Generative Adversarial Networks (GAN)

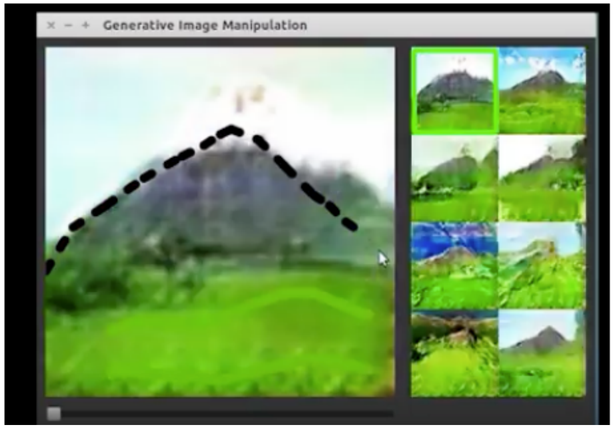
ECE57000: Artificial Intelligence, Fall 2019

David I. Inouye

Announcements

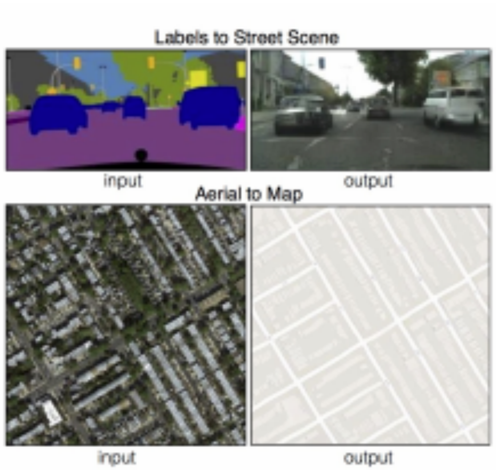
- HW3 due Monday
- Quiz on Wednesday

Why study generative models?



- Sketching realistic photos

- Style transfer

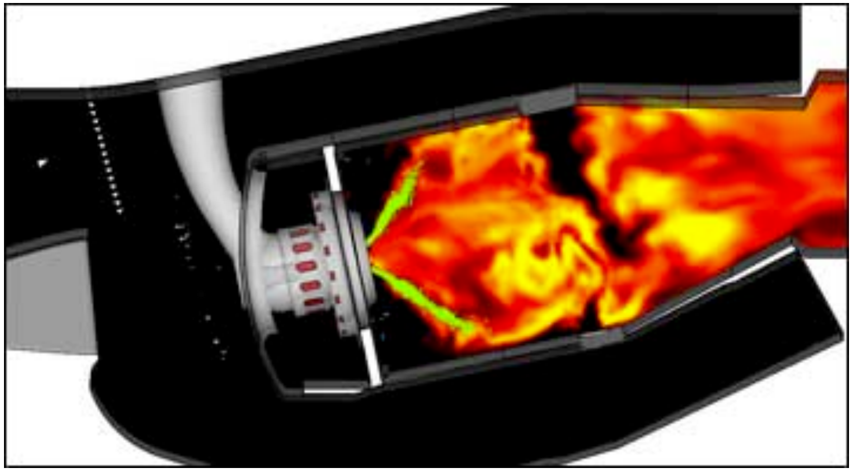


- Super resolution

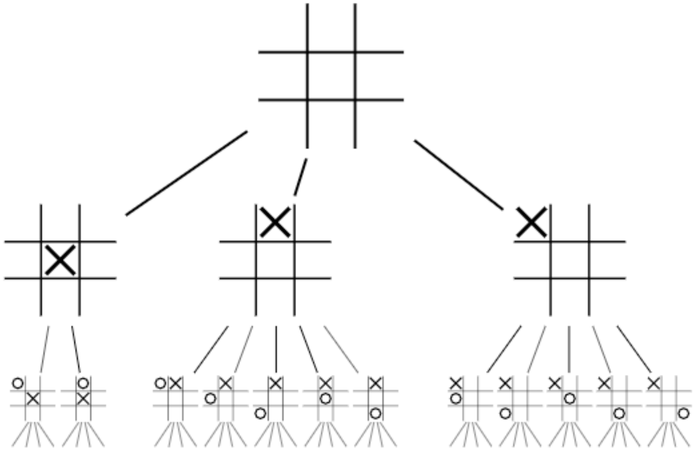
Much of material from: Goodfellow, 2012 tutorial on GANs.

