EECS 498 HW5

Author: Minxuan Chen

December 7, 2024

1 GAN

1.1 Task 2 Answer

At a high level, the goal of the generator G is to fool the discriminator D, such that D treats the fake/generated image $G(z)$ as if it were a real image. On the other hand, the goal of D is to distinguish real images x from fake/generated images $G(z)$. The output of D (i.e., $D(\cdot)$) is a score, where real images should receive higher scores than fake images.

With this understanding:

For the generator loss ($\mathcal{L}_G$), the objective is to improve G. If $G(z)$ closely resembles a real image, the value of $D(G(z))$ should increase. By introducing a negative sign before $D(G(z))$, the loss $\mathcal{L}_G$ decreases as $D(G(z))$ rises. Hence, a smaller $\mathcal{L}_G$ corresponds to better performance of G, pushing G to generate more realistic images.

For the discriminator loss ($\mathcal{L}_D$), the objective is to improve D. When $G(z)$ closely resembles a real image, $D(G(z))$ will increase, causing $\mathcal{L}_D$ to rise. Minimizing $\mathcal{L}_D$ forces D to assign lower scores to fake images, thereby enabling D to better distinguish fake images from real ones.

We can also give a rigorous explanation. From the form the these losses, we know here $D(\cdot)$ is the logits that D thinks an image is real, and we try to minimize these 2 losses.

Starting from the minimax objective function

$$\max_G \min_D \left\{ - \left[\mathbb{E}_{x\sim p_{data}} \log D(x) + \mathbb{E}_{z\sim p(z)} \log(1 - D(G(z))) \right] \right\}$$

The object function here is actually binary cross-entropy loss with real image=1 and fake image=0. And the output of D here is probability of real image.

For probability and logits, we have

$$p = \frac{1}{1 + \exp(-L)}$$

When training D, we use

$$\min_D \left\{ - \left[\mathbb{E}_{x\sim p_{data}} \log D(x) + \mathbb{E}_{z\sim p(z)} \log(1 - D(G(z))) \right] \right\}$$

after converting probability to logits, the loss of D is

$$\min_D \left\{ \left[\mathbb{E}_{x \sim p_{data}} \log(1 + \exp(-D(x))) + \mathbb{E}_{z \sim p(z)} \log(1 + \exp(D(G(z)))) \right] \right\}$$

When training G, we only care about the second term,

$$\max_G \left\{ - \left[\mathbb{E}_{z \sim p(z)} \log(1 - D(G(z))) \right] \right\}$$

however, since the gradient property of this expression is not good, we alternatively use

$$\min_G \left\{ - \left[\mathbb{E}_{z \sim p(z)} \log(D(G(z))) \right] \right\}$$

similarly convert probability to logits, we have

$$\min_G \left\{ \left[\mathbb{E}_{z \sim p(z)} \log(1 + \exp(-D(G(z)))) \right] \right\}$$

1.2 Task (optional extra credits)

Despite the two regularization techniques we have shown above, what are some other augmentation or regularization techniques that are functional in GAN? Please search for the literature online and list three or more here.

1. We can use gradient penalty to control the Lipchitz of discriminator.

Literature: Improved training of wasserstein GANs by Gulrajani, Ishaan and Ahmed, Faruk and Arjovsky, Martin and Dumoulin, Vincent and Courville, Aaron. link

2. We can use spectral normalization, which also aims to control the Lipchitz of D.

Literature: Spectral Normalization for Generative Adversarial Networks by Takeru Miyato, Toshiki Kataoka, Masanori Koyama, Yuichi Yoshida. link

3. CR forces the output of the discriminator to remain consistent when the input undergoes certain transformations (e.g., data augmentation). This approach encourages the discriminator to focus on invariant features, leading to more robust training.

Regularizing Generative Adversarial Networks under Limited Data by Hung-Yu Tseng, Lu Jiang, Ce Liu, Ming-Hsuan Yang, Weilong Yang. link

2 Diffusion Model

2.1 Task 3

The total time spent of DDPM is 355.428147315979 seconds (10 pictures).

The total time spent of DDIM is 17.853221654891968 seconds (10 pictures).

DDIM offers faster sampling and more deterministic outputs compared to DDPM. However, it may lack flexibility and sample diversity.

2.2 Task 5

The total time spend of DDIM+DPS is 220.55031895637512 seconds (3 pictures).

So, on average, for DDIM, 1.79 seconds are needed for 1 picture, and for DDIM+DPS, 73.52 seconds are needed for 1 picture.

The extra time cost of 1 picture is 71.73 seconds, about 1 minute 10 seconds.

Time step=800, 173.20 seconds (3 pictures), 57.73 seconds/1 picture.

Time step=500, 108.27 seconds (3 pictures), 36.09 seconds/1 picture.

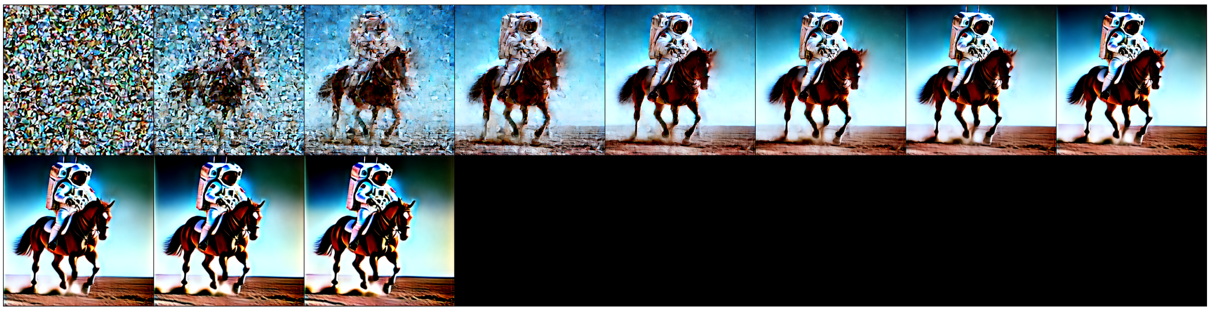
Time step=200, 44.40 seconds (3 pictures), 14.80 seconds/1 picture.

From these results, it can be seen that setting fewer time steps in DPS generally speeds up generation, but can lead to lower image quality, including reduced texture detail and increased artifacts, because the model does not have enough time to refine the sampling process.

2.3 Task 6

CFG=7.5

CFG=100



2.4 Task 7