Tips for Improving GAN

Martin Arjovsky, Soumith Chintala, Léon Bottou, Wasserstein GAN, arXiv prepring, 2017
JS divergence is not suitable

• In most cases, $P_G$ and $P_{data}$ are not overlapped.
• 1. The nature of data
   Both $P_{data}$ and $P_G$ are low-dim manifold in high-dim space.
   The overlap can be ignored.
• 2. Sampling
   Even though $P_{data}$ and $P_G$ have overlap.
   If you do not have enough sampling ......
What is the problem of JS divergence?

JS divergence is log2 if two distributions do not overlap.

Intuition: If two distributions do not overlap, binary classifier achieves 100% accuracy.

Same objective value is obtained. Same divergence.
Least Square GAN (LSGAN)

• Replace sigmoid with linear (replace classification with regression)

They don’t move.
Wasserstein GAN (WGAN): Earth Mover’s Distance

- Considering one distribution $P$ as a pile of earth, and another distribution $Q$ as the target
- The average distance the earth mover has to move the earth.

$W(P, Q) = d$
There many possible “moving plans”.

Using the “moving plan” with the smallest average distance to define the earth mover’s distance.

Source of image: https://vincentherrmann.github.io/blog/wasserstein/
WGAN: Earth Mover’s Distance

There many possible “moving plans”.

Using the “moving plan” with the smallest average distance to define the earth mover’s distance.

Source of image: https://vincentherrmann.github.io/blog/wasserstein/
A “moving plan” is a matrix. The value of the element is the amount of earth from one position to another.

Average distance of a plan $\gamma$:

$$B(\gamma) = \sum_{x_p, x_q} \gamma(x_p, x_q) \| x_p - x_q \|$$

Earth Mover’s Distance:

$$W(P, Q) = \min_{\gamma \in \Pi} B(\gamma)$$

The best plan...
Why Earth Mover’s Distance?

\[ D_f(P_{\text{data}} || P_G) \]

\[ W(P_{\text{data}}, P_G) \]

\[ JS(P_{G_0}, P_{\text{data}}) = \log 2 \]

\[ W(P_{G_0}, P_{\text{data}}) = d_0 \]

\[ JS(P_{G_50}, P_{\text{data}}) = \log 2 \]

\[ W(P_{G_50}, P_{\text{data}}) = d_{50} \]

\[ JS(P_{G_{100}}, P_{\text{data}}) = 0 \]

\[ W(P_{G_{100}}, P_{\text{data}}) = 0 \]
WGANG

Evaluate wasserstein distance between $P_{data}$ and $P_G$

$$V(G, D) = \max_{D \in 1-Lipschitz} \{E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)]\}$$

D has to be smooth enough.

Without the constraint, the training of D will not converge.

Keeping the D smooth forces D(x) become $\infty$ and $-\infty$
WGAN

Evaluate wasserstein distance between $P_{data}$ and $P_G$

$$V(G,D) = \max_{D \in 1-Lipschitz} \{E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)]\}$$

D has to be smooth enough. How to fulfill this constraint?

Lipschitz Function

$$\|f(x_1) - f(x_2)\| \leq K \|x_1 - x_2\|$$

Output change

Input change

K=1 for "1 – Lipschitz" Do not change fast

Weight Clipping [Martin Arjovsky, et al., arXiv, 2017]

Force the parameters w between c and -c

After parameter update, if $w > c$, $w = c$;

if $w < -c$, $w = -c$
**Improved WGAN (WGAN-GP)**

\[ V(G, D) = \max_{D \in 1-Lipschitz} \{ E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)] \} \]

A differentiable function is 1-Lipschitz if and only if it has gradients with norm less than or equal to 1 everywhere.

\[ D \in 1 - Lipschitz \iff \|\nabla_x D(x)\| \leq 1 \text{ for all } x \]

\[ V(G, D) \approx \max_D \{ E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)] \] \\
\[ \quad - \lambda \int_x \max(0, \|\nabla_x D(x)\| - 1)dx \} \]

