GANs

SCSC, Seoyoung Lim

Background

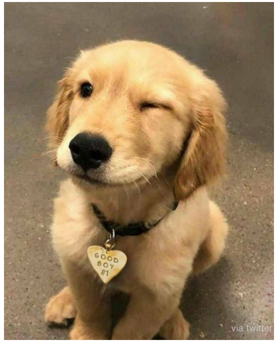
Discriminative Model vs Generative Model

Features Class

$$X \rightarrow Y$$

$$P(Y|X)$$

X가 강아지의 특징인 귀와 코라는
특징 일때, y라는 class인 개를
도출함



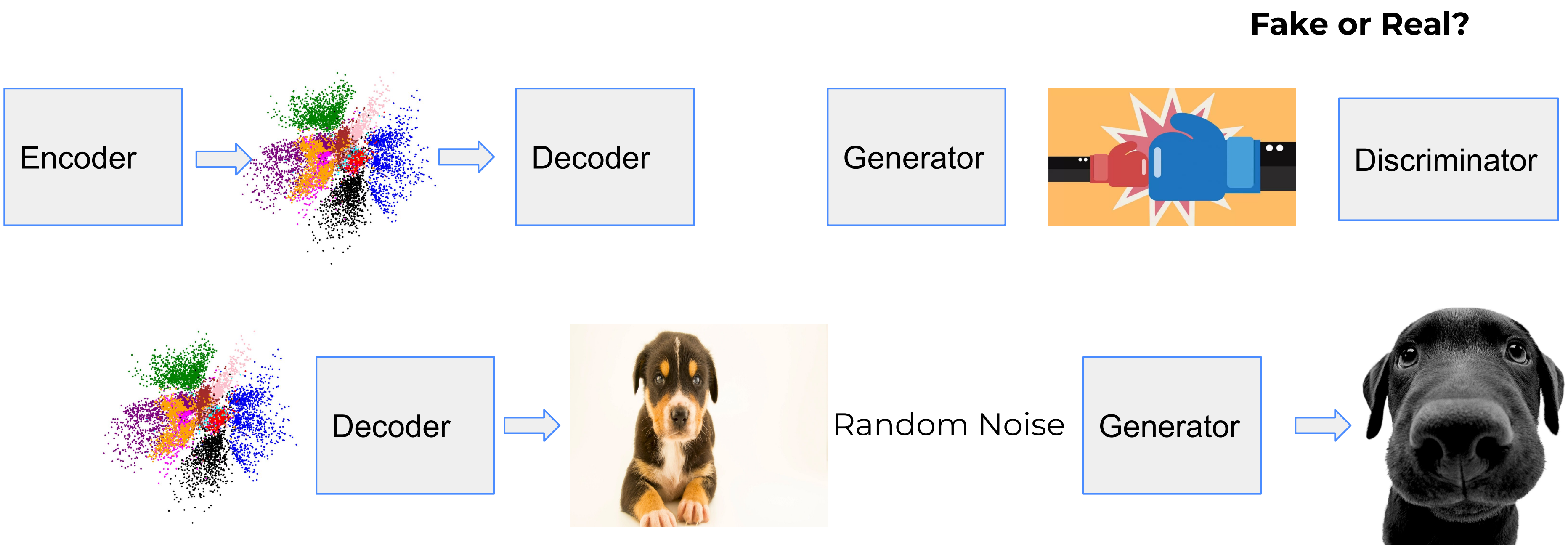
Noise Class Features

$$\xi, Y \rightarrow X$$

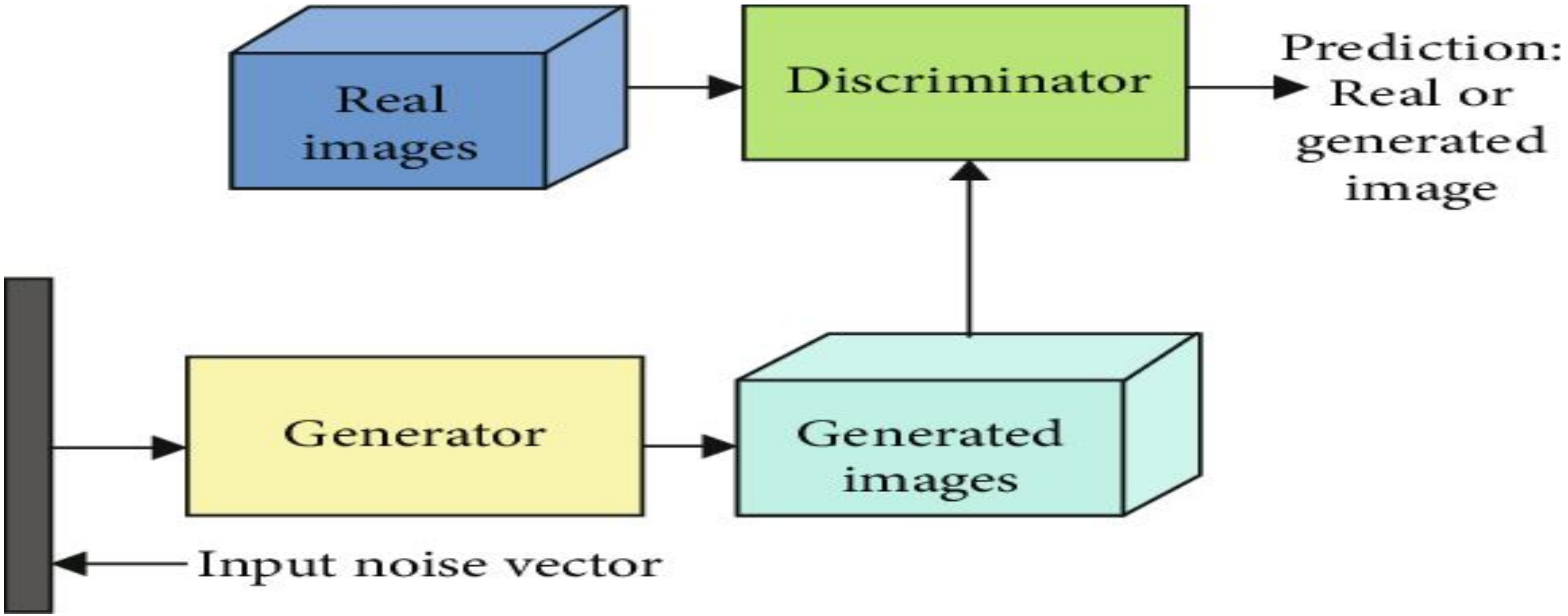
$$P(X|Y)$$

Background

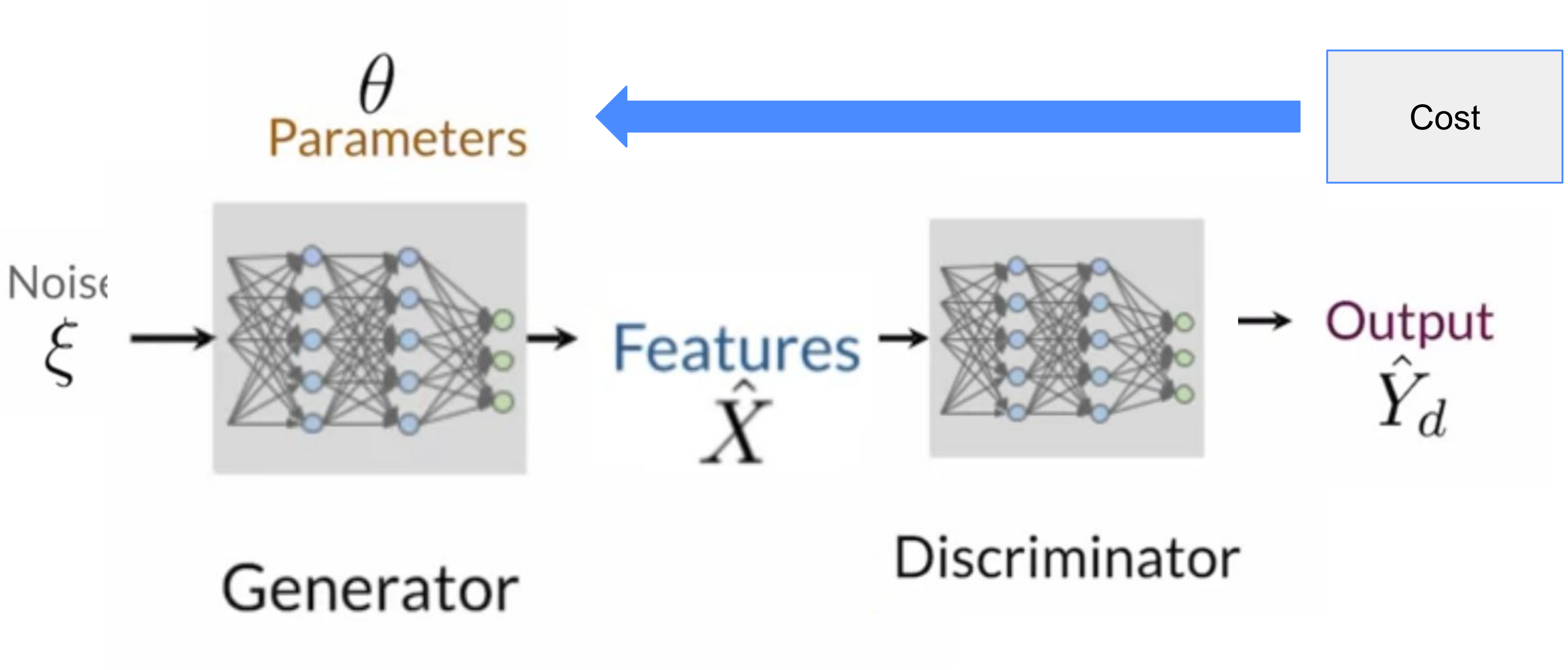
VAE vs GANs



GANs Model


Real
images
Discriminator
Prediction:
Real or
generated
image
Generator
Generated
images
Input noise vector

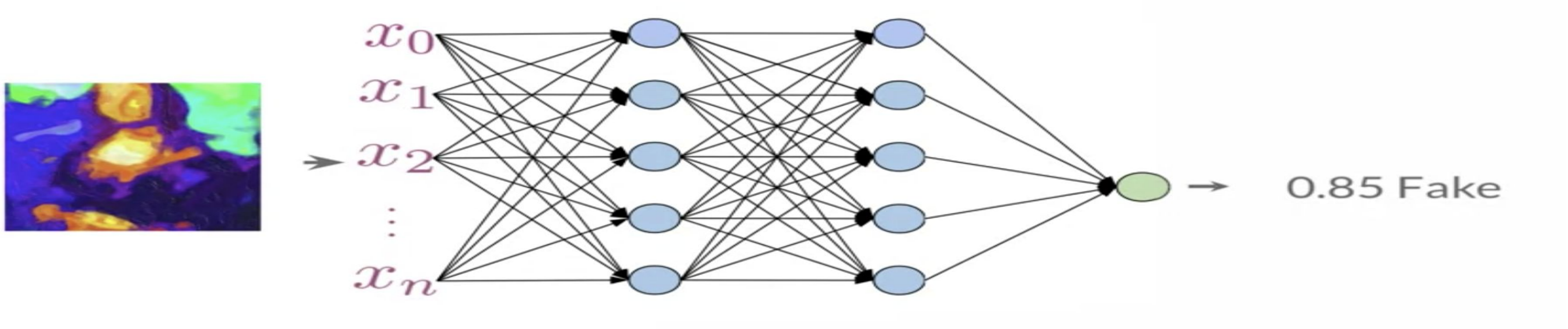
Generator


θ
Parameters
Cost
Noise
ξ
Features
$\hat{X}$
Output
$\hat{Y}_d$
Generator
Discriminator

Generator

1) The generator produces fake data
2) It learns the probability of features X
3) The generator takes as input noise (random features)

