Generative Adversarial Networks

2020.07.31 이재규

Introduction

INTRODUCTION

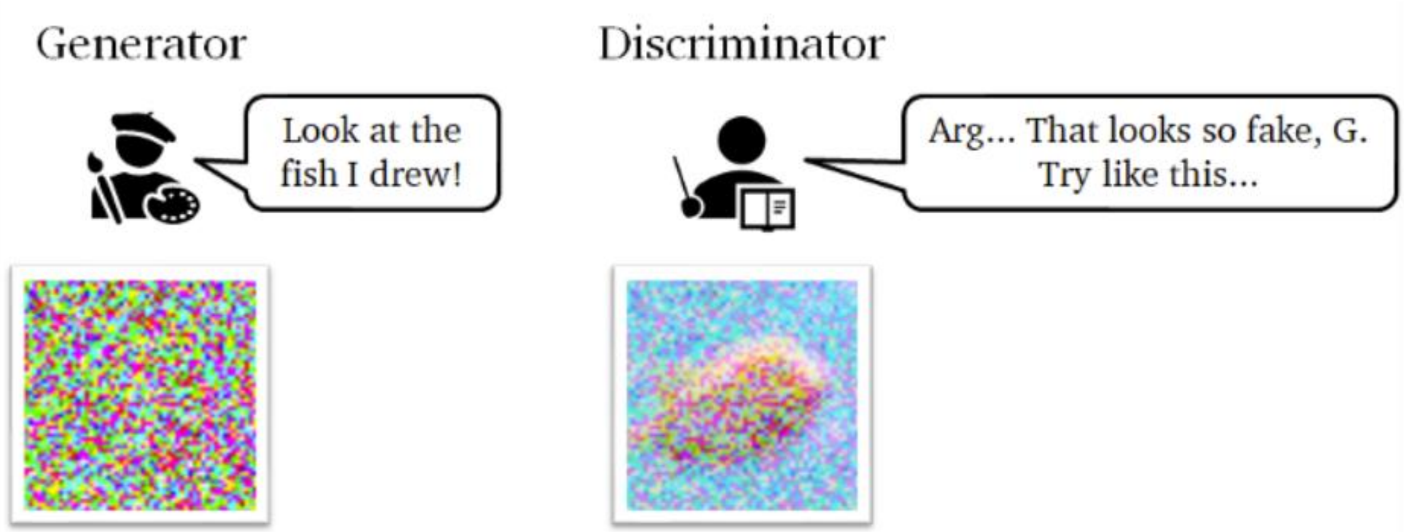
- **Why GAN?**

  - 기존 Deep generative model들은 maximum likelihood estimation과 관련된 전략들에서 발생하는 많은 확률 연산들을 근사하는 데 발생하는 어려움을 잘 해결 못함.

=> 이 논문에서 소개될 새로운 generative model은 이러한 어려움을 회피한다

INTRODUCTION

- **What is GAN?**
  - Ian Goodfellow(2014)가 제안한 Neural Network Model
  - Most Interesting Idea in the last ten years in Machine Learning!
    (Yann LeCun Facebook AI Director)
  - Generator와 Discriminator의 경쟁을 통한 학습



INTRODUCTION

- **Concept of GAN**

Generative Adversarial Networks

생성의 vs 적대적인

✓ 판별자는 전통적인 지도학습을 이용해 인풋을 진짜,가짜 구별하는 것을 학습함.

✓ 생성자는 판별자를 속일 정도의 진짜 같은 가짜 데이터를 생성하도록 훈련됨.

✓ 생성자를 위조지폐를 만드려하는 위조범처럼, 구별자는 진짜 지폐는 허용하고 위조지폐를 잡으려고 하는 경찰처럼 생각할 수 있음.

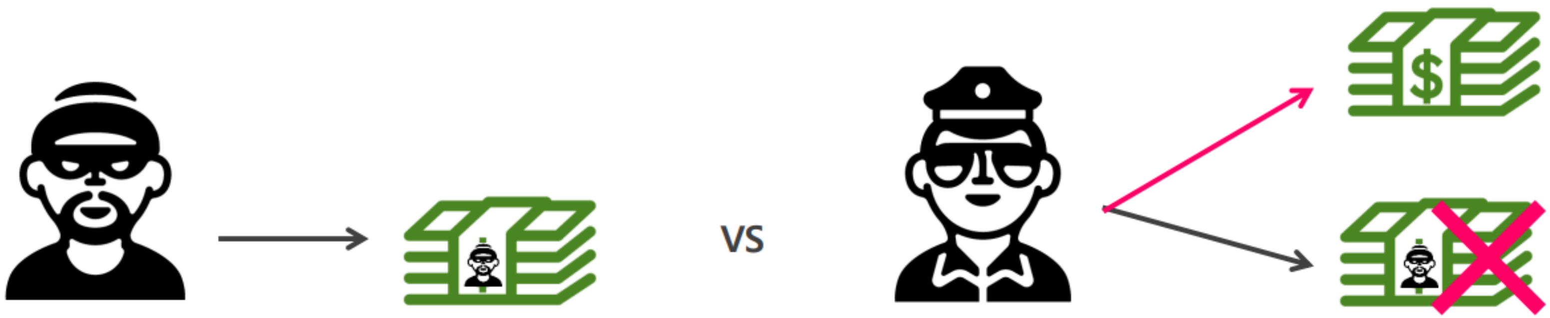
✓ 이 둘의 게임에서 이기기 위해 위조범은 진짜 지폐와 구별이 불가능한 위조 지폐를 만드는 것을 배워야함.

INTRODUCTION

- **Concept of GAN**

Generative Adversarial Networks

생성의 vs 적대적인



Goal: 진짜 지폐와 최대한 비슷한 위조지폐를 만들자!

Generator

Goal: 위조지폐와 진짜 지폐를 잘 구별해내자!

