Overview

1. GANs
2. Problems with training GANs
3. Lipschitz Discriminators & VRAL
4. Empirical Results
Generative Adversarial Networks

Architecture:

Value Function:

What’s really happening:

\[ \min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[ \log D(x) \right] + \mathbb{E}_{z \sim p_z(z)} \left[ \log (1 - D(G(z))) \right] \]

Goodfellow et. al, 2014
Generative Adversarial Networks

Original

Reconstructions

Dumoulin et. al, 2016
Problems with Training GANs

1. Mode Collapse

Salimans et. al, 2016; Che et. al, 2016
Problems with Training GANs

2. Problematic Gradients
Problems with Training GANs

2. Problematic Gradients

\[ \| \nabla D \| \text{ vanishes!} \]

\[ \| \nabla D \| \text{ explodes!} \]
The Lipschitz Constraint

A function \( f : X \rightarrow \mathbb{R} \) is \( K \)-Lipschitz if for every \( x_1, x_2 \in X \), \( f \) satisfies

\[
\frac{||f(x_1) - f(x_2)||}{||x_1 - x_2||} \leq K
\]
The Lipschitz Constraint

A function $f : X \to \mathbb{R}$ is $K$-Lipschitz if for every $x_1, x_2 \in X$, $f$ satisfies

$$\frac{||f(x_1) - f(x_2)||}{||x_1 - x_2||} \leq K$$

1-Lipschitz Discriminator:
Wasserstein GAN

• Real data lies on a low-dimensional manifold in a high-dimensional space, $P_r$

• Jenson-Shannon and KL divergences are not meaningful

• Use Wasserstein-1, or “earth-mover’s” objective instead

Arjovsky et. al, 2017
Wasserstein GAN

- Objective: \[ W(P_r, P_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{x \sim P_\theta}[f(x)] \]

- Use weight clipping to enforce Lipschitz constraint

Wasserstein GAN

Standard GAN

Arjovsky et. al, 2017
Another Interpretation: Variance

non-Lipschitz:

Lipschitz:
Variance regularized GANs: meta-discriminators

- 3 adversarial games:
  1. $G$ tries to fool $D$ by creating real-looking samples
  2. $D$ tries to fool $R$ by mimicking $\mathcal{N}(1,1)$ for real samples
  3. $D$ tries to fool $F$ by mimicking $\mathcal{N}(0,1)$ for fake samples
**Variance regularized GANs:** meta-discriminators

**Objective #1:**

\[
\min_G \mathbb{E}_{z \sim p_z}[(D(G(z)) - 1)^2] \text{ or } \min_G \left( \mathbb{E}_{z \sim p_z}[D(G(z))] - \mathbb{E}_{x \sim p_{data}}[D(x)] \right)^2
\]

**Objective #2:**

\[
\min_D \max_F V_F(F, D, G) = \mathbb{E}_{y \sim \mathcal{N}(0, 1)}[\log F(y)] + \mathbb{E}_{z \sim p_z}[\log (1 - F(D(G(z))))]
\]

**Objective #3:**

\[
\min_D \max_R V_R(R, D) = \mathbb{E}_{y \sim \mathcal{N}(1, 1)}[\log R(y)] + \mathbb{E}_{x \sim p_{data}}[\log (1 - R(D(x)))]
\]
Variance regularized GANs: density estimators

- Minimize the KL-divergence between $D$’s normalized output distribution and $\mathcal{N}(0,1)$ or $\mathcal{N}(1,1)$

- Use a parzen-window density estimator to approximate $D$’s normalized output distribution, $\tilde{p}_D$

$$
\mathcal{L}_D = \mathbb{E}_{z \sim p_z}[KL(\mathcal{N}(0,1)||\tilde{p}_D(G(z))))] + \mathbb{E}_{x \sim p_{data}}[KL(\mathcal{N}(1,1)||\tilde{p}_D(x))]
$$

- fit $D$’s output given fake samples to a gaussian
- fit $D$’s output given real samples to a gaussian
Well-trained Discriminators

Standard GAN  Least-Squares GAN  VRAL, meta-disc.

1:1

50:1
Why Large Training Ratios

- The more the discriminator is trained, the more reliable the learning signal

- If the discriminator becomes too strong, the generator may not learn at all

- **Goal:** ensure training methods are robust against large training ratios (e.g. 50 discriminator updates per generator update)
Learning & $D$ output

- Standard
- Standard, $-\log D$ loss
- Least Squares
- Wasserstein
- VRAL, meta-disc.
- VRAL, parzen window

50:1
Learning & Gradients

- Standard
- Standard, \(-\log D\) loss
- Least Squares
- Wasserstein
- VRAL, meta-disc.
- VRAL, parzen window

50:1

\(|\nabla D|\)

\(\text{Var} D(G(z))\)
Learning & Gradients

Standard

Least Squares

VRAL, meta-disc.

50:1
Alternatives to Weight Clipping

- Add noise to intermediate layers of the discriminator to promote variance

- Weight clipping leads to unstable gradient norms, instead use the following gradient penalty:

\[
\mathbb{E}_{x \sim \hat{p}_x}[(\|\nabla_x D(x)\| - 1)^2],
\]

\[
\hat{p}_x(x) = (1 - t) \cdot p_{data}(x) + t \cdot p_g(x)
\]

Salimans et. al, 2016; Gulrajani et. al, 2017
Conclusion

1. Overly strong discriminators emit vanishing gradients, making it difficult for the generator to learn

2. The Lipschitz constraint can be interpreted as forcing the discriminator to have variance in its output

3. Mode-matching allows a generator to learn the data distribution/manifold in the presence of a well-trained discriminator
Future Work

1. Are we actually enforcing a Lipschitz function by regularizing the discriminator?

2. Why our proposed method fails under certain choices of hyperparameters
Collaborators

Devon Hjelm

Yoshua Bengio