Discriminator



Discriminator

1) The discriminator is a classifier

2) It learns the probability of class Y

3) The probabilities are the feedback for the generator

논문

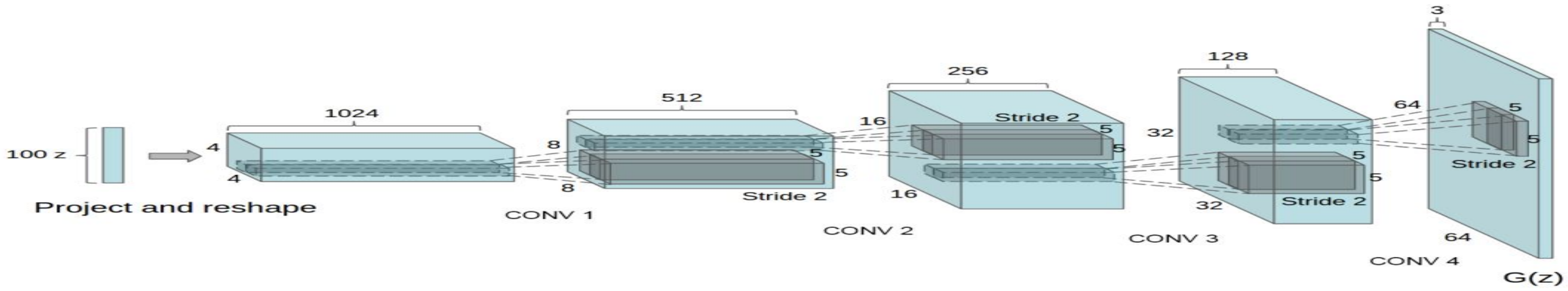
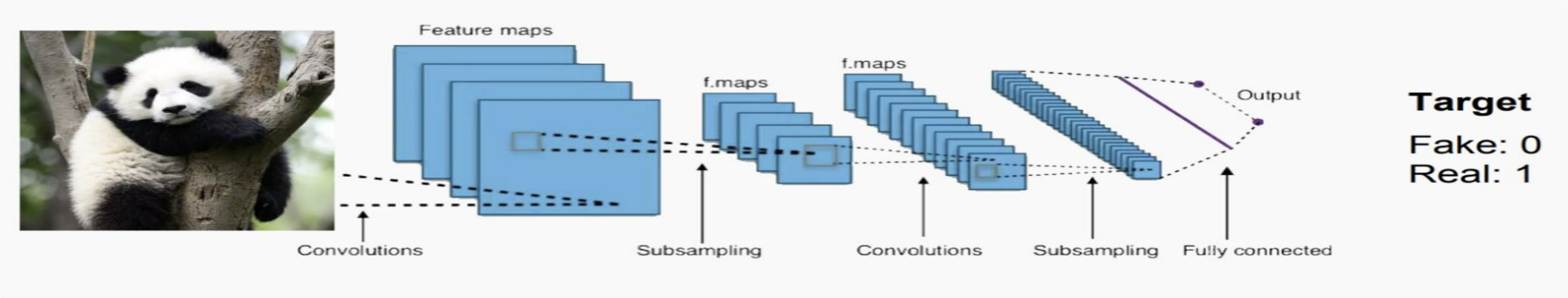


Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribu-

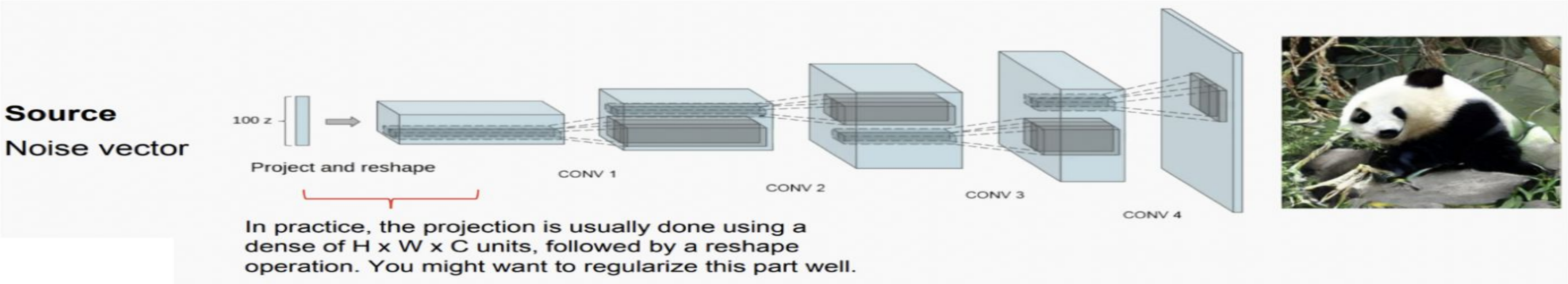
Table 1: CIFAR-10 classification results using our pre-trained model. Our DCGAN is not pre-trained on CIFAR-10, but on Imagenet-1k, and the features are used to classify CIFAR-10 images.

Model	Accuracy	Accuracy (400 per class)	max # of features units
1 Layer K-means	80.6%	63.7% (±0.7%)	4800
3 Layer K-means Learned RF	82.0%	70.7% (±0.7%)	3200
View Invariant K-means	81.9%	72.6% (±0.7%)	6400
Exemplar CNN	84.3%	77.4% (±0.2%)	1024
DCGAN (ours) + L2-SVM	82.8%	73.8% (±0.4%)	512

논문



Takes a vector of **noise *[N]*** and outputs an **image of *[H, W, C]***. The network has to perform synthesis. Again, we use a very minimalistic custom architecture.



Cost - 어떻게 학습?

BCE Cost Function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log h(x^{(i)},\theta) + (1-y^{(i)})\log(1-h(x^{(i)},\theta))]$$

Average loss of the whole batch

Cost - 어떻게 학습?