Why study generative models?

- Emulate complex physics simulations to be faster



- Reinforcement learning - Attempt to model the real world so we can simulate possible futures



Much of material from: Goodfellow, 2012 tutorial on GANs.

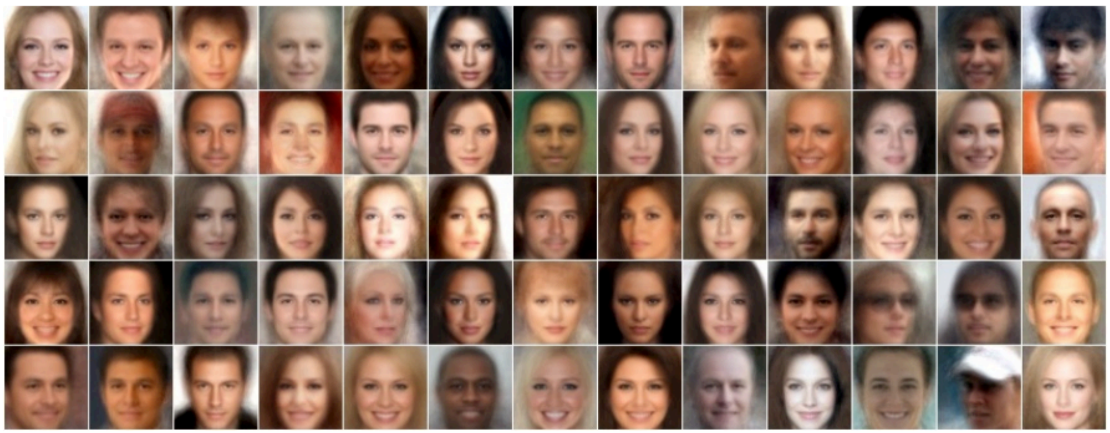
How do we learn these generative models?

- Primary classical approach is MLE
  - Density function is explicit
  - Density parameterized by θ
  - Examples: Gaussian, Mixture of Gaussians

- Problem: Classic methods cannot model very high dimensional spaces like images
  - Remember a 256x256x3 image is roughly 200k dimensions

Maybe not a problem: GMMs compared to GANs
http://papers.nips.cc/paper/7826-on-gans-and-gmms.pdf

- Which one is based on GANs?



Newer (not necessarily better) approach: Train generative model **without explicit density**

- GMMs had **explicit** density function
  (i.e., mathematical formula for density $p(x;\theta)$)

- In GANs, we just try learn a sample **generator**
  - **Implicit** density ($p(x)$ exists but cannot be written down)