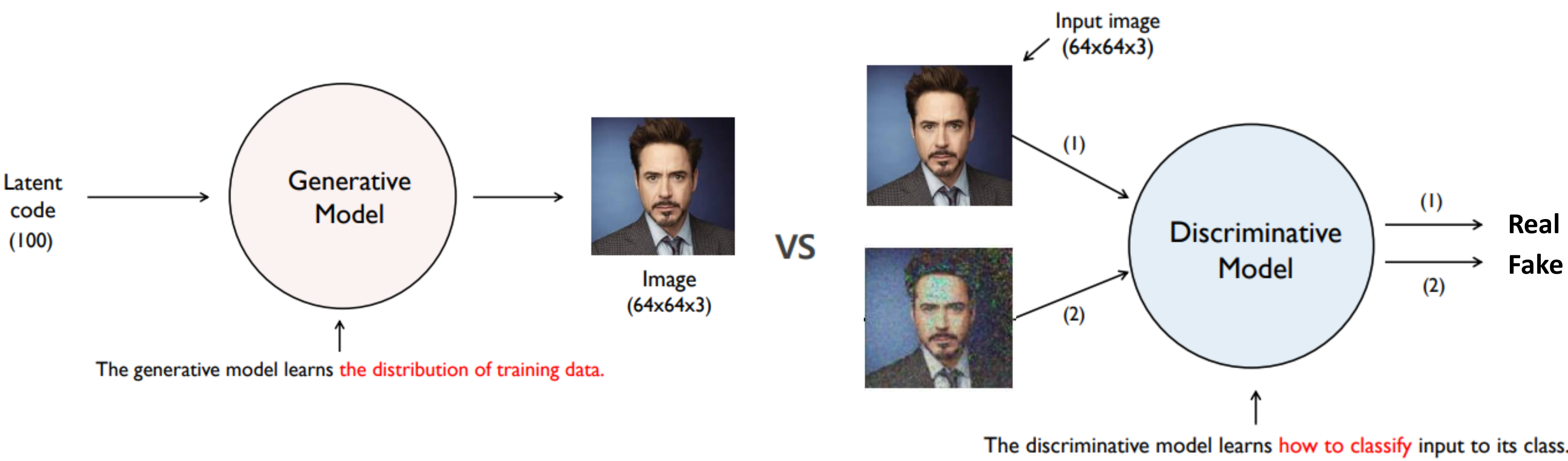
Discriminator

INTRODUCTION

- **Concept of GAN**

Generative Adversarial Networks

생성의 vs 적대적인



Generator **Discriminator**

✓ **생성자는 훈련 데이터와 동일한 분포에서 추출한 샘플들을 생성하는 방법을 배워야함.**

Model

Model

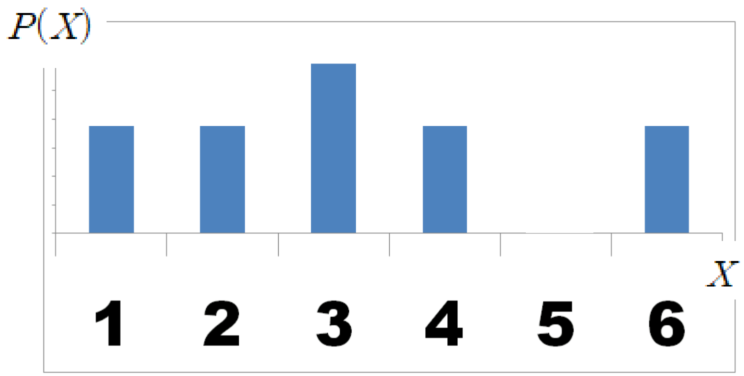
- **Probability Distribution**

Review Basics with Dice example

Random Variable

X	1	2	3	4	5	6
$P(X)$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{1}{6}$	$\frac{0}{6}$	$\frac{1}{6}$

Probablity mass function



Model

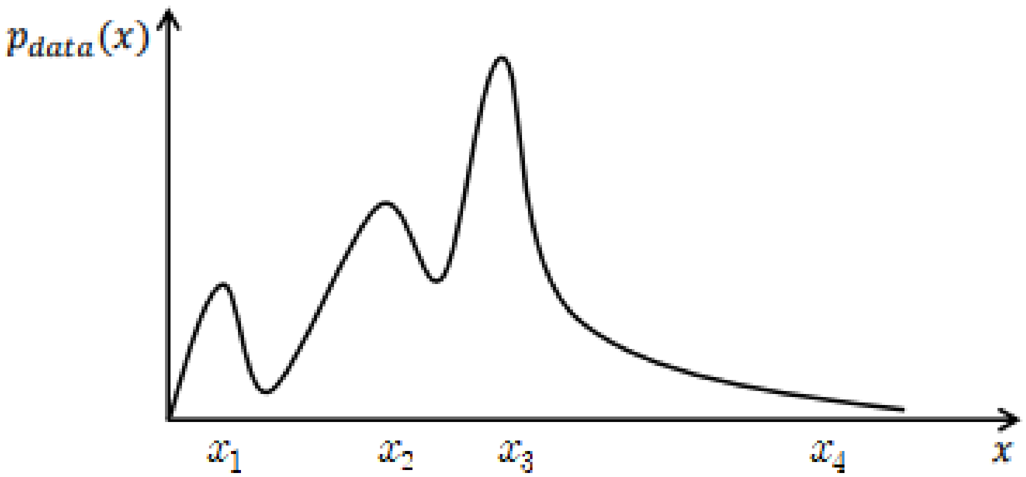
- **Probability Distribution**

What if ***X*** is actual images in the data?

