LOCK GAN

- GAN을 보안에 적용할수 있는 방안 고안.

1.GAN이란 무엇인가
2.LOCK GAN 소개
3.LOCK GAN 구현

GAN(생성적 적대 신경망) 소개

* 2016년에 논문이 나온 인공지능 이론이다.

위조지페를 잡는 경찰

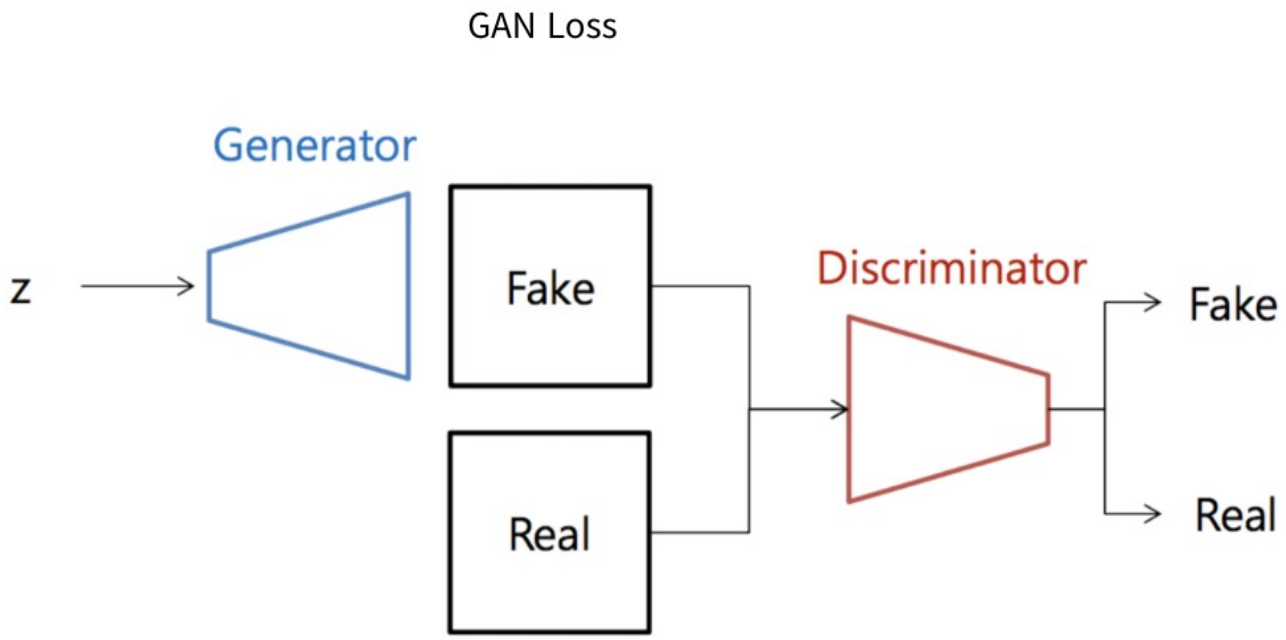
VS

위조지페를 만드는 도둑

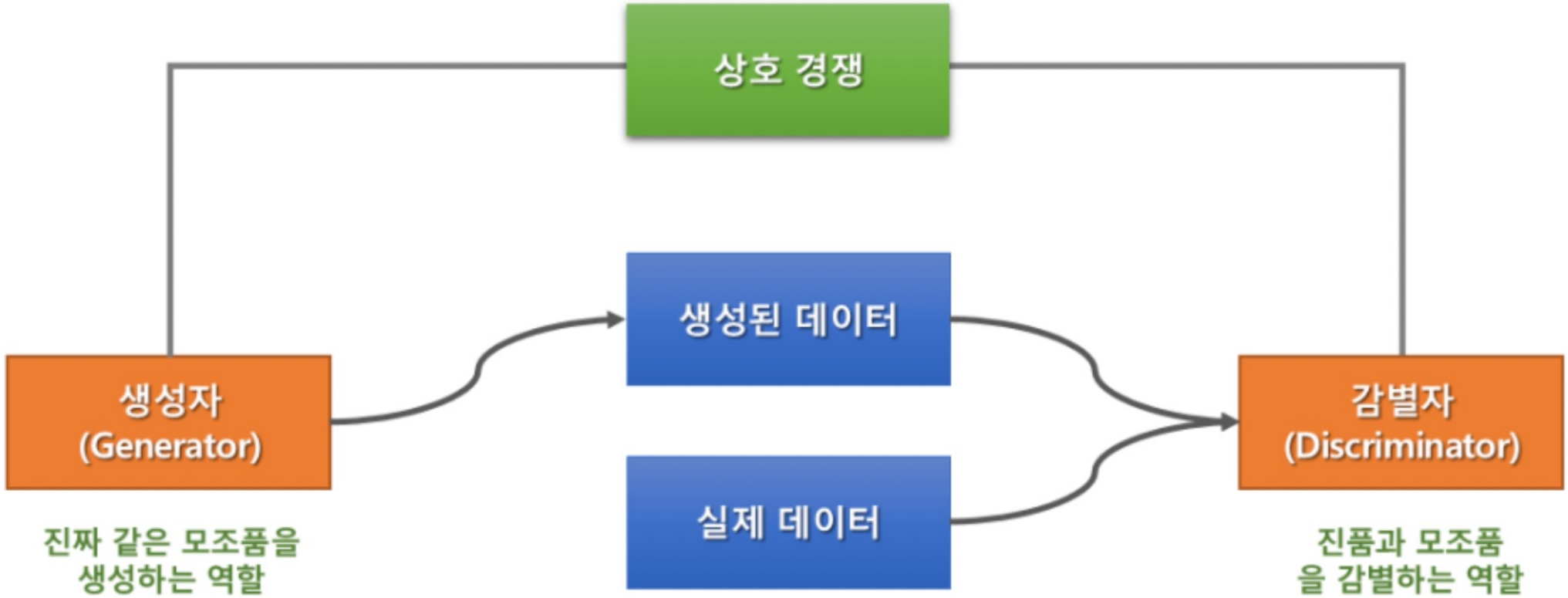
GAN 소개

위 예시를 구현하면 아래와 같다.

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



GAN 소개

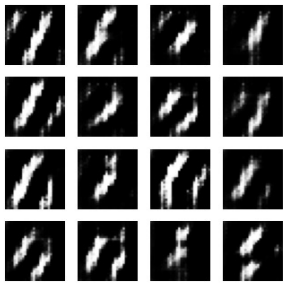

상호 경쟁
생성된 데이터
생성자
(Generator)
감별자
(Discriminator)
실제 데이터
진짜 같은 모조품을
생성하는 역할
진품과 모조품
을 감별하는 역할

GAN 종류

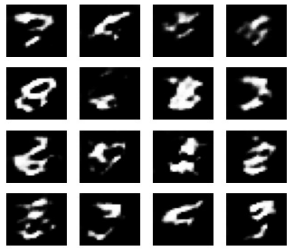
1. SRGAN(Super Resolution GAN)
   - 저해상도 이미지를 고해상도로 변환
2. Stack GAN
   - 입력된 문장과 단어를 해석해 이미지를 생성
3. 3D GAN
   - 2D 이미지를 3D 이미지로 변환함
4. Cycle GAN
   - 이미지를 바꿈 ex) 밤 -> 낮 , 경주마 -> 얼룩말

