Assignment 10-3

May 20, 2021

[1]:
```
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()

from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
import numpy as np
import matplotlib.pyplot as plt
from pathlib import Path
from keras.models import Sequential
from keras.layers import Embedding, Flatten, Dense
import os
from contextlib import redirect_stdout
import time
start_time = time.time()
from keras.layers import LSTM
# Needed the following as caused CUDA DNN errors
#physical_devices = tf.config.list_physical_devices('GPU')
#tf.config.experimental.set_memory_growth(physical_devices[0], True)

from keras.datasets import imdb
from keras.preprocessing import sequence
```

```
WARNING:tensorflow:From /opt/conda/lib/python3.8/site-
packages/tensorflow/python/compat/v2_compat.py:96: disable_resource_variables
(from tensorflow.python.ops.variable_scope) is deprecated and will be removed in
a future version.
Instructions for updating:
non-resource variables are not supported in the long term
```

[2]:
```
imdb_dir = Path('imdb/aclImdb/')
test_dir = os.path.join(imdb_dir, 'test')
train_dir = os.path.join(imdb_dir, 'train')

results_dir = Path('results').joinpath('model_1')
results_dir.mkdir(parents=True, exist_ok=True)
```

[3]:
```
max_features = 10000
maxlen = 500
batch_size = 32
max_words = 1000
training_samples = 200
validation_samples = 10000
```

[4]:
```
labels = []
texts = []

for label_type in ['neg', 'pos']:
    dir_name = os.path.join(test_dir, label_type)
    for fname in sorted(os.listdir(dir_name)):
        if fname[-4:] == '.txt':
            f = open(os.path.join(dir_name, fname), encoding="utf8")
            texts.append(f.read())
            f.close()
            if label_type == 'neg':
                labels.append(0)
            else:
                labels.append(1)
```

[5]:
```
tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(texts)
sequences = tokenizer.texts_to_sequences(texts)

print('Loading data... ')

word_index = tokenizer.word_index
print('Found %s unique tokens.' % len(word_index))

data = pad_sequences(sequences, maxlen=maxlen)
labels = np.asarray(labels)
print('Shape of data tensor:', data.shape)
print('Shape of label tensor:', labels.shape)

indices = np.arange(data.shape[0])
np.random.shuffle(indices)
data = data[indices]
labels = labels[indices]
```

```
Loading data…
Found 87393 unique tokens.
Shape of data tensor: (25000, 500)
Shape of label tensor: (25000,)
```

[6]:
```
#x_train
input_train = data[:training_samples]
#y_train
y_train = labels[:training_samples]

#x_val
input_test = data[training_samples: training_samples + validation_samples]
#y_val
y_test = labels[training_samples: training_samples + validation_samples]

print('input_train shape:', input_train.shape)
print('input_test shape:', input_test.shape)
```

```
input_train shape: (200, 500)
input_test shape: (10000, 500)
```

