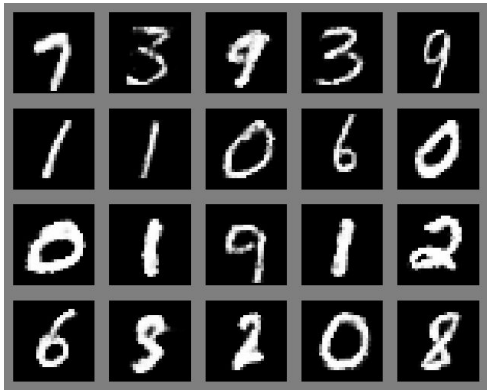
GANs

TJ Machine Learning

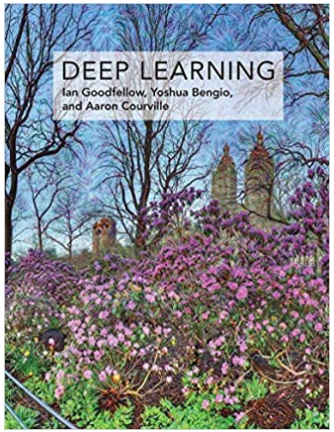
* A lot of the content in this lecture is from the GANs section of Google's Crash Course in Machine Learning.

History

- Introduced several times in bits and pieces during last several years but fully developed by Ian Goodfellow in 2014
- Idea developed at a bar and first came online at ~ 3 a.m.
- Rare occasion where idea panned out on the first try



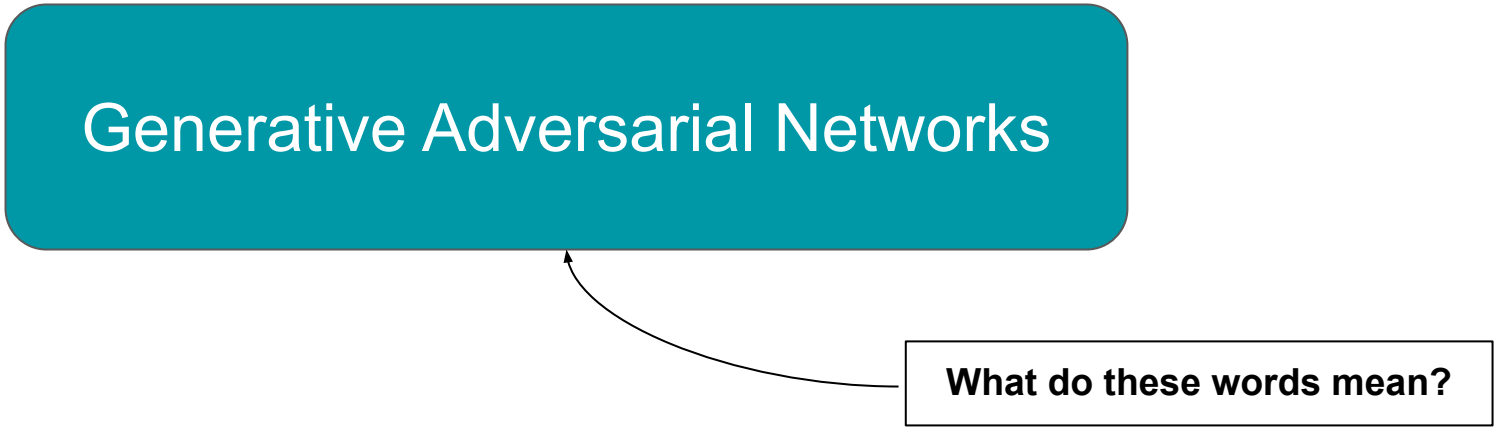
From GAN Paper



Purpose

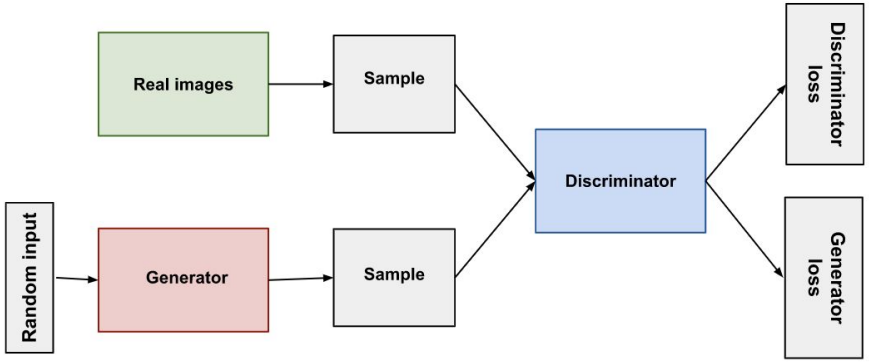
- **Provided:** Dataset of image (Ex: Faces)
- **Goal:** Generate synthetic images based on common features learned from the provided dataset
- https://www.whichfaceisreal.com/

