COGS 260: Assignment 4

Instructions

- Due on May 22, 11:59 PM. To be uploaded on TritonEd.
- Reference materials and some useful codes can be found at the end of the document.
- Write a report including: a) abstract, b) method, c) experiment, d) discussion, and e) references. You can follow leading conferences like CVPR (`http://cvpr2016.thecvf.com/submission/main_conference/author_guidelines`), NIPS (`https://papers.nips.cc/`), or ICML (`http://icml.cc/2016/?page_id=151`).
- Copy and paste your code as appendix section to your report.
- You are supposed to provide brief quantitative details and analysis of experiments wheneverasked to report the details of the experiment. Please DO NOT write long paragraphs. You are encouraged to use plots and images to explain your results.
- All parts of a question has to be at one place.
- Please start this assignment early so that you don't run into computational problems later on.

NOTE: In this assignment you are going to play with Recurrent Neural Networks. In the first problem you are going to learn how character-RNNs can be used to reproduce any text documents by predicting one character at a time. You will then apply the learnings from this problem in the next task where you will use RNNs to generate music.

1 Char RNN

Train a vanilla RNN on *tinyshakespeare* data (can be found under Data.zip on piazza). The network should predict the next character when given a sequence of characters as input.

For example: Say if the input sequence at time t is {'T','h','i','s',' ','i','s',' ','a',' ','b','o'} then the most likely output chacter at time t could be 'y'.

In a similar way, start with some valid sequence of characters (may be from the dataset provided in piazza) at t_0 and predict the next character. Use this predicted character along with the input sequence at t_0 as input sequence at t_1 to predict the next character. Repeat this step to produce at least 500-800 characters.

You may refer to [1] for more details about this problem. You may have to scroll down in the website to locate the part where sample output for Shakespeare is provided. A sample code to do this in python can be found in [2]. However, writing your own implementation won't be very difficult either.

Report the input and output representation of your network, network architecture and hyperparameters, i.e. all the information what one would need to reproduce the same network. Put at least 3

outputs generated after starting with different initial input sequence. Also, plot the training loss wrt. number of iterations (or epochs).

You may use a fraction of the dataset to overcome the computational challenges, but be sure to mention it in your report. You may optionally try 2-layered RNNs or may be LSTMs if you have the necessary computational power.

2 Music Composition

In this problem, I am going to guide you to do a small feasibility study of an interesting project which we did in the last quarter. To know more about the project you may refer [3] (`http://www.apurvapathak.me/files/CSE253_Project.pdf`). You can also extend this work in your final project.

You have seen how you can produce entire Shakespeare using char-RNNs. Can this be used to compose music? Well, why not if we can represent music as text? Music can be represented in ABC notation [4] which is a text format as shown below.

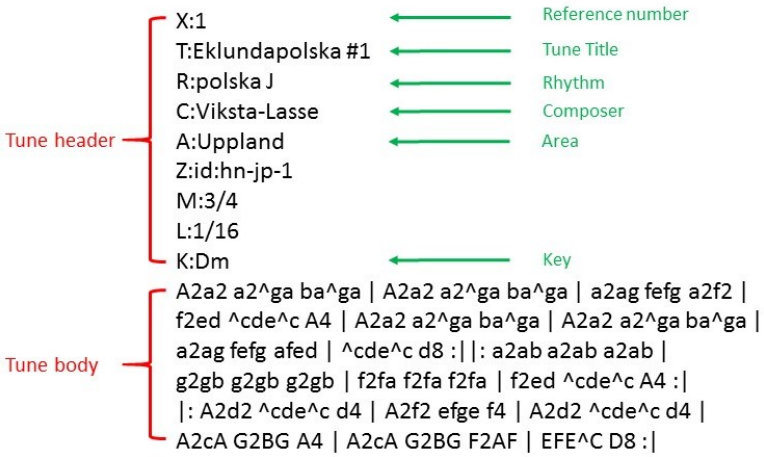

Figure 1: Abc notation of a music file

2.(a) Play the music

In this part, we are going to see how we can convert music from ABC notation to a playable format (.midi in this case) online. Go to the website `http://mandolintab.net/abcconverter.php`. Copy the text from *sample-music.txt* file (found under Data.zip) and hit *Submit*. Download the tune in *midi* format and play it on your computer.

2.(b) Dataset

I have prepared a small dataset specifically for this assignment by scraping tunes in ABC notation from *The Nottingham Collection* [5]. The tunes in ABC notation are then appended together to create a big text file, *music-dataset.txt*. Each tune is separated by a <start> and <end> token, as you can observe in the dataset.

2.(c) Compose music

Train a char-rnn (just like Problem 1) to generate music, one character at a time. Report the input-output representation, network architecture and hyperparameters. Also, report the plot of loss wrt. iterations. Report atleast 3 output music generated by your network (in both ABC and Midi

format) for different initial sequence. You can put a google drive link into your report for the music in midi format.

NOTE 1: The initial character sequence for reporting the output should end with the <start> token. (Can you guess why?)

NOTE 2: The <start> and the <end> tokens should be considered as single character rather than a sequence of characters (words). (Can you guess why?)

NOTE 3: You only need to turn-in part (c) in your report.

References

[1] Karpathy, Andrej. The Unreasonable Effectiveness of Recurrent Neural Networks. http://karpathy.github.io/2015/05/21/rnn-effectiveness/

[2] Karpathy, Andrej. Char-RNN code. https://gist.github.com/karpathy/d4dee566867f8291f086

[3] Pathak, Apurva, et. al. Context-Based Music Composition using RNN Generative Model. http://www.apurvapathak.me/files/CSE253_Project.pdf

[4] The abc music standard 2.1, http://abcnotation.com/wiki/abc:standard:v2.1.

[5] The Nottingham Collection. http://abc.sourceforge.net/NMD/.