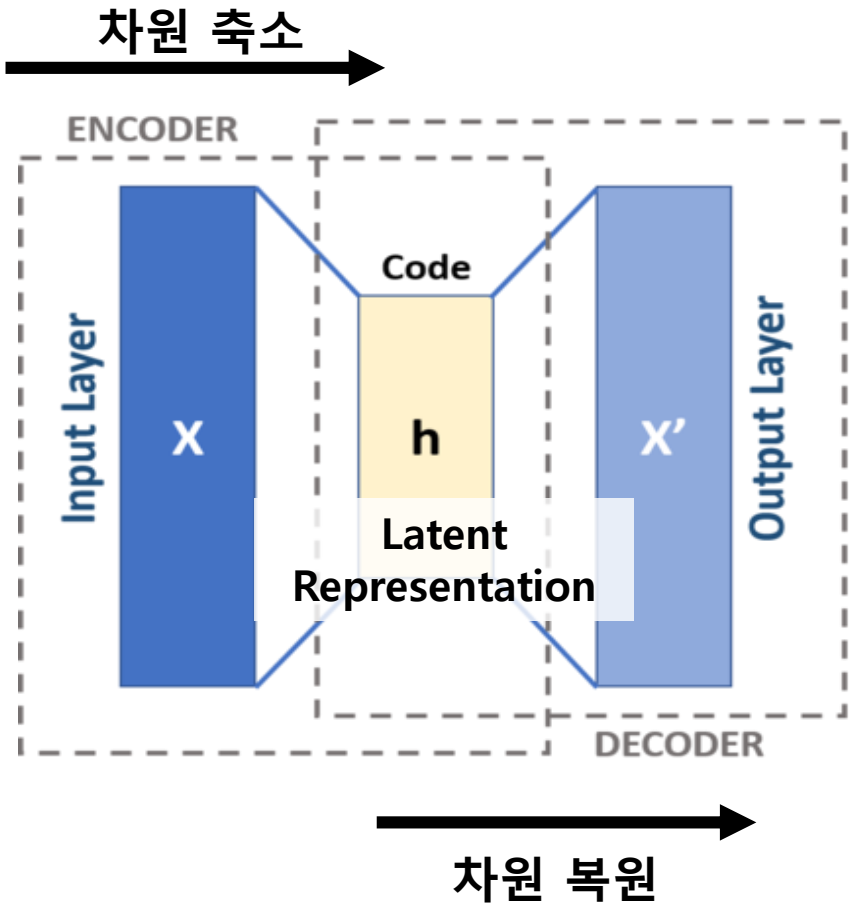
생성모델과 이상감지

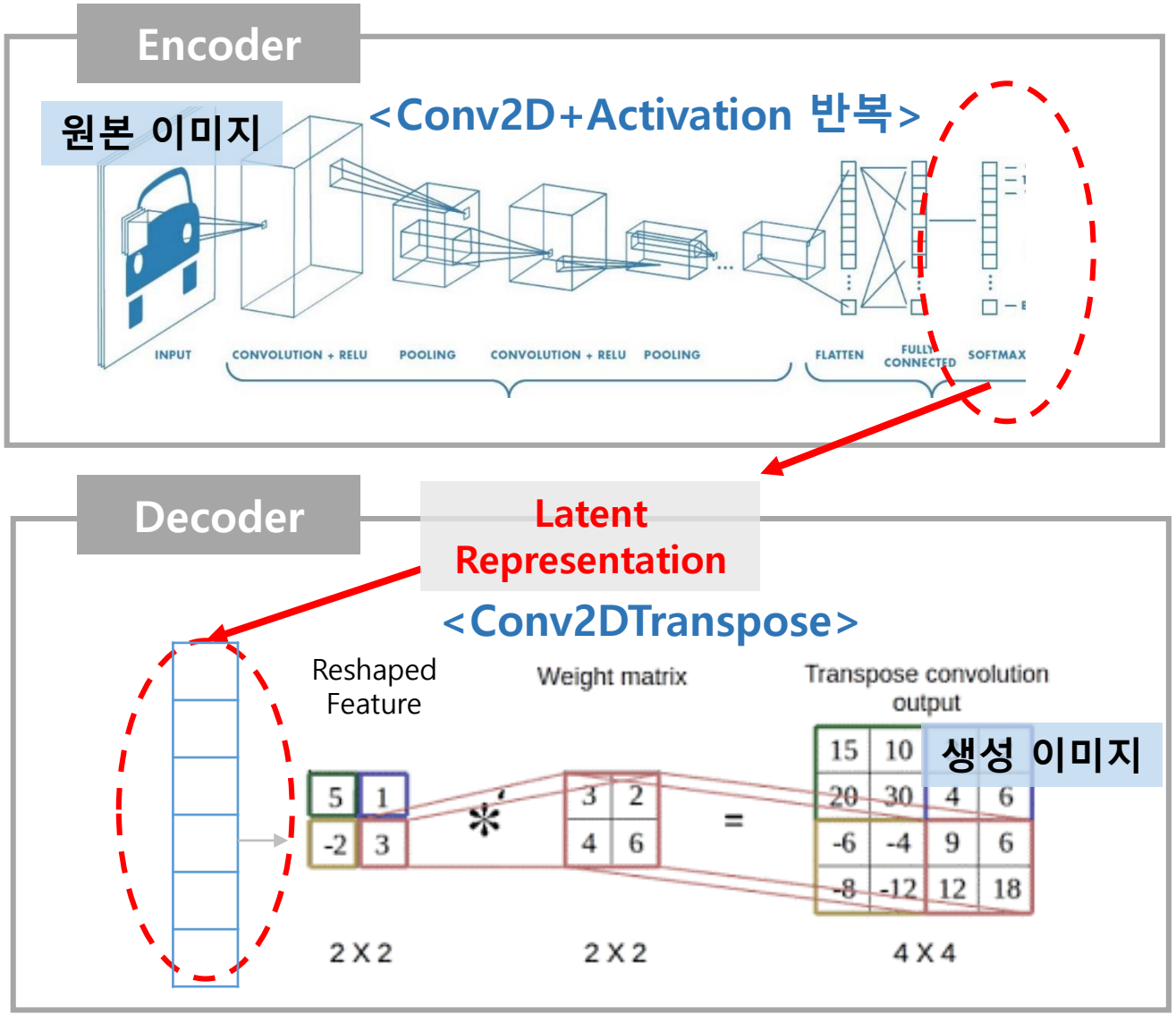
이론 정리_김유리

목차

1. Autoencoder & Variational Autoencoder
    + Anomaly Detection
    + LSTM with AE
2. GAN
    + Anomaly Detection
    + GAN의 한계와 극복

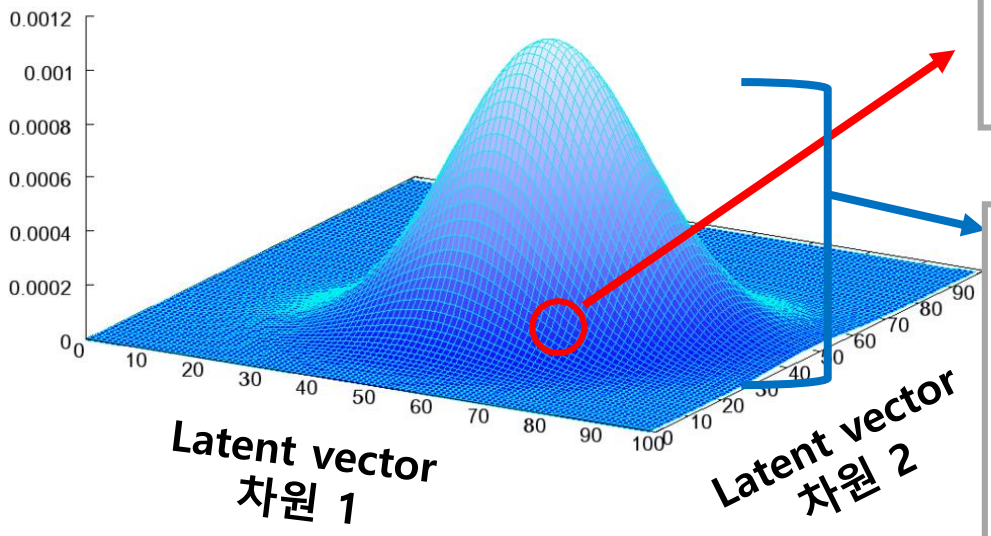
1.1. AE, VAE 공통


차원 축소
ENCODER
Input Layer
X
Code
h
Latent Representation
X'
Output Layer
DECODER
차원 복원


Encoder
원본 이미지
<Conv2D+Activation 반복>
INPUT
CONVOLUTION + RELU
POOLING
CONVOLUTION + RELU
POOLING
FLATTEN
FULLY CONNECTED
SOFTMAX
Decoder
Latent Representation
<Conv2DTranspose>
Reshaped Feature
Weight matrix
Transpose convolution output
생성 이미지
2 X 2
2 X 2
4 X 4

