 1/15
25000/25000 [==============================] - 111s 4ms/step - loss: 0
.4626 - acc: 0.7826 - val_loss: 0.4118 - val_acc: 0.8111
Epoch 2/15
25000/25000 [==============================] - 110s 4ms/step - loss: 0
.3058 - acc: 0.8765 - val_loss: 0.3849 - val_acc: 0.8336
Epoch 3/15
25000/25000 [==============================] - 110s 4ms/step - loss: 0
.2249 - acc: 0.9123 - val_loss: 0.4319 - val_acc: 0.8224
Epoch 4/15
25000/25000 [==============================] - 112s 4ms/step - loss: 0
.1608 - acc: 0.9400 - val_loss: 0.4549 - val_acc: 0.8248
Epoch 5/15
25000/25000 [==============================] - 122s 5ms/step - loss: 0
.1127 - acc: 0.9584 - val_loss: 0.6717 - val_acc: 0.8000
Epoch 6/15
25000/25000 [==============================] - 110s 4ms/step - loss: 0
.0806 - acc: 0.9719 - val_loss: 0.6483 - val_acc: 0.8239
Epoch 7/15
25000/25000 [==============================] - 110s 4ms/step - loss: 0
.0566 - acc: 0.9816 - val_loss: 0.7076 - val_acc: 0.8174
Epoch 8/15
25000/25000 [==============================] - 109s 4ms/step - loss: 0
.0452 - acc: 0.9850 - val_loss: 0.7800 - val_acc: 0.8208
Epoch 9/15
25000/25000 [==============================] - 111s 4ms/step - loss: 0
.0332 - acc: 0.9899 - val_loss: 0.8109 - val_acc: 0.8193
Epoch 10/15
25000/25000 [==============================] - 110s 4ms/step - loss: 0
.0225 - acc: 0.9932 - val_loss: 0.8807 - val_acc: 0.8155
Epoch 11/15
25000/25000 [==============================] - 112s 4ms/step - loss: 0
.0209 - acc: 0.9935 - val_loss: 1.0190 - val_acc: 0.8107
Epoch 12/15
25000/25000 [==============================] - 115s 5ms/step - loss: 0
.0176 - acc: 0.9948 - val_loss: 1.0302 - val_acc: 0.8100
Epoch 13/15
25000/25000 [==============================] - 114s 5ms/step - loss: 0
.0133 - acc: 0.9960 - val_loss: 1.1126 - val_acc: 0.8146
Epoch 14/15
25000/25000 [==============================] - 115s 5ms/step - loss: 0
```

```
.0121 - acc: 0.9959 - val_loss: 1.0701 - val_acc: 0.8088
Epoch 15/15
25000/25000 [==============================] - 112s 4ms/step - loss: 0
.0079 - acc: 0.9978 - val_loss: 1.1069 - val_acc: 0.8064
25000/25000 [==============================] - 12s 497us/step
Test score: 1.1069349014878274
Test accuracy: 0.8064
```

In [ ]: