Course: DD2424 - Assignment 4
Optional for Bonus Points

Álvaro Orgaz Expósito

May 27, 2019

1 Synthesize Donald Trump tweets instead of Harry Potter

For this bonus part, I take the same approach in assignment 4 and apply it to generating tweets of Donald Trump. Here we have the constraint that each sequence can be at most 140 characters long and for training the RNN I will use his tweets from 2009 to 2018 downloaded from the GitHub repository *https://github.com/bpb27/trump_tweet_data_archive*, concretely 36.307 tweets. Once I loaded the tweets, I analysed non-common ASCII characters (e.g. emojis, Chinese characters, etc.) to reduce the dimensionality of the alphabet.

Then, I initially preprocessed each tweet with the following steps to transform them into usable training data:

- Convert all tabs `'\t'` and newlines `'\n'` contained in the tweets to spaces ' '.
- Filter out all strange characters detected before.
- Convert all remaining HTML escape sequences in the tweets to their corresponding Unicode characters (e.g. &amp becomes &, this is fortunately trivially achieved by Python's function *html.unescape*.
- Add an end-of-tweet character (`'\n'`) and a start character `'\t'` which is useful because we can feed it to the synthesis procedure as an initial dummy character (which will hopefully lead to more sensible results than using a random character here).

Then, for training the RNN I initially did some adjustments I to the previous assignment implementation.

- For synthesizing text I set a hard limit of 140 characters and the stop character added in the preprocessing `'\n'`, then if it appears as next synthesized character the synthesized tweet is aborted.
- During training I re-initialize the hidden state *h0* to zero after each training tweet.
- I reduced the sequence length to 15 to guarantee that the network would be exposed to enough *end-of-tweet* sequences.
- For training each epoch, I iterate all tweets shuffling the tweets' order, set the parameter corresponding to the number of updates equal to infinite, and limit the maximum epochs to 1 for iterating each 1 only once each tweet.

In figure 1 we can find the learning curves for training corresponding to 5 epochs (iteration of all tweets), with learning rate 0.1 and sequence length 15. In the curves, you can see the smooth loss decreasing steadily until the end of training which means that we could continue the training to see if loss continue decreasing although after 400.000 updates the loss flattens.

Also, in listing 1 we can find the synthesized tweets (line wrapped to improve readability) that were generated by the partially trained RNN during several epochs. As you can see, from the end of epoch 1 the synthesized text includes words that makes sense like the full name 'DonalTrump' and after epoch 4 we see that the RNN uses the character '@' used by Twitter for tagging users in the synthesized characters '@realDonaldTrump', as well as shortened URLs. However, it seems that the RNN could be optimized a lot since the results do not seem Donald Trump tweets at all.

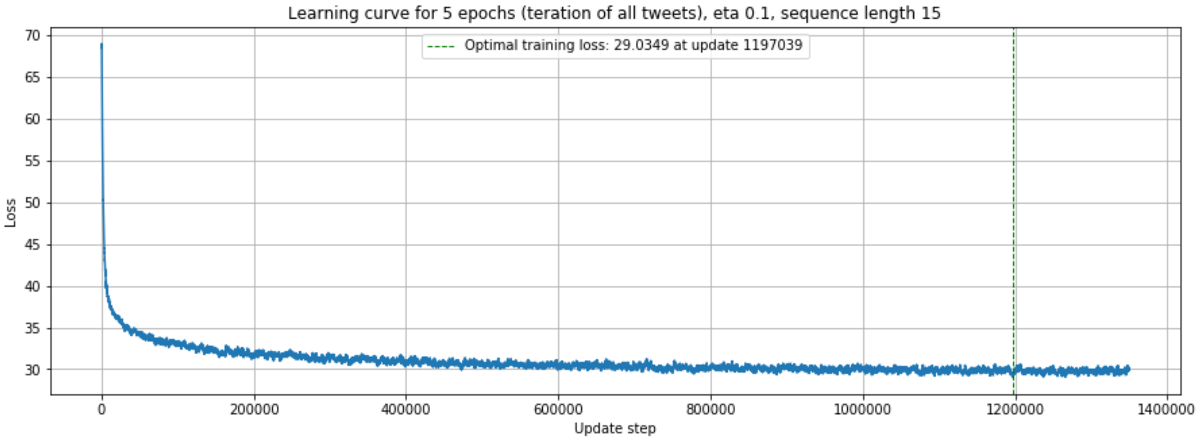

Figure 1: Learning curve for RNN with hidden state size 100, trained during 5 epochs with learning rate 0.1 and sequences of length 15.

```
Epoch 1 tweet 36307/36307 [==================================================]
Synthesized tweets:

The Mame. #Calise that Jountiy & dWars @cotFUMrRLEichace. It demidect Barssgity m
ay, 1 cornol to betticanify  Sawg oul sinly of @MIDAG Toums

@Obllab: he loonel a pleycatico D. Thanks in gederdary!  Thisk nathe. Tox Intting
cauts stiloon treating @reassolls Standy queth fig alte. S

A-ngonuy #DonaldTrump Amragn Ama im http://t.cv/$5pYBAUVML hodesccalaU
```

```
Epoch 2 tweet 36307/36307 [==================================================]
Synthesized tweets:

@Barensenqooketam

Cake wing, Donagars porte. You & af the engrity farw har hards mught an ard showe
n! Hight troeving be win tear an Helling in Amashs dree to

.@jvarmarly long puing faine! He umatrendinally - Lo.n: do keppide is calld tract
ing- goea sumpice. Lead News govald #Blasintlyabladle  Thin


Epoch 3 tweet 36307/36307 [==================================================]
Synthesized tweets:

"@SecripaJlay, USA 754 & Good jent & should rething for digh is with listeredthap
igald. Thats Crices: @realDonalDonaldonesteth Trupploir or

Thank you Tromal bit of have IP that nood http://t.co/V& EWFEDGELLAISN Toogh the
sof and innouirad." Itin on OGERE  WeRo 363 no ectick. Maki

Codresling tot just for nic on #LlenjokilebBpbrs20Y: Ariat Trump Goller on alken
@FiklDoldees call,-A doels dridorer" @lobandambew priciarly


Epoch 4 tweet 36307/36307 [==================================================]
Synthesized tweets:

"@bitarnthe: @realDonaldTrump The they I ver: @realDonaldTrOma. Was in it is stat
that the burines mored breal congrys air wiscar.  Wheo I.1

Stare some ofrenticon the was the Partyher Pose clavere. The lount. for jubl, to
it AHust offy.C. httpU: @vonien seare couvid ublepelefing D

"@jxa,026038 Diles and net’s and shouth the odead whtte 213: Go do show prony sam
e think but and on Cormigull Retticensenter & Crailles." Th


Epoch 5 tweet 36307/36307 [==================================================]
Synthesized tweets:

RT @FBUSS hameal oul came by Doratk look prive is and in ary your a We hand thate
cannitsor of With Ce UTWS1 seaming of #TREjShake_Got?" Cin

"@awovNearX no mork Agraush tell, brotes and Bighaticigur non--Faud Waking will w
umoras! http://t.co/mDk9Zd67Q4nERpqqfTochiacasicaGos drolic

Witha, u to knogelis, country, jung. fusses ledions. T ponem your sole. htAp  1s
it it ho and is and a teny has with windmint on the Apleben
```

Listing 1: Synthesized tweets after each epoch during the training in figure 1.