DD2424
Deep Learning in Data Science
Assignment 4

Franco Ruggeri, fruggeri@kth.se

May 29, 2020

1 Changes in implementation

Besides trivial modifications, here is a list of what I implemented to adapt the code to the new task:

- Special characters:
  - *Start of tweet*: I used the character '^', not present in the dataset, both as first character of each training tweet and for synthesizing (dummy input). The idea was to learn also the first characters of the tweets.
  - *End of tweet*: I used the null character (char(0) in MATLAB) both for the training data and for synthesizing. The idea was to learn the end of the tweets, so that the synthesized tweets could also be shorter than the maximum length.
- Data pre-processing:
  - I removed the retweets. We are not interested in them because they are not written by Trump.
  - I removed the twitter links (*http://t.co/something* or *https://t.co/something*) wrongly present at the end of some tweets. They are not part of the original tweets actually.
  - I replaced '&' with '&'.
  - I removed wrong empty tweets (there were a few).

  - I padded the tweets with end-of-tweet characters in order to make their lengths multiple of the sequence length used for training. In this way, all the characters are used and hopefully the network can learn also the ending part of the tweets, which would be otherwise discarded.

- Training with more than one sequence: I adapted the *AdaGrad* function to handle multiple sequences (tweets). One epoch is defined as an iteration over all the tweets and each tweet is of course split in sub-sequences of length *seq_length* in order to perform more than one update for each tweet. The hidden state is reset when a training tweet is terminated.

- Maximum length of synthesized text: the maximum length of each synthesized sequence is set to 140 characters. If the end-of-tweet character is sampled, the synthesis is stopped. If the end-of-tweet character is sampled at the beginning, the sampling is repeated.

2 Synthesize *Donald Trump tweets* with vanilla RNN

I trained a vanilla RNN using all the tweets available at trump tweet data archive, so from year 2009 to year 2018, with the following hyper-parameters: $m = 100$, $\eta = 0.1$, $seq\_length = 10$, $epochs = 2$. The smooth loss plot is shown in figure 1.

Table 1 shows the evolution of the synthesized tweets during the first 100000 updates. We can notice that a typical pattern of Twitter appears: the tag (*@username*).

Finally, figure 2 reports several tweets synthesized from the final vanilla RNN, using the start-of-tweet character as dummy input (see section 1). The spelling is quite good and the tweets contain some of Trump's favorite words, according to the most frequently used words in Trump tweets, such as *great*, *Obama*, *people*. The tweets, especially the spelling, could most probably be improved by tuning better the hyper-parameters (m, seq_length) and by training for a longer time. However, the tweets are not very coherent and this problem is hard to solve with a vanilla RNN, because it hardly learns long-term dependencies, so it hardly can generate whole coherent sentences.

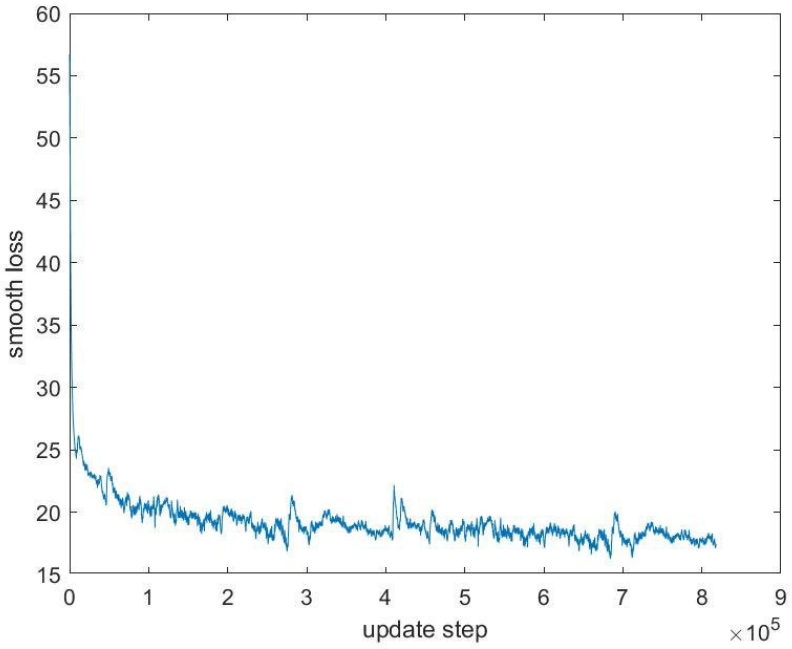

Figure 1: Smooth loss for a 2-epoch training run.