[7]:
```
model = Sequential()
model.add(Embedding(max_features, 32))
model.add(LSTM(32))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
history=model.fit(input_train, y_train, epochs=10, batch_size=32,␣
 ↪validation_data=(input_test, y_test))

result_model_file = results_dir.joinpath('pre_trained_glove_model_LSTM.h5')
model.save_weights(result_model_file)
```

```
WARNING:tensorflow:From /opt/conda/lib/python3.8/site-
packages/tensorflow/python/keras/initializers/initializers_v1.py:58: calling
RandomUniform.__init__ (from tensorflow.python.ops.init_ops) with dtype is
deprecated and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the
constructor
Train on 200 samples, validate on 10000 samples
Epoch 1/10
200/200 [==============================] - ETA: 0s - loss: 0.6900 - acc: 0.5800

/opt/conda/lib/python3.8/site-
packages/tensorflow/python/keras/engine/training.py:2325: UserWarning:
`Model.state_updates` will be removed in a future version. This property should
not be used in TensorFlow 2.0, as `updates` are applied automatically.
  warnings.warn('`Model.state_updates` will be removed in a future version. '

200/200 [==============================] - 23s 116ms/sample - loss: 0.6900 -
acc: 0.5800 - val_loss: 0.6936 - val_acc: 0.5041
Epoch 2/10
200/200 [==============================] - 23s 113ms/sample - loss: 0.6770 -
```

```
acc: 0.5600 - val_loss: 0.6945 - val_acc: 0.5041
Epoch 3/10
200/200 [==============================] - 23s 115ms/sample - loss: 0.6610 -
acc: 0.5600 - val_loss: 0.6931 - val_acc: 0.5042
Epoch 4/10
200/200 [==============================] - 20s 101ms/sample - loss: 0.6283 -
acc: 0.5850 - val_loss: 0.6773 - val_acc: 0.5517
Epoch 5/10
200/200 [==============================] - 21s 104ms/sample - loss: 0.6372 -
acc: 0.6950 - val_loss: 0.6968 - val_acc: 0.5395
Epoch 6/10
200/200 [==============================] - 21s 105ms/sample - loss: 0.5301 -
acc: 0.8300 - val_loss: 0.7583 - val_acc: 0.5660
Epoch 7/10
200/200 [==============================] - 20s 98ms/sample - loss: 0.5037 - acc:
0.8700 - val_loss: 0.6628 - val_acc: 0.5935
Epoch 8/10
200/200 [==============================] - 19s 94ms/sample - loss: 0.4773 - acc:
0.8800 - val_loss: 0.6550 - val_acc: 0.6384
Epoch 9/10
200/200 [==============================] - 19s 95ms/sample - loss: 0.3956 - acc:
0.9200 - val_loss: 0.6325 - val_acc: 0.6618
Epoch 10/10
200/200 [==============================] - 19s 93ms/sample - loss: 0.4260 - acc:
0.9150 - val_loss: 0.6586 - val_acc: 0.6496
```

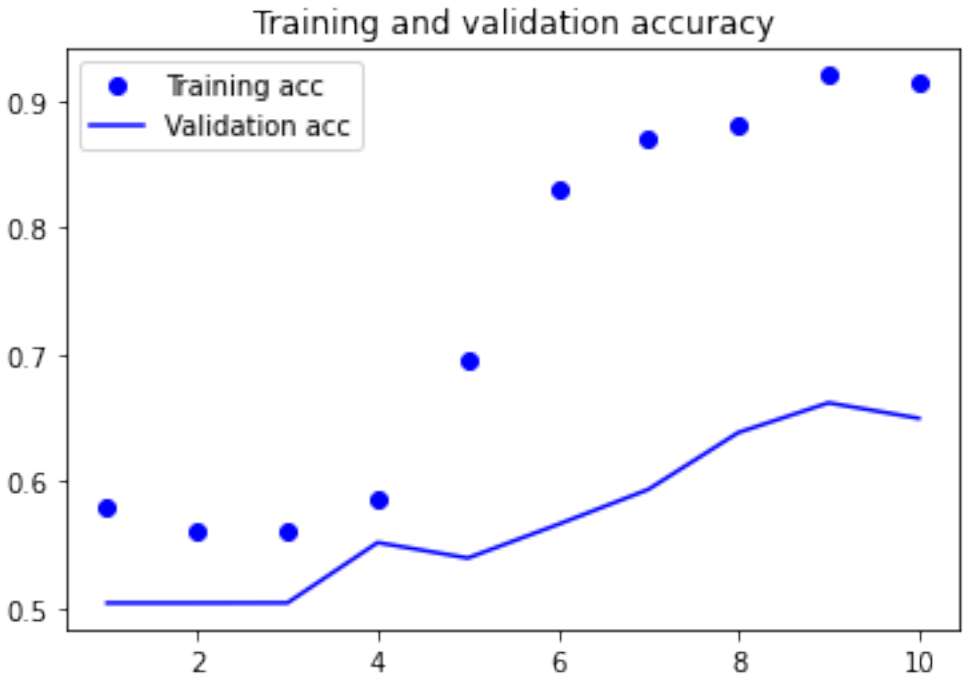
[8]:
```
# Save the summary to file
summary_file = results_dir.joinpath('Assignment_10.3_ModelSummary.txt')
with open(summary_file, 'w') as f:
    with redirect_stdout(f):
        model.summary()
```

[9]:
```
# Place plot here
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(1, len(acc) + 1)

plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()
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()
img_file = results_dir.joinpath('Assignment_10.3_Model Accuracy Validation.png')
plt.savefig(img_file)
plt.show()
```




Training and validation loss
Training loss
Validation loss
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
2
4
6
8
10

[ ]: