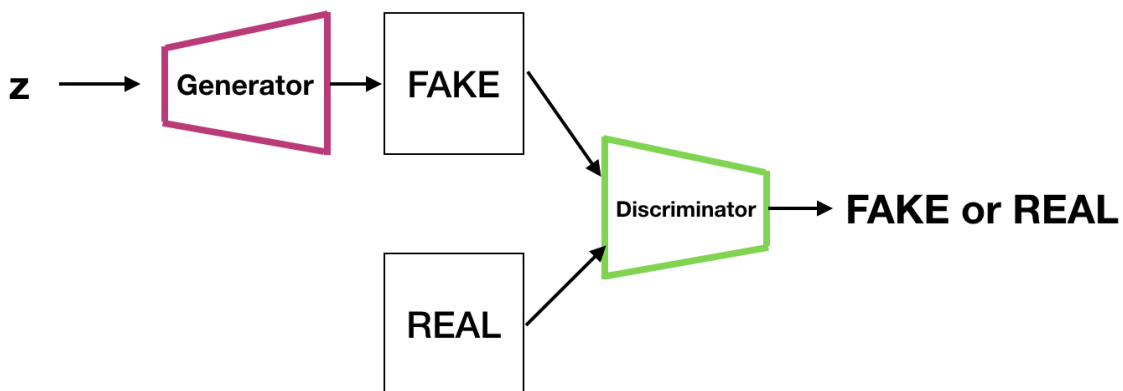




(Hayeon Kim) GAN Code Review

👤 발표자	㉟ 김하연
🔗 논문링크	https://arxiv.org/abs/1406.2661
🔗 발표동영상	http://gofile.me/6UrAI/b4QbLrKK2
⌵ 발표시간	10:00 ~ 10:30
📎 발표자료	

GAN의 구조



GAN은 우리가 가지고 있는 데이터와 비슷하게 데이터를 생성하는 것을 학습하는 **Generator** 와 Generator 가 생성한 데이터(fake)와 실제 우리가 가지고 있는 데이터(real)를 fake 인지 real 인지 구분하는 **Discriminator** 를 구분하는 두 신경망으로 구성

1. Generator : 가우시안 노이즈를 입력으로 받아, 이 데이터로 부터 데이터 생성
2. Discriminator : Generator 가 생성해낸 데이터와 기존에 가지고 있는 진짜 데이터를 입력으로 받아, 진짜 데이터를 1, 가짜데이터를 0으로 학습(진짜에 높은 확률, 가짜에 낮은 확률)

를)

Loss Function

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

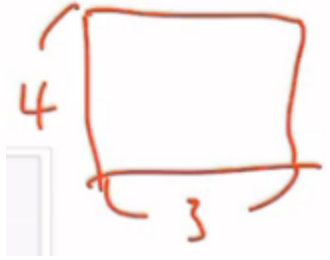
1. max D : Real을 입력으로 넣었을 때는, $\log(D(x))$ 의 기댓값이 최대가 되게 하고, $G(z)$ 즉, 가짜를 가짜라고 할 확률 $1-D(G(z))$ 는 최대가 되게끔 학습
2. min G : 가우시안 랜덤 변수를 받아 생성된 데이터를 $D(G(z))$, discriminator에 넣었을 때, $1-D(G(z))$, 즉 가짜라고 할 확률을 최소화하게끔 학습하는 것입니다.(즉, G는 D를 속이도록 최적화)
3. $\log(1-D(G(z)))$ 의 경우 saturation이 발생하므로 non-saturation방식을 채택하여 실제 구현에서의 criterion 은 Discriminator 가 사용하는 criterion(binary cross entropy loss)을 동일하게 사용합니다.

▼ pytorch view(reshape도 사용가능 !)

<https://wikidocs.net/52846>



`torch.Size([2, 2, 3])`



```
print(ft.view([-1, 3]))
```