Emily Sheetz
COMP-401, Capstone
Project Proposal

Generative Adversarial Networks with Memory for Text Generation

I. Abstract

Neural networks are powerful tools that researchers have used to push the capabilities of artificial intelligence. As the field of artificial intelligence progresses, neural networks are able to learn to perform tasks that are considered to be inherently human. In particular, generative adversarial networks (GANs) provide a framework for training neural networks to generate output, such as images, music, or text. While generating images is a computationally complex task, generating music or text is more complicated because meaningful outputs require long-term structure. Long-term structure is achieved by a type of recurrent neural network (RNN), including gated recurrent units (GRUs) and long short-term memory networks (LSTMs). Previous research exploring text generation using GANs has focused on comparing the effectiveness of regular RNNs and GRUs. When LSTMs are used to learn long-term structure, the application is in phrases or short poems. The purpose of this project is to combine approaches from previous works to train a GAN for text generation. The GAN framework will utilize LSTMs for memory and will be trained to generate variable length sentences of horror text at the character level. The quality of the generated text samples will be qualitatively evaluated through a survey in which participants will rank the quality of text samples from the GAN and from human authors. The aim of the project is to explore the possibilities of the GAN framework in generating text, which is personal, expressive, and requires intelligence in storytelling, making it a task that is closely associated with humanity.

This project proposal serves to outline socio-technical issues surrounding the project, describe the project specifications, discuss foundational material, and document the project plan that will be followed during the spring semester. The social issues related to GANs generating text force us to consider the significance of storytelling in our lives, specifically when eliciting emotion. The project description specifies the details of the project, focusing on the end product. Foundational material includes a description of the GAN framework, different recurrent networks used for long-term memory, and algorithms that aid the training process. The project plan includes checkpoints to be met throughout the semester, alternative plans in the event that the project need to be scaled back, opportunities for project extensions, and tools to be used during implementation. The project proposal is the culmination of all preparations that took place during this semester and is a starting point for implementation next semester.

II. Introduction

The opportunity to train neural networks to generate something new is an interesting and complex problem in artificial intelligence. The generative adversarial network—a framework in which two neural networks train simultaneously to generate new samples and improve the quality of generated samples—is a relatively new concept. A research team led by Ian Goodfellow wrote a paper on their groundbreaking work generating images with the new generative adversarial network (GAN) framework in 2014 (Goodfellow). Generating music or text using the GAN framework provides a different challenge, as the two adversarial networks must each have some form of memory in order to produce samples with long-term structure. Memory or long-term structure is achieved through some form of recurrent neural network, which feeds outputs from previous time steps through the network as inputs for the current state. The field of artificial intelligence is waiting for more research in training models to generate samples.

The goal of this project is to generate sentences of horror text at the character level using the GAN framework with long short-term memory networks—a type of recurrent neural network—as memory. The training process will be aided by curriculum learning, variable length outputs, and teacher helping. The network will be trained by horror texts available from Project Gutenberg, a free online collection of books that are public domain. Some texts that will be used for training the network include the works of Edgar Allen Poe, Robert Louis Stevenson, and Bram Stoker. The focus of this project on texts specifically from the horror genre is due to the availability of texts from Project Gutenberg, the researcher's personal interest in the genre, and the style of horror fiction. The quality of the generated samples will be evaluated qualitatively through a survey. Voluntary participants will be asked to rate the quality of randomly selected samples from both human authors and the project's model. The perceived quality of the generated samples from the model will be compared to the quality of the samples from human authors to gauge how well the built model learned to generate horror text.

The specifications for this project are motivated by related research, specifically with respect to how memory is achieved within the GAN framework. While recurrent neural networks are network architectures that allow for memory, researchers have found that training regular RNNs within the GAN framework is unreliable. Other types of RNNs such as gated recurrent units (GRUs) and long short-term memory networks (LSTMs) prove much more effective. LSTMs can selectively remember new information and forget information about previous states, allowing the network to learn more complex and long-term structure than regular RNNs (Donahue

2626). GRUs have similar features, but have fewer parameters, making them more computationally efficient than LSTMs (Rajeswar).