1.2. AE, VAE 차이점

차이점1. 잠재 공간 맵핑



AE:
잠재 공간의 한 포인트에 매핑
잠재 공간의 latent vector를 z'라 하자

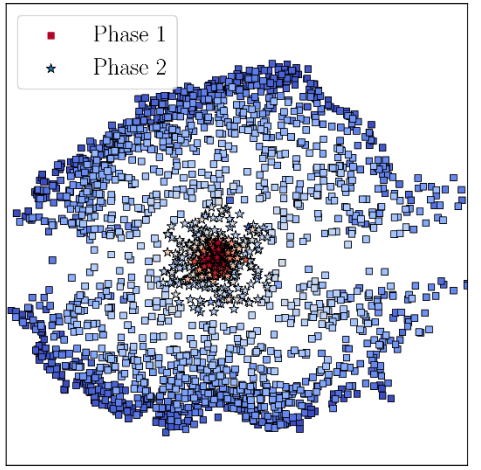
VAE:
랜덤샘플링으로 Noise를 추가한다.

정규분포 noise의 경우,
축소된 차원 벡터 x를 완전 연결층으로 연결해 정규분포의 모수 mu, sigma 생성 후 잠재 벡터 z 생성

mu=Dense_Layer(z')
sigma=Dense_Layer(z')

z=mu+sigma*epsilon,
epsilon~ N(0,1)
-> VAE의 latent vector는 z

<Autoencoder 2차원 매핑 예시>



- AE: 포인트로 매핑 -> 값이 연속적이지 않음
- 조밀하지 않은 잠재공간 상 포인트 디코딩 시 제대로 이미지 복원이 되지 않음,
  + 잠재 공간의 차원이 커질수록 빈 공간 많아짐
- VAE: 분포 이용해 연속 잠재 공간 생성

1.2. AE, VAE 차이점

차이점2. Loss function

Loss: 원본 이미지, 생성 이미지 간 차이
- AE: MAE or MSE (픽셀 값 자체)
- VAE: MAE(픽셀 값 자체)+KL divergence (잠재 벡터 분포)

1

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2 = \|\mathbf{x} - \sigma'(\mathbf{W}'(\sigma(\mathbf{W}\mathbf{x} + \mathbf{b})) + \mathbf{b}')\|^2$$

원 이미지 픽셀, 생성 이미지 픽셀 차이의 L1 or L2 loss

2

$$D_{KL}(N(\mu, \sigma)||N(0,1)) = \frac{1}{2}\Sigma(1 + log(\sigma^2) - \mu^2 - \sigma^2)$$

