

Assignment_10.3

August 3, 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
```

```
[5]: 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)
```

```
[6]: 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,)
```

```
[7]: #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)
```

```
[8]: 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)
```

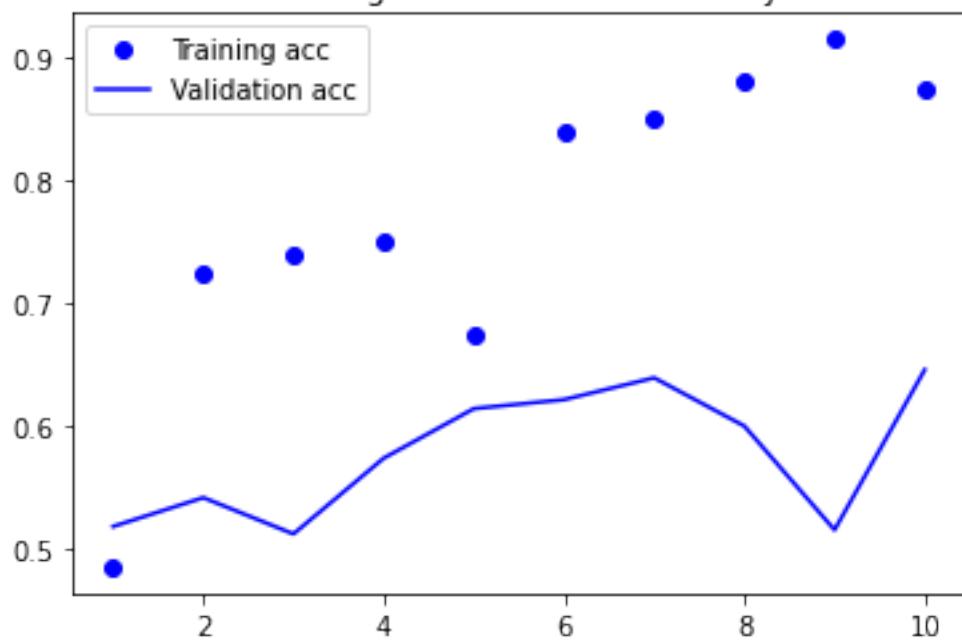
```
WARNING:tensorflow:From /opt/conda/lib/python3.8/site-
packages/tensorflow/python/keras/initializers/initializers_v1.py:58: calling
RandomUniform.__init__ (from tensorflow.python.ops.init_ops) with dtype is
deprecated and will be removed in a future version.
Instructions for updating:
Call initializer 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: 0.6937 - acc: 0.4850
/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 115ms/sample - loss: 0.6937 -
acc: 0.4850 - val_loss: 0.6929 - val_acc: 0.5180
Epoch 2/10
200/200 [=====] - 22s 111ms/sample - loss: 0.6881 -
acc: 0.7250 - val_loss: 0.6922 - val_acc: 0.5415
Epoch 3/10
200/200 [=====] - 22s 112ms/sample - loss: 0.6803 -
acc: 0.7400 - val_loss: 0.6899 - val_acc: 0.5119
Epoch 4/10
200/200 [=====] - 22s 110ms/sample - loss: 0.6442 -
acc: 0.7500 - val_loss: 0.6842 - val_acc: 0.5736
Epoch 5/10
200/200 [=====] - 22s 111ms/sample - loss: 0.6651 -
acc: 0.6750 - val_loss: 0.6786 - val_acc: 0.6140
Epoch 6/10
200/200 [=====] - 22s 109ms/sample - loss: 0.6131 -
acc: 0.8400 - val_loss: 0.6699 - val_acc: 0.6213
Epoch 7/10
200/200 [=====] - 22s 108ms/sample - loss: 0.5503 -
```

```
acc: 0.8500 - val_loss: 0.6606 - val_acc: 0.6391
Epoch 8/10
200/200 [=====] - 21s 104ms/sample - loss: 0.5279 -
acc: 0.8800 - val_loss: 0.6675 - val_acc: 0.5998
Epoch 9/10
200/200 [=====] - 21s 105ms/sample - loss: 0.4536 -
acc: 0.9150 - val_loss: 0.9446 - val_acc: 0.5152
Epoch 10/10
200/200 [=====] - 20s 102ms/sample - loss: 0.4545 -
acc: 0.8750 - val_loss: 0.6406 - val_acc: 0.6457
```

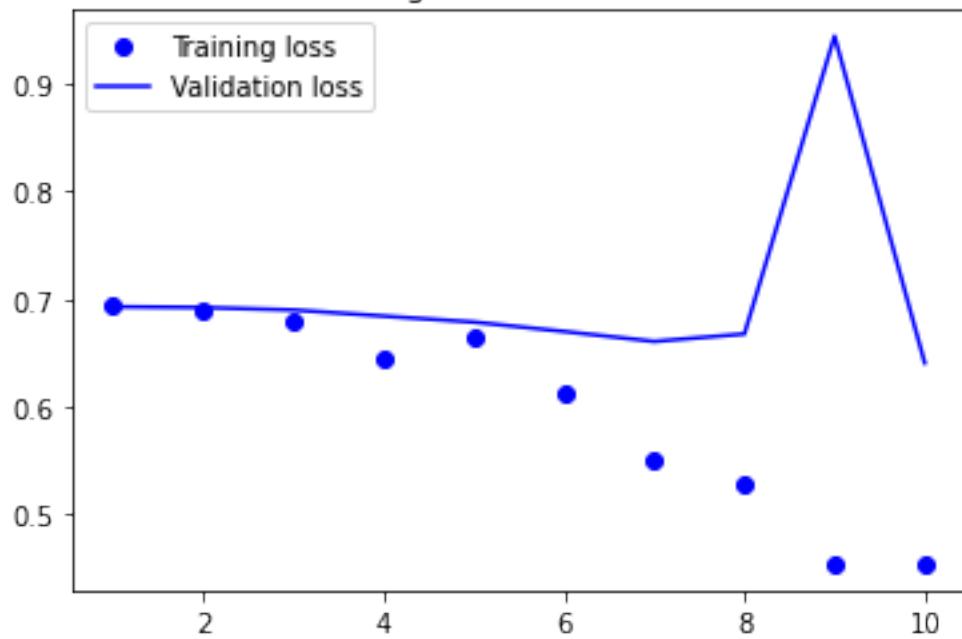
```
[9]: # Save the summary to file
summary_file = results_dir.joinpath('Assignment_10.3_ModelSummary.txt')
with open(summary_file, 'w') as f:
    with redirect_stdout(f):
        model.summary()
```

```
[10]: # 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 accuracy



Training and validation loss



[]: