

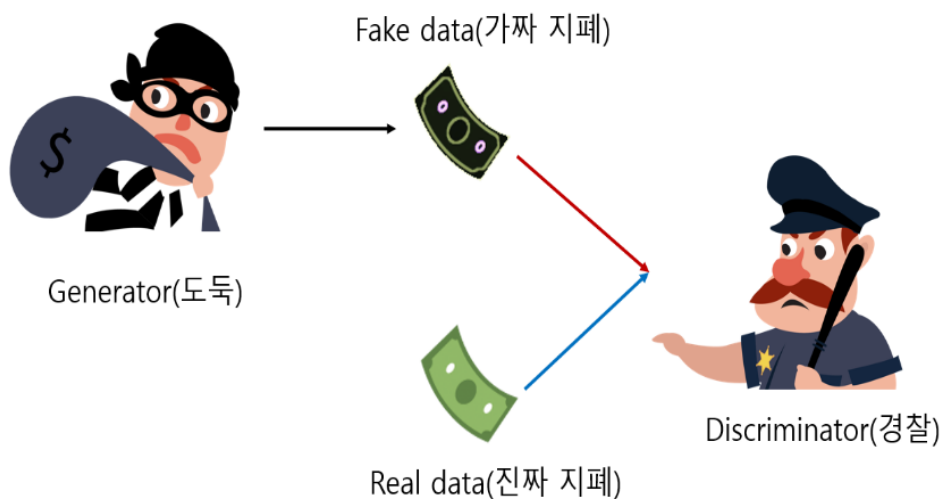
# 9주차 Gen Basic 과제1 -논문리뷰

## GAN : Generative Adversarial Networks



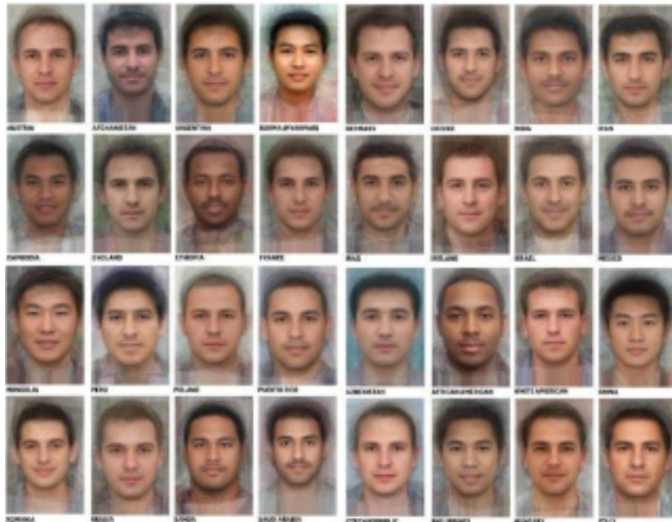
기본적으로 두 개의 다른 신경망 (Generator와 Discriminator)간의 적대적인 관계로 대립(Adversarial)하며 서로의 성능을 점차 개선해 나간다.

이미지를 생성하는 Generator와 이를 구별하는 Discriminator가 서로 적대적으로 학습하는 과정을 거쳐 Generator는 더욱 질이 좋은 이미지를 생산해내고, Discriminator는 해당 이미지가 real인지 fake인지 더 잘 구별하도록 학습을 하게 된다.



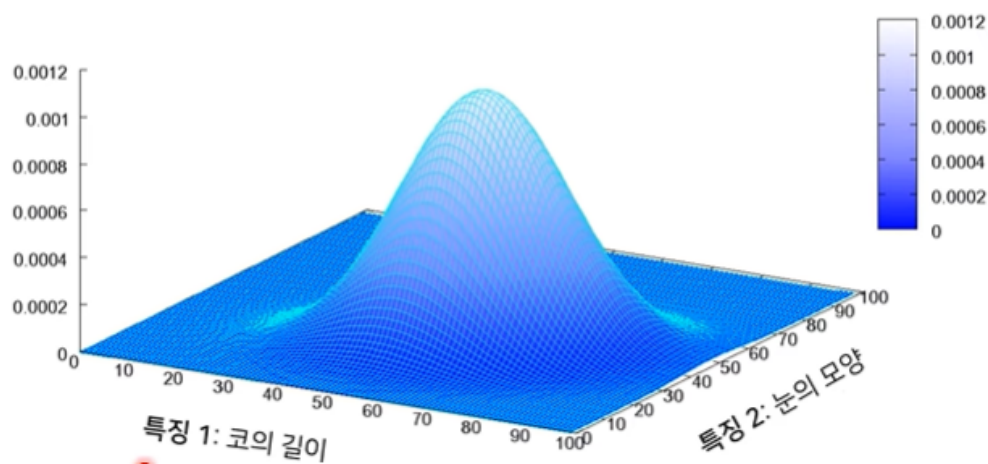
### Generative model (G)

즉, (Generative model) 모델 G는 우리가 갖고 있는 data x의 distribution을 알아내려고 노력한다. 만약 G가 정확히 data distribution을 묘사할 수 있다면 거기서 뽑은 sample은 완벽히 data와 구별할 수 없다. (생성 모델은 실존하지 않지만 있을 법한 이미지(자연어. 음성 다 가능)를 생성할 수 있는 모델을 의미한다.)

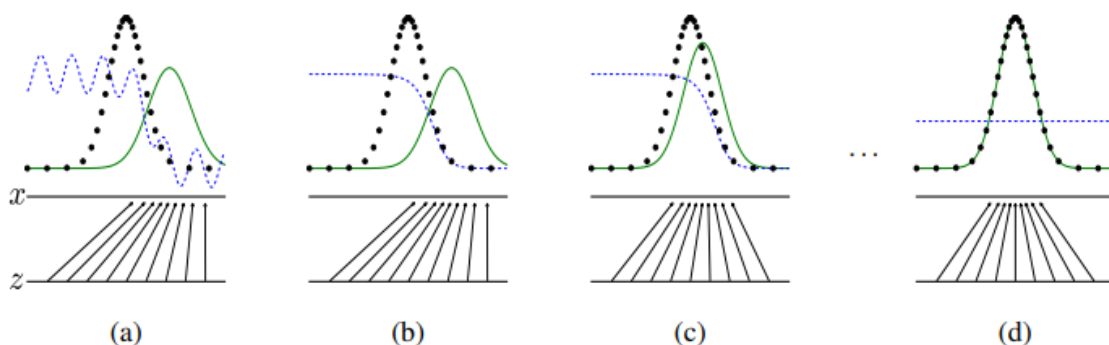


- 모든 이미지는 통계적인 data distribution이 있고, 이런 이미지 데이터는 다차원 특징 공간의 한 점으로 표현한다.

다양한 특징들이 각각 확률 변수가 되는 분포를 다변수 확률분포라고 한다.



- 모델 G는 원래 데이터(이미지)의 분포를 근사할 수 있도록 학습된다.



검은색 : 원본 데이터의 분포, 초록색 : 생성 모델의 분포 // 시간이 지나면서 생성 모델 G가 원본 데이터의 분포를 학습한다.

따라서 모델 G의 학습이 잘 되었다면 통계적으로 평균적인 특징을 가지는 데이터를 쉽게 생성할 수 있다.

## Discriminator model (D)

- 모델 D는 현재 자기가 보고 있는 sample이 training data에서 온 것(진짜)인지 혹은 G로부터 만들어진 것인지를 구별하여 각각의 경우에 대한 확률을 측정한다.
- D는 data로부터 뽑은 sample  $x$ 에 대해서  $D(x) = 1$ 이 되고, G에 임의의 noise distribution으로부터 뽑은 input 값인  $z$ 를 넣고 만들어진 sample에 대해서는  $D(G(z)) = 0$ 이 되도록 훈련해야 한다.

## Generative Adversarial Networks (GAN)

생성자(G)와 판별자(D) 두 개의 네트워크를 활용한 생성 모델 (따라서 Adversarial 적대적이라고 볼 수 있다.)

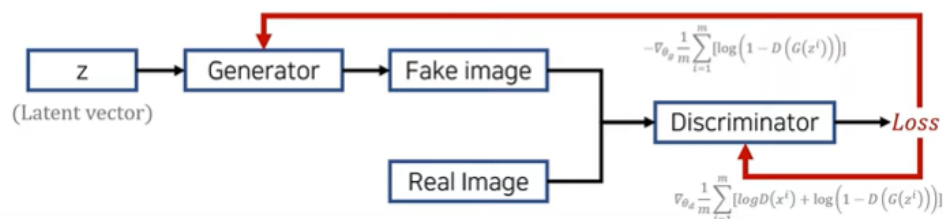
아래의 목적 함수 (objective function)를 통해 생성자는 이미지 분포를 학습할 수 있다.

$$\min_G \max_D V(D, G) = \min_{\theta_g} \max_{\theta_d} \left[ \mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log (1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Discriminator output for real data  $x$ 
Discriminator output for generated fake data  $G(z)$

- Discriminator는 기본적으로 분류기에 해당
- $p_{data}$  : 실제 데이터 분포 sample
- $D_{\theta_d}$  : 실제 sample에 대한 discriminator 분류 값
- $p(z)$  : 실제 데이터 분포 sample
- $D_{\theta_d}(G_{\theta_g}(x))$  : fake sample에 대한 디스크리미네이터의 샘플 값

판별자(D)는 학습을 할 때, 원본 데이터에 대해서는 1로 분류할 수 있도록 학습된다. 따라서 최대화 할 수 있도록 한다. 반면에 fake가 들어올 경우, 0으로 분류할 수 있도록 학습된다. 반면에 생성자(G)는 오른쪽 term만 해당되고, 이를 최소화시키는 것이다. 즉, 생성자가 만든 가짜 이미지가 판별자에 의해 진짜라고 인식될 수 있도록, 다시 말하면 생성자에 의해 만들어진 이미지를 판별자로부터 진짜, 사실이라고 인식될 수 있도록 학습된다는 것이다.



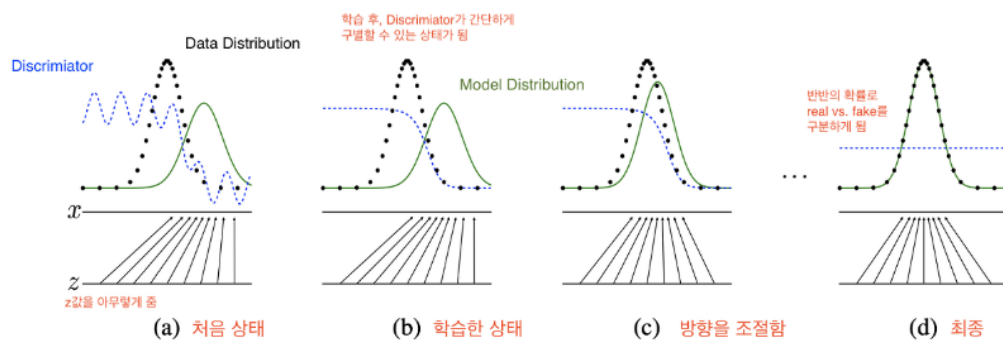
동일한 식에 대해서 G,D가 서로 다른 목적을 가지기 때문에, 게임이론에 기반하는 최적화 문제로 볼 수 있다.

$G \rightarrow \min, D \rightarrow \max$ 가 달성이 되면, 결과적으로 생성자 G는 굉장히 그럴싸한 이미지를 만들 수 있는 모델이 될 것이다.

## GAN의 수렴(학습) 과정

- 공식의 목표 (Goal of Formulation)

$P_g \rightarrow P_{data}, D(G(z)) \rightarrow 1/2$  (G(z)는 더이상 D에 의해 구별되지 않는다 따라서 50% 값만 갖게 된다.)



$P_g$ 가  $P_{data}$ 로 수렴하는 과정을 본 논문에서 제일 중요하게 다룬다.

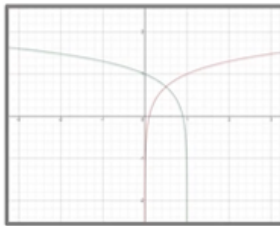
저자는 이론적인 동기로부터 수정을 한 것이 아니라 순수하게 실용적인 측면에서 적용을 한 것이라고 한다. 왜냐하면 학습 초기에 G가 생성하는 image들은 실제와는 거리가 먼 이상한 image들을 생성하기 때문에 D가 너무도 쉽게 이를 real image와 구별하게 되고 따라서  $\log(1 - D(G(z)))$  값이 매우 saturate하여 gradient를 계산해보면 아주 작은 값이 나오기에 학습이 매우 느리다.

## 증명 : Global Optimality 1

Proposition:  $D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$

Proof: For G fixed,

$$\begin{aligned} V(G, D) &= E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \\ &= \int_x p_{data}(x) \log(D(x)) dx + \int_z p_z(z) \log(1 - D(g(z))) dz \\ &= \int_x p_{data}(x) \log(D(x)) + p_g(x) \log(1 - D(x)) dx \end{aligned}$$



function  $y \rightarrow a \log(y) + b \log(1 - y)$  achieves its maximum in  $[0, 1]$  at  $\frac{a}{a + b}$

same as optimal control:  $\frac{\delta V(G, D)}{\delta D} [D^*(x)] = 0$

극값을 갖는 지점

## 증명 : Global Optimality 2

Proposition: Global optimum point is  $p_g = p_{data}$

Proof:

$$\begin{aligned} C(G) &= \max_D V(G, D) = E_{x \sim p_{data}(x)}[\log D^*(x)] + E_{z \sim p_z(z)}[\log(1 - D^*(G(z)))] \\ &= E_{x \sim p_{data}(x)} \left[ \log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right] + E_{x \sim p_g(x)} \left[ \log \frac{p_g(x)}{p_{data}(x) + p_g(x)} \right] \\ &= E_{x \sim p_{data}(x)} \left[ \log \frac{2 * p_{data}(x)}{p_{data}(x) + p_g(x)} \right] + E_{x \sim p_g(x)} \left[ \log \frac{2 * p_g(x)}{p_{data}(x) + p_g(x)} \right] - \log(4) \\ &= KL(p_{data} || \frac{p_{data}(x) + p_g(x)}{2}) + KL(p_g || \frac{p_{data}(x) + p_g(x)}{2}) - \log(4) \\ &= 2 * JSD(p_{data} || p_g) - \log(4) \end{aligned}$$

removed when  $p_g = p_{data}$

$KL(p_{data} || p_g) = \int_{-\infty}^{\infty} p_{data}(x) \log \left( \frac{p_{data}(x)}{p_g(x)} \right) dx$

$JSD(p || q) = \frac{1}{2} KL(p || \frac{p+q}{2}) + \frac{1}{2} KL(q || \frac{p+q}{2})$

목적 식을 새로운 함수 C에 대해서 표현한다. 또한 중간 부분에서 분자에 2씩 곱해주고 -log(4)를 취하는 과정은 계산에 편리성을 위해서이다. 뿐만 아니라 KL 수렴 공식을 이용하여 두 개의 기대값을 더한 값을 반환할 수 있다. 궁극적으로 JSD(젠슨-새넨 발산)를 이용하여 최적화 시킨다.

**JSD는 최소값을 0으로 가지기 때문에,  $P_g$ 가  $P_{data}$ 에 수렴하게 되면, JSD 값은 0을 반환하게 된다.**

## GAN 알고리즘

---

**Algorithm 1** Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator,  $k$ , is a hyperparameter. We used  $k = 1$ , the least expensive option, in our experiments.

---

**for** number of training iterations **do**

**for**  $k$  steps **do**

- Sample minibatch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\}$  from noise prior  $p_g(z)$ .
- Sample minibatch of  $m$  examples  $\{x^{(1)}, \dots, x^{(m)}\}$  from data generating distribution  $p_{\text{data}}(x)$ .
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[ \log D(x^{(i)}) + \log (1 - D(G(z^{(i)}))) \right].$$

**end for**

- Sample minibatch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\}$  from noise prior  $p_g(z)$ .
- Update the generator by descending its stochastic gradient:

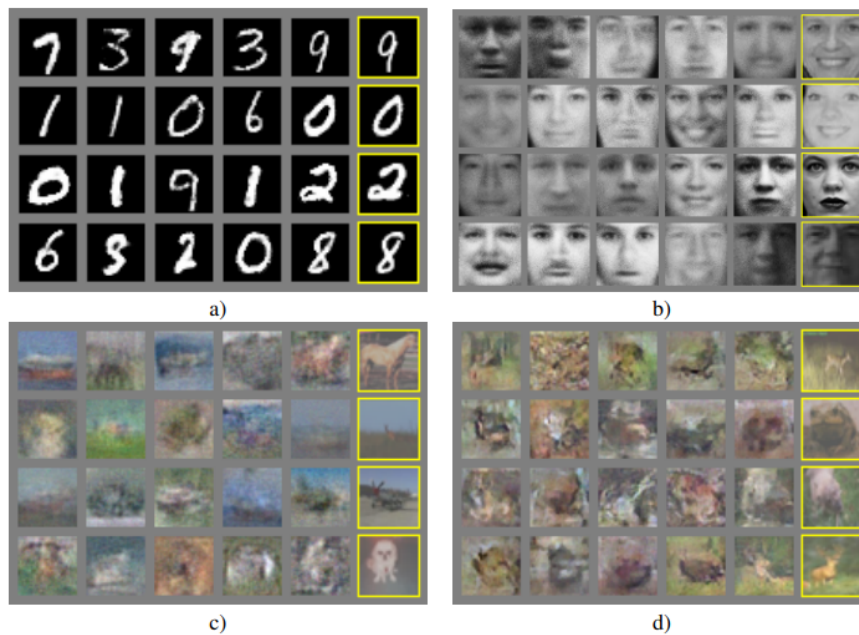
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log (1 - D(G(z^{(i)}))).$$

**end for**

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

---

## Results



- Not cherry-picked : 이미지를 별도로 선별해서 넣은 것이 아닌 랜덤하게 만든 것을 그대로 적용했다.
- Not memorized the training set : 단순히 훈련 데이터 셋을 암기한 것이 아니다.
- Competitive with the better generative models : 2014 당시 경쟁력있는 생성 모델이라고 한다.

- Image represent sharp : 상대적으로 흐리지 않고, 깔끔하게 나타난다고 한다.

▼ 참고자료

<https://dacon.io/en/competitions/official/235946/codeshare/5499>

<https://velog.io/@a01152a/논문-읽기-및-구현1-GAN>

<https://arxiv.org/pdf/1406.2661.pdf>

나동빈 유튜브 GAN : Generative ADversarial Networks (꼼꼼한 딥러닝 논문 리뷰와 코드 실습)