잠재 공간의 분포와 표준 정규 분포의 Kullback Leilber 발산

AE

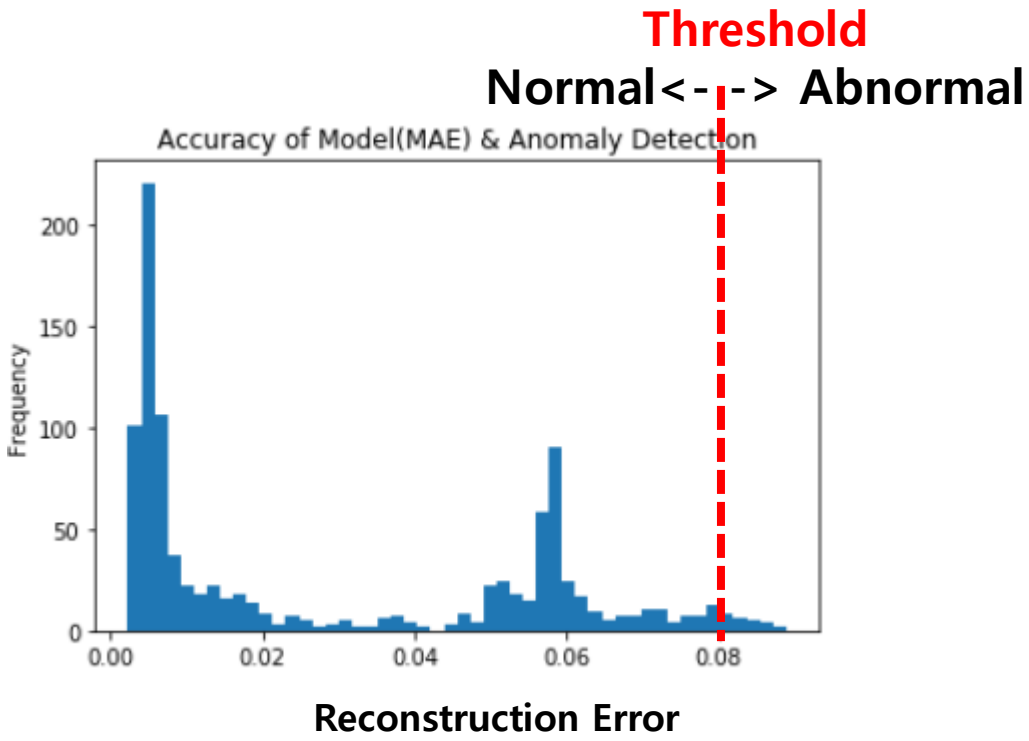
\+

VAE

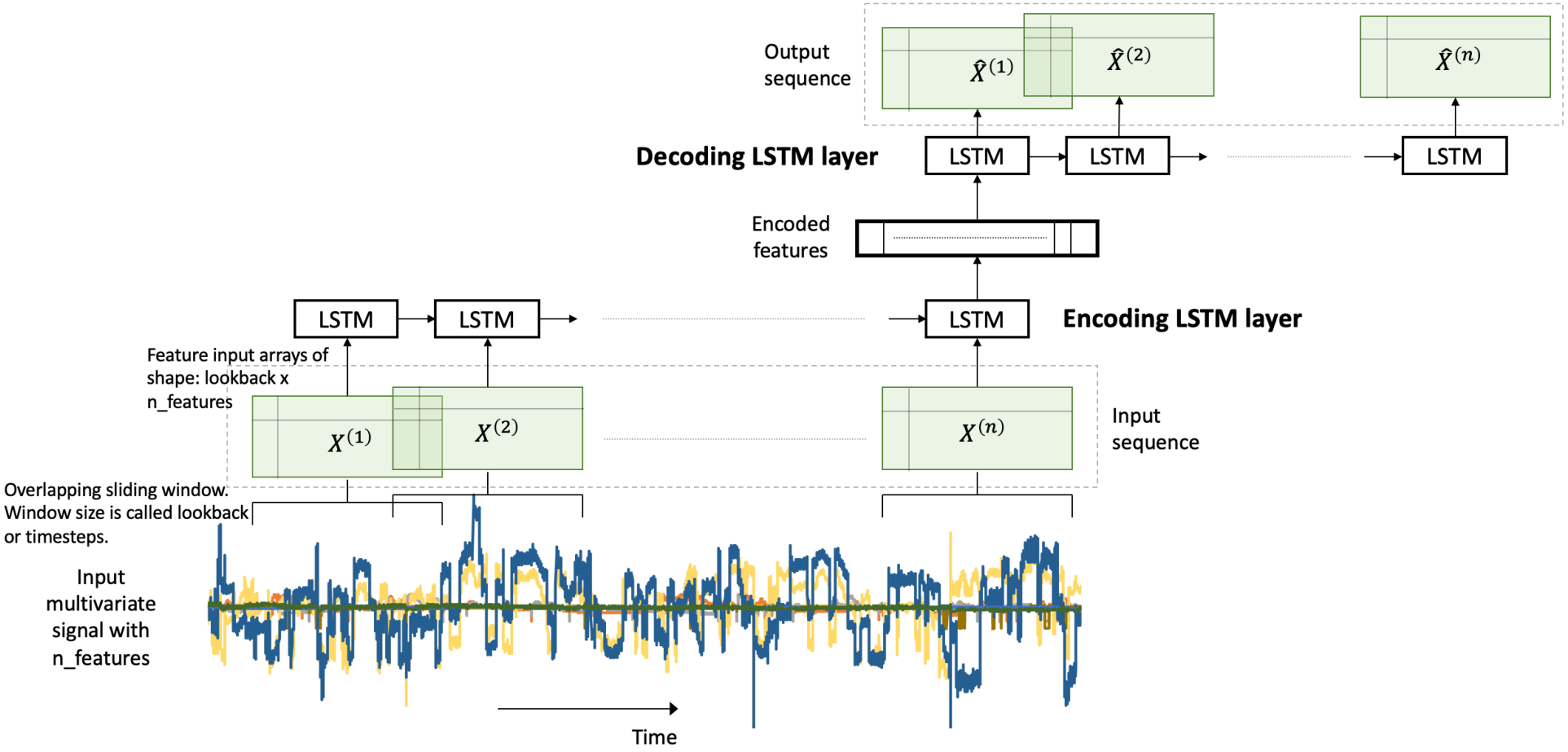
차원 축소 시 잠재 공간에 맵핑된 값들이 N(0,1) 분포에 가까워지도록 모델이 훈련된다.
-> 잠재 벡터가 N(0,1) 안에서 선택되도록 유도되고, 포인트 간 간격이 너무 멀어지지 않으며,
잠재 공간을 대칭, 효과적으로 사용하도록 유도됨.