Prefer \( \|\nabla_x D(x)\| \leq 1 \) for all \( x \)

\[ -\lambda E_{x \sim P_{penalty}}[\max(0, \|\nabla_x D(x)\| - 1)] \]

Prefer \( \|\nabla_x D(x)\| \leq 1 \) for \( x \) sampling from \( x \sim P_{penalty} \)
Improved WGAN (WGAN-GP)

\[ V(G, D) \approx \max_D \{ E_{x \sim P_{data}} [D(x)] - E_{x \sim P_G} [D(x)] \} \]

\[ -\lambda E_{x \sim P_{penalty}} [\max(0, \|\nabla_x D(x)\| - 1)] \}

“Given that enforcing the Lipschitz constraint everywhere is intractable, enforcing it **only along these straight lines** seems sufficient and experimentally results in good performance.”

Only give gradient constraint to the region between \( P_{data} \) and \( P_G \) because they influence how \( P_G \) moves to \( P_{data} \)
Simply penalizing overly large gradients also works in theory, but experimentally we found that this approach converged faster and to better optima.
Spectrum Norm

Spectral Normalization → Keep gradient norm smaller than 1 everywhere [Miyato, et al., ICLR, 2018]
Algorithm of WGAN

In each training iteration:

- Sample m examples \( \{x^1, x^2, \ldots, x^m\} \) from data distribution \( P_{data}(x) \)
- Sample m noise samples \( \{z^1, z^2, \ldots, z^m\} \) from the prior \( P_{prior}(z) \)
- Obtaining generated data \( \{\tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^m\} \), \( \tilde{x}^i = G(z^i) \)
- Update discriminator parameters \( \theta_d \) to maximize
  \[ \tilde{V} = \frac{1}{m} \sum_{i=1}^{m} D(x^i) - \frac{1}{m} \sum_{i=1}^{m} D(\tilde{x}^i) \]
  \[ \theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d) \]
- Sample another m noise samples \( \{z^1, z^2, \ldots, z^m\} \) from the prior \( P_{prior}(z) \)
- Update generator parameters \( \theta_g \) to minimize
  \[ \tilde{V} = \frac{1}{m} \sum_{i=1}^{m} \log D(x^i) - \frac{1}{m} \sum_{i=1}^{m} D(G(z^i)) \]
  \[ \theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g) \]

No sigmoid for the output of D
Weight clipping / Gradient Penalty ...

Repeat \( k \) times
Energy-based GAN (EBGAN)

- Using an autoencoder as discriminator $D$
  - Using the negative reconstruction error of auto-encoder to determine the goodness
  - **Benefit**: The auto-encoder can be pre-train by real images without generator.

[Junbo Zhao, et al., arXiv, 2016]
EBGAN

Auto-encoder based discriminator only gives limited region large value.

0 is for the best.

Do not have to be very negative

Hard to reconstruct, easy to destroy
Outlook:
Loss-sensitive GAN (LSGAN)
Reference

• Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative Adversarial Networks, NIPS, 2014


• Martin Arjovsky, Soumith Chintala, Léon Bottou, Wasserstein GAN, arXiv, 2017

• Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville, Improved Training of Wasserstein GANs, NIPS, 2017

• Junbo Zhao, Michael Mathieu, Yann LeCun, Energy-based Generative Adversarial Network, arXiv, 2016

• Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, Olivier Bousquet, “Are GANs Created Equal? A Large-Scale Study”, arXiv, 2017

• Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen Improved Techniques for Training GANs, NIPS, 2016
Reference

- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Sepp Hochreiter, GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, NIPS, 2017
- Xiang Wei, Boqing Gong, Zixia Liu, Wei Lu, Liqiang Wang, Improving the Improved Training of Wasserstein GANs: A Consistency Term and Its Dual Effect, ICLR, 2018
- Takeru Miyato, Toshiki Kataoka, Masanori Koyama, Yuichi Yoshida, Spectral Normalization for Generative Adversarial Networks, ICLR, 2018