With this point, ***X*** can be represented as a 64x64x3 dimensional vector.
(Number of pixels & value per pixel)

Model

- **Probability Distribution**

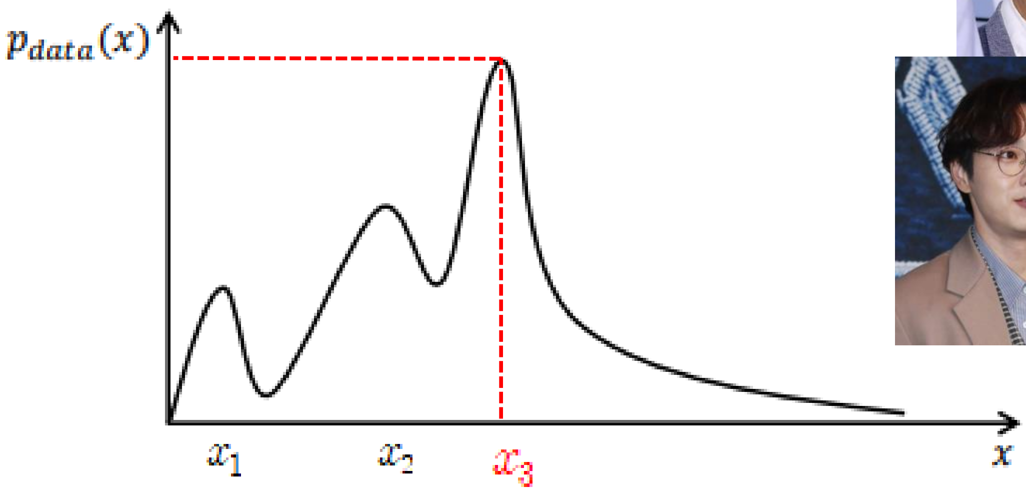


Pdata(x) (Probability density function) that represents the distribution of actual images.

(It is Very high dimensional distribution, so make example with low dimensional to represent)

Model

- **Probability Distribution**

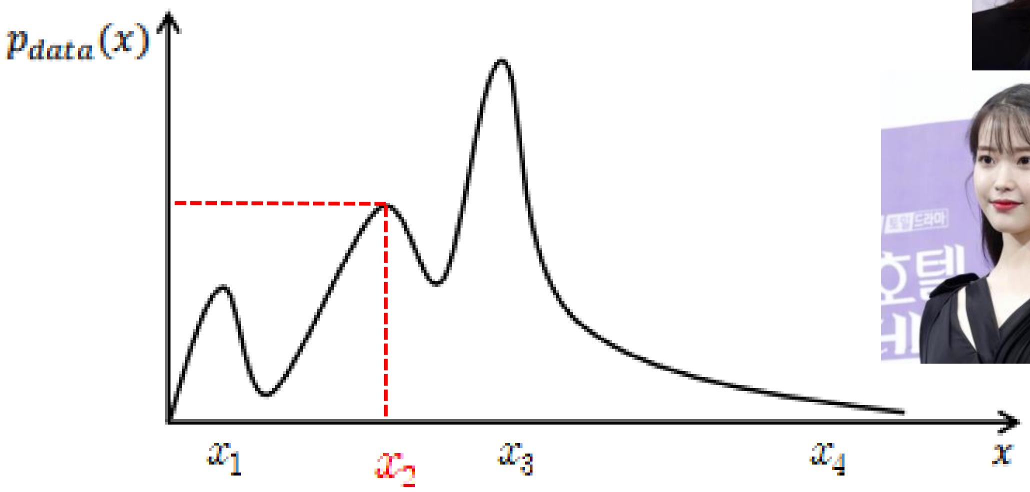


Take an example with human face image dataset.
This dataset may contain some images of men with glasses

X3 is a 64x64x3 high dimensional vector representing a man with glasses.
& the probability density is very high

Model

- **Probability Distribution**

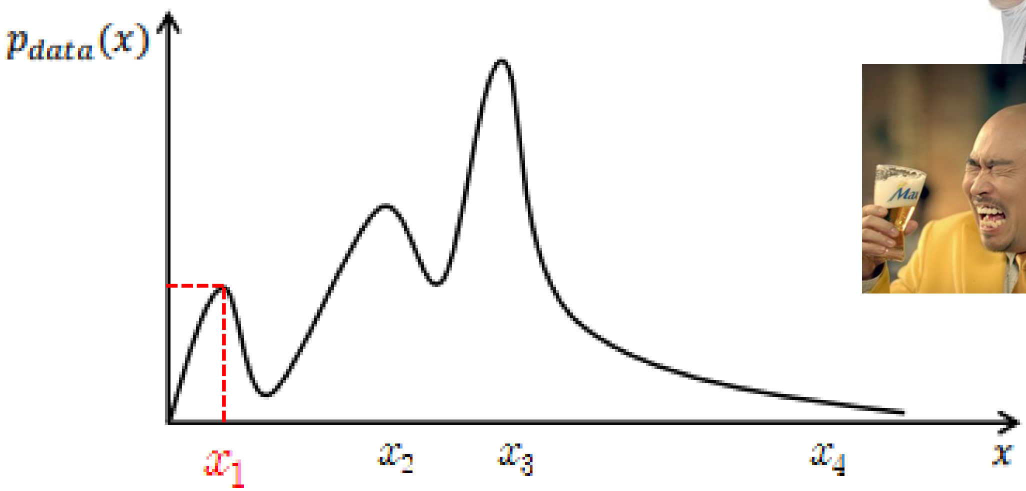


This dataset may contain some images of women with black hair

X2 is a 64x64x3 high dimensional vector representing a woman with black hair
& the probability density is high

Model

- **Probability Distribution**

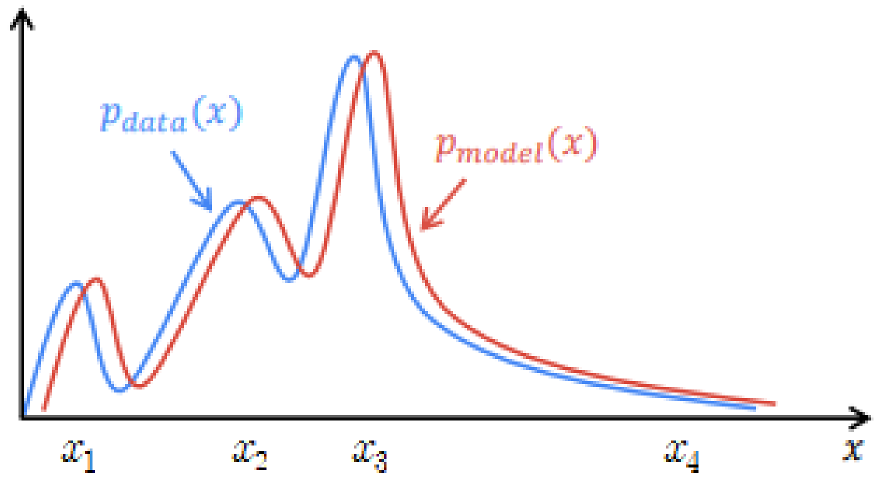


This dataset may contain some images of bald men

X1 is a 64x64x3 high dimensional vector representing a bald man
& the probability density is low

