Background and Motivation

Generative Adversarial Nets (GANs)

Given samples from \( p(x) \), to learn \( p_g(x) \approx p(x) \). \( D(x) \) outputs a binary unit.

\[
Z \sim p(z) \quad \rightarrow \quad G \quad X \sim p(X) \quad \rightarrow \quad D \quad A/R
\]

Under nonparametric assumptions, the global optimum is \( p_g(x) = p(x) \).

Semi-Supervised GANs

Given partially labeled data from \( p(x, y) \), to learn \( p_c(y|x) \) and \( p_g(x) \). \( D(x) \) outputs \(|Y| \) or \(|Y| + 1 \) dimensional units.

No game-theoretical analysis as in original GANs.

Problems of Two-Player Formulation

- An extreme case, where \( G \) is good but \( C \) is poor \([1]\)
- \( G \) converges, i.e. \( p_g(x) = p(x) \)
- \( D \) rejects \( x \sim p_g(x) \) with probability \( \downarrow \)
- \( C \) should accept it and classify it as certain class confidently
- Then \( C \) and \( G \) won’t be both optimal because of sharing parameters between \( D \) and \( C \)
- Cannot control what to generate
- The classifier predicts accurately while the generator cannot leverage such information

Triple-GAN

Rethink the Problem

- A dual perspective: \( p(x, y) = p(y)p(x|y) = p(x)p(y|x) \)
- Marginal distributions are known
- Conditional distributions are of interests

Three-Player Game

- Given partially labeled data, learn \( p_c(y|x) \) and \( p_g(x|y) \)

\[
X \sim p(x) \quad \rightarrow \quad (X, Y) \sim p(X, Y) \quad \rightarrow \quad (X, Y) \sim p(X, Y) \quad \rightarrow \quad G \quad Z \sim p_z(z) \quad \rightarrow \quad Y \sim p(y)
\]

where \( D(x) \) outputs a binary unit to justify \((x, y)\) pairs. A minimax game:

\[
\min_{C,G} \max_D \mathbb{E}_p[\log D(x, y)] + \alpha \mathbb{E}_p[\log (1 - D(x, y))] + (1 - \alpha) \mathbb{E}_p[\log (1 - D(x, y))].
\]

Game-Theoretical Analysis and Regularizations

- The optimum is \( p(x, y) = (1 - \alpha)p_g(x, y) + \alpha p_c(x, y) \).

Experiments

Classification

State-of-the-art classification results on MNIST, SVHN and CIFAR10.

Generation

Good \( C \) results in good \( G \)

Conditional Generation

Disentangle Classes and Styles

Figure 1: Same \( y \) for each row. Same \( z \) for each column.

Latent Space Interpolation