Iteration	Synthesized text
1	8 T 4 & a V a J _ 3 l 0 T 6 y k , 5 ? h % 0 M v T] D R G |* o % J b g
10000	–hibe—ht...
20000	@Mong dounde Trum, rarisit.
30000	I gratpTrompns sfe faod shohprlor iw toum. thing and laigntares carach's bowhe taveenmert forstordebinme's the thee Trey frice ar 20 bletele
40000	Thatrhe pqon wath fhat wo% deecens hpee!
50000	Owerad Dod!
60000	"@Rod jomeoted sers thativerty se naokge greupse?
70000	@cesayen14 Than shoreccest'l, int bow them & S at sfeihe wour. I oufwith thankent
80000	"@Tidlenth forer
90000	@fascIa_onJustis on Thanks! Lama"
100000	@reagaislaben: @realDonaldTrump @Antaundrins dee's!

Table 1: Evolution of the synthesized tweets during the first 100000 updates. The synthesis is done *before* the update, using the first character of the current sequence as first input.

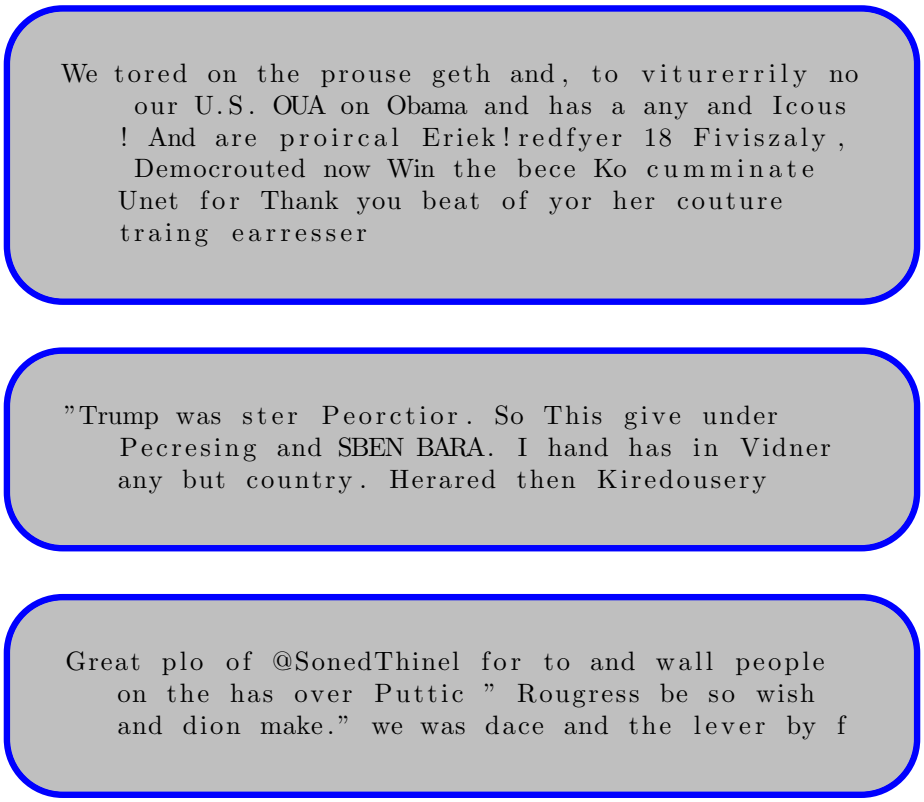

Figure 2: Tweets synthesized from the final vanilla RNN, using the start-of-tweet character as dummy input.