CS 7180 Special Topics in AI: Deep Learning
Northeastern University, Spring 2019

Assignment 2

Name: Sabbir Ahmad, NUID: 001860911
Email: ahmad.sab@husky.neu.edu

1 Poem Generation

1.1 Pre-processing

1.1.1 Character-based Model

For the character-based model, the pre-processing has been done by extracting the characters from the text file. At first, the text file was considered as the source as it was, and every possible consecutive 40-character sequences were created from the file. For each 40-character sequence the label was the next character in the file.
With this initial pre-processing, the model was trained for a number of iterations but the training was slow because of considering every possible consecutive 40-character sequences. Later semi-redundant sequence was used by picking sequences starting every n-th character for some values of n.
In the generated poem, I noticed unnecessary newlines and sonnet number, which is not needed to generate poem. So finally the sonnet number (integer before the start of each sonnet in the data) and the extraneous newlines were removed. And all the characters were converted to lowercase Using this modified data, the 40-character sequences were generated. Finally $n = 5$ was chosen. Finally, there were 48 different characters present in the data. Each character was mapped to an integer, and it was divided by the size of the unique character set.

1.1.2 Word-based Model

To improve the model, word-based model was considered, where each sequence was a series of words from the sonnets. For the pre-processing, NLTK library was used to tokenize the words. Words, numbers, newlines, punctuations were considered as words. There were 3338 unique words present in the data. After the data was tokenized as words, sequences of length 10 (containing 10 consecutive words) were generated by picking sequences starting every 5-th word. Each unique word was mapped to an integer, and it was divided by the size of the unique word set.

1.2 RNN Model

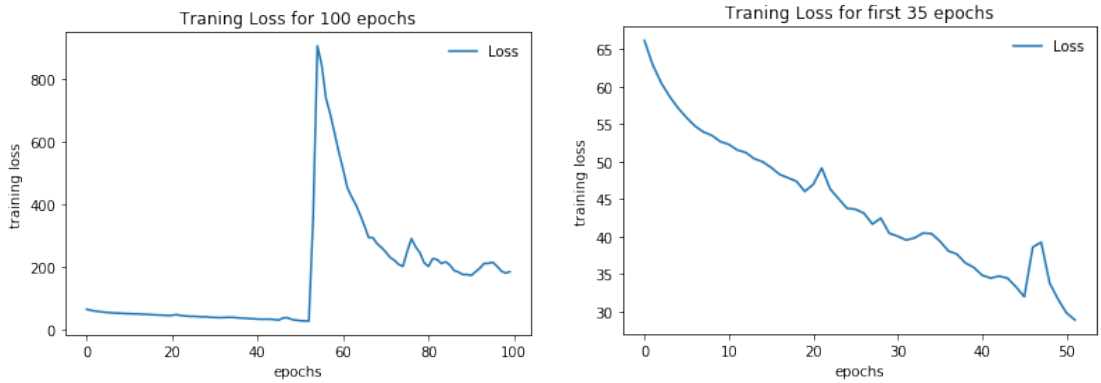
1.2.1 Baseline RNN Model

For the baseline characted-based LSTM model, a single layer of 150 LSTM units was used as the LSTM layer. After that, the lstm layer, a funnly connected layer is used which uses softmax to calculate the probability of the predicted character of the possible 48 character set.

Each input sequence is of length 40, and number of feature for each timestep is 1 (only the character at that timestep in the sequence). So each sequence is a tensor of size 40×1. Batch size of 32 is considered. So the RNN model takes input of tensor size $32 \times 40 \times 1$, and gives output of tensor size 32×48. Output for each sequence is of size 1×48 because there are 48 possible characters considered from the dataset. Cross Entropy loss with Adam optimizer with learning rat 0.01 was used to train the model for 100 epochs.

1.2.2 Traning Loss

The loss can be seen from Figure 1. The loss decreased up to 52 epochs, and after that the loss increased suddenly and never become lower than achieved as can be seen from Figure 1(a). The decreasing nature for the first 52 epochs can be seen from Figure 1(b).



(a) Training loss versus epoch for 100 epochs (b) Training loss versus epoch for first 35 epochs

Figure 1: Baseline Model training loss versus epoch.

1.2.3 Generated Poem

Whenever a lower training loss was achieved, the model was saved. After the training the best model was retrieved, and samples were generated with a seed of 40-characters. An example sonnet generated from the model using `thereby beauty's rose might never die,` is given below.

```
thereby beauty's rose might never die,
bod ths hermpse thfls tiun ooe hatme iy t:y iedt whetheh po ho sia teesu thov toa tose
inv fod moi maldmlef fo aidec psohd tha tith, oytuy lftiy fouia io sei whmn. on thyec
thot oh mat sia helme lo to theh thete,
thi colrr io to thef tooue,s tfeci rp to the golld than.
ther gnus oo ml shlt thetu,s oy
```

1.2.4 Different Temperatures

Using temperatures 1.5, 0.75, 0.25 with seed `shall i compare thee to a summer's day?` poems were generated by generating 400 characters from the seed. But all the poems were exactly same. The temperature was considered as an input to the softmax function. Rather than calculating $softmax(X)$, for a given temperature t the softmax was calculated as $softmax(X/t)$. The poems generated using the three temperatures are as follows. Only one of them is given below as they are same.

```
shall i compare thee to a summer's day?
osrn eeiaso fnd deenh: sw heaeth co thg,srm. hendite fr i vr,py lfee thdt, thanr iiati
why hnlr thlfrest ho f soiic fnananr
tioc, toatogette oo si soass iea tywen shite to toecc fote: saklni toa eoyo'to, cey hr
ohu ayu aai taeevn vhoe teot oy thas shn ieier shy sheferten thon mo coe iease loase
toeltgt,
then iy toi, ialdy fodd v
```

1.3 Improvement

1.3.1 Improved RNN Model

To improve the performance of the model, word-based sequence was considered rather than the character based one. At fist the same model was used with word-based sequences. Here, each input sequence is of length 10, and number of feature for each time timestep is 1 (only the word at that timestep in the sequence). So each sequence is a tensor of size 10×1, and with batch size 32, the input tensor size is $32 \times 10 \times 1$. Output tensor size for each sequence is 1×3338 as there are 3338 unique words.
Later to see the effect of more layers, two lstm layers were used with a dropout value of 0.2 and each lstm layer consists of 100 units. Learning rate 0.001 was used with Adam optimizer and Corss Entropy loss.

1.3.2 Training Loss

The model with two lstm layers was trained for 200 epochs. The training loss can be seen from Figure 2.

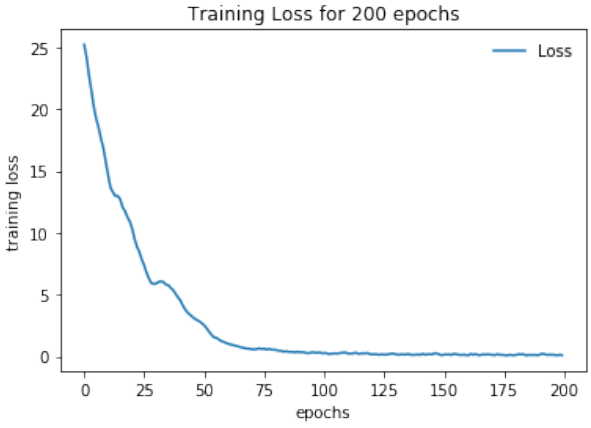


Figure 2: Improved Model training loss versus epoch.

1.3.3 Generated Poem

Using `trenches in thy beauty 's field ,\n thy youth 's` as the seed, the generated poem is as follow.

```
trenches in thy beauty 's field ,
thy youth 's title my invention gazed a
```

```
for heavily heart were self thine body ?


but am my shalt to have the ,

no no the to affections hast thing judgement ,
for how they spites 's my the his , as art
i hath i with hold marriage my have deep

but constant my which fair thee fiend mine ride friend ,
as when all counterpart face , me hand i , new is ,
and whilst that may face , death kept ,

eyes that
```

1.3.4 Different Temperatures

Using the same different temperatures with the same seed `shall i compare thee to a summer's day` 100 words were generated. The generated poems were different for this word-based model shown as follwos.

Temperature: 1.5

```
shall i compare thee to a summer 's day ?
do thou to record than , see all art who in composed nothing in ,
dream all
who is by that is , 2 a
if how do those self have the , , why onset stand in my deformed to feeding

the like of have with , pain knows ,
which shall time stamped becoming such earthly ,

that that the , the makes of to , old when fair ,
where of i this to change thou shade look ,
triumph in your praise of my be
```

Temperature: 0.75

```
shall i compare thee to a summer 's day ?
do thou to record than , thy dear art
and dig the so gave bastard i your be ,
to ten my of by i reason thou )
138 when be was saint soundless jewels -bettering and of , to thy that , and is of , t
more more the love in my , yet young ,
which i thou life it youth of part thine ,


if their doth doth hath hear in .
and constant some the muse
```

Temperature: 0.25

```
shall i compare thee to a summer 's day ?
do thou to to fading i such proceeds ,
in if be my art

and that beauty the had too ) ,
and since beauty is sway my of own ,
stealing stones they my ,


methinks may (sovereign too this they ) and i you true thing it when on ,
and in in days , dream as best all
for me fair ,
ah
all wise dead cruel my my , own dost ,
yet do others love and and and who that breathes of he
```

The improved model showed much better outputs as can be seen from the generated poems. The lines of the poems are correct in many cases, whereas, in the baseline model the generated words were not meaningful except some words. Also the training loss tends to decrease more in the word-based model, but in the baseline model suddenly there was a spike in the training loss.