RNN Competition

Nikhil Sardana

January 2018

1 Introduction

Welp, it looks like none of the officers are here. As a results, we've prepared this activity for you guys. There's a prize for the winner. Hopefully you find this fun.

Your task is to train a Recurrent Neural Network (or LSTM, or GRU) on a text, and use the trained model to generate a sample of text. Then, you will compete with your generated text in a tournament in which others must try to discern your generated text from a real sample from your selected work. You may work with a partner.

2 Choosing Your Text

Your text should be a relatively long (50,000+ words is generally recommended). If your text is literary, it must be in English, although there is no time period limitation. If your selected text is not literary (e.g. LaTeX, Python, Java, etc.), there are no restrictions.

3 Training the RNN

You can use any code from online to train your model. We've found these repositories, which are quite simple to setup and train. Simply clone the repository of your choice and follow the instructions.

If you would like to use Tensorflow: https://github.com/sherjilozair/char-rnn-tensorflow.

If you would like to use PyTorch (which is generally faster): https://github.com/spro/char-rnn.pytorch.

Note that if you are on Windows, then you must use Tensorflow. PyTorch is only compatible with Linux and MacOS.

To install Tensorflow on your laptop, follow these instructions: https://www.tensorflow.org/install/install_windows. In general, "pip3 install tensorflow" should work.

To install PyTorch on your laptop (or on one of the Linux machines in the room), follow the instructions here: http://pytorch.org/.

Installation should not take more than a couple minutes. Training an RNN using the Tensorflow repository on a 60,000 word text took my 4-core 3.6 GHz machine 16 minutes. As a result, if you do not have a quad-core laptop, it is highly recommended to use Pytorch. If you have a dual-core Windows machine, use one of the Linux machines in the classroom and install Pytorch, or SSH onto one of the high compute clusters, hpc1, hpc2, etc.

4 Generating The Sample

After training your RNN, generate 3-4 samples of text, each no less than 400 characters long. This should be a very quick process. You can provide any primer, as long as it is less than 50 characters long (including spaces). A primer is a piece of text you give to an RNN which the RNN uses to generate the following characters.

5 The Tournament

We wanted to make this a fun competition, not a boring exercise. So, this is what we came up with. One student is the administrator (this is decided by us beforehand), and will not participate.

1. The administrator assigns your team a number.
2. You select a text, train your RNN, and generate your sample as outlined in sections 2-4 above.
3. Choose two paragraphs of text your from source material (the text you trained your network on) of roughly similar length to your generated text.
4. Open up Notepad or a text editor. Paste in one of your generated samples, and the two real samples. Number these samples 1-3.
5. Write down your team number and which number is the generated text, and give them to the administrator.
6. Everyone will go around to the other teams, and attempt to guess which passage is the fake one. Write down your guess for each team, and hand these to the administrator.
7. The administrator will tally each team's votes. The teams with the least number of correct guesses (in other words, few people could tell the generated text apart from the real text), will move on.
8. Repeat steps 3-7, but with a different generated text sample and two different real text selections. When only one team remains, they are the victor. Students eliminated in previous rounds can still vote on others' texts.

6 FAQ

Some may ask: Can I modify my generated text to make it more realistic? The answer is: You can only remove things like line break characters "/r/n", for example, and replace them with line breaks. If your text uses HTML character codes (e.g. &amp; for &), you can also replace these. You get the idea. You cannot, however, modify your generated text if it just hasn't learned the correct format. For example, if your generated LaTeX doesn't compile, too bad! If your generated Java doesn't open or close its brackets, tough luck! Choose your text wisely! Good luck!

Others may ask: How many teams move on per round? The answer is the nearest power of 2 less than the current number of competitors. For example, if there are 12 teams competing, 8 move on to the next round, and 4 move on after that, then 2, then one winner is declared. If there are 16 teams competing, 8 move on, and 4 move on after that, etc. If there is a tie when there are supposed to be n competitors moving on, for example, if $n = 4$ and the 4th and 5th place teams have the same number of votes, they both move on. However, only $n/2$, or in this example, 2 teams, will move on in the following round.