Char-RNN

Reading Ref: http://karpathy.github.io/2015/05/21/rnn-effectiveness/ (http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

Text Generation

Download the Shakespear dataset from Andrej's Blog (https://cs.stanford.edu/people/karpathy/char-rnn/shakespear.txt)

Approach (broader view):

- load the dataset
- convert categorical values into some numerical representation; we'll create mapping of char to int
- setup the sequence logic

```
if the sequence length is 4,
HELL->O
WORL->D
```

- set up the LSTM architecture
- train it
- see the results

In [24]:

```
# import necessary libraries
import sys
import numpy as np
import pandas as pd
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from keras.callbacks import ModelCheckpoint
from keras.utils import np_utils
```

In [3]:

```
# loading the dataset
filename="dataset/text-generation/shakespear.txt"
raw_text= open(filename, 'r', encoding='utf-8').read()
raw_text= raw_text.lower()
```

In [9]:

```
# create mapping of all unique chars to integers
chars = sorted(list(set(raw_text)))
char_to_int = dict((c, i) for i, c in enumerate(chars))
int_to_char = dict((i, c) for i, c in enumerate(chars))
```

In [8]:

```
print(chars)
```

```
['\n', ' ', '!', "'", ',', '-', '.', ':', ';', '?', 'a', 'b', 'c', 'd',
'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z']
```

In [11]:

```
n_chars = len(raw_text)
n_vocabs = len(chars)
print("Total characters: ", n_chars)
print("Total vocab: ", n_vocabs)
```

```
Total characters:  99993
Total vocab:  36
```

In [13]:

```
seq_length = 100
dataX = []
dataY = []
for i in range(0, n_chars - seq_length , 1):
    seq_in = raw_text[i:i + seq_length]
    seq_out = raw_text[i + seq_length]
    dataX.append([char_to_int[char] for char in seq_in])
    dataY.append([char_to_int[seq_out]])

n_patterns = len(dataX)
print("Total patterns: ", n_patterns)
```

```
Total patterns:  99893
```

since LSTMs accept values in the form (no_of_sampels, time_steps, no_of_features), therefore reshape dataX to this form

In [14]:

```
# reshape dataX
X = np.reshape(dataX, (n_patterns, seq_length, 1))

# normalize
X = X/float(n_vocabs)

# one hot encoding using np_utils
y = np_utils.to_categorical(dataY)
```

In [16]:

```
X.shape
```

Out[16]:

```
(99893, 100, 1)
```

In [17]:

```
y.shape
```

Out[17]:

```
(99893, 36)
```

In [19]:

```
# define the LSTM model
model = Sequential()
model.add(LSTM(256, input_shape=(X.shape[1], X.shape[2]), return_sequences=True))
model.add(Dropout(rate=0.2))
model.add(LSTM(256))
model.add(Dropout(rate=0.2))
model.add(Dense(y.shape[1], activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam')
```

In [20]:

```
# define the checkpoints and callbacks
filepath="dataset/text-generation/saved_models/weights-imporvement-{epoch: 02d}-{loss:
 .4f}-from-class.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1, save_best_only=True,
mode='min')
callbacks_list = [checkpoint]
```

