Generating Poetry using Recurrent Neural Networks

medium.com/@uneeb/generating-poetry-using-recurrent-neural-networks-6a4fe7e3b3ac

August 22, 2023

U

Through recent research developments, artificial intelligence is becoming more able and important, especially in areas generally considered to be solely for humans, such as artwork, captioning images, text generation, etc. These developments, such as GPT-4 and DALL-E, make use of *neural networks*, a kind of computer programming that can learn through trial and error.

Before the development of these large language models, artwork, music, and text generation relied mostly on *recurrent neural networks*, which essentially were neural networks that processed data in the form of sequences. This allowed text, music, etc. to be generated as they are forms of data that occur sequentially (for example, a particular word must be based on whatever comes before it to make sense). One of the earliest developments in text generation was Char-RNN (created by Andrej Karpathy, who is still an avid deep learning researcher; check out his blog here). Karpathy used Char-RNN to generate essays in Paul Graham's style, Shakespeare, Wikipedia articles, and even Linux source code. Others have also used his code in more creative ways: generating music, raps, political speeches, as well as recipes.

Poetry Generation

Seeing the potential of Char-RNN, I myself attempted to create a poetry generator (this was back in 2021, when Char-RNN wasn't too outdated). Why poetry? Since it's a field of writing which is generally more creative, I felt that Char-RNN, from what I had seen, would be well-disposed to it.

On a more technical note, Torch-RNN, a kind of "upgrade" on Char-RNN was used. It doesn't impact the actual functioning of the neural network, mainly just improving performance and memory.

After setting up a Ubuntu subsystem (slightly outdated) and installing a few programs and libraries, training wasn't particularly difficult, although the process is extremely lengthy. For example, after adding a small input text file (~ 11 MB) to see how the training worked, it took a night for the model to train completely.

Nonetheless, I gathered various poetry texts from Project Gutenberg (a collection of public domain works) and added them all in to one text file which amounted to around 120 MB. Using this file, I trained the model and generated a few poems:

> The forest spells in this thou the winds
>
> The world against thou works, and while thou art
>
> Except the king of men even in the bands,
>
> And with the wind of old horrors, reason,
>
> The crimson — and to draw the destroy,
>
> And good a blood mine on the land of the shores and pity

Not particularly coherent, but quite creative for a computer to generate. Here is one where the network decided to capitalize…a title (maybe?):

> THE AID BANK BOOK OF THE TOTTER THEN MOOD:
>
> The hour of blood in thee will speak,
>
> And shall I that confessing the triumph,
>
> Art what how some inspirent motion of the forest.

The model also decided to invent a new adjective, as I don't believe "inspirent" is in any English dictionary.

> ‘But the numbers of the worn and all his hand,
>
> In the worm the blood of means of dreary ground,
>
> Which in the sun of the solong glimmer and rage
>
> ’Twas his flame.

Conclusion

All of these examples have two things in common: they aren’t very coherent, but they make up for that (somewhat) in creativity. Character-based RNNs (neural networks where the sequencing is done from previous to next character) tend to be more creative in comparison with more modern word-based neural networks, and this is because, as previously seen, they can use their own formatting, like adding random spaces, or create new words, etc.

Modern-day text generation revolves around a newer type of neural network known as Transformers, which are to some extent based on RNNs. And, as alluded to previously, word-based models are becoming more common than character-based ones, due to them inherently being more coherent (it won’t start generating new words, for example). These new developments have resulted in neural networks becoming much more capable, with advances such as OpenAI’s GPT series and others.

Despite RNNs falling out of use in the modern age, without them, it hardly seems possible that we would have ever been able to come this far without such a pivotal development. Because of this, Char-RNN will be remembered as an important point in the evolution of text generation models.

Further Resources

https://www.engadget.com/2015-12-02-neural-network-journalism-philip-k-dick.html (How I first learned about Char-RNN)

https://github.com/karpathy/char-rnn (The Char-RNN codebase on GitHub)

https://karpathy.github.io/2015/05/21/rnn-effectiveness/ (Andrej Karpathy’s original blog post)

https://github.com/jcjohnson/torch-rnn (The Torch-RNN codebase on GitHub)