There are several research projects that demonstrate the capabilities of these recurrent network architectures. Elliot Waite wrote of the research being done through Project Magenta in generating music. The research project explored different recurrent neural network architectures within the GAN framework to generate long-term structure within one or two measures of music (Waite). The research team led by Ofir Press generated phrases of text using GRUs within the GAN framework. Training of the network was aided with extensions for curriculum learning, variable length outputs, and teacher helping. The results of this research show that recurrent network architectures can be used with GANs to generate samples of text (Press et al.). Another project conducted by Xuerong Xiao showed that GRUs within the GAN framework trained on larger vocabularies performed better than regular RNNs trained on smaller vocabularies (Xiao). Sai Rajeswar's research team generated Chinese poetry using LSTMs within the GAN framework and implemented curriculum learning to aid the training process (Rajeswar).

This project will combine features of several research projects to explore a novel approach to text generation using GANs. The use of LSTMs for memory will avoid problems associated with regular RNNs within the GAN framework. Training the LSTMs will be aided by multiple training algorithms, which will allow for curriculum learning along with variable length outputs and teacher helping. The full sentences of text to be generated by the network are also longer than the samples generated in related research. By combining features of key previous works, this project will serve as an opportunity to extend research related to generative adversarial networks and text generation.

While this computer science capstone project is being implemented, a related project in mathematics will also be carried out. The mathematics project will explore methods to quantitatively evaluate the quality of generated text samples. An understanding of probability and natural language processing will allow the researcher to understand how text generation models—specifically n-gram models and Markov models—work and how to build such models. The focus of the project, however, will be understanding how methods for evaluating text generated by these models work. There is not a defined way to evaluate language models. Evaluation depends on the application of the model and the desired criteria of the text to be measured (Kawthekar et al. 1). The goal of the mathematics project is to understand how multiple evaluation methods can measure different characteristics of a text sample. More importantly, these evaluation methods will be used to make a comparison between text generated by probabilistic models and text from human authors.

The task of generating text using generative adversarial networks is not only of technical interest within the field of artificial intelligence, but it is also significant in a philosophical sense. Human beings use stories to communicate with one another. As artificial intelligence becomes more powerful, it is more common for people to interact with artificially intelligent agents. However, these agents are typically considered to be an "alien sort of intelligence" (Riedl). If artificial intelligence can learn how to tell stories, it will become a more familiar form of intelligence and therefore become more useful to us. Furthermore, works of fiction have "sociocultural knowledge encoded" into them from "different cultures and societies" (Riedl). Teaching artificial intelligence through fiction not only allows it to learn narrative intelligence, but also about the values and norms of a culture. Artificial intelligence learning to tell stories will provide the powerful capacity for more human-like intelligence.

Horror fiction in particular is personal and human. The goal of horror fiction is to "elicit an emotional reaction that includes some aspect of fear or dread." As author Douglas Winter said, "Horror is an emotion" (Horror Writers Association). Horror texts are compelling because they create powerful emotions in their readers. What one reader finds scary is dependent on who they are. A horror story "forces us to confront who we are, to examine what we are afraid of, and to wonder what lies ahead down the road of life." Horror "reminds us of how little we actually know and understand" (Horror Writers Association). Horror fiction is closely related to our humanity and our emotions. This makes horror a particularly interesting genre through which to explore the storytelling capabilities of artificial intelligence. This project will provide an opportunity to explore what generative adversarial networks can do, evaluate how well neural networks can produce quality horror text that elicits powerful emotions, and seek human-like intelligence in an intelligent agent through storytelling.

