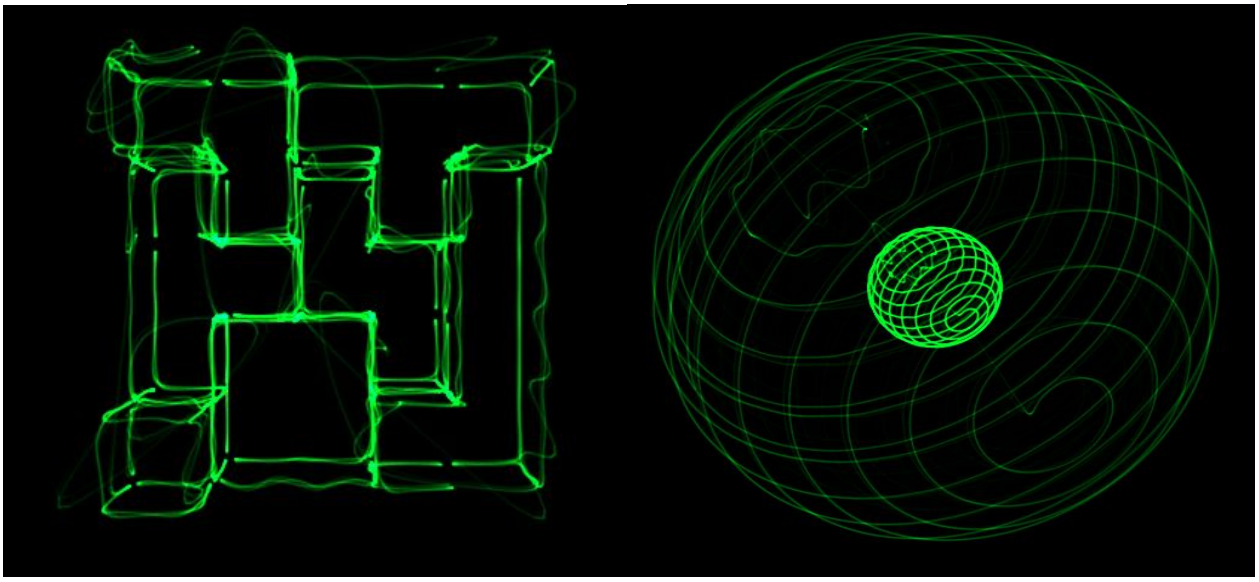
Machine Learning for the Arts
UCSD SPRING 2019
FINAL PROJECT

OSCILLOSCOPE PODCAST



Ryan Bez, A14773143

DESCRIPTION

Concept: I decided to revisit project 1, generative texts, for this final. I had also wanted to explore generative speech more, but due to timing and technical issues outlined below that ambition had to be scrapped. The podcast that I chose to base this on, Reply All, is all about the internet and weird technology surrounding it or influenced by it. Because of this I was inspired to use the openness of them hosting their podcast's text transcripts on the internet and using that to fuel my machine learning text generation. In the future I think it would be interesting to work on it further and present it to the podcast as I think it may be something they are interested in discussing. Since podcasts are primarily audio based I also wanted to have fun with the audio by interlacing special visuals that would appear when played on an oscilloscope. I chose to do this because using sound to produce an image on an oscilloscope is fascinating to me and it is always fun to show other people that this is possible.

Technique/Process:

- To begin I reran my web scraper, Reply All Web Scraper.ipynb, to gather all the new episodes that had come out since the first project to have additional material to train on. However, not many new episodes have come out since then and it added a little over 50,000 more characters into my training set for a total of 8.9 million characters. I also added in some additional code to learn more about how the transcripts are structured and found that the average length for a line in the transcript is 107 characters.
- This training set was then used in Final Training and Text Generation.ipynb, which is a modified version of the RNN sample we looked at in class. It is modified to correctly read in each of the transcript files generated in the last step, and the generate_text function has been modified to accept additional variables allowing the user to specify the number of characters to generate and the temperature of the generation each time the function is called.

- Training takes about 2 minutes per epoch for this large training set and in my previous work on project 1 I found diminishing returns when training on too many epochs. So in the interest of time and efficiency I trained for 50 epochs.
- From there I generated a handful of .txt files using various amounts of characters to generate, temperatures, and start strings. These are included in the gen_text.tar.gz. Additionally, inspired by one of the other student's techniques from their project 1, some of the output files had their strings generated by using the previously generated string as input.
- Poring through these files I chose strings and phrases that were of interest to me, whimsical, and appeared to have a "back and forth" between the different hosts. I compiled them in a separate file and this raw script can be found in "Unfiltered_script.pdf."
- This was then artistically edited to correct minor spelling/naming issues as well as formatting it more like an official transcript from the podcast with host's full names used the first time they speak, then reduced to just their first name in later responses. Then I designed a stylized script, "Script.pdf," to mimic the script being displayed on an oscilloscope and hosted it at tiny.cc/ece188 for the class to follow along with during my presentation.
- Now it was time to give voice to the script. Originally I had intended to use tacotron2 and/or DeepVoice3 to give me multiple voices to apply to the various hosts to say their lines. Unfortunately I could not get either of these to work on datahub, jupyterhub, or colab due to various issues with installs and imports.
- However, I still wanted the audio aspect to my project to be able to tie it all together and have something to play on an oscilloscope. Reminiscing on my younger years where it was fun just to have Microsoft Sam say random things on an old copy of Windows I turned to online text-to-speech generators. [1][2]

- Fortunately, text-to-speech technology has massively improved since my younger years and there are a plethora of voices at your disposal. This was relieving because though I was unable to get tacotron2/DeepVoice3, having command of all these voices allowed me to have creative fun and use voices that were reminiscent of the host's actual voices and allowed each host to sound unique from each other.
- These voices were recorded into Ableton Live 10 and arranged to follow the script. Additionally, for the [MUSIC] breaks in the script excerpts from Jerobeam Fenderson's "Blocks" and "Asteroids" from his Oscilloscope Music album were edited into the arrangement. [3] When played on an oscilloscope these provide a visually stunning element where normally podcasts are restricted to an auditory domain.
- This final mix was exported from Ableton as "Final.wav" which is meant to be played on an analog oscilloscope. Without access to an analog oscilloscope it may alternatively be played on "Oscilloscope.exe" which is a program mean to replicate an analog oscilloscope and is available for open source use. [4]

Result: Final.wav and Script.pdf which were exhibited to the class. Final.wav may also be played on the included "Oscilloscope.exe."

Reflection: The resulting script was chosen for clarity, whimsy, and the "back and forth" that I was looking for from the myriad of generated texts. I think my techniques of using the RNN and Ableton were appropriate for the project, but I am not wholly satisfied with the final result due to technical issues with tacotron2/DeepVoice3 and OsciStudio that I had previously used. Given these issues I am pleased with my application of the available online text-to-speech programs which allowed me to remain faithful to the host's original voices and give each one a unique character and voice to speak their lines. In the future, given more time, I would ideally become more skilled in using OsciStudio and use it to produce visually stunning music breaks, and possibly even character portraits, of my own.

REFERENCE:

[1] Natural Readers, an online text-to-speech program
https://www.naturalreaders.com/online/

[2] TTS Reader, an online text-to-speech program
https://ttsreader.com/

[3] Jerobeam Fenderson, Oscilloscope Music, 2016
https://oscilloscopemusic.com/watch.php

[4] Hans Raber, Oscilloscope, 2017
https://oscilloscopemusic.com/osci.php

CODE: https://github.com/ucsd-ml-arts/ml-art-final-ryan_bez