CS 4824/ECE 4424: Generative Adversarial Networks

Acknowledgement:
Many of these slides are derived from Tom Mitchell, Pascal Poupart, Pieter Abbeel, Eric Eaton, Carlos Guestrin, William Cohen, and Andrew Moore.

Generative networks

- Neural networks are typically used for classification or regression
  - Input: data
  - Output: class or prediction
- Can we design neural networks that can generate data?
  - Input: random vector
  - Output: data

Generative networks

- Several types of generative networks

  - Boltzmann machines
  - Sigmoid belief networks
  - Variational autoencoders
  - **Generative adversarial networks**
  - Generative moment matching networks
  - Sum-product networks
  - Normalizing flows
  - …

Generative Adversarial Networks

- Approach based on game theory
- Two networks:
  - Generator $g(\boldsymbol{z}; \boldsymbol{W}_g) \rightarrow \boldsymbol{x}$
  - Discriminator $d(\boldsymbol{x}; \boldsymbol{W}_d) \rightarrow \Pr(\boldsymbol{x}\ is\ real)$
- Objective

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

$$= \min_{W_g} \max_{W_d} \sum_n \log d(x_n; W_d) + \log\left(1 - d(g(z_n; W_g); W_d)\right)$$

Generative Adversarial Networks

- Approach based on game theory
- Two networks:
  - Generator $g(\boldsymbol{z}; \boldsymbol{W}_g) \rightarrow \boldsymbol{x}$
  - Discriminator $d(\boldsymbol{x}; \boldsymbol{W}_d) \rightarrow \Pr(\boldsymbol{x}\ is\ real)$
- Objective

discriminator

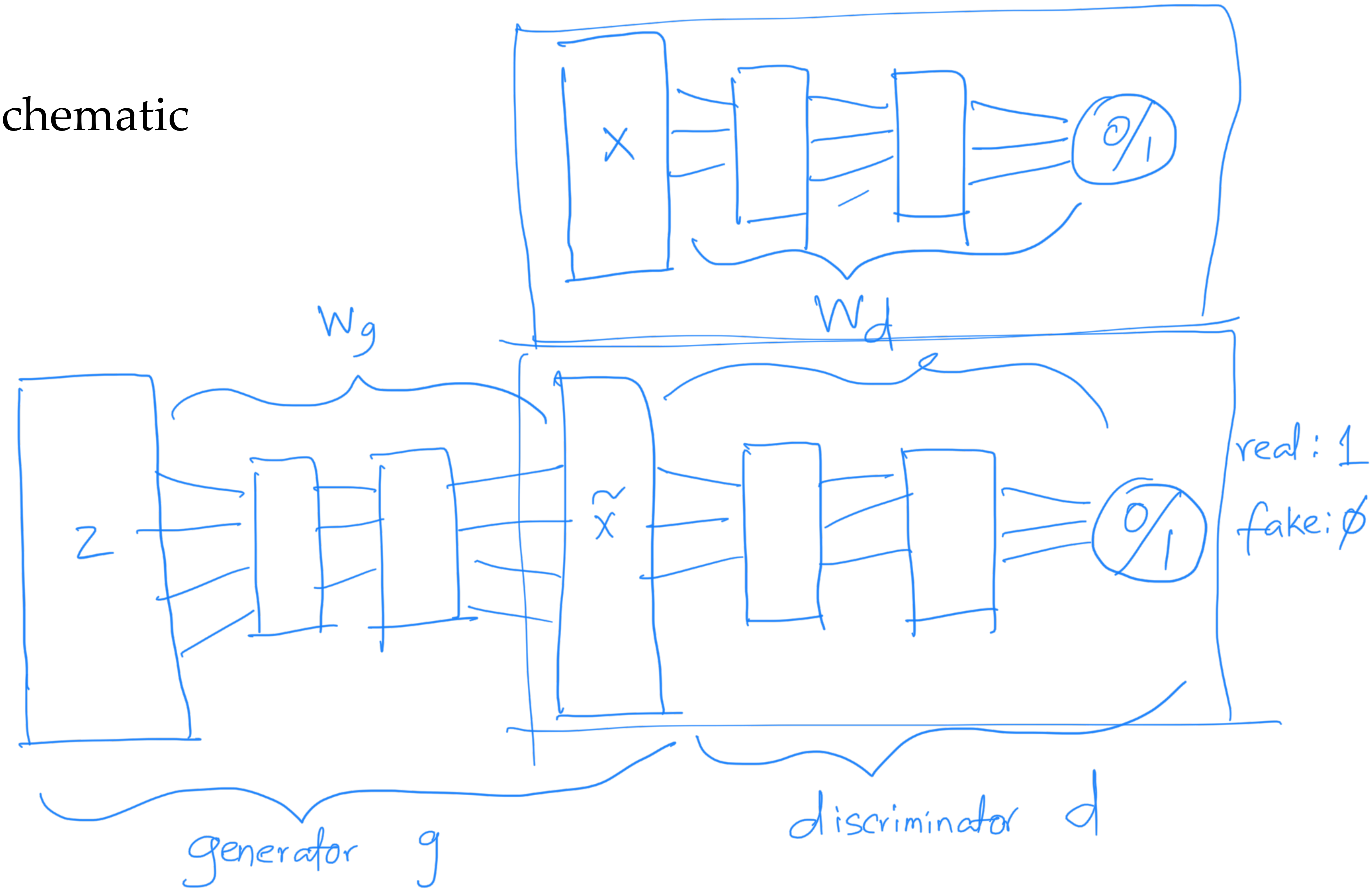
$$\min_{\boldsymbol{W}_g} \max_{\boldsymbol{W}_d} \sum_n \log \Pr(\boldsymbol{x}_n\ is\ real; \boldsymbol{W}_d) + \log \Pr\left(g\left(\boldsymbol{z}_n; \boldsymbol{W}_g\right)\ is\ fake; \boldsymbol{W}_d\right)$$