In [21]:

```
# training the LSTM

model.fit(X, y, epochs=50, batch_size=64, callbacks=callbacks_list) # do try out at dif
f epoch and batch sizes
```

```
WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\tensorf
low\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.
Epoch 1/50
99893/99893 [==============================] - 892s 9ms/step - loss: 2.881
0

Epoch 00001: loss improved from inf to 2.88098, saving model to dataset/te
xt-generation/saved_models/weights-imporvement- 1- 2.8810-from-class.hdf5
Epoch 2/50
99893/99893 [==============================] - 868s 9ms/step - loss: 2.588
9

Epoch 00002: loss improved from 2.88098 to 2.58888, saving model to datase
t/text-generation/saved_models/weights-imporvement- 2- 2.5889-from-class.h
df5
Epoch 3/50
99893/99893 [==============================] - 911s 9ms/step - loss: 2.431
9

Epoch 00003: loss improved from 2.58888 to 2.43191, saving model to datase
t/text-generation/saved_models/weights-imporvement- 3- 2.4319-from-class.h
df5
Epoch 4/50
99893/99893 [==============================] - 882s 9ms/step - loss: 2.325
7

Epoch 00004: loss improved from 2.43191 to 2.32569, saving model to datase
t/text-generation/saved_models/weights-imporvement- 4- 2.3257-from-class.h
df5
Epoch 5/50
28672/99893 [=======>......................] - ETA: 9:23 - loss: 2.2666
```

```
---------------------------------------------------------------------------
-
KeyboardInterrupt                         Traceback (most recent call las
t)
<ipython-input-21-1b591a68507f> in <module>
      1 # training the LSTM
      2
----> 3 model.fit(X, y, epochs=50, batch_size=64, callbacks=callbacks_list
)

C:\ProgramData\Anaconda3\lib\site-packages\keras\engine\training.py in fit
(self, x, y, batch_size, epochs, verbose, callbacks, validation_split, val
idation_data, shuffle, class_weight, sample_weight, initial_epoch, steps_p
er_epoch, validation_steps, **kwargs)
   1037                                         initial_epoch=initial_epoc
h,
   1038                                         steps_per_epoch=steps_per_
epoch,
-> 1039                                         validation_steps=validatio
n_steps)
   1040
   1041     def evaluate(self, x=None, y=None,

C:\ProgramData\Anaconda3\lib\site-packages\keras\engine\training_arrays.py
in fit_loop(model, f, ins, out_labels, batch_size, epochs, verbose, callba
cks, val_f, val_ins, shuffle, callback_metrics, initial_epoch, steps_per_e
poch, validation_steps)
    197                     ins_batch[i] = ins_batch[i].toarray()
    198
--> 199                 outs = f(ins_batch)
    200                 outs = to_list(outs)
    201                 for l, o in zip(out_labels, outs):

C:\ProgramData\Anaconda3\lib\site-packages\keras\backend\tensorflow_backen
d.py in __call__(self, inputs)
   2713                 return self._legacy_call(inputs)
   2714
-> 2715             return self._call(inputs)
   2716         else:
   2717             if py_any(is_tensor(x) for x in inputs):

C:\ProgramData\Anaconda3\lib\site-packages\keras\backend\tensorflow_backen
d.py in _call(self, inputs)
   2673             fetched = self._callable_fn(*array_vals, run_metadata=
self.run_metadata)
   2674         else:
-> 2675             fetched = self._callable_fn(*array_vals)
   2676         return fetched[:len(self.outputs)]
   2677

C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\client\sessio
n.py in __call__(self, *args, **kwargs)
   1437           ret = tf_session.TF_SessionRunCallable(
   1438               self._session._session, self._handle, args, status,
-> 1439               run_metadata_ptr)
   1440         if run_metadata:
   1441           proto_data = tf_session.TF_GetBuffer(run_metadata_ptr)

KeyboardInterrupt:
```

In [22]:

```
# Generating Text from pretrained/post training your LSTM

filename = "dataset/text-generation/saved_models/weights-improvement-38-1.4392-bigger-l
stm.hdf5" # add the name of your best trained saved model's name here
model.load_weights(filename)
model.compile(loss='categorical_crossentropy', optimizer='adam')
```

In [26]:

```
# set up a random seed for starting
start = np.random.randint(0, len(dataX)-1)
pattern = dataX[start]

print("INPUT SEED:")
print("\"", ''.join([int_to_char[val] for val in pattern]), "\"")
print()
# generate characters from the generated output of LSTM
for i in range(1000):
    x = np.reshape(pattern, (1, len(pattern), 1))
    x = x/float(n_vocabs)
    prediction = model.predict(x, verbose=0)
    index = np.argmax(prediction)
    result = int_to_char[index]
    seq_in = [int_to_char[value] for value in pattern]
    sys.stdout.write(result)
    pattern.append(index)
    pattern = pattern[1: len(pattern)]
print("\nTHE END.")
```

```
INPUT SEED:
" nd let me leave thee with him,
and be of such a taodless snaver'd hoes
and carved to us as i have sp "

ope to mea
and the his crmtght the sioge would not do your crmpheret and the seam of
heaven of the comd beao the lingdom of the comd beaote to mea
and then the thoughts to see the commons have been a partinn of the comsam
y donmand in the fallant cotrt:
i will nake the fallant cotnt:

pis toby belly of come oo the lind of your grace in stcriond and the ootie
r of the commons of the crow of all the place of her face with the comd be
aote to mea
and then the thoughts to see the comd beai'

dlrwn:
and the hand as i will sake the fallow of a cood that i do not ii sould be
puising of the fallant cotrt:
i will nake the fallant cotnt:

sis hugh evans:
yhat is the freet stand for the world and the several stane and the severa
l stanes and poene for the hands and ooe exteen the will in the dorrticr,
and so mook iis crowses in the comd beao the lingdom of the comd beaote to
the thoughts and his sond in the crowhres stord, and bll the halds and ooe
exeeute the siow of tomething tome and stand that wo
THE END.
```

In [ ]:

```
```