Model

- **Probability Distribution**

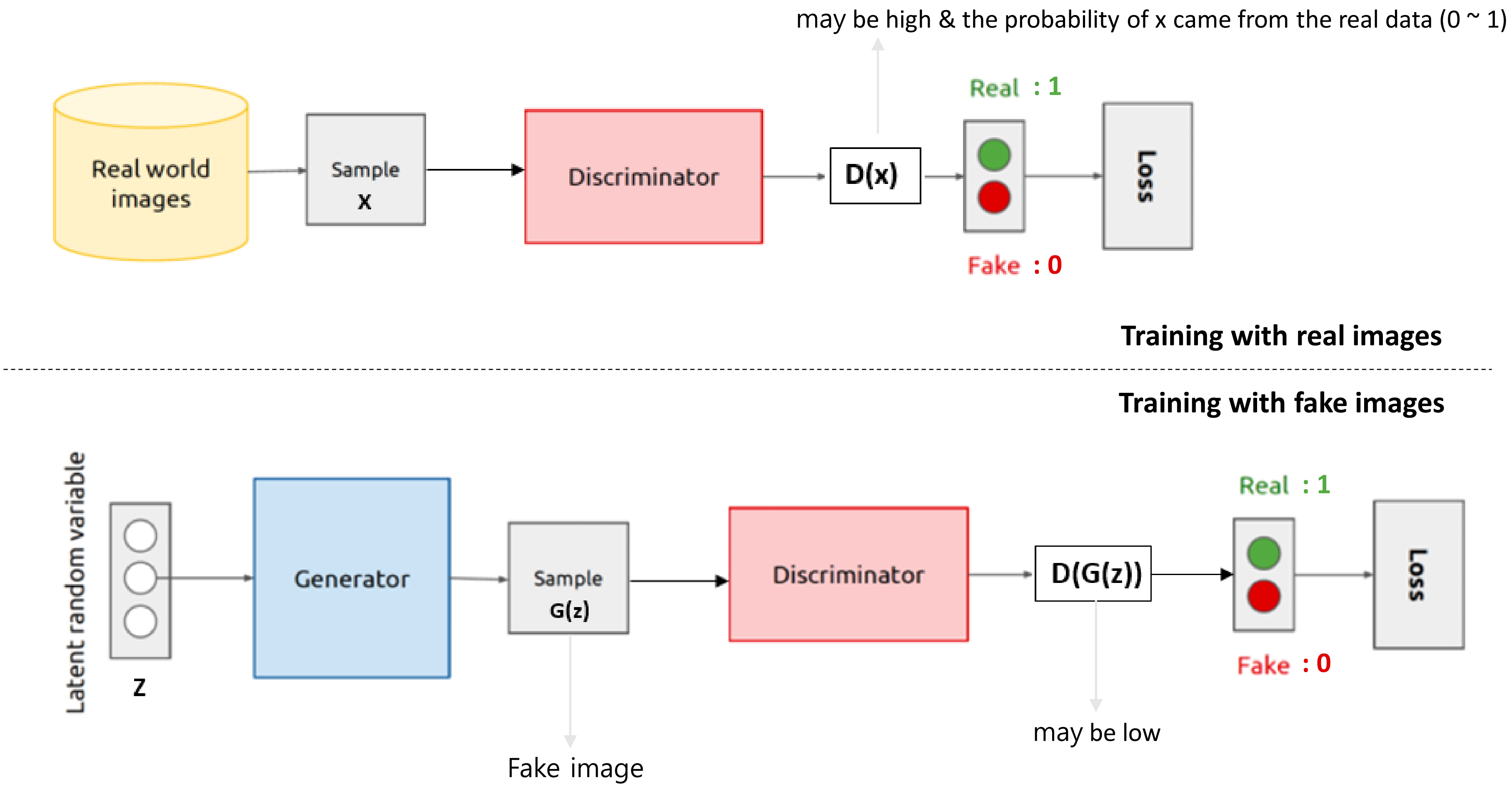


The goal of the generative model is to find a ***Pmodel(x)*** that approximates ***Pdata(x)*** well

Pmodel(x) : Distribution of images generated by the model
Pdata(x) : Distribution of actual images

