# Wasserstein GAN (WGAN)

Behrooz Azarkhalili[1]

[1]Life Language Processing Lab, University of California, Berkeley
[1]azarkhalili@behrooz.tech

## Introduction

Wasserstein GAN (WGAN) is an improvement over the original Generative Adversarial Networks (GANs), designed to solve issues like training instability and mode collapse. WGAN replaces the Jensen-Shannon (JS) divergence used in the original GAN with the Wasserstein distance (also known as Earth Mover's Distance) to measure the distance between the real data distribution and the generated data distribution. This change improves training stability and reduces the likelihood of mode collapse.

## Generative Adversarial Networks (GANs)

GANs consist of two neural networks:

- **Generator** $G$: Takes random noise $z$ as input and generates synthetic data.
- **Discriminator** $D$: Distinguishes between real data from the dataset and fake data generated by $G$.

The original GAN objective is:

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

## Wasserstein Distance

The Wasserstein distance measures the cost of transforming one probability distribution into another. Formally, for two distributions $P_r$ (real) and $P_g$ (generated), the Wasserstein distance is:

$$W(P_r, P_g) = \inf_{\gamma\in\Pi(P_r,P_g)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma}[\|\mathbf{x} - \mathbf{y}\|]$$

where $\Pi(P_r, P_g)$ is the set of joint distributions whose marginals are $P_r$ and $P_g$. This distance provides smoother gradients and reduces mode collapse.

## Lipschitz Continuity and the Critic

To compute the Wasserstein distance, the Critic $C$ (replacing $D$) must satisfy the 1-Lipschitz condition:

$$|C(\mathbf{x}_1) - C(\mathbf{x}_2)| \leq \|\mathbf{x}_1 - \mathbf{x}_2\|$$

This ensures the critic remains within the bounds necessary for Wasserstein distance computation.

## Weight Clipping and Gradient Penalty

To enforce the Lipschitz constraint, WGAN originally used weight clipping. However, weight clipping can lead to poor convergence. An alternative approach, Gradient Penalty (WGAN-GP), was introduced:

$$L_C = \mathbb{E}_{\mathbf{x} \sim P_r}[C(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{z}}}[C(G(\mathbf{z}))] + \lambda \mathbb{E}_{\hat{\mathbf{x}} \sim P_{\hat{\mathbf{x}}}}[(\|\nabla_{\hat{\mathbf{x}}} C(\hat{\mathbf{x}})\|_2 - 1)^2]$$

where $\lambda$ is the penalty coefficient and $\hat{\mathbf{x}}$ is interpolated between real and fake samples.

## Loss Functions for WGAN

The critic's loss is:

$$L_C = \mathbb{E}_{\mathbf{x} \sim P_r}[C(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{z}}}[C(G(\mathbf{z}))]$$

The generator's loss is:

$$L_G = -\mathbb{E}_{\mathbf{z} \sim P_{\mathbf{z}}}[C(G(\mathbf{z}))]$$

## Conclusion

WGAN improves GAN training by using the Wasserstein distance, offering more stable gradients and reducing the likelihood of mode collapse. With the addition of Gradient Penalty, WGAN-GP enhances the model's performance, especially in enforcing the Lipschitz constraint, making it a powerful approach for generative modeling.