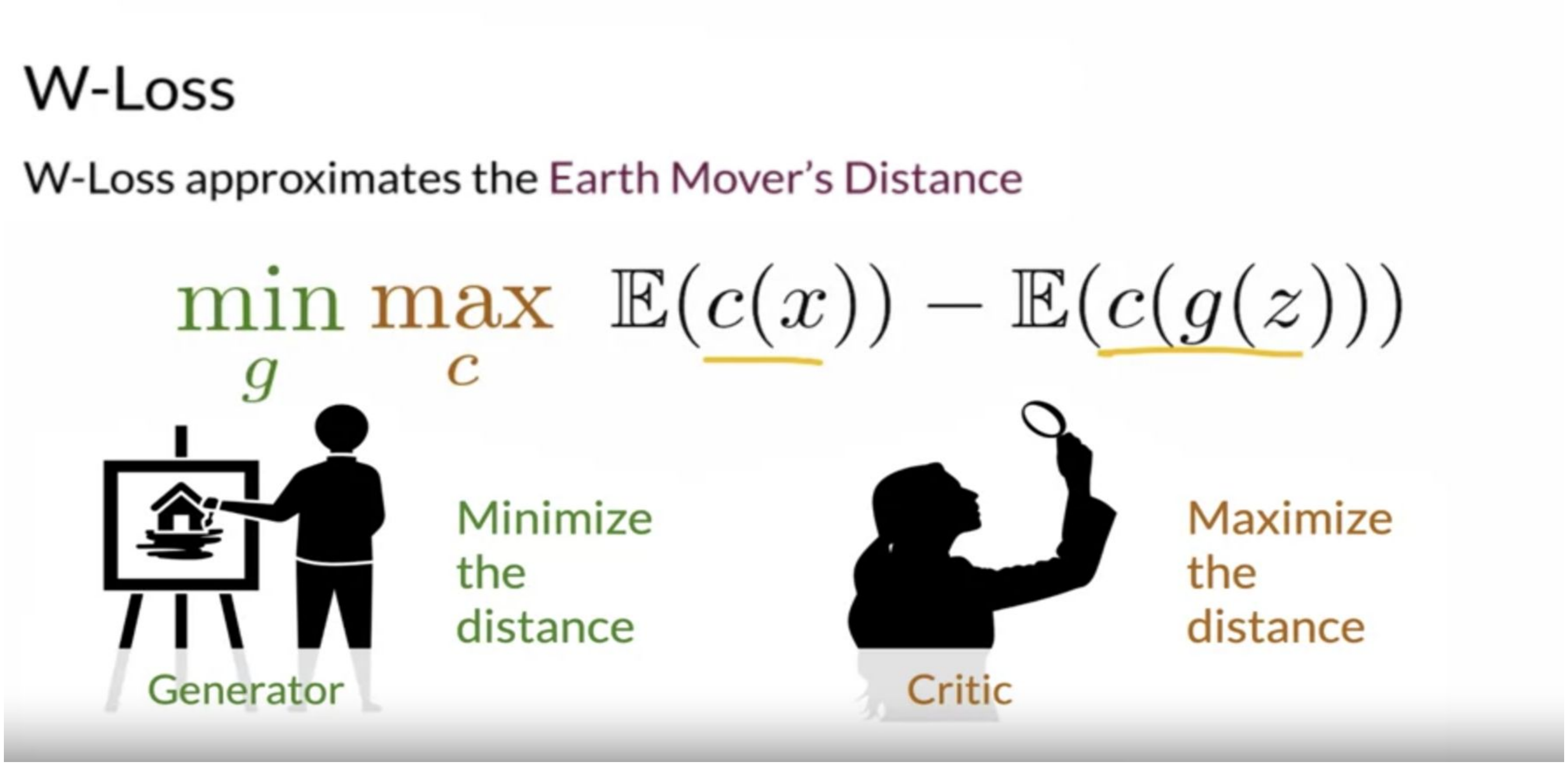
$$\min_{G} \max_{D} V(D,G) = E_{x \sim p_{data}(x)}[log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

$D(x) = 1$일 때 최대

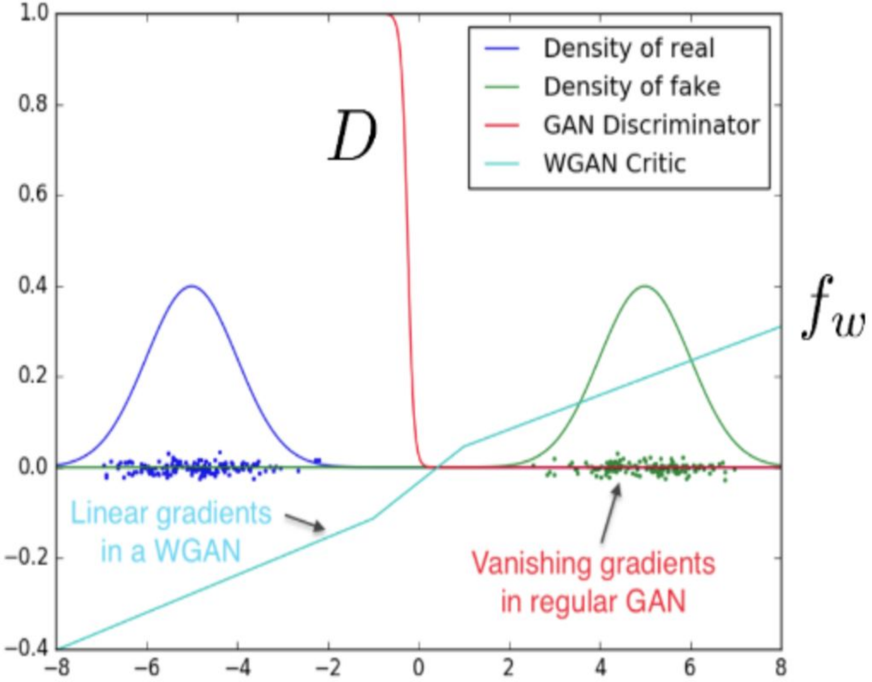
$D(G(z)) = 1$일 때 최소

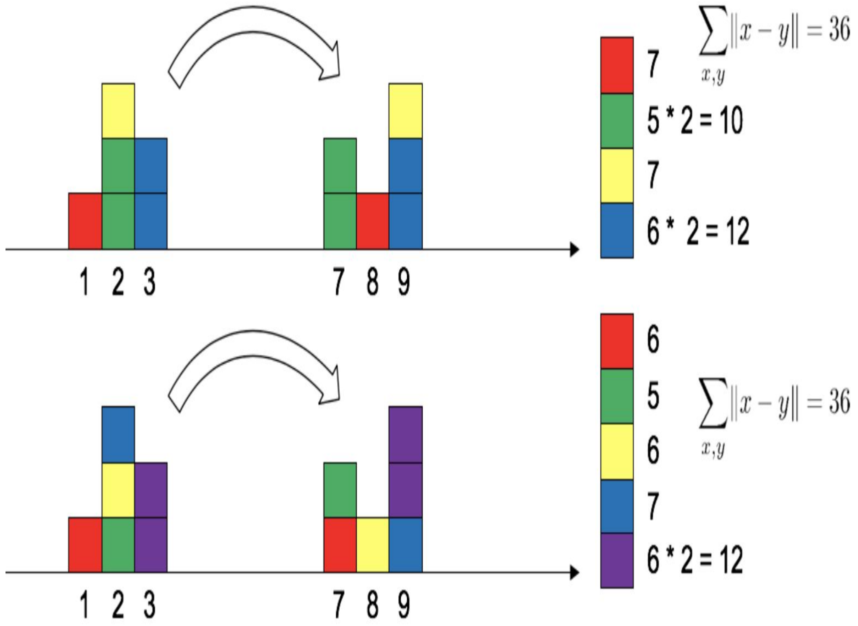
$D(G(z)) = 0$일 때 최대

Cost - 어떻게 학습?