III. Project Description

The purpose of this project is to extend previous works by combining features of models used for text generation. Specifically, LSTMs will be used within the GAN framework to generate a sentence of horror text on the character level. Several algorithms will be implemented to aid the training process. Curriculum learning allows the network to move from generating short sequences of characters to increasingly longer sequences. Variable length outputs will set a maximum sequence length and allow the network to generate sequences of any length below the maximum, thereby training the network to be more robust. Teacher helping conditions the network on sequences of characters that appear in the training dataset (Press et al.). The goal is that the training algorithms will help the network to generate full length sentences, which are generally longer samples than have been generated in related works. The quality of the generated text samples will be evaluated through a survey in which participants will be asked to rate the quality of sentences of horror text generated by the GAN and by human authors.

For neural networks to learn meaningful relationships that can generalize well, sufficient training data must be obtained. Horror texts for this project will be gathered from Project Gutenberg. Texts that will be used for training include:

- *Twenty-Five Ghost Stories*, an anthology edited by W. Bob Holland
- *Metamorphosis* by Franz Kafka
- *The Trial* by Franz Kafka
- *The Shunned House* by H. P. Lovecraft
- *The Works of Edgar Allen Poe*, Volumes 1 – 5 by Edgar Allen Poe
- *Varney the Vampire: Or the Feast of Blood* by Thomas Preskett Prest
- *The Strange Case of Dr. Jekyll and Mr. Hyde* by Robert Louis Stevenson
- *Dracula* by Bram Stoker
- *Dracula's Guest* by Bram Stoker
- *The Jewel of Seven Stars* by Bram Stoker
- *The Lady of the Shroud* by Bram Stoker
- *Lair of the White Worm* by Bram Stoker
- *The Man* by Bram Stoker

These particular texts were selected for author name recognition as well as the variety of horror elements. This will create a GAN that is trained on multiple types of horror characters and trained on texts that are known to elicit strong emotions from readers.

The goal is that by the end of the project, a random feature vector can be fed into the network to output a variable length sentence of horror text. Several samples will be collected, along with randomly selected sentences from the training dataset. A survey will be created asking participants to rank the quality of the random samples from the GAN and the human authors. Participants will be asked to rank the samples on a scale from one to five, one being poor quality horror text and five being high quality. The participants will not know which samples are from the GAN and which are from human authors. To keep the survey to a manageable length but provide sufficient measurements of the quality of the samples, there will be ten text samples from each source. The survey will define a quality horror text sample as one that elicits some sort of emotion from the readers, but will be intentionally vague on the definition of "quality." The purpose of the survey results are to provide a qualitative evaluation of the text generated by the GAN which can be compared to the qualitative evaluation of text from human authors. Because the project will serve as an opportunity for exploring the capabilities of artificial intelligence in the realm of storytelling, the success of the project will be measured by undirected human reactions to the generated text.

IV. Foundations

The foundations of this project lie in the field of natural language processing. Natural language processing (NLP) is the field of study that allows us to "characterize and explain" how humans and computers "acquire, produce, and understand language" (Manning and Schütze 3). For computers to generate text, they must have a way of learning the rules, structure, and meaning that govern language.

The GAN framework has provided a way for computers to generate language. GANs were originally proposed in 2014 by the research team led by Ian Goodfellow (Goodfellow et al.). GANs consist of two neural networks, a generator *G* and a discriminator *D*. These networks are adversaries. *D* is trying to determine whether samples are from the dataset or fom the generator, meaning they are real or fake, respectively. Meanwhile *G* is trying to generate better samples to trick *D* into thinking its samples are real. Training ends when the generator's samples are nearly indistinguishable from those in the dataset, meaning the discriminator is guessing whether they are real or fake (Heinrich).

A visualization of the GAN framework can be seen in Appendix A. This image portrays the GAN framework as it would be used to generate sample images (Heinrich). While this is not the application of the GAN framework for this project, it is useful to visualize how the training process works. To begin, a vector of random numbers called the *z* vector is fed into the generator. The elements of the *z* vector will, through training, come to correspond to particular features in the generated samples. The generator creates a sample from the *z* vector. The discriminator is alternatively fed samples from the dataset and from the generator. The discriminator examines the sample and classifies it as either real—from the dataset—or fake—from the generator. Based on its guess, the discriminator is optimized through back-propagation to minimize the error in its guesses and better discern the difference between the two sample sources. The classification of the discriminator is also back propagated through the generator to train the generator to create better samples and maximize the likelihood that the discriminator will make a mistake.

