Generative Adversarial Networks

INSTRUCTOR: HONGJIE CHEN

JUNE 27TH 2022

Generative Networks

- Typical neural networks
  - Input: data
  - Output: class for classification, or real values for prediction

- What about we design neural networks
  - Input: random vector
  - Output: data

Types of Generative Networks

- Generative Adversarial Networks (GAN)
- Boltzmann machines
- Sigmoid belief networks
- Variational autoencoders
- And more…

GAN

- An adversary as one in a game
- Train two networks
  - Generator $g(\mathbf{z}; W_g) \rightarrow \mathbf{x}$
  - Discriminator $d(\mathbf{x}; W_d) \rightarrow Pr(\mathbf{x} \text{ is real})$
- Objective function

Objective of GAN

- The definition of *real* depends on the games

- Train two networks
  - Generator $g(\mathbf{z}; W_g) \rightarrow \mathbf{x}$
  - Discriminator $d(\mathbf{x}; W_d) \rightarrow Pr(\mathbf{x} \text{ is real})$

- Objective function

$$\min_{W_g} \max_{W_d} \sum_n \log Pr(x_n \text{ is real}; W_d) + \log Pr(g(z_n; W_g) \text{ is fake}; W_d) = \min_{W_g} \max_{W_d} \sum_n \log d(x_n; W_d) + \log(1 - d(g(z_n; W_g); W_d))$$

Training GAN

- Min-max optimization
- Repeat until convergence
    - Sample $z_1, \ldots, z_N$ from $Pr(z)$
    - Sample $x_1, \ldots, x_N$ from training set
    - Update discriminator by ascending its stochastic gradient

$$\boxed{\nabla_{W_d}\left(\frac{1}{N}\sum_{n=1}^{N}\log d(x_n; W_d)\right.} + \log(1 - d(g(z_n; W_g); W_d)))$$

    - Sample $z_1, \ldots, z_N$ from $Pr(z)$
    - Update generator by descending its stochastic gradient

$$\nabla_{W_d}\left(\frac{1}{N}\sum_{n=1}^{N}\log(1 - d(g(z_n; W_g); W_d))\right)$$

Sufficient and Converged Training

- Generator will achieve true data distribution
- Discriminator give 0.5/0.5 prediction on generated samples