W-Loss
W-Loss approximates the Earth Mover's Distance
$\min_{g} \max_{c} \; \mathbb{E}(c(x)) - \mathbb{E}(c(g(z)))$
Minimize the distance
Generator
Maximize the distance
Critic

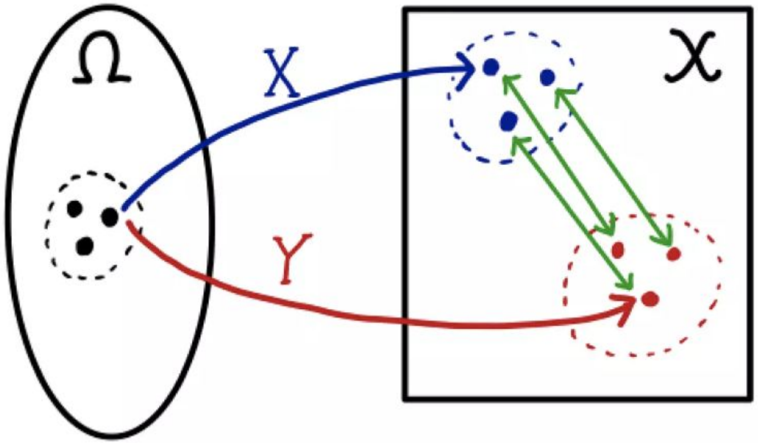
Cost - 어떻게 학습?- Wasserstein GAN





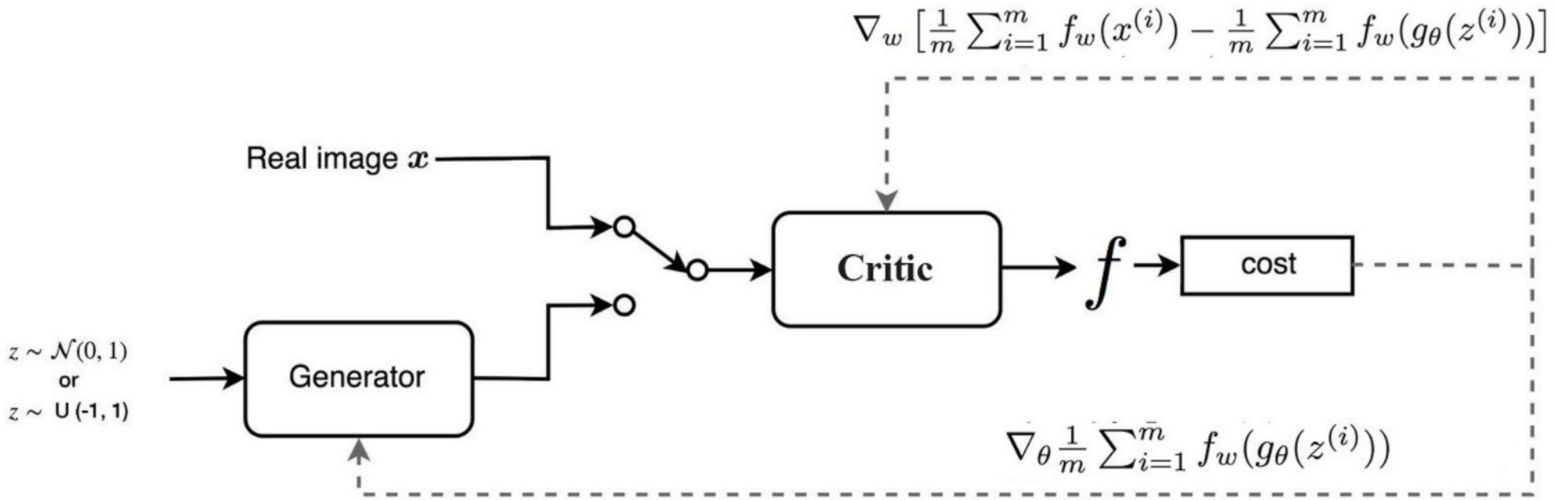