Because neural networks work with numeric data, it is important to represent text—whether on the character level or the word level—numerically. One solution is that rather than having the generator output a character or word, the generator will output a probability distribution over all possible characters or words. The character corresponding to the greatest probability in the distribution can later be interpreted as the character output by the generator. When feeding sequences of characters into the discriminator, the samples from the generator will be represented as sequences of probability distributions while the samples from the dataset are one-hot vectors. This means that the intended character or word is given a probability of one, while all other possible characters or words are assigned a probability of zero (Press et al.). By representing the vocabulary of all characters or words as a probability distribution, the problem of text generation can effectively be tackled by neural networks.

When generating text using the GAN framework, both the generator and discriminator need memory. In this project, the generator will create a sentence of text one character at a time. Any output character will need to be conditioned on the sequence of previous outputs in order to generate a sentence with long-term structure and meaning. Similarly, the discriminator will process a sentence of text one character at a time to determine whether the sample is real or fake. A type of network architecture that allows for memory is the RNN. RNNs feed outputs from previous time steps back into the network as inputs at current time steps. This means that the output of the network at any given time is conditioned on all previous outputs (Medsker). In the context of text generation, any character output by the network is conditioned on all previous output characters, ensuring long-term structure in the text.

However, when used within the GAN framework, RNNs may not train effectively. Researchers have found that for the purpose of text generation, regular RNNs do not train very well because certain parameters cannot easily be normalized. However, GRUs and LSTMs, both of which are types of RNNs, prove to be more effective at training and learning complex long-term structure within text (Press et al.; Rajeswar). LSTMs can selectively remember new information and forget information from previous states, allowing them to learn more complex and longer structure within text (Donahue 2626). GRUs are comparable to LSTMs in terms of the structure learned, but contain fewer parameters, making them more computationally efficient (Rajeswar). Because of the lack of research using LSTMs within the GAN framework, LSTMs will be used to achieve memory for this project.

There are several algorithms that can be used to aid the training process. These extensions include curriculum learning (CL), variable length outputs (VL), and teacher helping (TH). All of these algorithms were used to train a GAN with GRUs for memory to generate 32 character sequences of text (Press et al). CL trains the GAN on shorter character sequences and incrementally increases to longer sequences. The generator will generate sequences of length one in the first training stage while the discriminator receives sequences of length one. In the second stage, the networks generate or receive sequences of length two, and so on. The curriculums continue until the desired character sequence length is reached (Press et al). VL allows the network to work with sequences less than or equal to the maximum length determined by the current curriculum. For example, if the current curriculum trains the network to output sequences of ten characters, VL will allow the network to output sequences less than or equal to ten characters. This ensures that the network can handle outputs of variable length (Press et al.) TH conditions the network to generate long character sequences through short sequences that appear in the training dataset. A sequence of characters from the dataset is fed into the generator, which is expected to then produce a probability distribution over the final character in the sequence. This training extension suffers from exposure bias, based on the sorts of sequences in the training dataset. However, once the discriminator is fed this sequence of real characters with a fake final character, the generator learns to create better samples (Press et al.). In this project, all of these extensions will be used in the training process of the GAN in order to create a model that generalizes beyond samples seen in the training data.

V. Implementation Plan and Timeline

The timeline for project implementation during the spring semester is based on biweekly checkpoints. The checkpoints for the semester are described below.