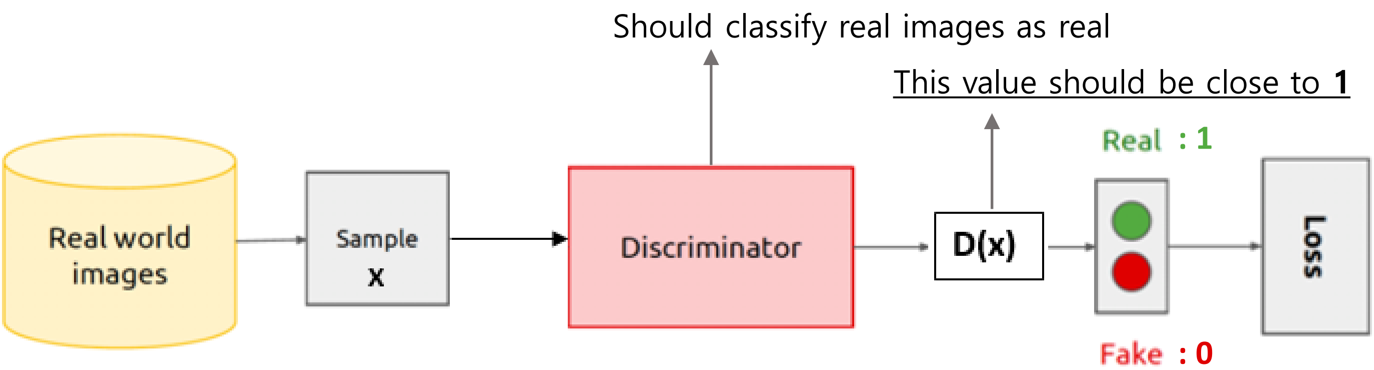
Model

- **Structure of GAN**



Model

- **Structure of Discriminator - real**

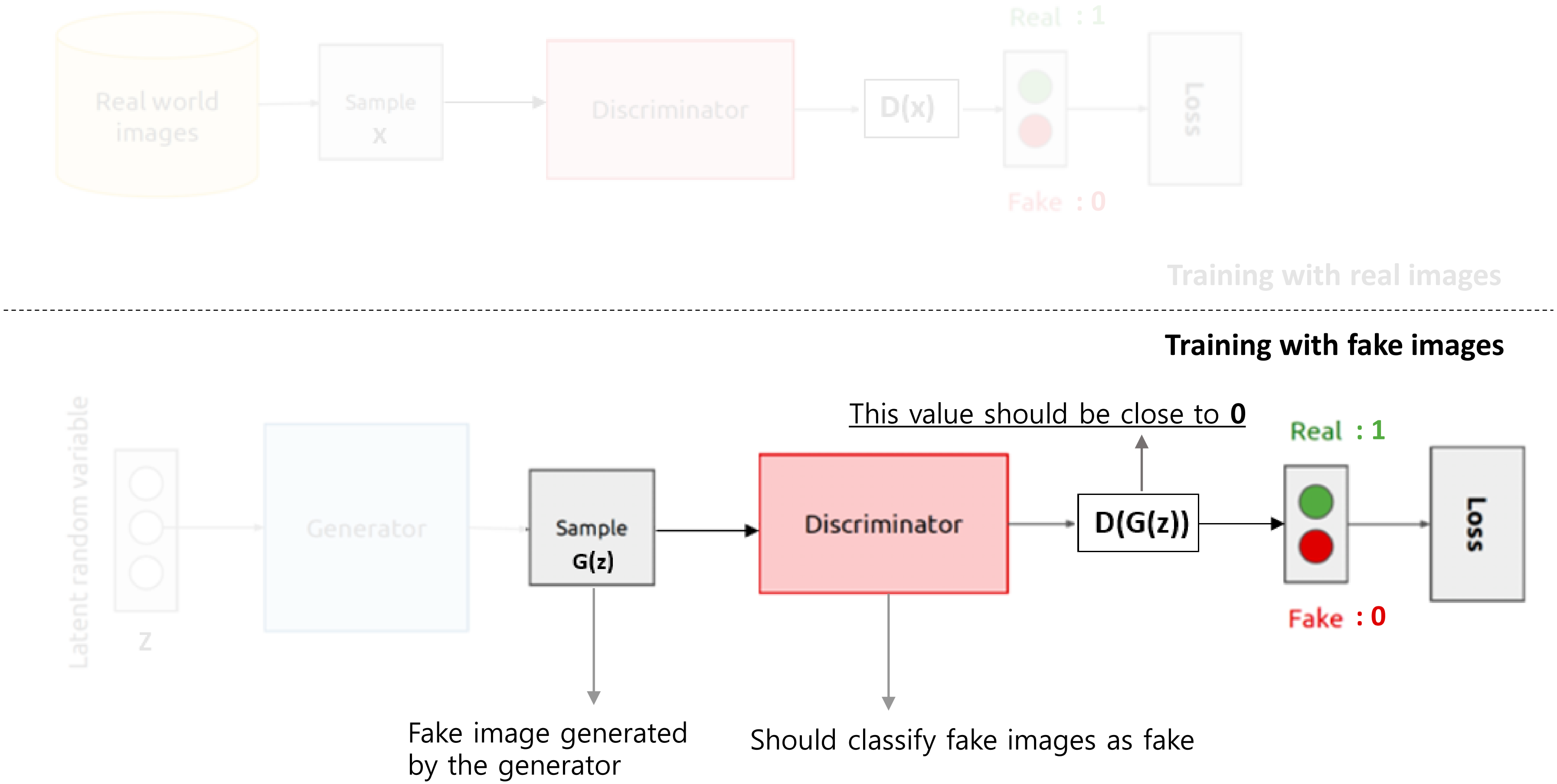


Training with real images

Training with fake images

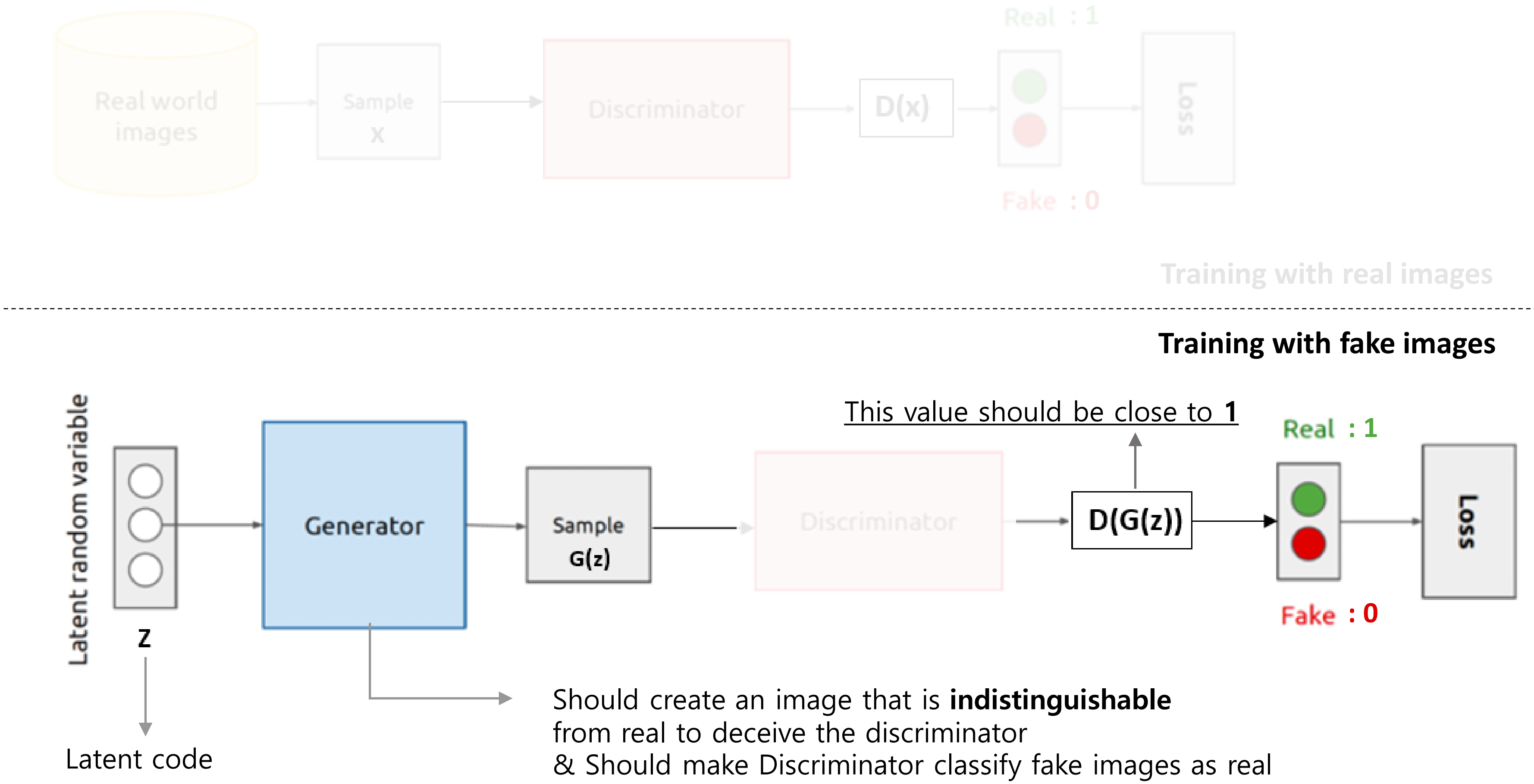
Model

- **Structure of Discriminator - fake**



