assignment11

February 11, 2022

Assignment 11 Using section 8.1 in Deep Learning with Python as a guide, implement an LSTM text generator. Train the model on the Enron corpus or a text source of your choice. Save the model and generate 20 examples to the results directory of dsc650/assignments/assignment11/

[1]:

```
import os
import json
from pathlib import Path

import keras
import numpy as np

import random
import sys
```

```
Using TensorFlow backend.
```

[16]:

```
current_dir = Path(os.getcwd()).absolute()
results_dir = current_dir.joinpath('results')
results_dir.mkdir(parents=True, exist_ok=True)
data_dir = current_dir.joinpath('data')
data_dir.mkdir(parents=True, exist_ok=True)
corpus_data_dir = data_dir.joinpath('corpus')

print(current_dir)
print(results_dir)
print(corpus_data_dir)
```

```
c:\Users\saman\git_repos\dsc650\dsc650\assignments\assignments11
c:\Users\saman\git_repos\dsc650\dsc650\assignments\assignments11\results
c:\Users\saman\git_repos\dsc650\dsc650\assignments\assignments11\data\corpus
```

[3]:

```
# Downloading and parsing the initial text
def download_file(tgt_filename, path_to_file):
        path = keras.utils.get_file(
                tgt_filename,
                origin=path_to_file)
        print('Downloaded into: ', path)
        corpus = open(path,encoding = 'utf-8').read().lower()
        return(corpus)
```

[4]:

```
filename = '3090-0.txt'
web_site = 'https://www.gutenberg.org/files/3090/3090-0.txt'

# filename = 'nietzsche.txt'
# web_site ='https://s3.amazonaws.com/text-datasets/nietzsche.txt'

text = download_file(filename, web_site)
print('Corpus length:', len(text))
```

```
Downloaded into:  C:\Users\saman\.keras\datasets\3090-0.txt
Corpus length: 2730110
```

[ ]:

```
# text = '''As a boy, the author dreamed of wonderful municipal playgrounds,
# of organizations giving the boys opportunity to camp in the open,
# of zoological and botanical gardens planned and adapted to the
# understanding of youth. His busy life as a civil engineer, surveyor,
# and work in the open gave him no opportunity to develop his dreams, but
# at the end of a five year tour of the United States and Canada, made
# over fifty years ago, he drifted into New York City and was shocked
# beyond expression by the almost total lack of breathing spaces for our
# boys, in the greatest of American cities. True, it then had Central
# Park; but fifty years ago Central Park was out among the goats, only to
# be reached by a long and tiresome horse car journey.'''
```

[5]:

```
# Vectorizing sequences of characters

maxlen = 60 # extract sequences of 60 characters
step = 3 # sample a new sequence every 3 characters
sentences = [] # holds the extracted sequences
next_chars = [] # holds the targets (in this case the next character)
for i in range(0, len(text) - maxlen, step):
        sentences.append(text[i: i + maxlen])
        next_chars.append(text[i + maxlen])
print('Number of sequences:', len(sentences) )
```

```
Number of sequences: 910017
```

[ ]:

```
sentences[:10]
```

[ ]:

```
next_chars[:10]
```

[6]:

```
# list of unique characters in the corpus
chars = sorted(list(set(text)))
print('Unique characters:', len(chars))
# Dictionary that maps unique characters to their index in the list "chars"
char_indices = dict((char, chars.index(char)) for char in chars)
```

```
Unique characters: 60
```

[ ]:

```
'  '.join(chars)
```

[ ]:

```
char_indices
```

[7]:

```
# One-hot encodes the characters into binary arrays

x = np.zeros((len(sentences), maxlen, len(chars)), dtype=np.bool)
y = np.zeros((len(sentences), len(chars)), dtype=np.bool)
for i, sentence in enumerate(sentences):
        for t, char in enumerate(sentence):
                x[i, t, char_indices[char]] = 1
        y[i, char_indices[next_chars[i]]] = 1
```

[8]:

```
print('shape of input', x.shape)
print('shape of target', y.shape)
```

```
shape of input (910017, 60, 60)
shape of target (910017, 60)
```

[9]:

```
# Single layer LSTM model for next-character prediction
# Tis network is a single LSTM layer followed by a Dense classifier and softmax␣
 ↪over all possible characters
from keras import layers

model = keras.models.Sequential()
model.add(layers.LSTM(128, input_shape=(maxlen, len(chars))))
model.add(layers.Dense(len(chars), activation = 'softmax'))
model.summary()
```

```
Model: "sequential_1"

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 128)               96768

_________________________________________________________________
dense_1 (Dense)              (None, 60)                7740
=================================================================
Total params: 104,508
Trainable params: 104,508
Non-trainable params: 0

_________________________________________________________________
```

