Generative Adversarial Networks

Daniel Hackney 

Timothy Neale 

Advisor:
Badri Velllambi 

UNIVERSITY OF Cincinnati
College of Engineering and Applied Science

Objective:

To research and design a functional Generative Adverserial Network, and use it to generate pictures of animals similar to the efficacy of existing GANs like thispersondoesnotexist.com.

Once created, experiment with the input data and neural network to see how it affects the resulting data.

Process:

We started with a DCGAN optimized for 28x28 pixel, black and white, images. From there, we modified it to convert a local dataset into usable data, upscaled the images, added color, and optimized it to run using GPUs.

Once we had reasonable generated images, we experimented with the generator to attempt to remedy instability and errors with our encoding. We finished by getting our loss functions to plot after training.

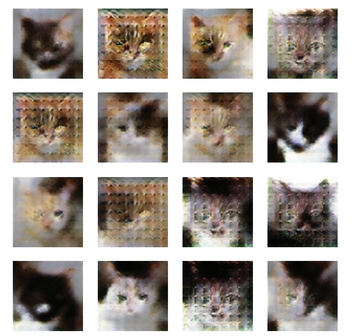
Outcomes:

Our final product is a series of generated 128x128 RGB images of cats, and a gif showing the training of the GAN over 1000 epochs. Our final runtime is ~100 minutes, but the generator can then be used to create new cat faces in <6 seconds.

Some stability and optimizations could be added to increase the quality of the outputs, but this was beyond the scope of our project, given the time constraint and our lack of expertise with machine learning.

If we were to start this project again with our expanded knowledge, we could generate cats to convince viewers that they were real. We could also get better results starting from training checkpoints. We could train using multiple animal faces, creating a bimodal distribution of input data.

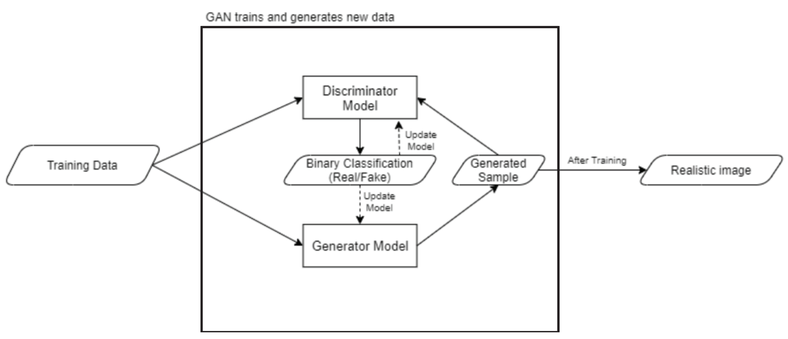
Initial 28x28 B&W output



Final 128x128 RGB output

Design:

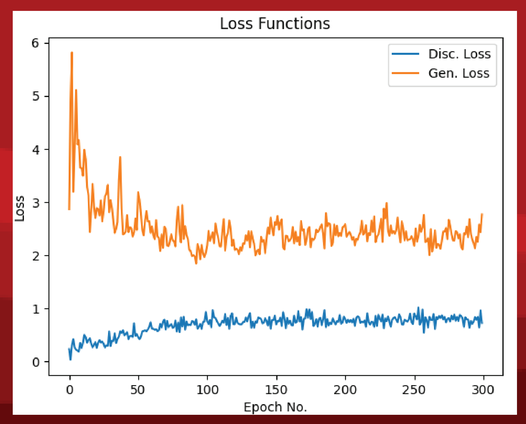
A GAN is an unsupervised neural network composed of a generator and discriminator. The former seeks out patterns in the input to create new data elements, while the latter accepts or rejects these elements based on past experience. These models improve eachother in a minimax game until the disciminator can only guess.



Our GAN is deep convolutional, and consists of a series of 2D transpose layers followed by batch normalization and leaky rectified linear unit activation functions.

Challenges:

- **Computational Intensity** - Needed to utilize the Ohio Super Computer (OSC)
- **Technical Experience** - Limited exposure to neural networks and machine learning
- **Project Scope** - Project objectives shifted as research into GANs was carried out
- **Limited Data** - Overestimated number of large, publicly available datasets
- **Instability** - Only a specific range of iterations was yielding covincing cat images



A perfect GAN will have loss functions that converge. Clearly our GAN stops learning after about 200 epochs. This can be improved with various modifications to our GAN's functionality.

References:

https://www.tensorflow.org/tutorials/generative/dcgan
https://thispersondoesnotexist.com/
https://www.kaggle.com/andrewmvd/animal-faces
https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf
https://arxiv.org/pdf/1701.07875.pdf
https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/