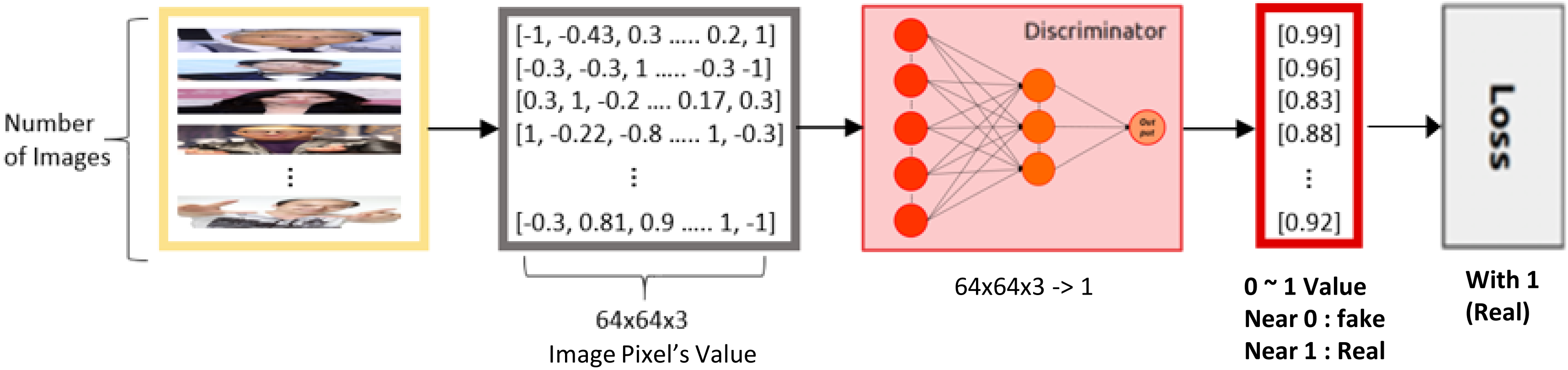
Model

- **Structure of Generator - fake**



Model

- **Actual flow of GAN**

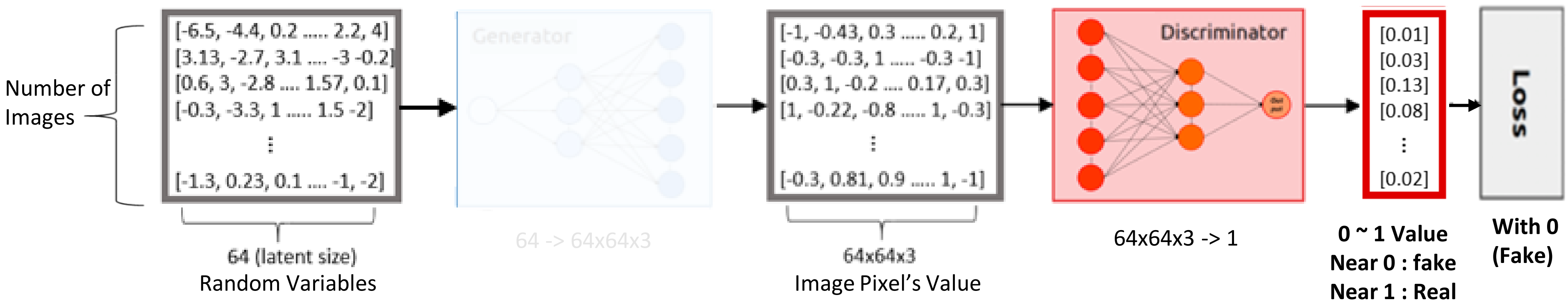


Training Discriminator with **real images**

Training Discriminator with **fake images**

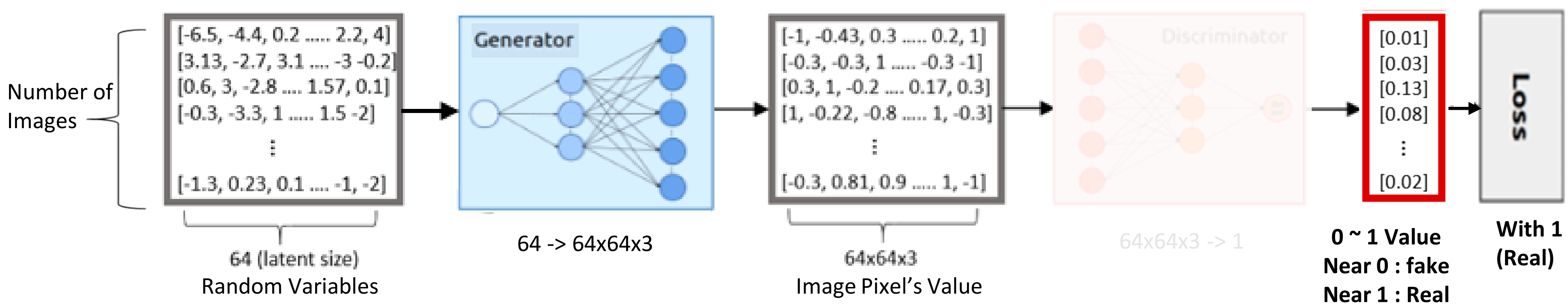