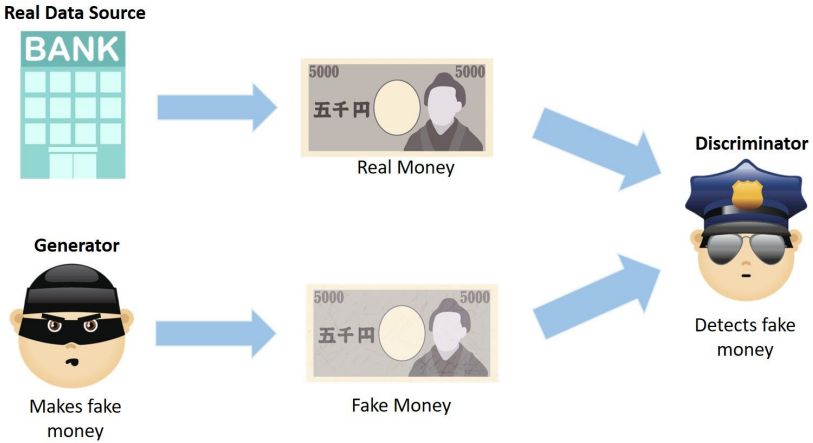
How do we do this?



Overview of GAN Structure

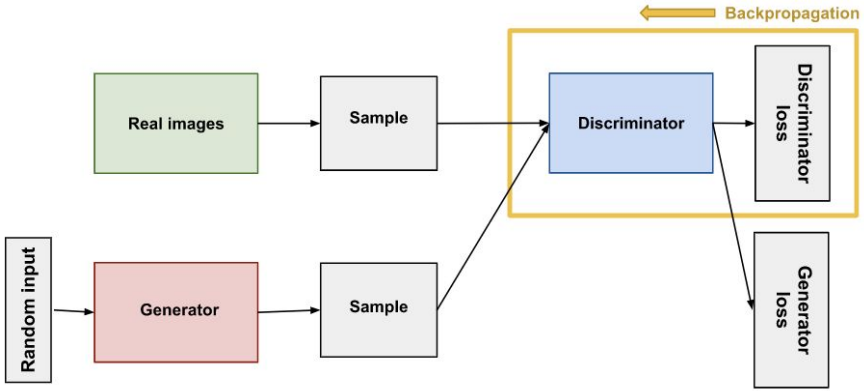
- Pitting two neural networks, the **Generator** and **Discriminator**, against each other
- **Generator** is like a counterfeiter printing fake money
- **Discriminator** is like a cop trying to distinguish real money between fake money
- Both systems (hopefully) get better together during training





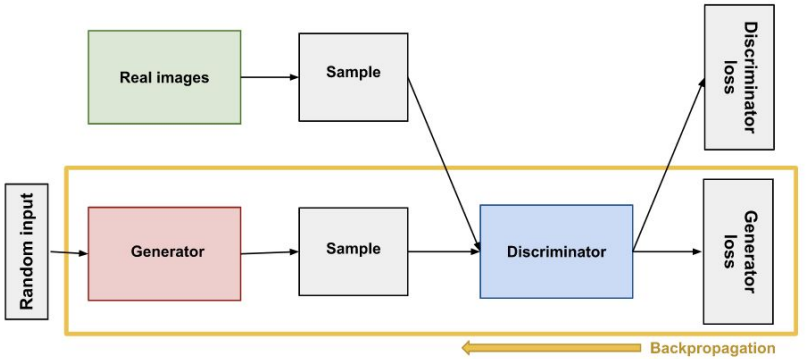
Discriminator

- Training data of discriminator consists of two types:
  - Real data (from our training set)
  - Fake/Generated data from our Generator
- Training:
  - Classifies both real images from training dataset and fake images coming from our generator
  - Discriminator should get penalized for wrong prediction about image



Generator

- Random input is required since the generator is a neural network
  - Some GANs use tailored inputs to shape the output of the generator
- Note: Our generator's performance is determined by how good (or bad) the discriminator is doing
- Training the generator:
  - Sample random noise and generate image
  - Get discriminator output for given image
  - Calculate loss (how did discriminator classify?)
  - Perform backpropagation on generator weights
    - Don't change discriminator weights!



GAN Training

- When training a GAN, we have to train two different networks
- The GAN training algorithm looks like:
  - Train the discriminator for 1 epoch
  - Train generator for 1 epochs
  - Repeat above steps
- GAN convergence is hard to pinpoint because we have two networks that depend on each other

- For those who have taken TJ's AI class, the GAN training process can be formalized as a two-player minimax game (just like tic-tac-toe). Check the GAN paper for more info.

Discriminator Loss

- Binary Classification Task → Binary Cross-Entropy
- Labels → $y = [1, 1, 1, 0, 0, 0]$
- Predictions → $\hat{y} = [0, 1, 1, 1, 0, 1]$
- 1: Real Image (i.e. from dataset)
- 0: Fake Image (i.e. generated by generator)
- Only backpropagate through the **discriminator** with this loss!

Generator Loss Function

- What is the goal of the generator?
- How can we measure this goal?

Generator Loss Function

- **Objective:** Trick the discriminator into predicting 1 (real) as the output for the generated images
- Labels → $y = [1, 1, 1]$
- Discriminator Predictions → $\hat{y} = [0, 0, 1]$
- Only backpropagate through the **generator** with this loss!

Applications of GANs

- DeepFakes
- Generating new training data to improving training accuracy
- Image Colorization
- Improving Image Resolution