PS12

January 24, 2023

0.1 Recurrent Neural Networks

[1]:
```
import tensorflow as tf
from tensorflow.keras.datasets import imdb
import numpy as np
```

0.2 Loading IMDB

[2]:
```
from tensorflow.keras.datasets import imdb

(X_train,y_train), (X_test,y_test) = imdb.load_data(num_words=10000)
```

[3]:
```
# Get IMDB dictionary
word_idx_dict = imdb.get_word_index()

#print(word_idx_dict.keys())
#print(word_idx_dict.values())
```

0.3 Data preprocessing: Padding

[4]:
```
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

[5]:
```
print(len(X_train[0]))
```

```
218
```

[6]:
```
max_len = 256
X_train_pad = pad_sequences(X_train,value=0, padding='post', maxlen=256)
```

[7]:
```
print(X_train_pad.shape)
print(len(X_train_pad[0]))
print(X_train_pad[0].shape)
print(X_train_pad[0])
```

```
(25000, 256)
256
(256,)
[   1   14   22   16   43  530  973 1622 1385   65  458 4468   66 3941
```

```
   4  173   36  256    5   25  100   43  838  112   50  670    2    9
  35  480  284    5  150    4  172  112  167    2  336  385   39    4
 172 4536 1111   17  546   38   13  447    4  192   50   16    6  147
2025   19   14   22    4 1920 4613  469    4   22   71   87   12   16
  43  530   38   76   15   13 1247    4   22   17  515   17   12   16
 626   18    2    5   62  386   12    8  316    8  106    5    4 2223
5244   16  480   66 3785   33    4  130   12   16   38  619    5   25
 124   51   36  135   48   25 1415   33    6   22   12  215   28   77
  52    5   14  407   16   82    2    8    4  107  117 5952   15  256
   4    2    7 3766    5  723   36   71   43  530  476   26  400  317
  46    7    4    2 1029   13  104   88    4  381   15  297   98   32
2071   56   26  141    6  194 7486   18    4  226   22   21  134  476
  26  480    5  144   30 5535   18   51   36   28  224   92   25  104
   4  226   65   16   38 1334   88   12   16  283    5   16 4472  113
 103   32   15   16 5345   19  178   32    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0]
```

[8]:
```
#word_idx_dict_new['<PAD>']=0
#word_idx_dict_new
```

[9]:
```
#print(decode_review(X_train_pad[0]))
```

[10]:
```
X_test_pad = pad_sequences(X_test, value=0, padding='post',maxlen=max_len)
```

[11]:
```
print(X_train_pad.shape)
print(X_test_pad.shape)
```

```
(25000, 256)
(25000, 256)
```

0.4 Simple RNN

[13]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Embedding
from tensorflow.keras.layers import SimpleRNN

basic_rnn = Sequential()
basic_rnn.add(Embedding(input_dim=10000, output_dim=100, input_length=256))
basic_rnn.add(SimpleRNN(128))
basic_rnn.add(Dense(1,activation='sigmoid'))
```

[13]:
```
# validation dataset
X_train_pad, X_val_pad = X_train_pad[:20000], X_train_pad[20000:]
y_train, y_val = y_train[:20000], y_train[20000:]
```

[14]:
```
basic_rnn.compile(optimizer='adam',loss='binary_crossentropy',metrics='acc')
#history_basic_rnn = basic_rnn.fit(X_train_pad,y_train,
#                                  validation_data=(X_val_pad,y_val),
#                                  epochs=10)
history_basic_rnn = basic_rnn.fit(X_train_pad,y_train, epochs=10)
```

```
Epoch 1/10
782/782 [==============================] - 24s 30ms/step - loss: 0.6980 - acc:
0.5087
Epoch 2/10
782/782 [==============================] - 25s 32ms/step - loss: 0.6946 - acc:
0.5164
Epoch 3/10
782/782 [==============================] - 25s 33ms/step - loss: 0.6777 - acc:
0.5491
Epoch 4/10
782/782 [==============================] - 26s 33ms/step - loss: 0.6458 - acc:
0.5723
Epoch 5/10
782/782 [==============================] - 27s 35ms/step - loss: 0.6249 - acc:
0.6053
Epoch 6/10
782/782 [==============================] - 27s 34ms/step - loss: 0.6476 - acc:
0.5698
Epoch 7/10
782/782 [==============================] - 27s 34ms/step - loss: 0.6417 - acc:
0.5758
Epoch 8/10
782/782 [==============================] - 26s 34ms/step - loss: 0.6218 - acc:
0.5865
Epoch 9/10
782/782 [==============================] - 27s 34ms/step - loss: 0.5929 - acc:
0.6069
Epoch 10/10
782/782 [==============================] - 27s 34ms/step - loss: 0.5744 - acc:
0.6210
```

[15]:
```
test_performance = basic_rnn.evaluate(X_test_pad,y_test)
print(test_performance)
```

```
782/782 [==============================] - 8s 10ms/step - loss: 0.7149 - acc:
0.5500
[0.7149398922920227, 0.5500400066375732]
```

0.5 LSTM

[23]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Embedding
from tensorflow.keras.layers import LSTM

LSTM_model = Sequential()
LSTM_model.add(Embedding(input_dim=10000,output_dim=100,input_length=256))
LSTM_model.add(LSTM(128))
LSTM_model.add(Dense(1,activation='sigmoid'))
LSTM_model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 256, 100)          1000000

 lstm (LSTM)                 (None, 128)               117248

 dense_2 (Dense)             (None, 1)                 129

=================================================================
Total params: 1,117,377
Trainable params: 1,117,377
Non-trainable params: 0
_________________________________________________________________
```

[24]:
```
LSTM_model.compile(optimizer='adam',
                   loss='binary_crossentropy',
                   metrics='acc')

LSTM_model.fit(X_train_pad,y_train, epochs=10)
```

```
Epoch 1/10
625/625 [==============================] - 91s 144ms/step - loss: 0.6758 - acc:
0.5514
Epoch 2/10
625/625 [==============================] - 87s 140ms/step - loss: 0.6407 - acc:
0.5856
Epoch 3/10
625/625 [==============================] - 88s 141ms/step - loss: 0.6165 - acc:
0.6079
Epoch 4/10
625/625 [==============================] - 86s 138ms/step - loss: 0.5761 - acc:
0.6523
Epoch 5/10
```

```
625/625 [==============================] - 87s 139ms/step - loss: 0.4452 - acc:
0.7847
Epoch 6/10
625/625 [==============================] - 85s 136ms/step - loss: 0.2666 - acc:
0.8983
Epoch 7/10
625/625 [==============================] - 85s 136ms/step - loss: 0.1760 - acc:
0.9370
Epoch 8/10
625/625 [==============================] - 86s 138ms/step - loss: 0.1161 - acc:
0.9640
Epoch 9/10
625/625 [==============================] - 85s 137ms/step - loss: 0.0744 - acc:
0.9805
Epoch 10/10
625/625 [==============================] - 86s 138ms/step - loss: 0.0536 - acc:
0.9872
```

[24]:
```
<keras.callbacks.History at 0x17731d55280>
```

[76]:
```
#test_performance = LSTM_model.evaluate(X_test_pad,y_test)[1]
test_performance = LSTM_model.evaluate(X_test_pad,y_test)
print(test_performance)
```

```
782/782 [==============================] - 22s 28ms/step - loss: 0.5110 - acc:
0.8502
782/782 [==============================] - 22s 28ms/step - loss: 0.5110 - acc:
0.8502
[0.5110498666763306, 0.8502399921417236]
```