- *Checkpoint 1*: By the end of this checkpoint, the script that will handle text preprocessing will be created. The goal of text preprocessing is to edit the raw text files from Project Gutenberg so that the GAN can effectively learn from the data. Punctuation and capitalization will have to be removed from the text because it may affect how words are distinguished from one another. For example, in Edgar Allen Poe's *The Masque of the Red Death*, if punctuation and capitalization were not ignored, "death," "Death," and "death." would all be considered different words. Text preprocessing will also include breaking texts into sentences, which will serve as individual samples in the training dataset. While the script for text preprocessing will handle much of the preprocessing automatically, there will likely be some formatting that will need to be done manually before the dataset is ready to train the GAN. By the end of this checkpoint, it will be understood how much preprocessing will need to be done manually.
- *Checkpoint 2*: This checkpoint will focus on determining the architecture of the network. The network architectures used in previous works—particularly relating to how LSTMs are used within the GAN framework—will influence the network architecture design for this project. One specific reference for network architecture will be the code used in the research conducted by Press et al., which was available on GitHub. This semester, the

researcher attempted to run the code from this project, but ran into compatibility problems between versions of Python. By Checkpoint 2, these issues will be resolved and the researcher will have trained the GAN with GRUs created by Press et al. in their research. If time allows, this GAN will be extended to train on horror text rather than the dataset used in the original work. This will give an estimate on training time and experience working with a particular network architecture.

- *Checkpoint 3*: By this checkpoint, the code for the GAN framework will be largely written, using the code used in research by Press et al. as a guide. The code necessary to give the generator and discriminator memory through LSTMs will also be started. Tutorials on machine learning and LSTMs in Python and TensorFlow—the selected tools for implementing this project—will be sued as a reference.
- *Checkpoint 4*: This checkpoint will focus on finishing the code for the LSTMs in the generator and discriminator and implementing the learning algorithms, specifically curriculum learning, variable length outputs, and teacher helping. Once again, code used in research by Press et al. will be used as reference. The interactions between LSTMs and the curriculum learning algorithm will be informed by work done by Rajeswar. The final poster presentation, which will be given on Scholar's Day in the spring, will also be started.
- *Checkpoint 5*: The training of the GAN will be the focus of this checkpoint. Because the network is performing such a computationally expensive task, it is assumed that there will only be time to train the network once. However, more specific training time estimates will be obtained by the end of Checkpoint 2. While waiting on the network to train, a significant amount of time will be dedicated to putting the final poster presentation together.

- *Checkpoint 6*: During the final checkpoint, the evaluation survey will be created and distributed via email and social media platforms. By the end of the checkpoint, all responses obtained will be used to give a qualitative measure of the quality of the text samples generated by the GAN. The poster presentation will be finished, including the results of the survey and the conclusions of the project.

Checkpoints are structured to maximize success in the event of an unforeseen obstacle. For example, one of the goals of Checkpoint 2 is to extend the network—a GAN using GRUs for memory—from a previous project to train on horror text. In the event that the objectives of later checkpoints are not met, the project can continue through extensions of this network. Even though the ultimate goal of using LSTMs for memory within the GAN framework may not be met, there will be a working network trained on horror text.

If time allows, the developed network can also be quantitatively evaluated using the methods investigated in the related mathematics capstone project, as described in the introduction. While the focus of this project is the qualitative evaluation of artificial intelligence participating in the human task of storytelling, it would be interesting and informative to compare the qualitative and quantitative metrics. This could give a better understanding of what each metric measures and whether the metrics cannot accurately represent some characteristic of the text.

Tools required for this project include Python and TensorFlow. Documentation and tutorials will be used as reference. The code used in research by Press et al.—a work which significantly influenced the specifications of this project—was implemented in Python and TensorFlow, and will be used as a guide throughout implementation.

To begin working with the foundational material for the project, the researcher put together a technical presentation exploring GANs and LSTMs in Python and TensorFlow. Due to incompatibility between Python versions—code used by Press et al. was written in Python 2.7 while the version installed on the department server is Python 3—the reference GAN using GRUs could not be run this semester. One of the goals for the spring semester is to continue troubleshooting and fixing the incompatible functionality to run this code by the end of Checkpoint 2. However, the researcher was able to run a tutorial from the TensorFlow website on LSTMs (TensorFlow). The purpose of the tutorial was to train an LSTM to model the English language, based on the work of Zaremba et al. Though the task of language modeling is less computationally expensive than text generation, it was nevertheless useful to start working with recurrent networks in Python and TensorFlow. The results of the experiments running the tutorial showed that longer training time created a model that better represented the language. However, even the lightweight language modeling task took almost two and a half hours to train for a relatively small network architecture. The experience of the technical presentation confirmed the need to carefully budget time for training the network while providing an opportunity to work with LSTMs using the tools for the project.

