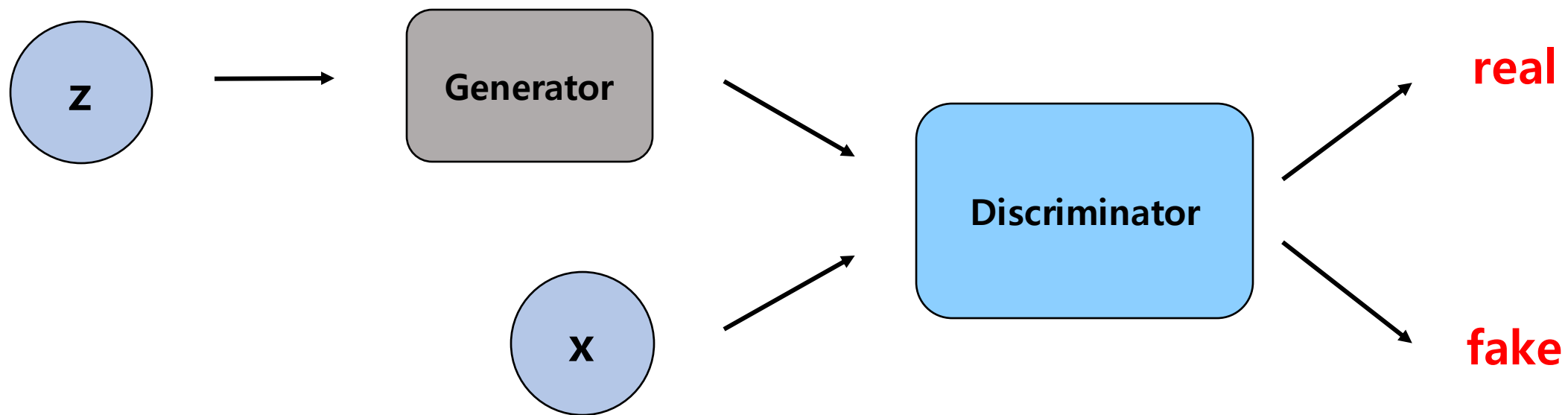


Generative Adversarial Nets

(GAN)

1. 구조



2. 용어 정리

- $z(\text{noise})$: 정규분포를 따르는 임의의 수로 이루어진 배열
- Generator(생성자) : 입력 받은 z 로 이미지를 생성
- x : 실제 이미지 배열
- Discriminator(판별자) : 실제 이미지는 1, 가짜 이미지(생성자가 생성한 이미지)는 0으로 판별
- latent space : z 가 존재하는 공간.

z 를 입력받은 생성자가 data를 생성하므로 무엇으로든 변할 수 있는 것들이 모여 있는 공간

3. 동작 원리

1. 생성자가 임의의 $\text{noise}(z)$ 를 입력 받으면 fake 데이터를 생성한다.
2. 판별자에 real data, fake data를 입력하고, $\text{fake} = 0$, $\text{real} = 1$ 을 출력한다.
3. 생성자는 최대한 real data에 가깝게 fake data를 생성 하도록 학습한다.
4. 판별자는 fake data 인지, real data인지 최대한 잘 구분하도록 학습한다.
5. 생성자는 진짜에 가깝게, 판별자는 가짜를 잘 구분하도록 서로 적대적으로 학습하여 결국 생성자와 판별자 서로가 경쟁적으로 발전하여 둘 모두의 성능이 점점 향상된다.

4. Loss

GAN에서 사용하고 있는 loss function은 minmax loss 이다.

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

$\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})]$: real data를 판별자에 넣었을 때 나온 값에 log 취했을 때 얻는 기대값

$\mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))]$: 1 - fake data를 판별자에 넣었을 때 나온 값에 log 취했을 때 얻는 기대값

4. Loss

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

판별자 D의 $V(D, G)$ 에 대한 이상적인 결과

$D(\mathbf{x}) = 1$ 이므로 첫번째 항은 0이 된다. fake data 를 잘 구분했을때 $D(G(\mathbf{z})) = 0$ 이 되므로 두번째 항도 0이 되는 것이 가장 이상적이다.

즉, 판별자의 입장에서 $V(D, G) = 0$ 이 이상적이다.

4. Loss

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

생성자 G의 $V(D, G)$ 에 대한 이상적인 결과

첫번째 항은 생성자의 성능과 관계가 없으므로 생각하지 않고, 두번째 항만 생각한다. fakedata를 판별자가 구분하지 못할 정도로 잘 만들었다면 $D(G(\mathbf{z})) = 1$ 이 되므로 두번째 항은 $-\infty$ 가 된다.

즉, 생성자의 입장에서 $V(D, G)$ 이 $-\infty$ 로 수렴하는 것이 이상적이다.

4. Loss

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

정리

판별자 D는 training sample과 fake data를 잘 구분할 확률을 최대화 하기 위해 학습

-> $V(D, G)$ 의 최댓값이 나오도록 학습

생성자 G는 training sample과 fake data를 판별자가 구분할 수 없도록 학습

-> $V(D, G)$ 의 최솟값이 나오도록 학습

* 판별자 D는 fake or real 이진분류 하므로 binary crossentropy 사용