4/22 Project Progress Check In

Liz Parker, Aren Dalloul, and Colin Craighead

Project Topic: Building a Generative Adversarial Network

A GAN is made of two models: generator and discriminator

The **generator** is a neural network that takes noise as input and outputs a sample of data that follows the desired distribution - that distribution is determined by the training data.

The **discriminator** is a neural network that takes a data sample as input and outputs a probability that the data sample is real - determining if the input data was generated or from the real data distribution. A probability closer to 1 suggests the input is real, while a probability closer to 0 suggests the input is fake/generated.

The two networks are “adversaries” pitted against each other. Using a log loss metric, the generator is trying to maximize the outputs of the discriminator (more inputs classified as real), and the discriminator is trying to minimize the probability that a generated image is incorrectly classified as real.

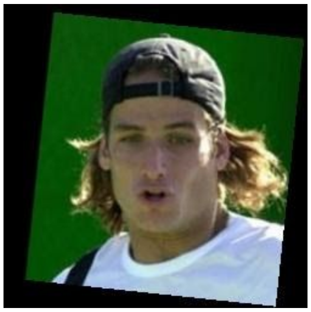
Data Sources

We are interested in using a GAN to generate images. There are many comprehensive image datasets at our disposal, including:

Labeled Faces in the Wild from University of Massachusetts

(http://vis-www.cs.umass.edu/lfw/)

These images are adjusted for alignment



Stanford Dogs Dataset

(http://vision.stanford.edu/aditya86/ImageNetDogs/)

Our biggest question is how to convert these datasets provided as folders and .JPGs of varying size into an input that can be interpreted by our models.

Our Approach

As an introduction to GANs, we will be doing research to understand how they work. This will include reading Ian Goodfellow's original paper introducing GANs (found at https://arxiv.org/abs/1406.2661) and a later update with training techniques (found at https://arxiv.org/abs/1606.03498)

There are a few python/tensorflow examples for implementing a GAN that generates handwritten numbers, trained by the MNIST dataset

https://github.com/StackAbuse/Introduction-to-GANs-with-Python-and-TensorFlow/blob/master/gan.py

https://github.com/diegoalejogm/gans/blob/master/1.%20Vanilla%20GAN%20TensorFlow.ipynb

https://github.com/eriklindernoren/Keras-GAN