Cost - 어떻게 학습?- Wasserstein GAN

Wasserstein distance 는 이렇게 여러가지 γ 중에서 $d(X, Y)$ 의 기대값이 가장 작게 나오는 확률분포를 취합니다! 😃



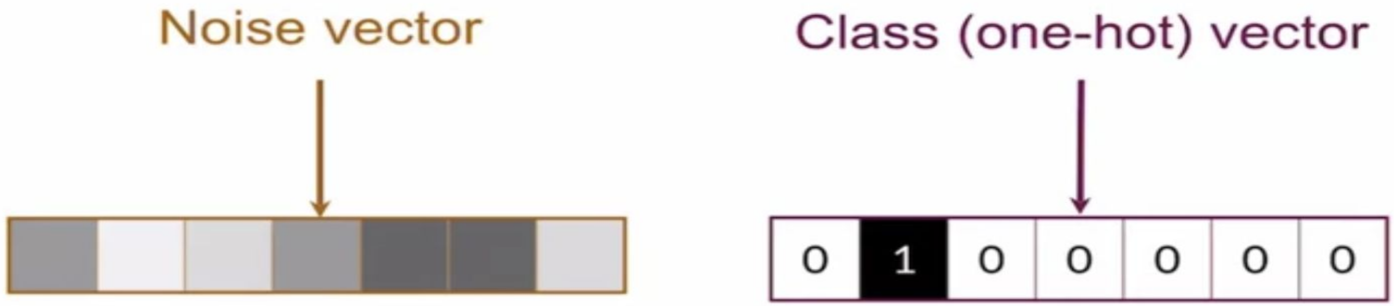
Cost - 어떻게 학습?- Wasserstein GAN

WGAN

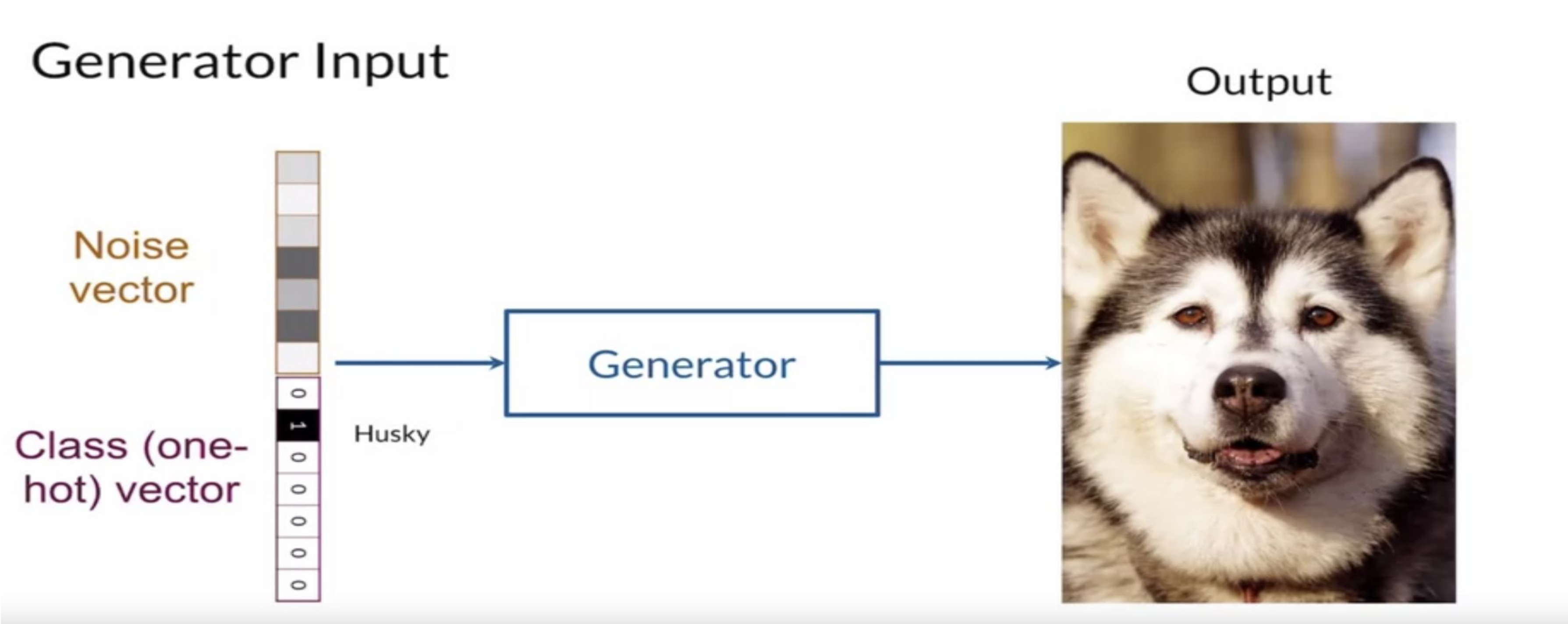


출처 : GAN — Wasserstein GAN & WGAN-GP

Conditional GANs



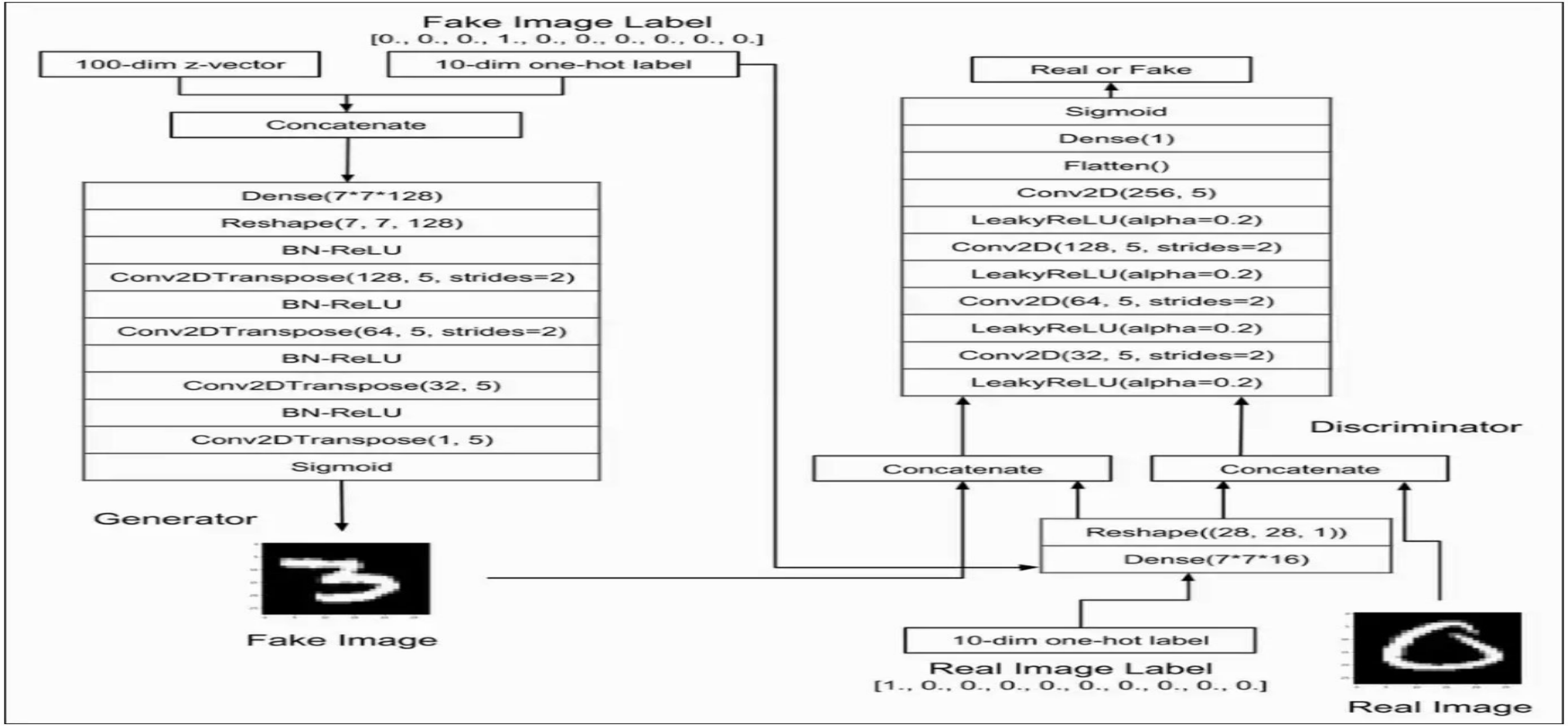