$$\equiv \min_{\boldsymbol{W}_g} \max_{\boldsymbol{W}_d} \sum_n \log d(\boldsymbol{x}_n; \boldsymbol{W}_d) + \log\left(1 - d\left(g(\boldsymbol{z}_n; \boldsymbol{W}_g); \boldsymbol{W}_d\right)\right)$$

generator

Generative Adversarial Networks

- Schematic



GAN training

- We have a min-max optimization
  - Optimize the discriminator by stochastic gradient ascent
    - Optimize the generator by stochastic gradient descent

GAN training

- Repeat until convergence
  - For k steps do
    - Sample $\mathbf{z}_1, \ldots, \mathbf{z}_N$ from $\Pr(\mathbf{z})$
    - Sample $\mathbf{x}_1, \ldots, \mathbf{x}_N$ from training set
    - Update discriminator by ascending its stochastic gradient

$$\nabla_{\boldsymbol{W}_d}\left(\frac{1}{N}\sum_{n=1}^{N}\left[\log d(\boldsymbol{x}_n;\boldsymbol{W}_d) + \log\left(1 - d\left(g\left(\boldsymbol{z}_n;\boldsymbol{W}_g\right);\boldsymbol{W}_d\right)\right)\right]\right)$$

- Sample $\mathbf{z}_1, \ldots, \mathbf{z}_N$ from $\Pr(\mathbf{z})$
- Update generator by descending its stochastic gradient

$$\nabla_{\boldsymbol{W}_g}\left(\frac{1}{N}\sum_{n=1}^{N}\log\left(1 - d\left(g\left(\boldsymbol{z}_n;\boldsymbol{W}_g\right);\boldsymbol{W}_d\right)\right)\right)$$