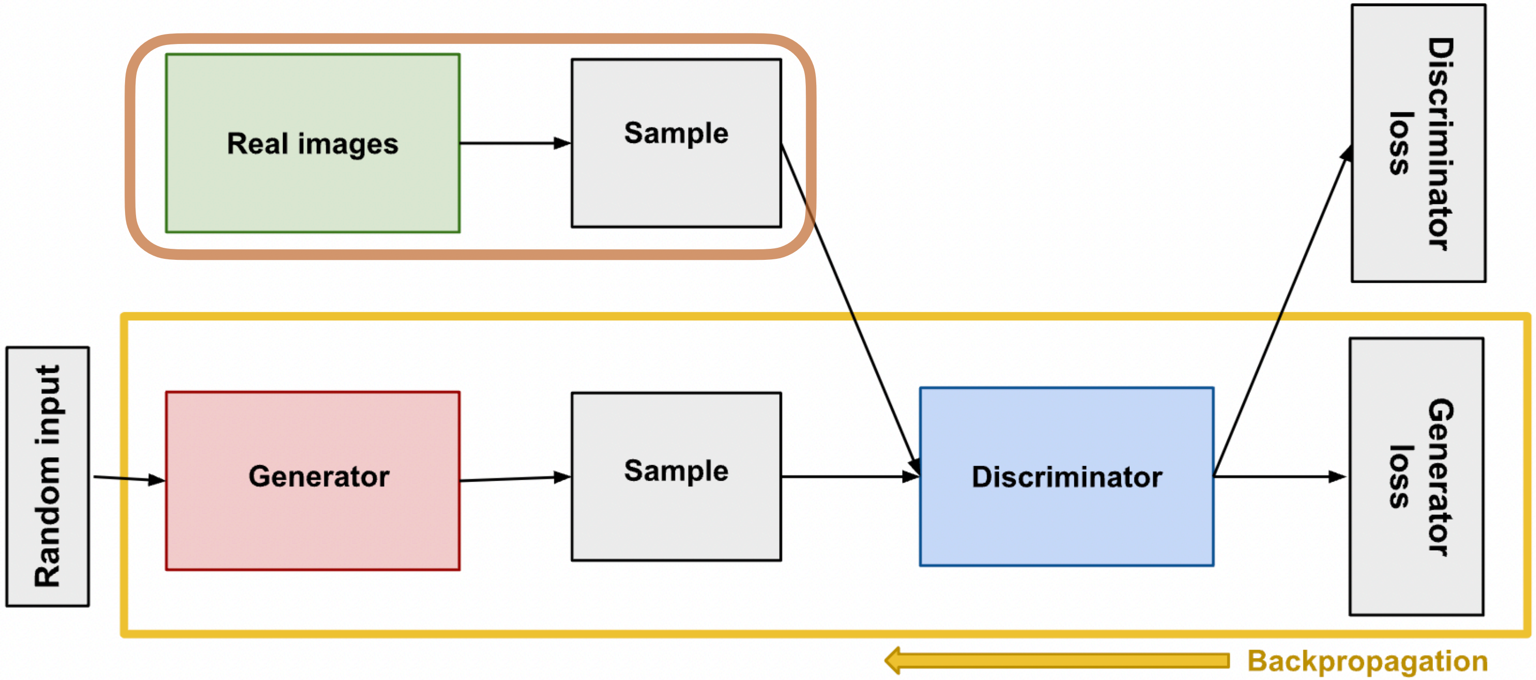

Figure: credit

Deepfake with GAN

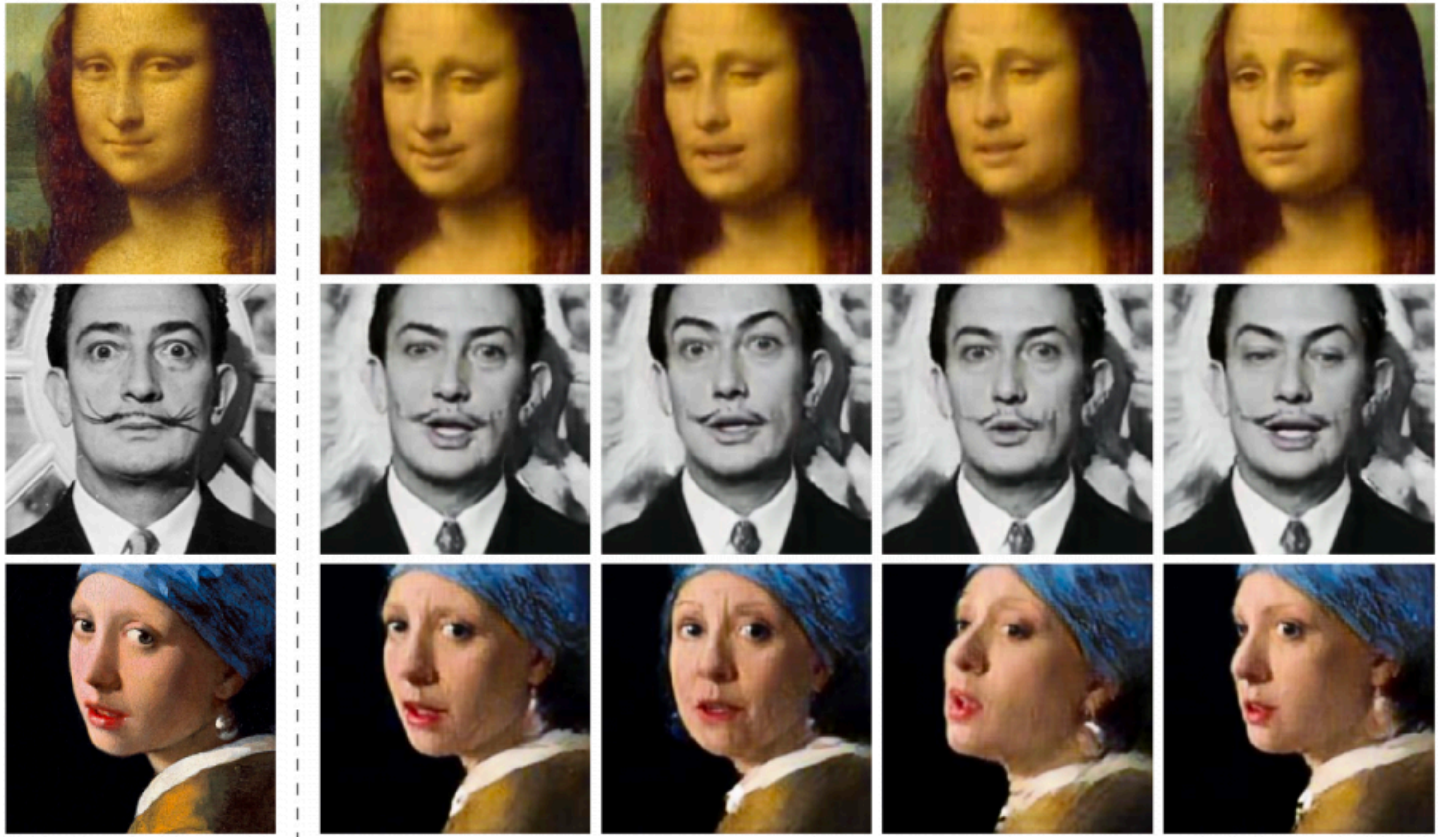

Figure: credit