Model

- **Actual flow of GAN**



Training **Discriminator** with fake images

Training **Generator** with fake images



Model

- **Objective Function**

Sample x from real data distribution

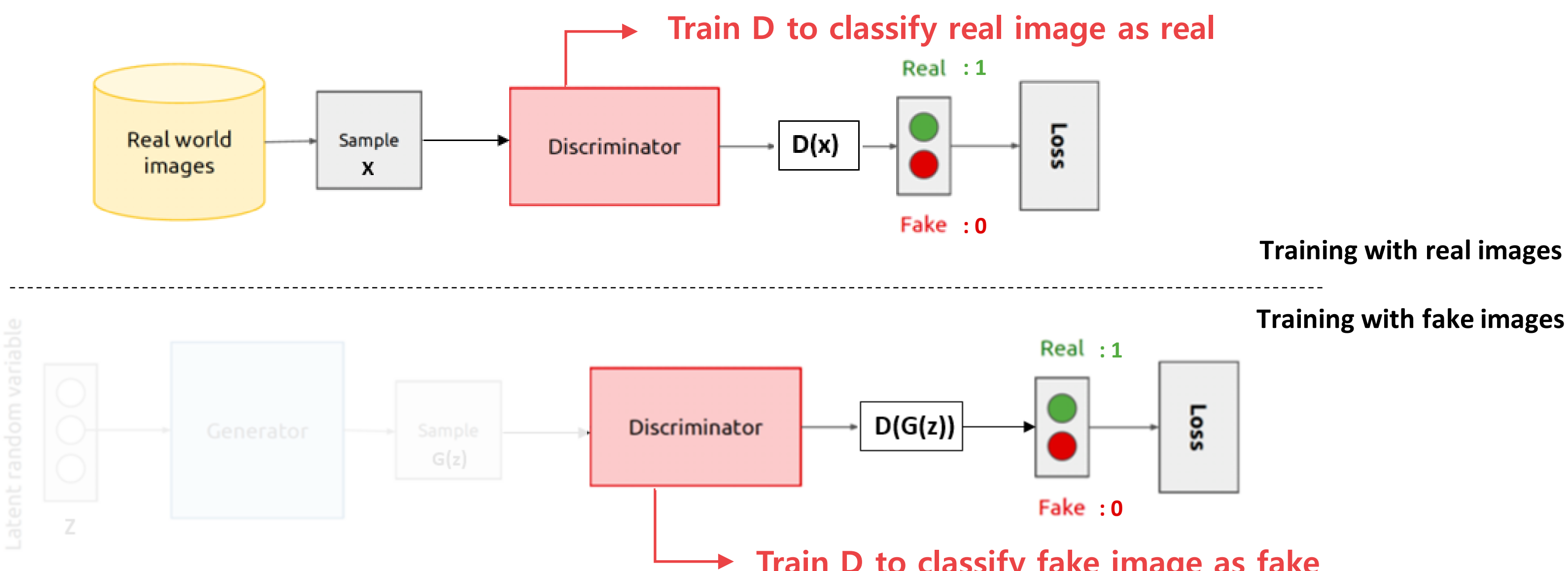
Sample latent code z from Gaussian distribution

