Introduction to machine learning

Anime faces generation by GAN

Juo-Yu,Wang jyw319
Chia-Yuan, Kuo cyk324

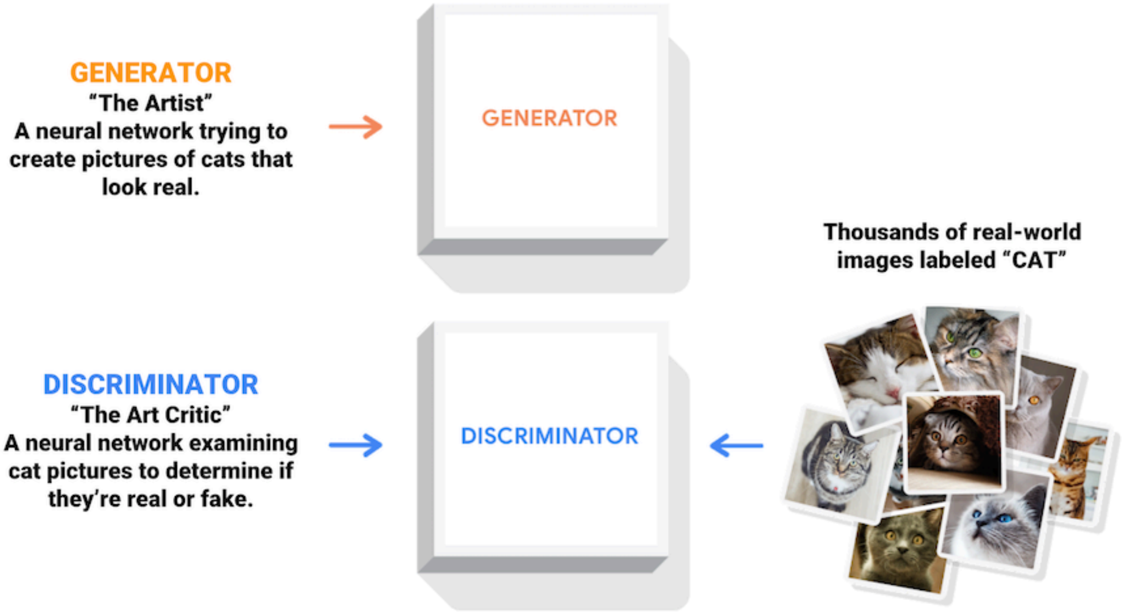
Abstract

In this project, we are trying to build a Generative Adversarial Networks (GANs) and use it to generate anime faces. We have applied the Deep Convolutional GAN (DCGAN) method to the model. To avoid the fast convergence of the Discriminator, the Generator network is updated twice for each network update.

Introduction

GAN

Generative adversarial networks (GANs) are deep neural net architectures comprised of two nets, pitting one against the other (thus the "adversarial"). One neural network, called the generator, generates new data instances, while the other, the discriminator, evaluates them for authenticity; i.e. the discriminator decides whether each instance of data that it reviews belongs to the actual training dataset or not.


GENERATOR
"The Artist"
A neural network trying to create pictures of cats that look real.
GENERATOR
Thousands of real-world images labeled "CAT"
DISCRIMINATOR
"The Art Critic"
A neural network examining cat pictures to determine if they're real or fake.
DISCRIMINATOR

(https://tensorflow.org/alpha/tutorials/generative/images/gan1.png)

