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 \( g(z; W_g) \to x \)
  - Discriminator \( d(x; W_d) \to \Pr(x \text{ is real}) \)

- Objective
Generative Adversarial Networks

- Approach based on game theory

- Two networks:
  - Generator \( g(z; W_g) \rightarrow x \)
  - Discriminator \( d(x; W_d) \rightarrow \Pr(x \text{ is real}) \)

- Objective

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

\[
\equiv \min_{W_g} \max_{W_d} \sum_n \log d(x_n; W_d) + \log \left(1 - d(g(z_n; W_g); W_d)\right)
\]
Generative Adversarial Networks

- 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 $z_1, \ldots, z_N$ from $Pr(z)$
    - Sample $x_1, \ldots, x_N$ from training set
    - Update discriminator by ascending its stochastic gradient
      \[
      \nabla_{W_d} \left( \frac{1}{N} \sum_{n=1}^{N} \left[ \log d(x_n; W_d) + \log \left(1 - d(g(z_n; W_g); W_d)\right) \right] \right)
      \]
  - Sample $z_1, \ldots, z_N$ from $Pr(z)$
  - Update generator by descending its stochastic gradient
    \[
    \nabla_{W_g} \left( \frac{1}{N} \sum_{n=1}^{N} \log \left(1 - d(g(z_n; W_g); W_d)\right) \right)
    \]
GAN training

- In the limit (with sufficiently expressive networks, sufficient data and global convergence)
  \[ \Pr(x|z; W_g) \rightarrow \text{true data distribution} \]
  \[ \Pr(x \text{ is real}; W_d) \rightarrow 0.5 \text{ (for real and fake data)} \]

- 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