[13]:

```
# Model compilation configuration
optimizer = keras.optimizers.RMSprop(lr=0.01)
model.compile(
        loss='categorical_crossentropy',
        optimizer=optimizer,
        metrics=['acc']         )
```

```python
# fits the model for one iteration on the data
history = model.fit(
        x, y,
        batch_size=128,
        epochs=10,
        validation_split=0.2)
#saving the model
model.save_weights('LSTM_3090-0.h5')
```

```
Train on 728013 samples, validate on 182004 samples
Epoch 1/10
728013/728013 [==============================] - 1558s 2ms/step - loss: 1.4971 -
acc: 0.5444 - val_loss: 1.5081 - val_acc: 0.5440
Epoch 2/10
728013/728013 [==============================] - 1708s 2ms/step - loss: 1.4585 -
acc: 0.5550 - val_loss: 1.4975 - val_acc: 0.5465
Epoch 3/10
728013/728013 [==============================] - 1503s 2ms/step - loss: 1.4362 -
acc: 0.5613 - val_loss: 1.4851 - val_acc: 0.5525s - loss:
Epoch 4/10
728013/728013 [==============================] - 1392s 2ms/step - loss: 1.4222 -
acc: 0.5648 - val_loss: 1.4774 - val_acc: 0.5540
Epoch 5/10
728013/728013 [==============================] - 1056s 1ms/step - loss: 1.4135 -
acc: 0.5676 - val_loss: 1.4697 - val_acc: 0.5565
Epoch 6/10
728013/728013 [==============================] - 1017s 1ms/step - loss: 1.4056 -
acc: 0.5701 - val_loss: 1.4766 - val_acc: 0.5586
Epoch 7/10
728013/728013 [==============================] - 1344s 2ms/step - loss: 1.3994 -
acc: 0.5714 - val_loss: 1.4714 - val_acc: 0.5589
Epoch 8/10
728013/728013 [==============================] - 1193s 2ms/step - loss: 1.3953 -
acc: 0.5726 - val_loss: 1.4731 - val_acc: 0.5591
Epoch 9/10
728013/728013 [==============================] - 1249s 2ms/step - loss: 1.3907 -
acc: 0.5744 - val_loss: 1.4659 - val_acc: 0.5587
Epoch 10/10
728013/728013 [==============================] - 1010s 1ms/step - loss: 1.3875 -
acc: 0.5746 - val_loss: 1.4804 - val_acc: 0.5590
```

[14]:

```python
# Function to sample the next character given the model's prediction

def sample(preds, temperature):
        preds = np.asarray(preds).astype('float64')
        preds = np.log(preds) / temperature
        exp_preds = np.exp(preds)
        preds = exp_preds / np.sum(exp_preds)
        probas = np.random.multinomial(1, preds, 1)
        return np.argmax(probas)
```

[17]:

```
# Text Generation loop
# Trains the model for n epochs
for epoch in range(1, 21):
        print('epoch', epoch)
        with open(f'{results_dir}\\file_{epoch}.txt', 'w') as wf:
                # fits the model for one iteration on the data
            # model.fit(x, y, batch_size=128, epochs=1)

                start_index = random.randint(0, len(text) - maxlen - 1)
                generated_text = text[start_index: start_index + maxlen]
                print('--- Generating with seed: "' + generated_text + '"')

                wf.write(f'--- Generating with seed: {generated_text}')
                for temperature in [0.2, 0.5, 1.0, 1.2]:
                        print('------ temperature:', temperature)
                        wf.write(f'\n ------ temperature: {temperature} \n')
                        # sys.stdout.write(generated_text)
                        # We generate 400 caracters

                        for i in range(250):
                                sampled = np.zeros((1, maxlen, len(chars)))
                                for t, char in enumerate(generated_text):
                                        sampled[0, t, char_indices[char]] = 1
                                # Load and predict using the model on the sampled text
                                model.load_weights('LSTM_3090-0.h5')
                                preds = model.predict(sampled, verbose=0)[0]
                                next_index = sample(preds, temperature)
                                next_char = chars[next_index]

                                generated_text += next_char
                                generated_text = generated_text[1:]

                                # sys.stdout.write(next_char)
                                wf.write(f'{next_char}')
                                sys.stdout.flush()
                        #print()
```

```
epoch 1
--- Generating with seed: "
to look at them. people scarcely greeted her; only a few me"
------ temperature: 0.2
------ temperature: 0.5
```

```
------ temperature: 1.0
```

```
C:\Users\saman\.conda\envs\dsc650\lib\site-packages\ipykernel_launcher.py:5:
RuntimeWarning: divide by zero encountered in log
  """
```

```
------ temperature: 1.2
epoch 2
--- Generating with seed: "an go and get them. we must take good care that our
son-
in-"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 3
--- Generating with seed: "he withstood time like an old tree; but
celeste, especially "
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 4
--- Generating with seed: "o has just left-you know, that excellent man
whom i have mad"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 5
--- Generating with seed: "t she began to cry, without
knowing why. the young man was n"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 6
--- Generating with seed: "at age, and
more than once asked doctor chenet, emphasizing "
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 7
--- Generating with seed: "ou can choose . . . you can choose . . . yes, you can
choose"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
```

```
------ temperature: 1.2
epoch 8
--- Generating with seed: "tain was a bold man of an
inventive mind, and this was the p"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 9
--- Generating with seed: "like an answer to a cry uttered by my soul, to that
vague an"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 10
--- Generating with seed: " to him. he stooped down
and clasped her closely in his arms"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 11
--- Generating with seed: " and tranquil, father, mother, and
child walking on the boul"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 12
--- Generating with seed: "ion which
did not exist, and he would answer nothing. i felt"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 13
--- Generating with seed: "od motionless, and did not open their lips.

the prussian, p"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 14
--- Generating with seed: "good school, and once he felt his steed between his
legs he
"
```

```
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 15
--- Generating with seed: "elder, at last, feeling that he was lost, murmured
once
more"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 16
--- Generating with seed: "ses of france belong to us!"

the others, who were quite dru"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 17
--- Generating with seed: "ispered scandal, the mind soiled by all the filth
that
is ta"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 18
--- Generating with seed: " mystery.

a door opened and a servant entered, holding on a"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 19
--- Generating with seed: "t with them a
fine rabbit-stolen, doubtless-and they made si"
------ temperature: 0.2
------ temperature: 0.5
------ temperature: 1.0
------ temperature: 1.2
epoch 20
--- Generating with seed: "caravan, to explain matters, said: "she has been
somewhat il"
------ temperature: 0.2
------ temperature: 0.5
```

```
------ temperature: 1.0
------ temperature: 1.2
```

[ ]:

```

```