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

D should maximize *V(D,G)*

Maximum when *D(x) = 1*

Maximum when *D(G(z)) = 0*



Model

- **Objective Function**

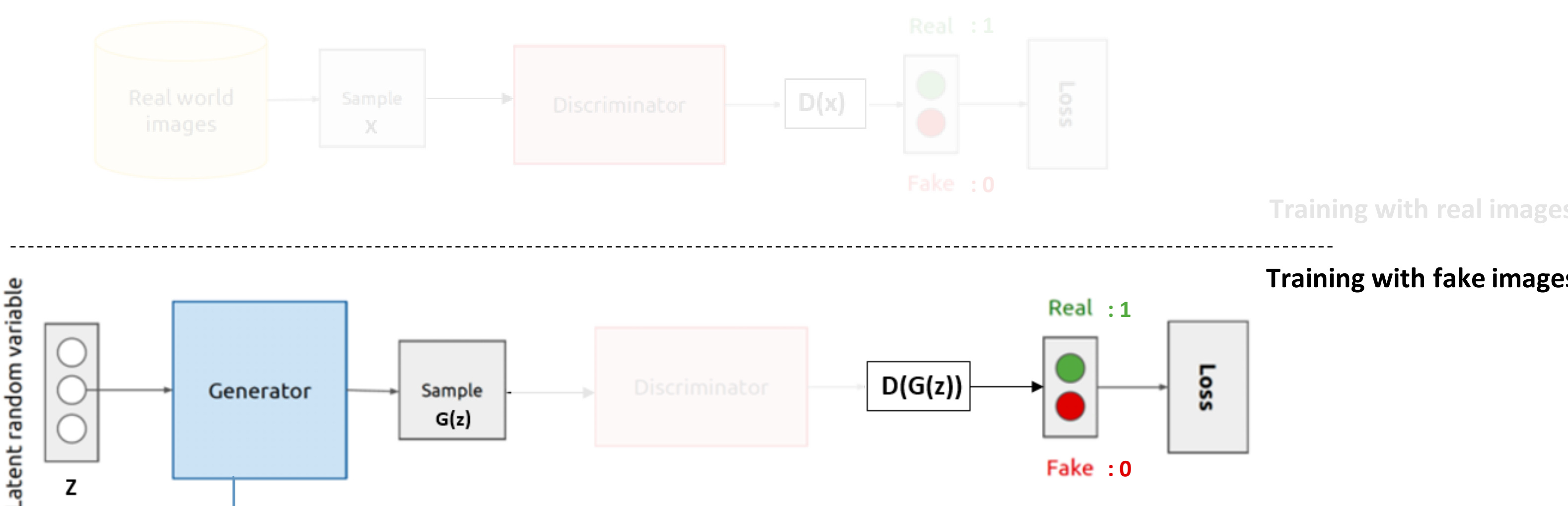
G is independent of this part

Sample latent code z from Gaussian distribution

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

G should minimize *V(D,G)*

Maximum when *D(G(z)) = 1*



Train G to generate fake image that D classify it real

Model

- **Why Objective Function works?**

$$\min_G \max_D V(D,G) \xrightarrow{\text{Same}} \min_{G,D} JSD\left(p_{data}||p_g\right)$$

Because it actually same with minimizing distance between the real data distribution & the model distribution

$$D_G^*(x) = arg\max_D V(D) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

$$arg\min_G V(D_G^*, G) = arg\min JSD\left(p_{data}||p_g\right)$$

$$\rightarrow\ when,\ p_{data}(x) = p_g(x)$$

$$\rightarrow So, D_G^*(x)\ will\ converge\ to\ \frac{1}{2}$$

Conclusion

Conclusion

- **Results**

- a) MNIST b) TFD c,d) CIFAR-10

Conclusion

- **Limitations of GAN**

  - Difficult to train model
  (because it is minimax optimization problem)

  *In training procedure, If one of them trained powerfully,
  other one did not train well and almost stopped.

  *Mode Collapse problem – generator makes similar images again

  - There is no obvious standard to stop training

Appendix

Appendix

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

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

$$= E_{x \sim p_{data}(x)}[\log D(x)] + E_{x \sim p_g(x)}[\log(1 - D(x))]$$

$$= \int_x p_{data}(x) \cdot \log D(x) + p_g(x) \cdot \log(1 - D(x)) \, dx$$

→ 이 적분안의 식이 최대가 되게하는 값 찾으면 된다.

$$p_{data}(x) \cdot \log D(x) + p_g(x) \cdot \log(1 - D(x))$$

$$\frac{d}{dD(x)}$$

$$p_{data}(x) \cdot \frac{1}{D(x)} + p_g(x) \cdot \frac{-1}{1 - D(x)}$$

$$= \frac{p_{data}(x) \cdot (1 - D(x)) - p_g(x) \cdot D(x)}{D(x) \cdot (1 - D(x))} = \frac{p_{data}(x) - (p_{data}(x) + p_g(x)) \cdot D(x)}{D(x) \cdot (1 - D(x))}$$

$\Rightarrow \quad D(x) = \dfrac{p_{data}(x)}{p_{data}(x) + p_g(x)}$ 일 때 0 이므로, 극대값을 가짐.

$$\therefore D^*(x) = p_{data}(x) / p_{data}(x) + p_g(x) \quad //$$

Appendix

$$\min_G \max_D V(D,G) \Rightarrow \min_G V(D^*,G)$$

$$\min_G V(D^*,G) = E_{x\sim p_{data}(x)}[\log D^*(x)] + E_{x\sim p_g(x)}[\log(1-D^*(x))]$$

$$D^*(x) = \frac{p_d}{p_d+p_g}$$

$$= \int_x p_{data}(x)\cdot\log D^*(x)\,dx + \int_x p_g(x)\cdot\log(1-D^*(x))\,dx$$

$$= \int_x p_{data}(x)\cdot\log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\,dx + \int_x p_g(x)\cdot\log\frac{p_g(x)}{p_{data}(x)+p_g(x)}\,dx$$

$$= -\log 4 + \log 4 + \int_x p_{data}(x)\cdot\log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\,dx + \int_x p_g(x)\cdot\log\frac{p_g(x)}{p_{data}(x)+p_g(x)}\,dx$$

$$= -\log 4 + \int_x p_{data}(x)\cdot\log\frac{p_{data}(x)}{\frac{p_{data}(x)+p_g(x)}{2}}\,dx + \int_x p_g(x)\cdot\log\frac{p_g(x)}{\frac{p_{data}(x)+p_g(x)}{2}}\,dx$$

$$= -\log 4 + KL\left(p_{data} \,\middle\|\, \frac{p_{data}+p_g}{2}\right) + KL\left(p_g \,\middle\|\, \frac{p_{data}+p_g}{2}\right)$$

$$= -\log 4 + 2\cdot JSD(p_{data}\,\|\,p_g)$$

이 식을 최소화 하는 것과 동일

& $p_{data} = p_g$ 일 때

이 식은 0이 됨.

$\therefore$ $p_{data} = p_g$ 일 때 최소

$$\therefore \min_G \max_D V(D,G) = \min_{G,D} JSD(p_{data}\,\|\,p_g)$$

Thank You