1.3. AE, VAE + Anomaly Detection

정상 데이터로 AE or VAE 학습



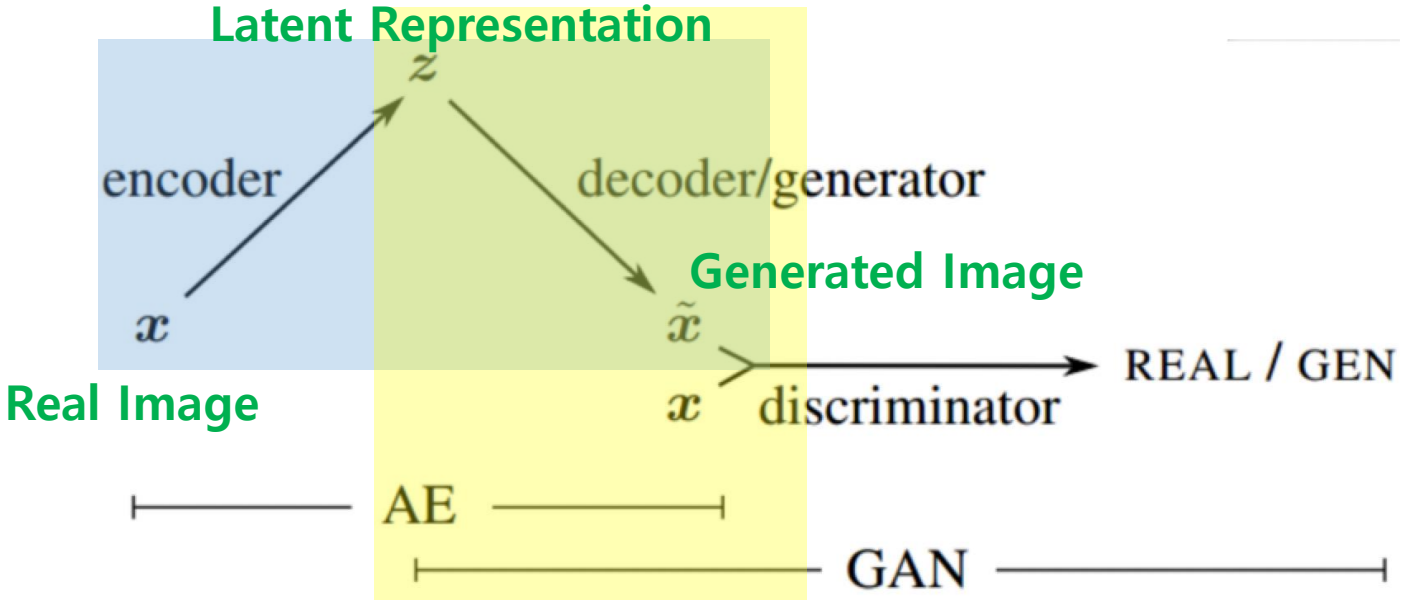
1.4. AE with LSTM



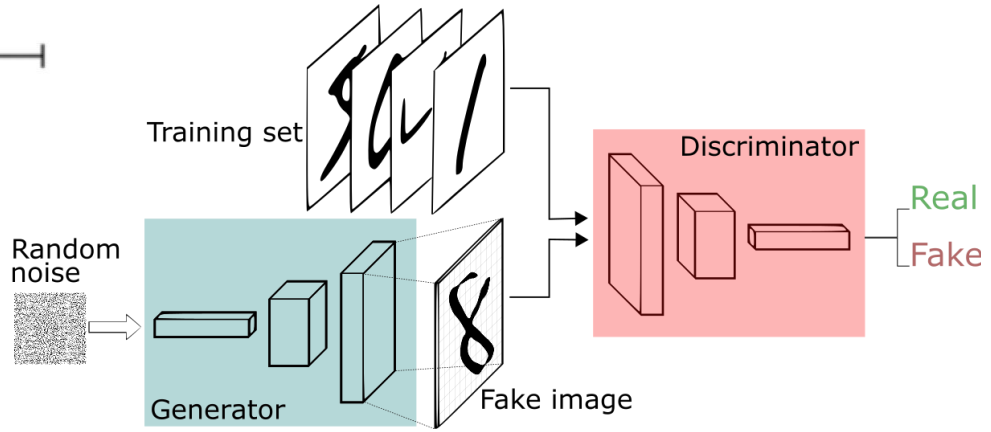
ConvLSTM2D (KERAS)

2.1. GAN, AE(or VAE) 차이점

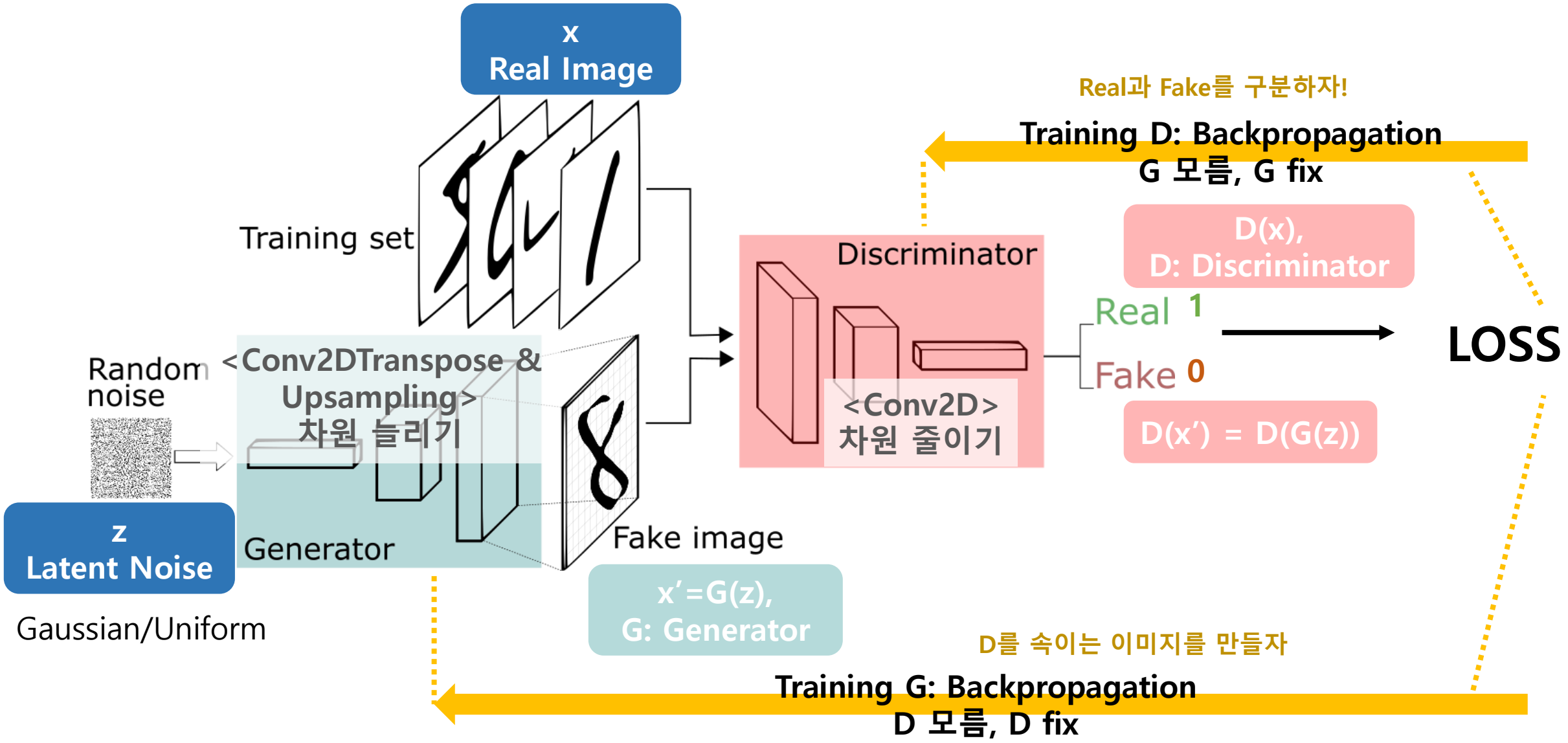
AE: 실제 이미지(x)와 생성 이미지(x')를 잠재 공간에 직접 매핑(z)



GAN: 랜덤 잡음 z로부터 이미지 생성(x'),
대신 discriminator 속이도록 훈련



2.2. GAN


x
Real Image
Training set
Real과 Fake를 구분하자!
Training D: Backpropagation
G 모름, G fix
D(x),
D: Discriminator
Discriminator
Real 1
Fake 0
LOSS
<Conv2D>
차원 줄이기
D(x') = D(G(z))
Random
noise
<Conv2DTranspose &
Upsampling>
차원 늘리기
z
Latent Noise
Generator
Fake image
x'=G(z),
G: Generator
Gaussian/Uniform
D를 속이는 이미지를 만들자
Training G: Backpropagation
D 모름, D fix

GAN's formulation

$$\min_G \max_D V(D,G)$$

- It is formulated as a **minimax game**, where:
  - The Discriminator is trying to maximize its reward $\boldsymbol{V(D,G)}$
  - The Generator is trying to minimize Discriminator's reward (or maximize its loss)

$$V(D,G) = \mathbb{E}_{x\sim p(x)}[\log D(x)] + \mathbb{E}_{z\sim q(z)}\left[\log\left(1 - D(G(z))\right)\right]$$

- The Nash equilibrium of this particular game is achieved at:
  - $P_{data}(x) = P_{gen}(x) \;\; \forall x$
  - $\mathrm{D}(x) = \frac{1}{2} \;\; \forall x$

게임 이론에서 경쟁자 대응에 따라
최선의 선택을 하면
서로가 자신의 선택을 바꾸지 않는 균형상태

2.3. GAN's formulation

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

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

end for

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

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

end for

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

Discriminator updates

Generator updates

D, G의 min, max 대결!