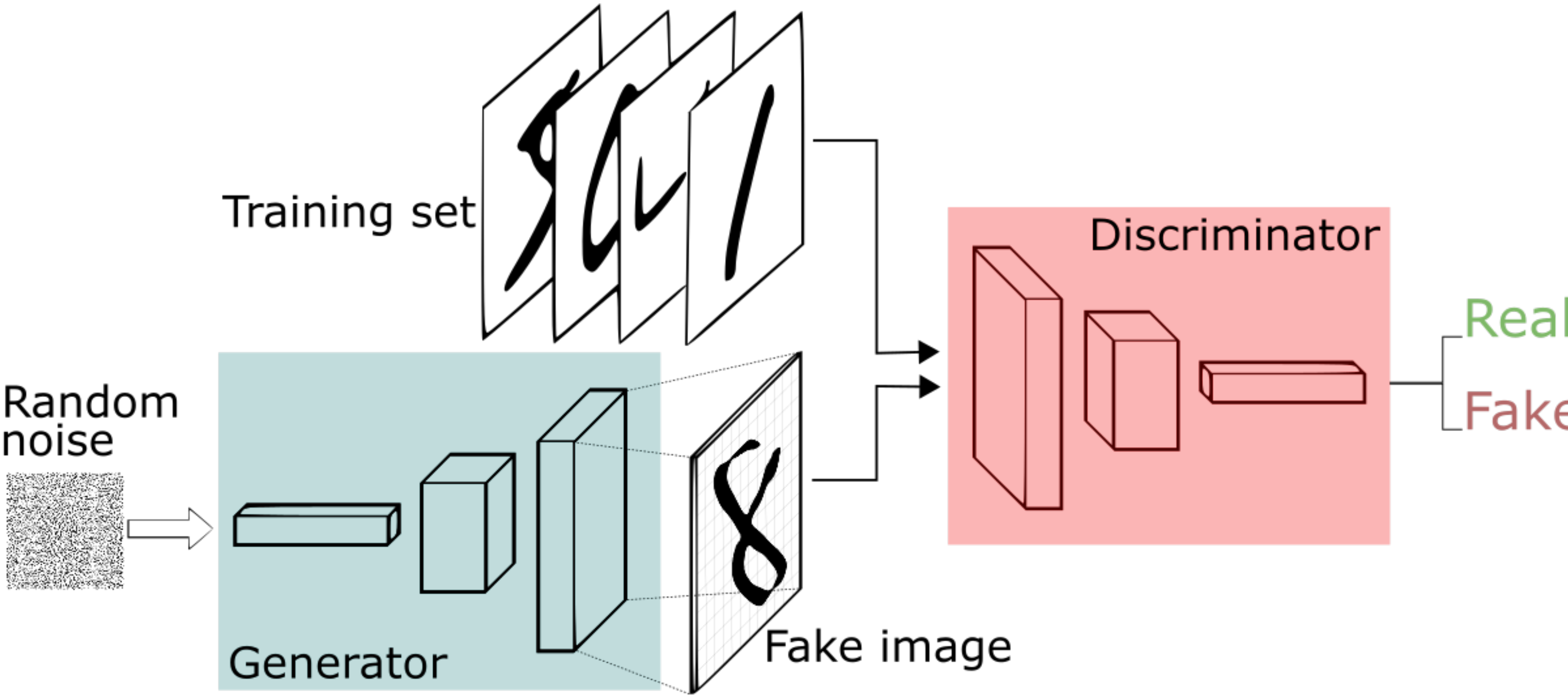
- Sample generation is simple
  - Let $x \sim p_{\text{data}}(x)$, which is the true data distribution
  - $z \sim p_z$, e.g., $z \sim \mathcal{N}(0, I) \in \mathbb{R}^{100}$
  - $G_\theta(z) = \hat{x} \sim \hat{p}_g(x)$
  - Where G is a deep neural network

How do we learn this implicit generative model? Intuition: Competitive game between two players

- Intuition: Competitive game between two players
  - Counterfeiter is trying to avoid getting caught
  - Police is trying to catch counterfeiter

- Analogy with GANs
  - Counterfeiter = Generator denoted G
  - Police = Discriminator denoted D

<u>How</u> do we learn this implicit generative model? Train two deep networks simultaneously



https://www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394/

How do we learn this implicit generative model? Intuition: Competitive game between two players

- Abstract formulation as minimax game

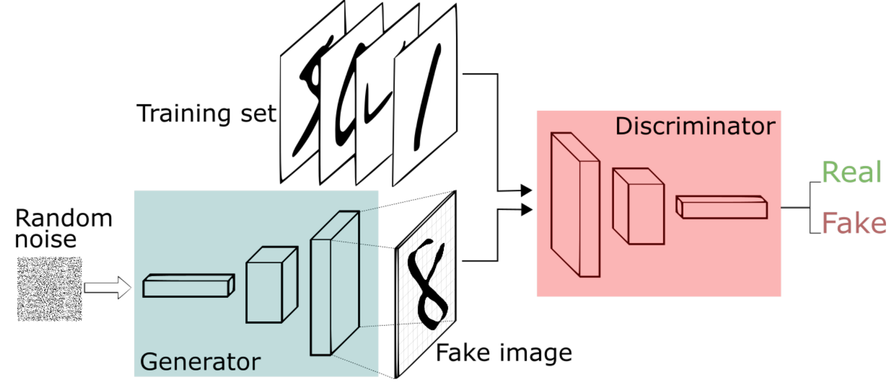
$$G^* = \arg\min_G \max_D V(D, G)$$

$$G^* = \arg\min_G \left(\max_D V(D, G)\right)$$

$$G^* = \arg\min_G C(G)$$

- Minimax: “Minimize the ***worst case*** (max) loss”
  - Counterfeiter goal: “Minimize chance of getting caught assuming the best possible police.”