GAN training

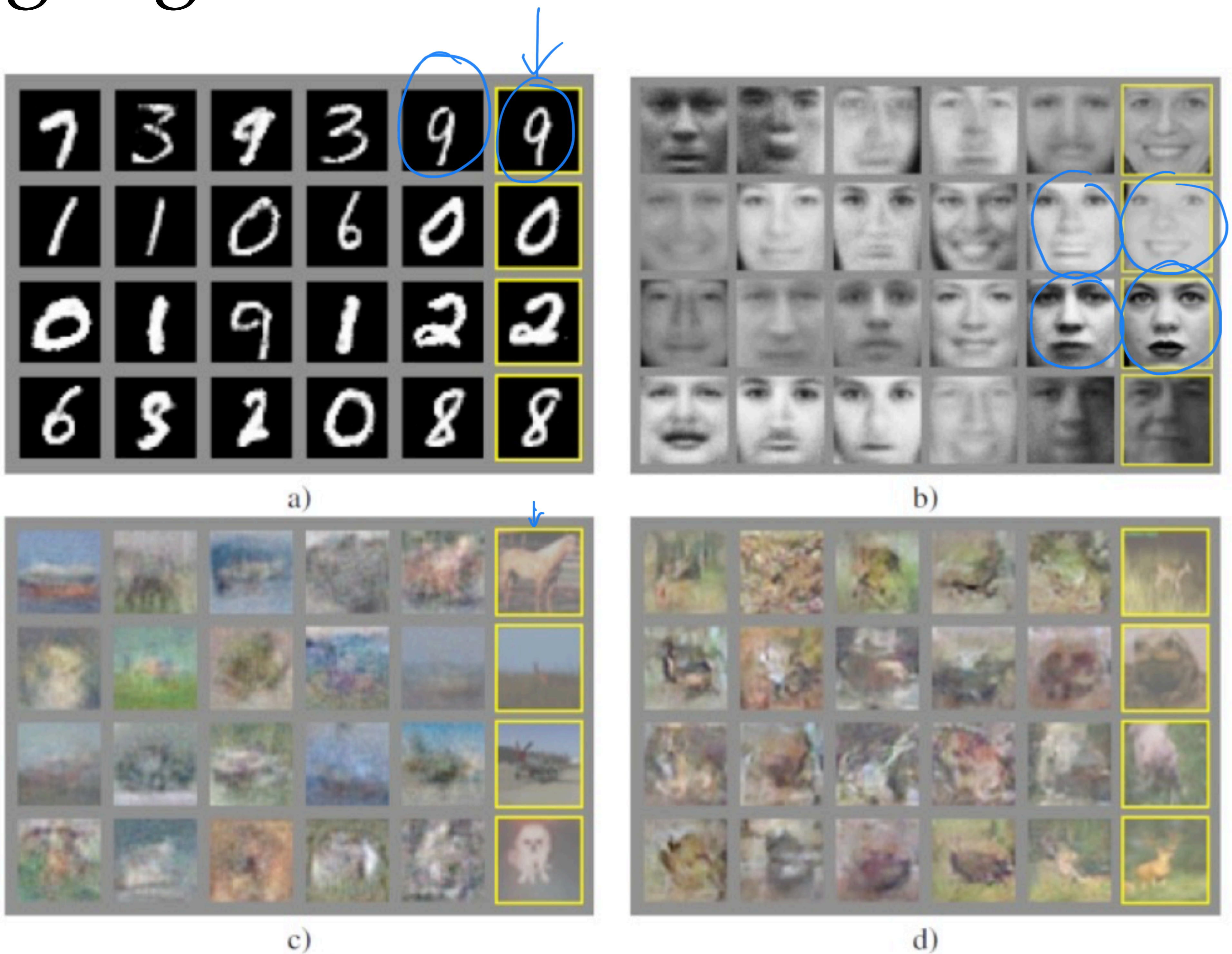
- In the limit (with sufficiently expressive networks, sufficient data and global convergence)

$$\Pr\left(\boldsymbol{x} \middle| \boldsymbol{z}; \boldsymbol{W}_g\right) \rightarrow \textit{true data distribution}$$

$$\Pr(\boldsymbol{x}\ is\ real; \boldsymbol{W}_d) \rightarrow 0.5 \text{ (for real and fake data)}$$

- Problems in practice:
  - Imbalance: one network may dominate the other
  - Local convergence

Images generated with GANs training



◦ Right columns are nearest neighbour training examples of adjacent column