Conditional GANs



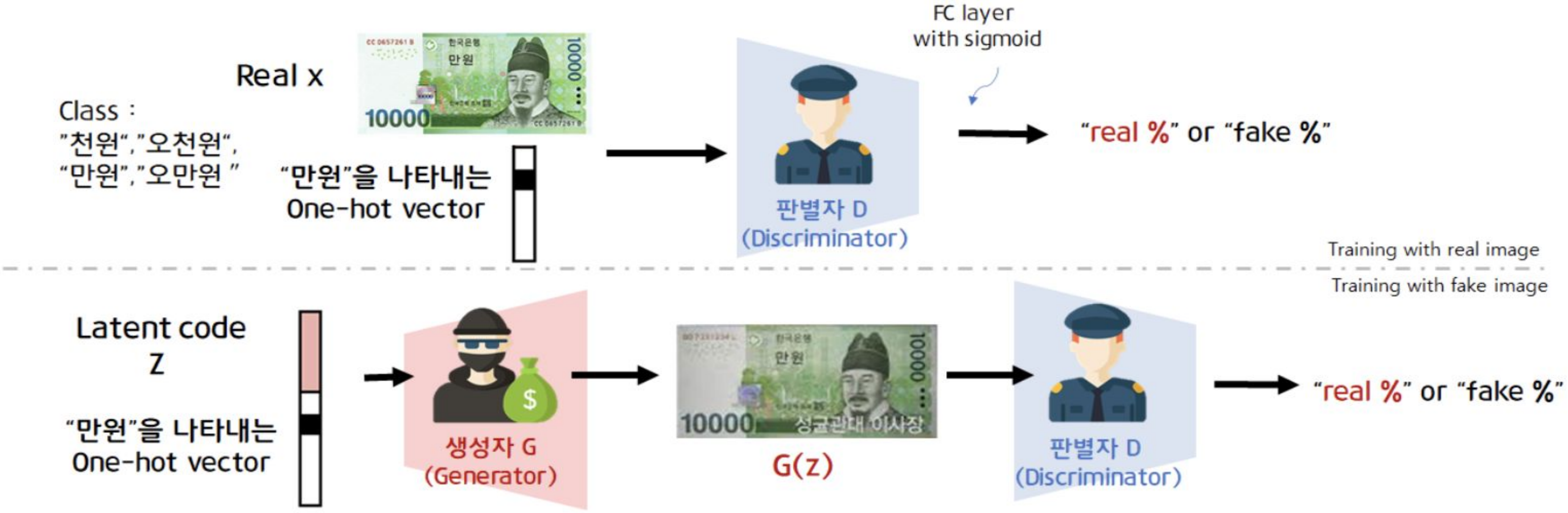
Conditional GANs

1) The class is passed to the generator as one-hot vectors
2) The class is passed to the discriminator as one-hot matrices
3) The size of the vector and the number of matrices represent the number of classes

