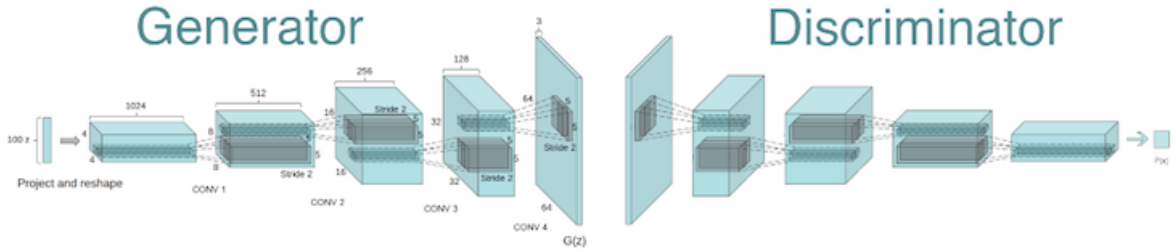
Greyson Gerhard-Young

Context

In the modern era, AI models are becoming increasingly sophisticated at producing realistic content. A common example of this is DeepFakes, which permeate social networks and make the already arduous task of fact checking content even more difficult. This project mirrors those attempts by generating fake synthesized images in an attempt to better understand the underlying technology.

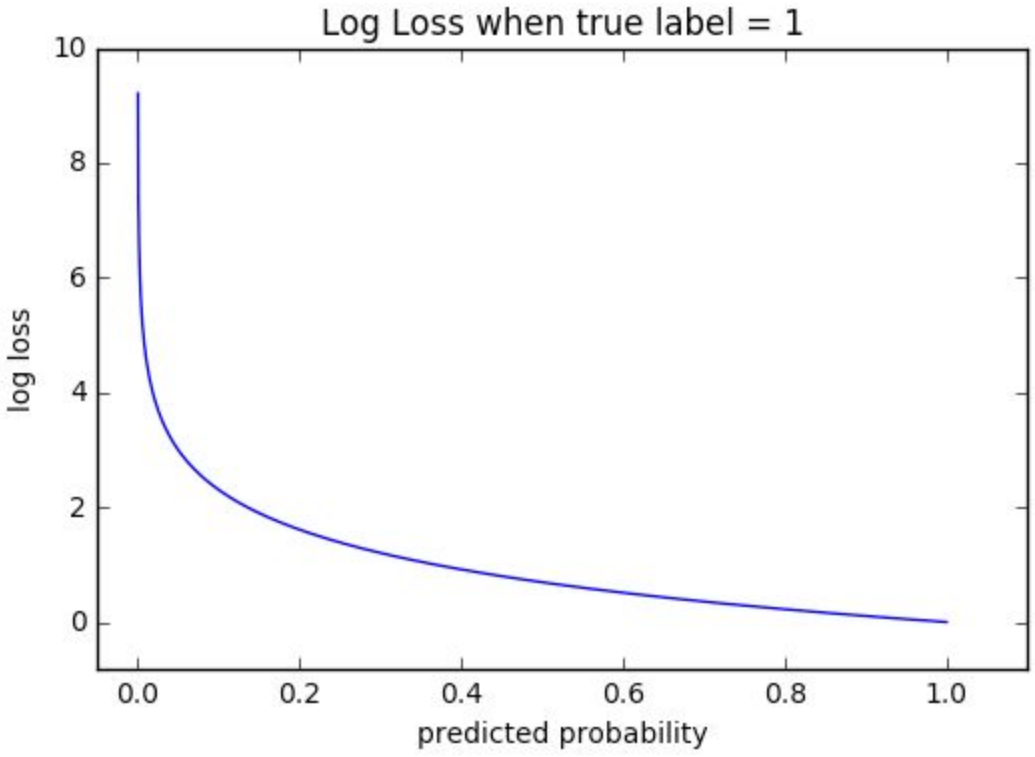
Research

Many of the current thought leaders in the field have experienced success with adversarial architectures. This consists of a generator model that learns to produce fake images, along with a corresponding discriminator model that tries to distinguish between real and fake. The generator uses an up-convolutional model to map from a code to an image, while the discriminator reverses the process to produce probabilities (representing how likely it is that that image is fake). The architecture is shown below, and modeled after this paper by Redford, Metx, and Chintala (https://arxiv.org/abs/1511.06434).



Design Process/Tradeoffs

When building a deep learning model like this one, there are a number of important design factors to consider (hyperparameters, size of layers, etc.). We tried iterations of the model with MSE and Maximum Likelihood as the loss function, but ultimately went with the author's recommendation of slightly modified versions of binary cross-entropy loss.



With regards to hyperparameters, there are almost infinite possible combinations to test out. You can increase layer size in an attempt to help the model glean more information, or add dropout layers as a means to prevent overfitting. The largest constraint on this particular problem was computing power. A single run of the model in a GCP

environment could take up to 5 hours, so it was important to methodically take these considerations into account.

The combination that best maximized resources was one with layers of the following sizes:

64 → 128 → 256 → 512

512 → 256 → 128 → 64 → result

It's also important to note that these dense layers were separated by LeakyRelu/Batch Normalization layers. These are essential to capturing the 'magic' of Deep Learning, since they provide non-linear properties.

Images in the midst of training resembled human features, but weren't exceedingly accurate.

However, by the latter stages of the process, features became much more realistic.

Disclaimer: these are fairly low resolution images, since we were constrained from a computational perspective

Concluding Thoughts

This project demonstrated that with enough computing power, it isn't incredibly difficult to synthesize an existing database of images into something new that's fairly realistic. I also learned just how important it is to thoroughly understand the latest research in the field of deep learning, as it provides incredible templates for future tweaks. I'd love to experiment with similar techniques in a more video centric context.