Deep Learning Project: Text Generation

Omar Youssif

Area of Study

For this project, I'm trying to analyze how effective it is to use a model (Generative Adversarial Networks) that is primarily used for image generation for text generation. The focus of the project is to try to see whether this notion that GANs can be used for generating realistic human-sounding text is true.

I approached this project primarily because of an interest I have in Natural Language Processing (NLP). In a previous software engineering internship, I worked with the Azure environment trying to generate text based on speech converted to text. I'm interested in seeing evolving segments of NLP/ML research. I've done some research about some new fields to explore and it seems like GANs are generally new but also extremely new for text generation.

Background Reading

GANs have had an extraordinary performance for images and also have been a central topic when it comes to images. However, some researchers and students tried experimenting with them for text generation purposes and claimed to have promising performance. Consequently, I wanted to see how true that is and see this, particularly from the scale of a class project. Seemingly, the difficulty with GANs was the training and how unstable it is. It is also difficult to get to a working state and not as intuitive as an alternative like Recurrent Neural Networks (RNNs) which would be more suitable for this task.

For this to work, a little more explanation is needed for how GANs work. They are primarily a competition between two neural networks, the generator, and the discriminator. It is a type of unsupervised learning and relies on the generator for learning. The generator takes in random noise as input to produce similar text to the training data. The discriminator tries to classify real and fake data well. The generator will constantly try to fool the discriminator and the discriminator will constantly be better at classifying.

From my background reading, it seems like there is an acknowledgment of bad performance yet somehow still incredibly significant accomplishments. Better alternatives that constantly popped up included transformers, particularly a pre-trained one (such as GPT models). The motivation is to try to see if this is an interesting point of research for future work. Particularly because in the context of image generation, there was a boom that caused a lot of significant work to be done.

Implementation

I chose my dataset to train my data to be the Amazon Q&A Dataset. I have data files for every Amazon product that include all the questions and answers and are also separated by product category. The reasoning behind this is that I believe that it is scalable where I can include additional or fewer reviews depending on how big I want the dataset to be. Naturally, this dataset will be extremely large since Amazon is one of the largest online retailers or marketplaces in the world. The data sits at around 1,400,000 pairs of questions and answers as it filters out the questions with no answers. I also believe that because it is inherently human-written, it's the most human-like data possible that is not very difficult to wrangle. A metric I used to gauge how expansive the dataset will be after I scale it down is the number of unique words (or vocabulary size).

I also chose to implement a generator, discriminator, and GAN and text dataset as their own classes and call the training function on them. I have a short print of 100 words generated from the model using beam search. I chose beam search as my primary algorithm for generating text based on the GAN's model output. It has a higher chance of getting global optimums while still being very efficient due to being able to limit beam sizes. I have also left some hyperparameters in after realizing that they should be scalable depending on the input size, which is sometimes different. I automated this process, so we don't need to worry about dimensions with different data sizes.

I also chose Binary Cross Entropy as my loss function. Typically in text generation, there is no binary selection as tokens can fall into a magnitude of classes, but with this context, we have a more nuanced choice. We are dealing with neural networks trying to classify text into real and fake, so it is appropriate to use in this situation.

For my optimization algorithm, I chose Adam. It seems like an appropriate choice because of its momentum and adaptive learning rate while being very efficient and fast. This allows for stability and also being able to converge quite quickly. This is crucial given the nature of our training and how large the dataset is.

Evaluation

For this project, I had a lot of experimenting with different structures and optimization algorithms, loss functions, and text generation strategies. My implementation seems to be incredibly efficient as it can process a large number of unique vocabulary words as input. From some research and conversations, it seems huge models such as ChatGPT were trained on around 100,000 unique vocabulary words. I've been able to do 10,000-15,000 vocabulary words though it took around an hour or two to complete.

My model converges at a fast pace and requires around 40-50 epochs to do so, though it is difficult to see a noticeable difference between 3 epochs and 30 epochs in terms of text quality (assessed by me) but there is a difference in loss. One interesting aspect regarding the hyperparameters is that it was incredibly obvious just from the first or second epoch whether some were obviously not optimal. The learning rate causes extreme instability and I've found that 1e-5 to be the perfect one (which thankfully is what I started with, so it didn't take a lot of time to find out). I can selectively choose word training size and also output size. The generation of text takes a few seconds so it is easy to customize.

The generated text seems to represent the sampled data, but it seems difficult to make some coherent statements from it. I have some doubts about this being a text generation issue, but rather it is related to the training itself. I believe the model structure heavily struggles with sentence coherence despite it being able to generate an output. I was able to find a sample of text generated also by a GAN at a Stanford Deep Learning class, but it seems like it was a project at a much larger scale and also had a lot of tweaks to specifically cater to this environment.

The Stanford GAN text:

“To melinda falls aboard bartok when escapes the two gun, they warns it to be police.”

My GAN's text:

“to who we here idea capable wife mobillink wierd cover still protein still herself go to 36091 argonia super pfc spare teacher musa over port 6th listed artist be musa info water item pfc item still psi steep get tyrosine versus item the narrow spinosa glycerin rev sloped additional materials cover increase psi item wednesday year d300 still smearing panthenol exactly find panthenol pfc d300 traditional audio we weapon fit additional ready 1 retina pair super item service glycerin past professional into glycerin again which smearing case thie label holes go taught both pleased pretty citrus 69 ship to port”

From this text, I can tell that I had an incorrect assumption about the dataset. It seems like there is a particular laziness with grammar and sentence structure/coherence with the questions and answers. Even though that is very “human”, it makes it very difficult to have proper structure which I do think affects the outcomes. It also doesn't help when a word is used a lot (which makes sense in the context of a review), such as “pfc” in this excerpt. When I use my sampler that takes in question-and-answer pairs from across multiple categories, the issue gets a little better, but it is still noticeable.

Given that I didn't have the resources to get a quantitative measure of human evaluations due to the scope of the project, I relied on myself in terms of evaluation. I explored alternative texts that shy away from the issues with training using Amazon data and it seems noticeably more coherent. I got texts from the Gutenberg project and got Pride and Prejudice as the input dataset, and got the following text.

“resign dispense bearing thin thin admirer bearing embracing diffused recurring dispense recurring historically confidante bearing recurring recurring bully 84 convey interpreted warranties dispense _his_ admirer recurring unforgiving unforgiving 84 drab commendations originated dispense dispense bearing bearing arch recurring 84 heroines interpreted bearing captivation bearing piqued 84 replacement bearing dispense existence recurring piqued recurring fluctuating recurring piqued till recurring replacement honour embracing expected circumspect bully dispense dispense unheard honour convey inter surveying unheard embracing diffused surveying bearing bearing voice_ embracing gulf copies admirer dispense bully manage bully embracing recurring invention piqued manage dispense unforgiving manage bearing despised bearing seventies thin 84”

I believe that this is an interesting area of research particularly due to the background research I’ve seen. Some novel techniques could allow text to be more coherent and it’d be interesting to see what are the possibilities with that. However, from my implementation, it seems like the “great performance” was heavily exaggerated. It is still quite impressive that it can generate text with a huge resemblance to the original data, but it is a much less attractive choice than using transformers. It was interesting to see the text first-hand work on this and try different options. I’m excited to learn more about the applications of GANs in the future.

References

- Unsupervised Text Generation Using Generative Adversarial Networks
    - http://cs230.stanford.edu/projects_winter_2021/reports/70709277.pdf