VI. Conclusion

The goal of this computer science capstone project is to build a generative adversarial network and train it to generate sentences of horror text one character at a time. In order to achieve long-term structure within the text samples, the generator and discriminator in the GAN will both be long short-term memory networks. These networks will output or take in sequences of probability distributions over all possible characters, which allows the task of text generation to be represented numerically and handled by neural networks. Extensions allowing for curriculum learning, variable length outputs, and teacher helping will aid the training process and help the network generalize to samples not seen in the training data. The quality of the generated samples will be measured qualitatively through a survey in which participants will be asked to rank the quality of texts from the GAN and from human authors. This qualitative measurement will provide a way of determining how well the network carries out the human task of storytelling.

The work to implement this project during the spring semester will be challenging yet rewarding. While there are sure to be obstacles given that the researcher has never worked with neural network architectures, the opportunity to experience artificial intelligence research will ultimately be the driving force of the project. Planning biweekly checkpoints, possible ways to scale back the project, and opportunities to extend the project has forced the researcher to carefully consider and prioritize realistic goals. No matter which goals are met, the final product of this capstone project will serve as an exploration of the capabilities of artificial intelligence. The quality of the text generated by the GAN will carry implications about artificial intelligence participating in human tasks, what intelligence means, and how we express our own humanity.

Bibliography

Bird, Steven, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media. 2009.

Clark, Alexander, Chris Fox, and Shalom Lappin, eds. *The Handbook of Computational Linguistics and Natural Language Processing*. John Wiley & Sons. 2013.

Creswell, Antonia, et al. "Generative Adversarial Networks: An Overview." *arXiv preprint arXiv:1710.07035*. 2017.

Donahue, Jeffrey, et al. "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

Goodfellow, Ian, et al. "Generative Adversarial Nets." *Advances in Neural Information Processing Systems*. 2014.

Heinrich, Greg. "Photo Editing with Generative Adversarial Networks (Part 1)." *NVIDIA, Parallel Forall*, devblogs.nvidia.com/parallelforall/photo-editing-generative-adversarial-networks-1/. 2017. Accessed Aug. 2017.

Heinrich, Greg. "Photo Editing with Generative Adversarial Networks (Part 2)." *NVIDIA, Parallel Forall*, devblogs.nvidia.com/parallelforall/photo-editing-generative-adversarial-networks-2/. 2017. Accessed Aug. 2017.

Horror Writers Association. "What is Horror Fiction?" *Horror Writers Association*, http://horror.org/horror-is.htm. 2009. Accessed Nov. 2017.

Kawthekar, Prasad, Raunaq Rewari, and Suvrat Bhooshan. "Evaluating Generative Models for Text Generation." *Stanford University*. 2017.

Manning, Christopher D., and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. Vol. 999. Cambridge: MIT Press. 1999.

McKeown, Kathleen. *Text Generation*. Cambridge University Press. 1992.

Medsker, Larry, and Lakhmi C. Jain, eds. *Recurrent Neural Networks: Design and Applications*. CRC Press. 1999.

Mirza, Mehdi, and Simon Osindero. "Conditional Generative Adversarial Nets." *arXiv preprint arXiv:1411.1784*. 2014.

NLTK Project. "Natural Language Toolkit." *NLTK Project*, http://www.nltk.org/. 2017. Accessed Oct. 2017.

Press, Ofir, et al. "Language Generation with Recurrent Generative Adversarial Networks without Pre-training." *arXiv preprint arXiv:1706.01399*. 2017.