The discriminator seeks to be optimal classifier



- The original GAN:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim p_z}\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$

- **Given a fixed *G***, the optimal discriminator is the optimal classifier between images

$$\mathrm{C(G)} = \max_D \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{x\sim \hat{p}_g}\left[\log\left(1 - D(x)\right)\right]$$

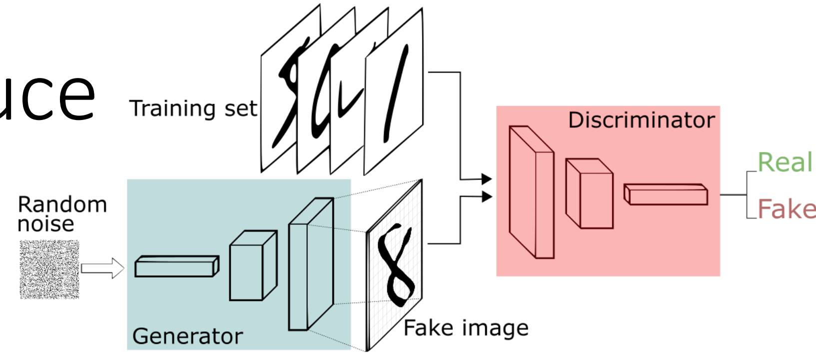
$$\mathrm{C(G)} = \max_D \mathbb{E}_{\tilde{x},\tilde{y}}\left[\tilde{y}\log D(\tilde{x}) + (1-\tilde{y})\log\left(1 - D(\tilde{x})\right)\right]$$

$$\mathrm{C(G)} = \max_D \mathbb{E}_{\tilde{x},\tilde{y}}[\log \hat{p}_D(\tilde{y}|\tilde{x})]$$

$$D^*(\tilde{x}) = p^*(\tilde{y}=1|\tilde{x}) = \frac{p_{data}(\tilde{x})}{p_{data}(\tilde{x}) + \hat{p}_g(\tilde{x})}$$

$$\text{where } \tilde{x} \sim \begin{cases} p_{\text{data}}(x) & \tilde{y} = 1 \\ \hat{p}_g(x) & \tilde{y} = 0 \end{cases}$$

The generator seeks to produce data that is like real data



- The original GAN:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim p_z}\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$

- **Given that the inner maximization is perfect**, the optimal generator G^* will generate samples that perfectly mimic the true distribution:

$$C(G) = \max_D V(D,G)$$

$$C(G) = JSD\left(p_{data}, \hat{p}_g\right) + constant$$

$$\arg\min_G C(G) = \arg\min_G JSD\left(p_{data}, \hat{p}_g\right)$$

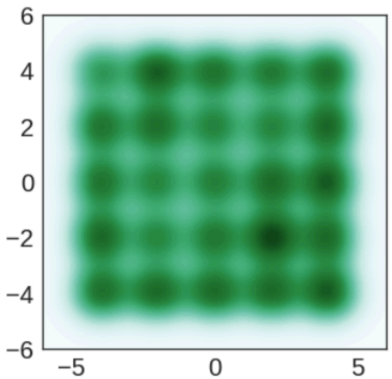
$$JSD\left(p_{data}, \hat{p}_g\right) \geq 0 \text{ and } = 0 \text{ if and only if } p_{data} = \hat{p}_g$$

Great! But wait… This theoretical analysis depends on critical assumptions

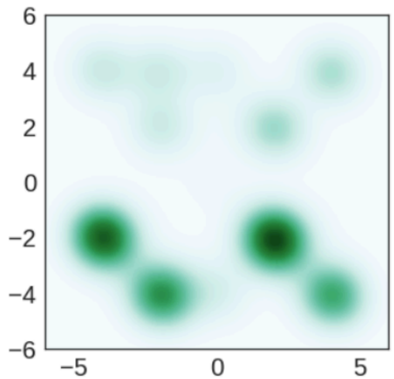
1. Assumptions on possible D and G
    1. Theory – All possible D and G
    2. Reality – Only functions defined by a neural network
2. Assumptions on optimality
    1. Theory – Both optimizations are solved perfectly
    2. Reality – The inner maximization is only solved approximately, and this interacts with outer minimization
3. Assumption on expectations
    1. Theory – Expectations over true distribution
    2. Reality – Empirical expectations over finite sample; for images, much of the high-dimensional space does not have samples

- **GANs can be very difficult/finicky to train**

Mode collapse is one of the key problems with GANs



(f) True Data



(g) GAN

http://papers.nips.cc/paper/6923-veegan-reducing-mode-collapse-in-gans-using-implicit-variational-learning.pdf

https://software.intel.com/en-us/blogs/2017/08/21/mode-collapse-in-gans

Visualize GANs

- https://poloclub.github.io/ganlab/
- DCGAN Tutorial
  https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html