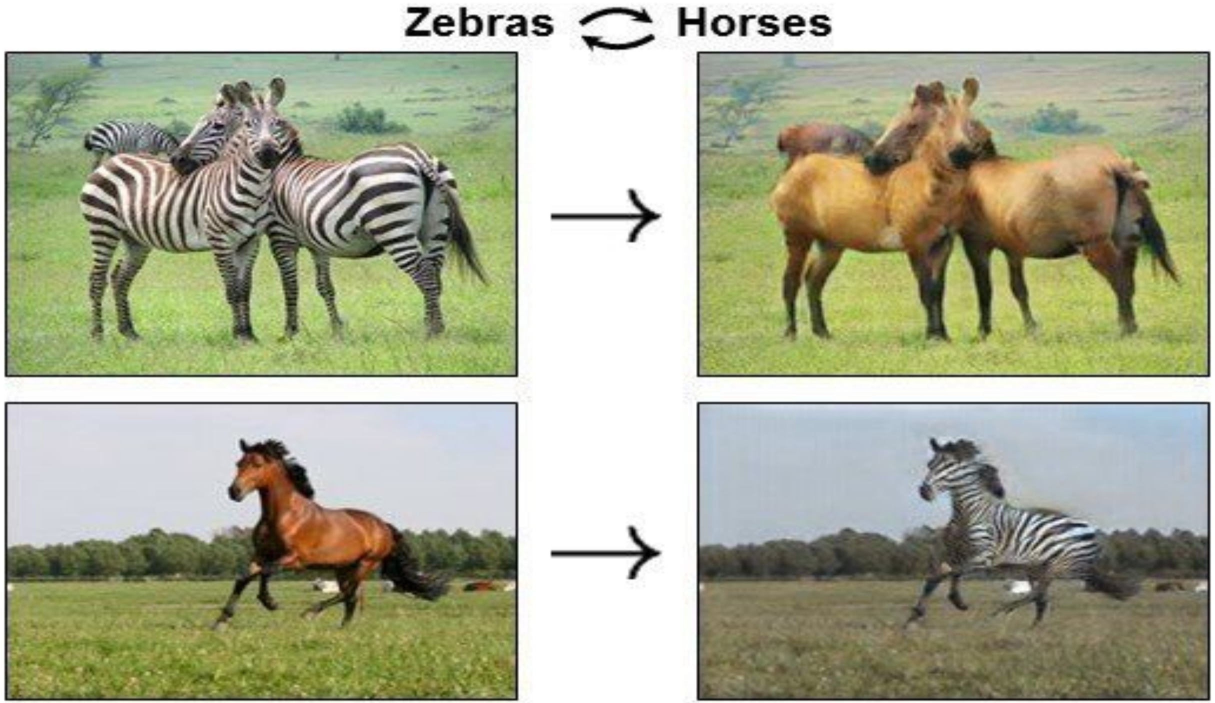
CGANs


Fake Image Label
[0., 0., 0., 1., 0., 0., 0., 0., 0., 0.]
100-dim z-vector
10-dim one-hot label
Concatenate
Dense(7*7*128)
Reshape(7, 7, 128)
BN-ReLU
Conv2DTranspose(128, 5, strides=2)
BN-ReLU
Conv2DTranspose(64, 5, strides=2)
BN-ReLU
Conv2DTranspose(32, 5)
BN-ReLU
Conv2DTranspose(1, 5)
Sigmoid
Generator
Fake Image
Real or Fake
Sigmoid
Dense(1)
Flatten()
Conv2D(256, 5)
LeakyReLU(alpha=0.2)
Conv2D(128, 5, strides=2)
LeakyReLU(alpha=0.2)
Conv2D(64, 5, strides=2)
LeakyReLU(alpha=0.2)
Conv2D(32, 5, strides=2)
LeakyReLU(alpha=0.2)
Discriminator
Concatenate
Concatenate
Reshape((28, 28, 1))
Dense(7*7*16)
10-dim one-hot label
Real Image Label
[1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
Real Image

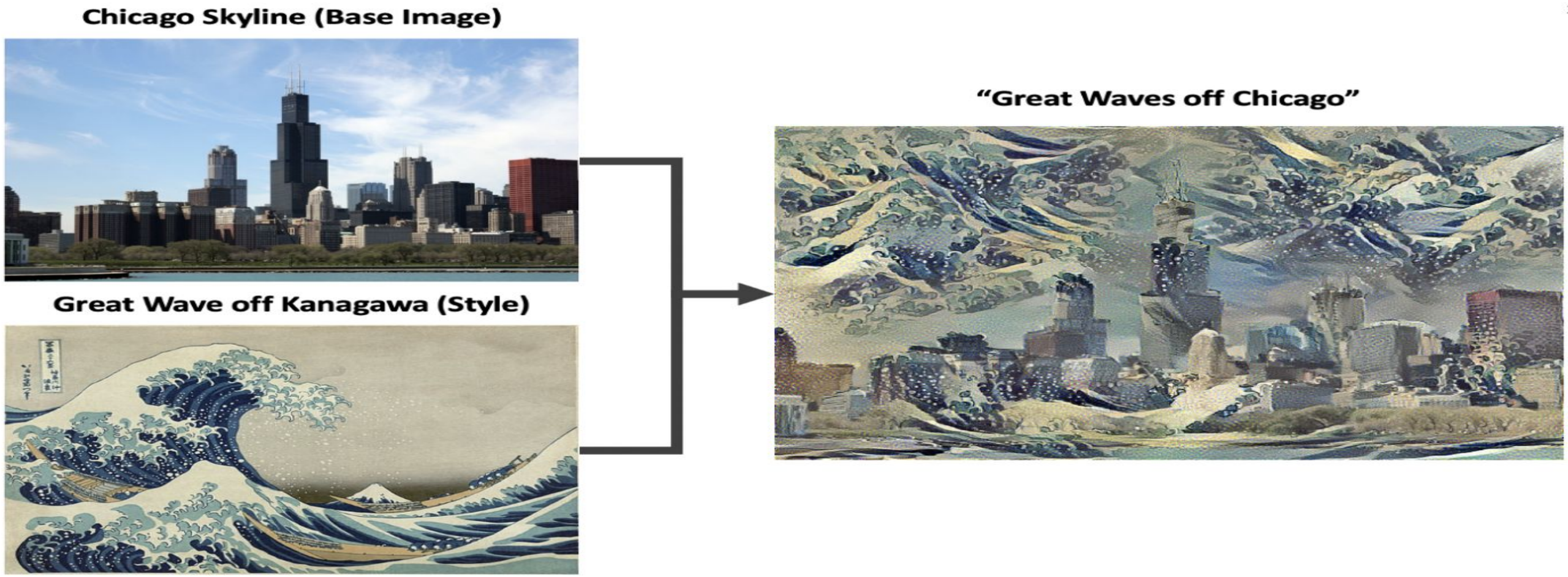
CGANs


Real x
Class :
"천원","오천원",
"만원","오만원 "
"만원"을 나타내는
One-hot vector
FC layer
with sigmoid
판별자 D
(Discriminator)
"real %" or "fake %"
Training with real image
Training with fake image
Latent code
Z
"만원"을 나타내는
One-hot vector
생성자 G
(Generator)
G(z)
판별자 D
(Discriminator)
"real %" or "fake %"

Application



Application - Cycle GANs



Combining the style of the famed "Great Wave off Kanagawa" with the Chicago skyline.

References - influential papers

- DCGAN https://arxiv.org/pdf/1511.06434v2.pdf
- Wasserstein GAN (WGAN) https://arxiv.org/pdf/1701.07875.pdf
- Conditional Generative Adversarial Nets (CGAN) https://arxiv.org/pdf/1411.1784v1.pdf
- Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks (LAPGAN) https://arxiv.org/pdf/1506.05751.pdf
- Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network (SRGAN) https://arxiv.org/pdf/1609.04802.pdf
- Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks (CycleGAN) https://arxiv.org/pdf/1703.10593.pdf
- InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets https://arxiv.org/pdf/1606.03657
- DCGAN https://arxiv.org/pdf/1704.00028.pdf
- Improved Training of Wasserstein GANs (WGAN-GP) https://arxiv.org/pdf/1701.07875.pdf
- Energy-based Generative Adversarial Network (EBGAN) https://arxiv.org/pdf/1609.03126.pdf
- Autoencoding beyond pixels using a learned similarity metric (VAE-GAN) https://arxiv.org/pdf/1512.09300.pdf
- Adversarial Feature Learning (BiGAN) https://arxiv.org/pdf/1605.09782v6.pdf
- Stacked Generative Adversarial Networks (SGAN) https://arxiv.org/pdf/1612.04357.pdf
- StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks https://arxiv.org/pdf/1710.10916.pdf
- Learning from Simulated and Unsupervised Images through Adversarial Training (SimGAN) https://arxiv.org/pdf/1612.07828v1.pdf