D=1, real
D=0, fake

진짜 이미지

가짜 이미지

Real Image, Generated
이미지 구분 잘하는 것을 Maximize

Discriminator가 가짜 이미지 구분하는 값을 minimize,
즉 D가 가짜 구분 못하도록 G 학습

2.4. GAN + Anomaly Detection

<Mapping new Images to the Latent Space>

GAN은 실제 이미지(x)를 latent representation(z)으로 매핑하지 않는다.
Random Noise z로부터 가짜 이미지를 생성하도록 모형 구조가 형성되어있다.

따라서 test set에 대해 이상 탐지를 하기 위해서는
개별 이미지의 $\hat{z}$를 찾은 후 앞서 학습한 GAN 모형에 넣어야 한다.
(개별 이미지의 각 $\hat{z}$로 이미지를 생성(G($\hat{z}$))했을 때 원래 이미지로 잘 복원이 되는가)

λ는 대략 0.1정도

Overall Loss $$\mathcal{L}(\mathbf{z}_\gamma) = (1-\lambda)\cdot\mathcal{L}_R(\mathbf{z}_\gamma) + \lambda\cdot\mathcal{L}_D(\mathbf{z}_\gamma)$$

γ : γth backpropagation iteration

Where $$\mathcal{L}_R(\mathbf{z}_\gamma) = \sum|\mathbf{x} - G(\mathbf{z}_\gamma)|$$ (Residual Loss): Generator 입장 Loss

$$\mathcal{L}_D(\mathbf{z}_\gamma) = \sum|\mathbf{f}(\mathbf{x}) - \mathbf{f}(G(\mathbf{z}_\gamma))|$$ (Discrimination Loss), f: Disc. function

앞서 훈련한 GAN 모형($\hat{D}$, $\hat{G}$) & 적합한 latent representation($\hat{z}$)를 Overall Loss 식에 넣는다.

$$\rightarrow \hat{z} = argmin_z L(z)$$

Anomaly Score $$A(\mathbf{x}) = (1-\lambda)\cdot R(\mathbf{x}) + \lambda\cdot D(\mathbf{x})$$

2.5. GAN의 한계와 극복

<한계 & 극복>

원본이미지(x)와 생성이미지(x')가 매핑된 관계가 아니고, two players(G,D)라서 훈련이 어렵다.

1. Vanishing Gradient

$$\min_G \max_D V(D,G)$$

$$V(D,G) = \mathbb{E}_{x\sim p(x)}[\log D(x)] + \boxed{\mathbb{E}_{z\sim q(z)}\left[\log\left(1 - D(G(z))\right)\right]}$$

-> Minimize $\boxed{-\mathbb{E}_{z\sim q(z)}\left[\log D\left(G(z)\right)\right]}$ for Generator instead **로 바꾼다**

2. Backpropagation 수렴 문제: G, D 두 player의 최적화 문제 -> SGD는 내쉬균형 보장 못함

3. 모드 붕괴: 새로운 이미지 생성 못함

2&3 해결책:

- **Mini-Batch GANs**
- **Supervision with labels**

- **Some recent attempts :-**
  - Unrolled GANs
  - W-GANs

Appendix

Appendix. (1) 정규분포 간 KL 발산 계산, univariate

<Definition of KL>

$$D_{\mathrm{KL}}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

<KL of Normal Dist. Case>

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$X_1 \sim N(\mu_1, \sigma_1^2)$ from another $X_2 \sim N(\mu_2, \sigma_2^2)$

$$D_{\mathrm{KL}}(X_1 \parallel X_2) = \frac{(\mu_1 - \mu_2)^2}{2\sigma_2^2} + \frac{1}{2}\left(\frac{\sigma_1^2}{\sigma_2^2} - 1 - \ln\frac{\sigma_1^2}{\sigma_2^2}\right)$$

$$\int [\log(p(x)) - log(q(x))]\, p(x) dx$$

$$= \int \left[-\tfrac{1}{2}\log(2\pi) - \log(\sigma_1) - \tfrac{1}{2}\left(\tfrac{x-\mu_1}{\sigma_1}\right)^2 + \tfrac{1}{2}\log(2\pi) + \log(\sigma_2) + \tfrac{1}{2}\left(\tfrac{x-\mu_2}{\sigma_2}\right)^2 \right]$$

