Cat Face Generation using Generative Adversarial Network

Outline

- Overview
- GAN Objective
- GAN Training
- LSGAN
- Implementation

Generative adversarial networks - Overview

- Train two networks with opposing objectives:
  - **Generator:** learns to generate samples
  - **Discriminator:** learns to distinguish between generated and real samples

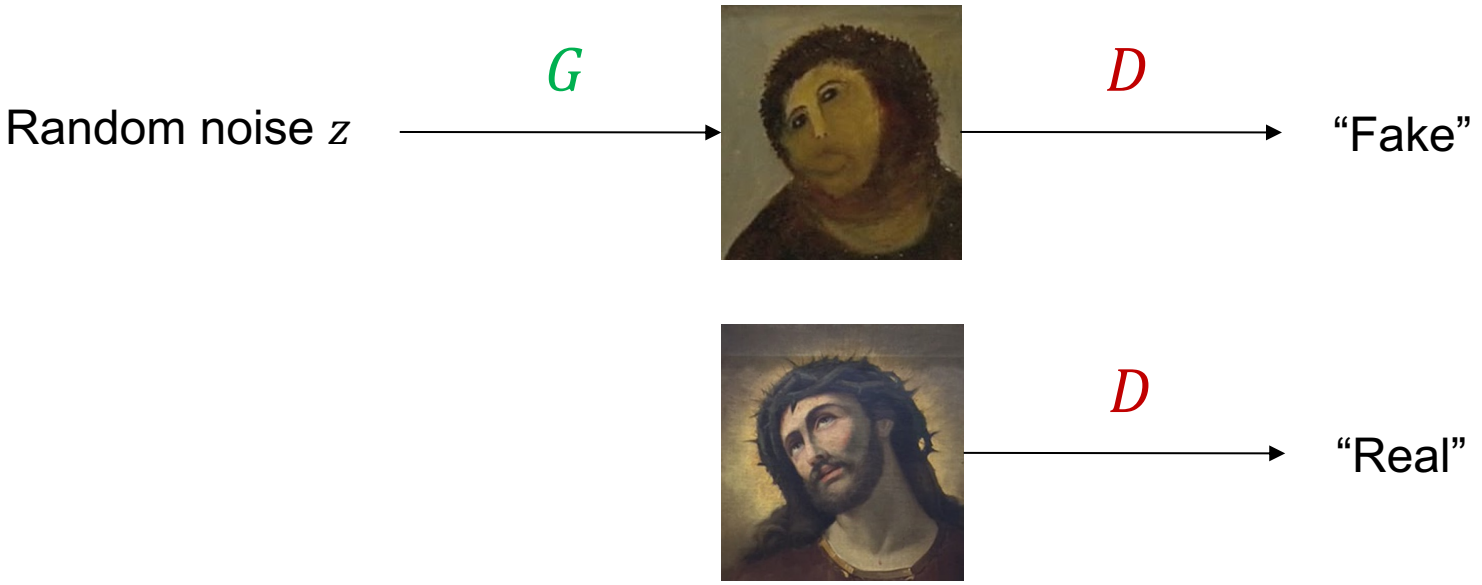
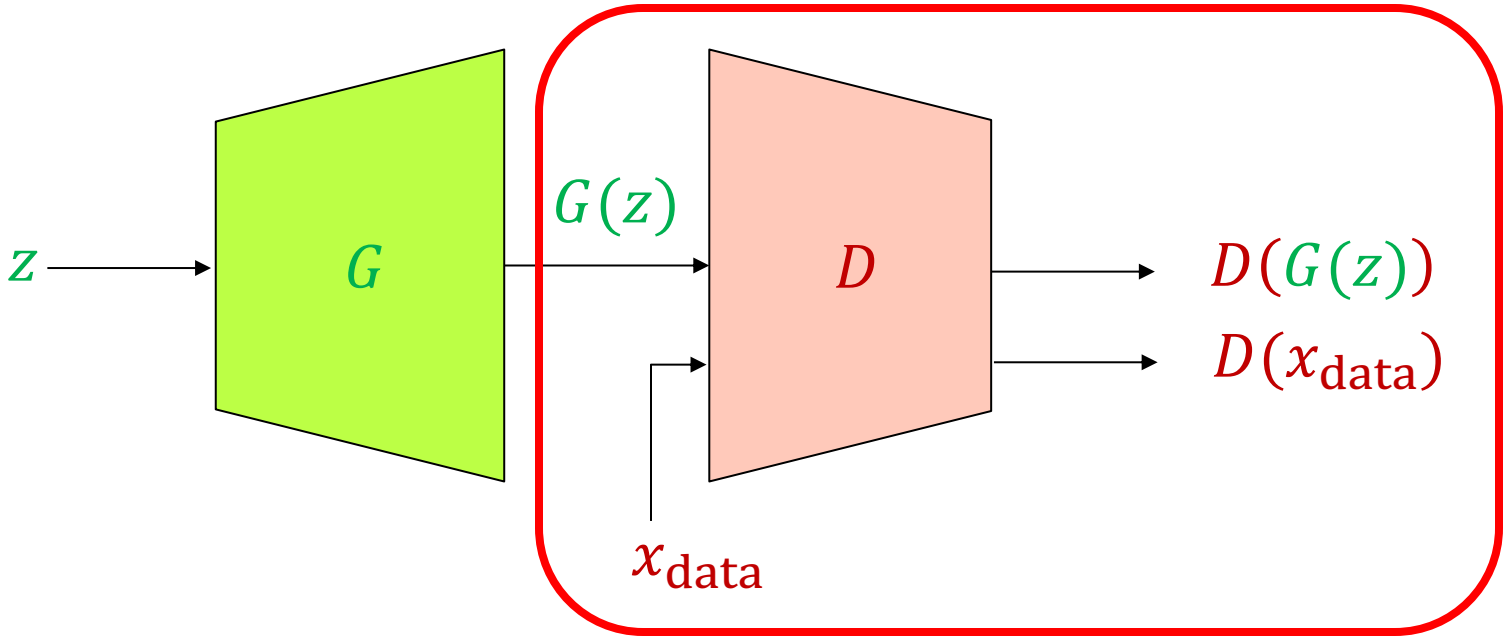