Deep Convolutional GANs

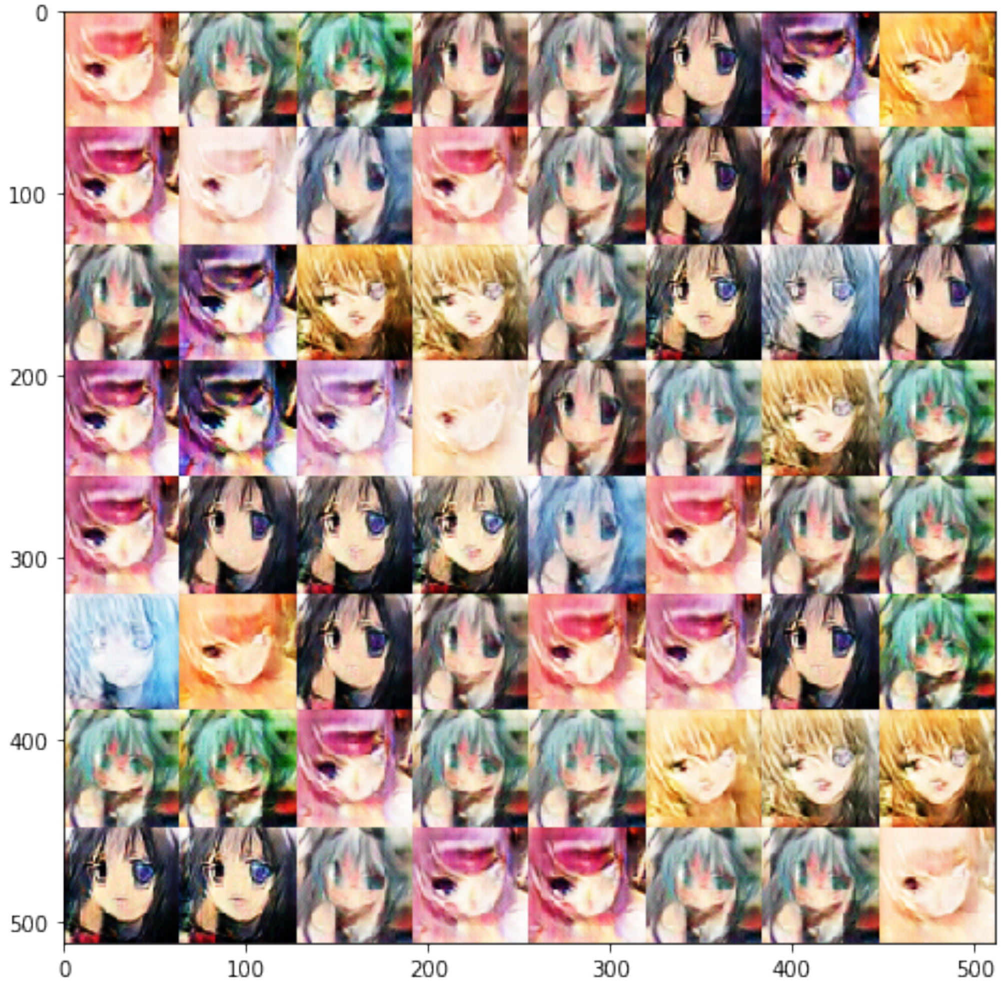
The Difference between the Simple GAN and the DCGAN is the generator of the simple GAN is a simple fully connected network, and the generator of the DCGAN uses the transposed convolution technique to perform up-sampling of 2D image size.

Procedure

We used the google gcp to run the jupyter file.
The data set we used (includes 30,000 pictures of anime faces):

We created a generator and discriminator model using different parameters, the parameters can be found on the project_DCGAN_result file.
The code and results are stored in the appended google drive file.
https://drive.google.com/drive/folders/1n2hSxLEwXD4Hl9crvli3C_OPusEe3JeT?usp=sharing

Results



As we can see from the output results, the outputs progressed from the initial noise filled pictures to blurry images with some facial feature resemblance.

Discussion

1. Firstly, we tried changing the optimizer from the Adam optimizer to the gradient descent optimizer, the results came out pretty bad. We have trained the model for several epochs, however the results did not get any better, we did not see anything resembling an anime face.
2. According to the output, we have noticed that similar faces were generated. To mitigated this issue, we could apply some sort of penalty to the model in order to penalize the generation of pictures that look alike.
3. The nature of the GAN makes it really hard to train, we can see this from the resulting pictures. One way to possibly mitigate this issue is to use the Wasserstein GAN (WGAN). The WGAN learns no regardless of the generator performance and the gradient will come out smoother.

References

http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2018/Lecture/CGANtip.pdf
https://github.com/soumith/ganhacks
https://arxiv.org/abs/1406.2661
https://github.com/tjwei/GANotebooks
https://github.com/carpedm20/DCGAN-tensorflow