$$\times \tfrac{1}{\sqrt{2\pi}\sigma_1} \exp\left[-\tfrac{1}{2}\left(\tfrac{x-\mu_1}{\sigma_1}\right)^2 \right] dx$$

$$= \int \left\{ \log\left(\tfrac{\sigma_2}{\sigma_1}\right) + \tfrac{1}{2}\left[\left(\tfrac{x-\mu_2}{\sigma_2}\right)^2 - \left(\tfrac{x-\mu_1}{\sigma_1}\right)^2 \right] \right\} \times \tfrac{1}{\sqrt{2\pi}\sigma_1} \exp\left[-\tfrac{1}{2}\left(\tfrac{x-\mu_1}{\sigma_1}\right)^2 \right] dx$$

$$= E_1 \left\{ \log\left(\tfrac{\sigma_2}{\sigma_1}\right) + \tfrac{1}{2}\left[\left(\tfrac{x-\mu_2}{\sigma_2}\right)^2 - \left(\tfrac{x-\mu_1}{\sigma_1}\right)^2 \right] \right\}$$

$$= \log\left(\tfrac{\sigma_2}{\sigma_1}\right) + \tfrac{1}{2\sigma_2^2} E_1\left\{(X-\mu_2)^2\right\} - \tfrac{1}{2\sigma_1^2} E_1\left\{(X-\mu_1)^2\right\}$$

$$= \log\left(\tfrac{\sigma_2}{\sigma_1}\right) + \tfrac{1}{2\sigma_2^2} E_1\left\{(X-\mu_2)^2\right\} - \tfrac{1}{2}$$

(Now note that
$(X-\mu_2)^2 = (X-\mu_1+\mu_1-\mu_2)^2 = (X-\mu_1)^2 + 2(X-\mu_1)(\mu_1-\mu_2) + (\mu_1-\mu_2)^2$)

$$= \log\left(\tfrac{\sigma_2}{\sigma_1}\right) + \tfrac{1}{2\sigma_2^2}\left[E_1\left\{(X-\mu_1)^2\right\} + 2(\mu_1-\mu_2) E_1\left\{X-\mu_1\right\} + (\mu_1-\mu_2)^2 \right] - \tfrac{1}{2}$$

$$= \log\left(\tfrac{\sigma_2}{\sigma_1}\right) + \tfrac{\sigma_1^2 + (\mu_1-\mu_2)^2}{2\sigma_2^2} - \tfrac{1}{2}$$

Appendix. (1) 정규분포 간 KL 발산 계산, multivariate

2.1 Multivariate normal KL divergence

First, consider two multivariate normal distributions over the $k \times 1$ vector x specified by

$$\begin{aligned} p(x) &= \mathrm{N}(x; \mu_1, \Sigma_1) \\ q(x) &= \mathrm{N}(x; \mu_2, \Sigma_2) \end{aligned} \tag{5}$$

According to equation (4), the KL divergence of P from Q is defined as

$$\mathrm{KL}[P||Q] = \int_{\mathbb{R}^k} \mathrm{N}(x; \mu_1, \Sigma_1) \ln \frac{\mathrm{N}(x; \mu_1, \Sigma_1)}{\mathrm{N}(x; \mu_2, \Sigma_2)} \mathrm{d}x \ . \tag{6}$$

Using the multivariate normal density function

$$\mathrm{N}(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right] \ , \tag{7}$$

it evaluates to (Duchi, 2014)

$$\mathrm{KL}[P||Q] = \frac{1}{2}\left[(\mu_2-\mu_1)^T \Sigma_2^{-1}(\mu_2-\mu_1) + \mathrm{tr}(\Sigma_2^{-1}\Sigma_1) - \ln\frac{|\Sigma_1|}{|\Sigma_2|} - k\right] \ . \tag{8}$$

Appendix. 참고 자료

- AE with LSTM
https://medium.com/neuronio/an-introduction-to-convlstm-55c9025563a7

https://github.com/keras-team/keras/blob/master/examples/conv_lstm.py

https://towardsdatascience.com/lstm-autoencoder-for-extreme-rare-event-classification-in-keras-ce209a224cfb

- **GAN's Formulation**
https://slazebni.cs.illinois.edu/spring17/lec11_gan.pdf

- **AnoGAN**
https://arxiv.org/pdf/1703.05921.pdf

- **책**
미술간에 GAN 딥러닝 실전 프로젝트:
코드 github(keras 기반):
https://github.com/davidADSP/GDL_code/tree/master/models