An Introduction to Deep Learning With Python

[6.5] LSTM using reversed sequence

Prof. Yuzo Iano

pgs: 220 - 222

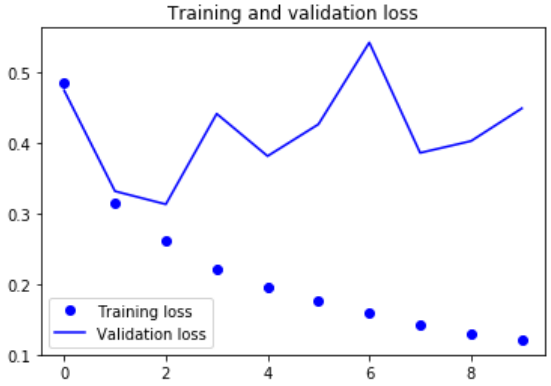
Training and evaluating an LSTM using reversed sequence

In [1]:

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

max_features = 10000
maxlen = 500

(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=max_features)

x_train = [x[::-1] for x in x_train]
x_test = [x[::-1] for x in x_test]

x_train = sequence.pad_sequences(x_train, maxlen=maxlen)
x_test = sequence.pad_sequences(x_test, maxlen=maxlen)

model = Sequential()
model.add(Embedding(max_features, 128))
model.add(LSTM(32))
model.add(Dense(1, activation='sigmoid'))
model.summary()

model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['acc'])

history = model.fit(x_train, y_train,
epochs=10,
batch_size=128,
validation_split=0.2)
```

```
Using TensorFlow backend.

WARNING:tensorflow:From C:\Users\pablo\AppData\Roaming\Python\Python36\site-packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 128)         1280000
_________________________________________________________________
lstm_1 (LSTM)                (None, 32)                20608
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 33
=================================================================
Total params: 1,300,641
Trainable params: 1,300,641
Non-trainable params: 0
_________________________________________________________________
WARNING:tensorflow:From C:\Users\pablo\AppData\Roaming\Python\Python36\site-packages\tensorflow\python\ops\math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Train on 20000 samples, validate on 5000 samples
Epoch 1/10
20000/20000 [==============================] - 114s 6ms/step - loss: 0.4864 - acc: 0.7686 - val_loss: 0.4752 - val_acc: 0.8182
Epoch 2/10
20000/20000 [==============================] - 118s 6ms/step - loss: 0.3147 - acc: 0.8794 - val_loss: 0.3321 - val_acc: 0.8638
Epoch 3/10
20000/20000 [==============================] - 210s 11ms/step - loss: 0.2620 - acc: 0.9041 - val_loss: 0.3135 - val_acc: 0.8754
Epoch 4/10
20000/20000 [==============================] - 269s 13ms/step - loss: 0.2207 - acc: 0.9199 - val_loss: 0.4420 - val_acc: 0.8542
Epoch 5/10
20000/20000 [==============================] - 285s 14ms/step - loss: 0.1963 - acc: 0.9290 - val_loss: 0.3819 - val_acc: 0.8404
Epoch 6/10
20000/20000 [==============================] - 285s 14ms/step - loss: 0.1772 - acc: 0.9377 - val_loss: 0.4270 - val_acc: 0.8376
Epoch 7/10
20000/20000 [==============================] - 202s 10ms/step - loss: 0.1599 - acc: 0.9464 - val_loss: 0.5431 - val_acc: 0.8494
Epoch 8/10
20000/20000 [==============================] - 196s 10ms/step - loss: 0.1422 - acc: 0.9514 - val_loss: 0.3865 - val_acc: 0.8686
Epoch 9/10
20000/20000 [==============================] - 193s 10ms/step - loss: 0.1297 - acc: 0.9568 - val_loss: 0.4032 - val_acc: 0.8672
Epoch 10/10
20000/20000 [==============================] - 147s 7ms/step - loss: 0.1212 - acc: 0.9606 - val_loss: 0.4497 - val_acc: 0.8594
```

In [3]:
```
import matplotlib.pyplot as plt

loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(loss))

plt.figure()

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()

plt.show()
```

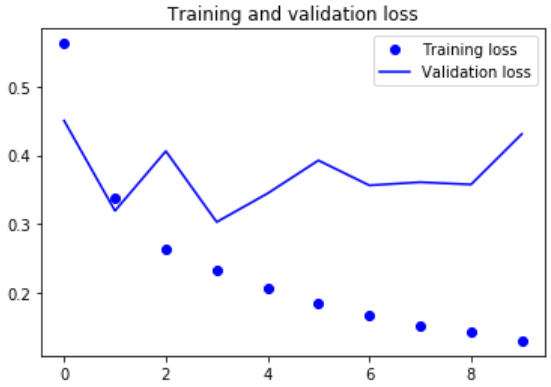


Training and evaluating a bidirectional LSTM

In [4]:
```
from keras.layers import Bidirectional

model = Sequential()
model.add(Embedding(max_features, 32))
model.add(Bidirectional(LSTM(32)))
model.add(Dense(1, activation='sigmoid'))
model.summary()

model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])

history = model.fit(x_train, y_train,
                    epochs=10,
                    batch_size=128,
                    validation_split=0.2)
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, None, 32)          320000
_________________________________________________________________
bidirectional_1 (Bidirection (None, 64)                16640
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 65
=================================================================
Total params: 336,705
Trainable params: 336,705
Non-trainable params: 0
_________________________________________________________________
Train on 20000 samples, validate on 5000 samples
Epoch 1/10
20000/20000 [==============================] - 202s 10ms/step - loss: 0.5622 - acc: 0.7172 - val_loss: 0.4
503 - val_acc: 0.8164
Epoch 2/10
20000/20000 [==============================] - 226s 11ms/step - loss: 0.3374 - acc: 0.8713 - val_loss: 0.3
196 - val_acc: 0.8814
Epoch 3/10
20000/20000 [==============================] - 314s 16ms/step - loss: 0.2650 - acc: 0.9017 - val_loss: 0.4
061 - val_acc: 0.8754
Epoch 4/10
20000/20000 [==============================] - 309s 15ms/step - loss: 0.2330 - acc: 0.9181 - val_loss: 0.3
033 - val_acc: 0.8834
Epoch 5/10
20000/20000 [==============================] - 312s 16ms/step - loss: 0.2073 - acc: 0.9254 - val_loss: 0.3
448 - val_acc: 0.8906
Epoch 6/10
20000/20000 [==============================] - 311s 16ms/step - loss: 0.1849 - acc: 0.9356 - val_loss: 0.3
927 - val_acc: 0.8766
Epoch 7/10
20000/20000 [==============================] - 279s 14ms/step - loss: 0.1687 - acc: 0.9415 - val_loss: 0.3
565 - val_acc: 0.8810
Epoch 8/10
20000/20000 [==============================] - 251s 13ms/step - loss: 0.1521 - acc: 0.9487 - val_loss: 0.3
611 - val_acc: 0.8730
Epoch 9/10
20000/20000 [==============================] - 181s 9ms/step - loss: 0.1441 - acc: 0.9511 - val_loss: 0.35
77 - val_acc: 0.8526
Epoch 10/10
20000/20000 [==============================] - 178s 9ms/step - loss: 0.1312 - acc: 0.9565 - val_loss: 0.43
10 - val_acc: 0.8660
```

In [5]:
```
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(loss))

plt.figure()

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()

plt.show()
```



Pablo Minango

- pablodavid218@gmail.com