Generative Lecture 5: Generative Adversarial Networks (GAN)

Asrl · 2026年11月18日 · generative

This note introduces GANs through distribution matching and adversarial training, then explains Wasserstein distance, WGAN, and the use of weight clipping and gradient penalties.

原文：https://asrl.top/posts/generative-gan



文章封面

Generative Adversarial Networks (GAN)

1. f-divergence

Definition

Let $f:(0,\infty)\to\mathbb{R}$ be a convex function with $f(1)=0$. We define:

$$D_f(P\|Q)=\int q(x)f\left(\frac{p(x)}{q(x)}\right)\,dx.$$

This divergence measures the difference between two distributions P and Q.

Examples

When $f(t) = t \log t$, we have the KL divergence:

$$D_f(P\|Q) = \int p(x) \log \frac{p(x)}{q(x)}\, dx = D_{\mathrm{KL}}(P\|Q).$$

When $f(t) = \frac{1}{2}|t-1|$, we have the total variation distance:

$$D_f(P\|Q) = \frac{1}{2}\int |p(x) - q(x)|\, dx = \mathrm{TV}(P, Q).$$

When $f(t) = \frac{1}{2}\left[t \log t - (t+1)\log \frac{t+1}{2}\right]$, we have the Jensen-Shannon (JS) divergence:

$$D_f(P\|Q) = \frac{1}{2} D_{\mathrm{KL}}(P\|M) + \frac{1}{2} D_{\mathrm{KL}}(Q\|M) = \mathrm{JS}(P\|Q), \qquad M = \frac{P+Q}{2}.$$

2. GAN

From JS divergence to a discriminator

With an optimal discriminator, the original GAN minimax objective is equivalent to minimizing the JS divergence between the model and real data distributions.

During training, our goal is to minimize $\mathrm{JS}(P^*\|P_\theta)$, but it is difficult to evaluate $P^*(x)$ and $P_\theta(x)$ directly.

$$D^*(x) = \frac{P^*(x)}{P^*(x) + P_\theta(x)} \in [0, 1].$$

The ratio lies between 0 and 1 and, with equal class priors, represents the probability that x is real. We can train a binary classifier to estimate it.

Adversarial training

We can reformulate the objective as a minimax problem:

$$\min_G \max_D V(D, G), \qquad V(D, G) = \mathbb{E}_{x\sim P^*}[\log D(x)] + \mathbb{E}_{z\sim P_z}[\log(1 - D(G(z)))].$$

$$\max_D V(D, G) = 2\,\mathrm{JS}(P^*\|P_\theta) - \log 4.$$

Finally, we train both $D_{\theta'}$, the discriminator, and G_θ, the generator. While training $D_{\theta'}$, we maximize $V(D, G)$; while training G_θ, we minimize it, keeping the other network fixed. This is known as adversarial training.

Counterfeiters and police

An analogy from the original GAN paper helps explain this process. Counterfeiters try to make fake money look real, while the police try to distinguish fake money from real money. The generator plays the role of the counterfeiters, and the discriminator plays the role of the police. The goal is to produce counterfeits that are indistinguishable from real money.

Training difficulties

GAN training is generally nonconvex–nonconcave and sensitive to the balance between the two networks. Mode collapse occurs when the generator produces samples from only a few modes of the data distribution.

3. Wasserstein distance

Intuition: moving probability mass

The Wasserstein-1 distance is the minimum cost of moving probability mass to transform one distribution into another. Moving mass m from x to y costs $m\|x-y\|_2$:

$$W_1(P,Q) = \inf_{\gamma\in\Gamma(P,Q)} \mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|_2]\,.$$

Here, $\Gamma(P,Q)$ is the set of transport plans with marginals P and Q.

A form we can optimize

For distributions with finite first moments, Kantorovich–Rubinstein duality gives

$$W_1(P,Q) = \sup_{f\in F} \{\mathbb{E}_P[f] - \mathbb{E}_Q[f]\}\,,$$

where F is the set of all 1-Lipschitz functions: $|f(x)-f(y)| \le \|x-y\|_2$.

4. WGAN

Training idea

WGAN follows the same adversarial idea as GAN, using a critic to estimate Wasserstein distance. The critic $f_{\theta'}$ outputs a real-valued score without a sigmoid:

$$J(\theta',\theta) = \mathbb{E}_{x\sim P^*}[f_{\theta'}(x)] - \mathbb{E}_{z\sim P_z}[f_{\theta'}(G_\theta(z))].$$

Alternate between maximizing J over θ' with the generator fixed and minimizing J over θ with the critic fixed. The critic should satisfy the 1-Lipschitz constraint; two practical approaches are given below.

Weight clipping

The original WGAN clips each critic parameter after an update:

$$\theta'_i \leftarrow \mathrm{clip}(\theta'_i, -c, c).$$

With Lipschitz activations, this bounds the critic's Lipschitz constant, though not necessarily by 1. Clipping can restrict the critic's capacity.

Gradient penalty: WGAN-GP

WGAN-GP replaces clipping with a penalty on the critic's input gradient at points between real and generated samples:

$$\hat{x} = \epsilon x + (1-\epsilon) G_\theta(z), \qquad \epsilon \sim \mathrm{Uniform}[0,1].$$

With $\lambda > 0$, the critic minimizes

$$L_C^{\mathrm{GP}} = -J(\theta', \theta) + \lambda \mathbb{E}_{\hat{x}} \left[\left(\|\nabla_{\hat{x}} f_{\theta'}(\hat{x})\|_2 - 1 \right)^2 \right].$$

This softly encourages gradient norms near 1 at sampled points. The penalty is used only for critic updates.