Figure adapted from F. Fleuret

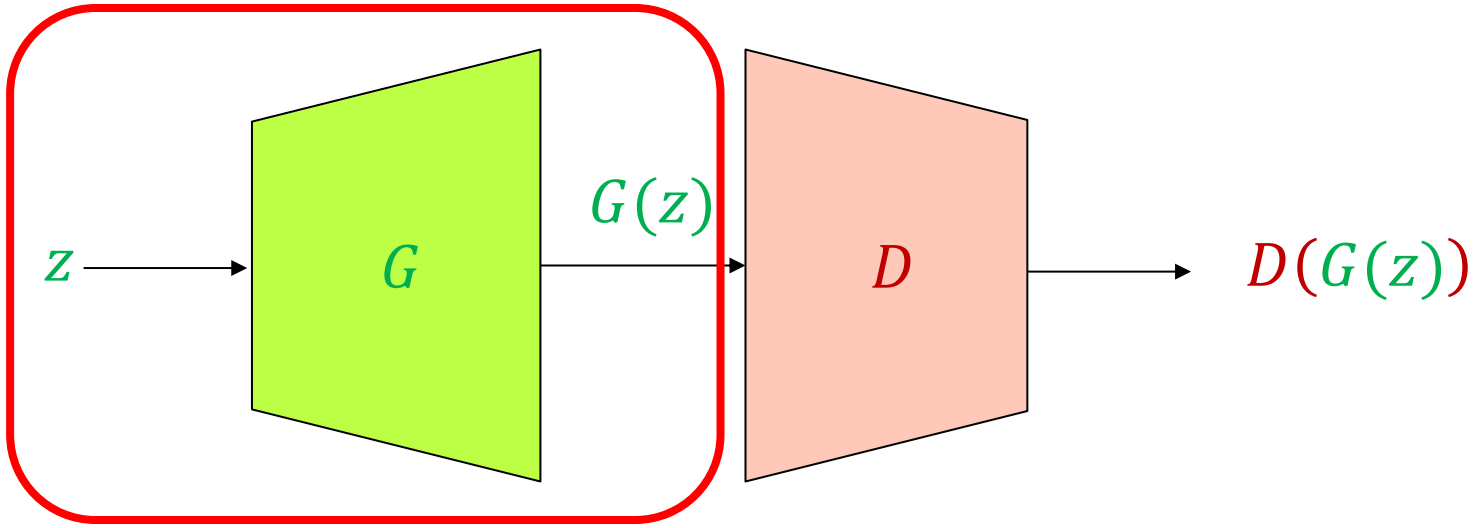
GAN: Schematic picture

- Update discriminator: push $D(x_{\text{data}})$ close to 1 and $D(G(z))$ close to 0



GAN: Schematic picture

- Update generator: increase $D(G(z))$



GAN objective

$$V(G, D) = \mathbb{E}_{x \sim p_{\text{data}}} \log D(x) + \mathbb{E}_{z \sim p} \log(1 - D(G(z)))$$

- The discriminator wants to correctly distinguish real and fake samples:

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

- The generator wants to fool the discriminator:

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

- Train the generator and discriminator jointly

GAN training

$$V(G, D) = \mathbb{E}_{x \sim p_{\text{data}}} \log D(x) + \mathbb{E}_{z \sim p} \log(1 - D(G(z)))$$

- Alternate between
  - *Gradient ascent* on discriminator:

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

  - *Gradient descent* on generator (minimize log-probability of generator samples being labeled “fake”):

$$G^* = \arg\min_G V(G, D)$$
$$= \arg\min_G \mathbb{E}_{z \sim p} \log(1 - D(G(z)))$$

  - In practice, do *gradient ascent* on generator (maximize log-probability of generator samples being labeled “real”):

$$G^* = \arg\max_G \mathbb{E}_{z \sim p} \log(D(G(z)))$$

GAN training algorithm

- Update discriminator:
  - Repeat for k steps:
    - Sample mini-batch of noise samples $z_1, \dots, z_m$ and mini-batch of real samples $x_1, \dots, x_m$
    - Update parameters of D by stochastic gradient ascent on

$$\frac{1}{m}\sum_{m}[\log D(x_m) + \log(1 - D(G(z_m)))]$$

- Update generator:
  - Sample mini-batch of noise samples $z_1, \dots, z_m$
  - Update parameters of G by stochastic gradient ascent on

$$\frac{1}{m}\sum_{m}\log D(G(z_m))$$

- Repeat until happy with results

Least Squares GAN (LSGAN)

- Use least squares cost for generator and discriminator
  - Equivalent to minimizing Pearson χ^2 divergence

$$L_D = \mathbb{E}_{x \sim p_{\text{data}}}(D(x) - 1)^2 + \mathbb{E}_{z \sim p}(D(G(z)))^2$$

Push discrim. response on real data close to 1

Push response on generated data close to 0

$$L_G = \mathbb{E}_{z \sim p}(D(G(z)) - 1)^2$$

Push response on generated data close to 1

Link to the Project Code

https://github.com/vporwal3/Cat-Face-Generation-using-Generative-Adversarial-Network-