Project Gutenberg. "Free ebooks – Project Gutenberg." *Project Gutenberg*, www.gutenberg.org/. 2017. Accessed Oct. 2017.

Rajeswar, Sai, et al. "Adversarial Generation of Natural Language." *arXiv preprint arXiv:1705.10929*. 2017.

Reed, Scott, et al. "Generative Adversarial Text to Image Synthesis." *arXiv preprint arXiv:1605.05396*. 2016.

Riedl, Mark. "Why Artificial Intelligence Should Read and Write Stories." *Huffington Post*, www.huffingtonpost.com/mark-riedl/why-artificial-intelligen_b_8287478.html. 2015. Accessed Oct. 2017.

Sutskever, Ilya, James Martens, and Geoffrey E. Hinton. "Generating Text with Recurrent Neural Networks." *Proceedings of the 28th International Conference on Machine Learning* (ICML-11). 2011.

TensorFlow. "Getting Started With TensorFlow." *TensorFlow*, www.tensorflow.org/get_started/get_started. 2017. Accessed Oct. 2017.

TensorFlow. "Recurrent Neural Networks." *TensorFlow*, www.tensorflow.org/tutorials/recurrent. 2017. Accessed Oct. 2017.

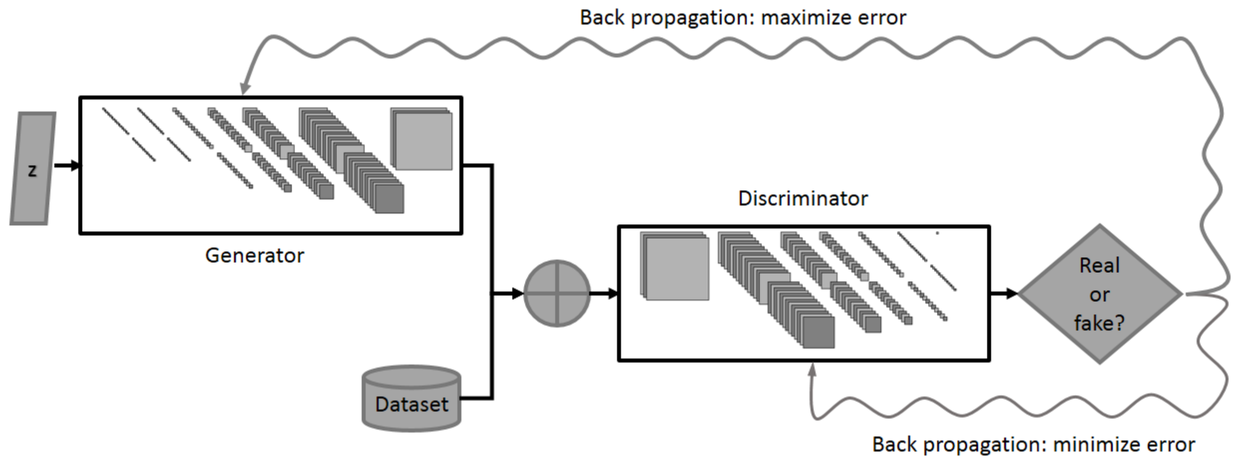
Waite, Elliot. "Generating Long-Term Structure in Songs and Stories." *Magenta*, magenta.tensorflow.org/2016/07/15/lookback-rnn-attention-rnn. 2016. Accessed Aug. 2017.

Xiao, Xuerong. "Text Generation using Generative Adversarial Training." *Stanford University*. 2017.

Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent Neural Network Regularization." *arXiv preprint arXiv:1409.2329*. 2014.

Appendix A

GAN Framework



This image, from Greg Heinrich's article "Photo Editing with Generative Adversarial Networks (Part 1)," shows a visual representation of the GAN framework. This visualization does not represent the purposes of this project, as the generator and discriminator in the picture above are both convolutional neural networks that either generate or classify images. However, the image is useful in the context of this project because it shows how samples—in this case, sequences of characters that comprise a sentence—will be built up by the generator and broken down by the discriminator.