GAN 구현

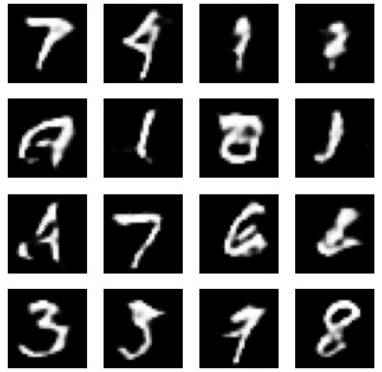
DCGAN



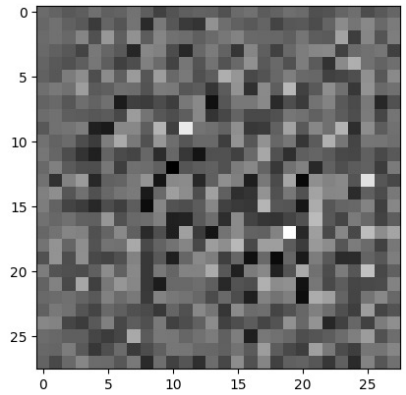
3 epoch

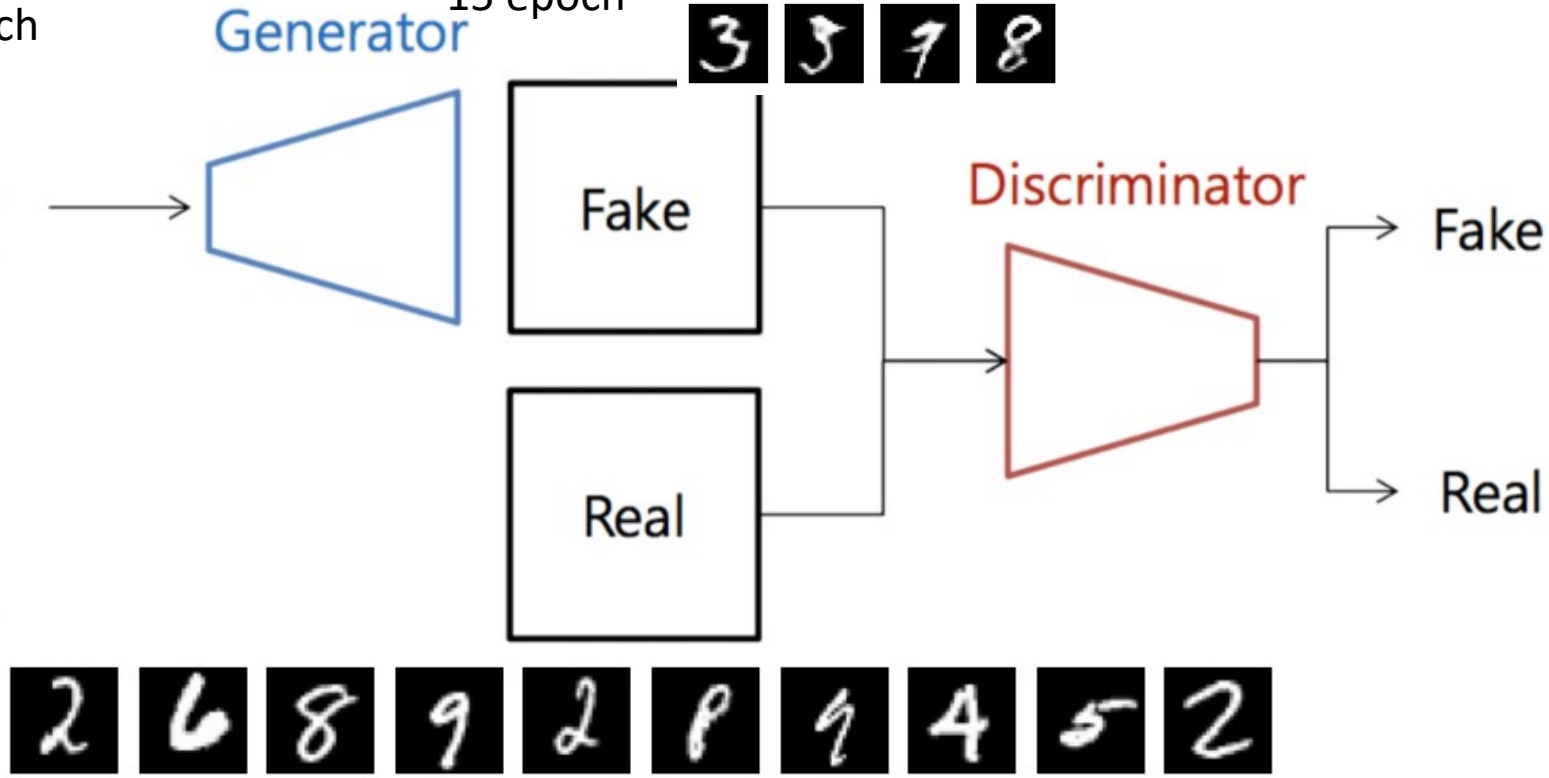


15 epoch



60 epoch





GAN 종류

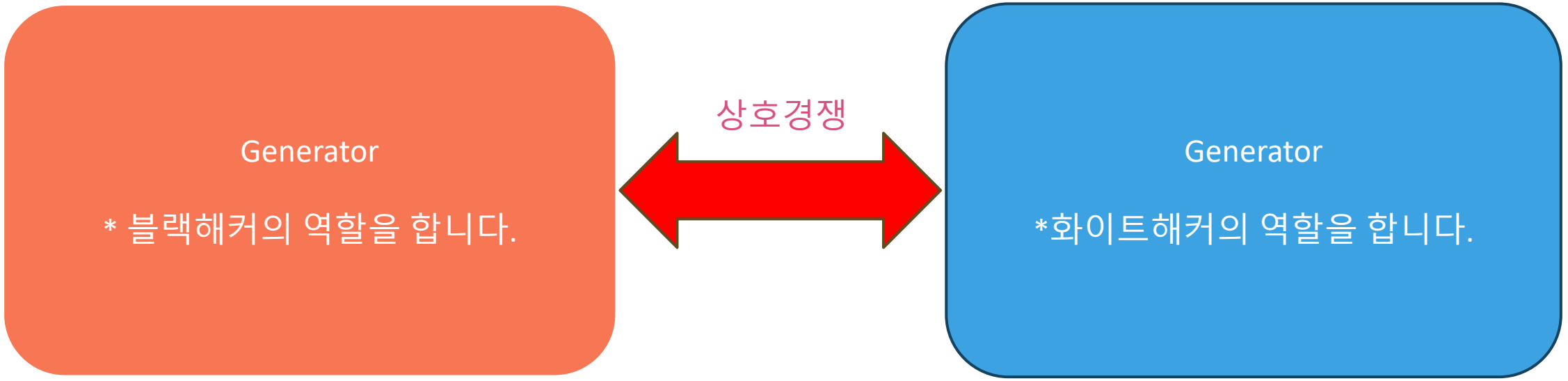
1. SRGAN(Super Resolution GAN)
   - 저해상도 이미지를 고해상도로 변환
2. Stack GAN
   - 입력된 문장과 단어를 해석해 이미지를 생성
3. 3D GAN
   - 2D 이미지를 3D 이미지로 변환함
4. Cycle GAN
   - 이미지를 바꿈 ex) 밤 -> 낮 , 경주마 -> 얼룩말

LOCK GAN

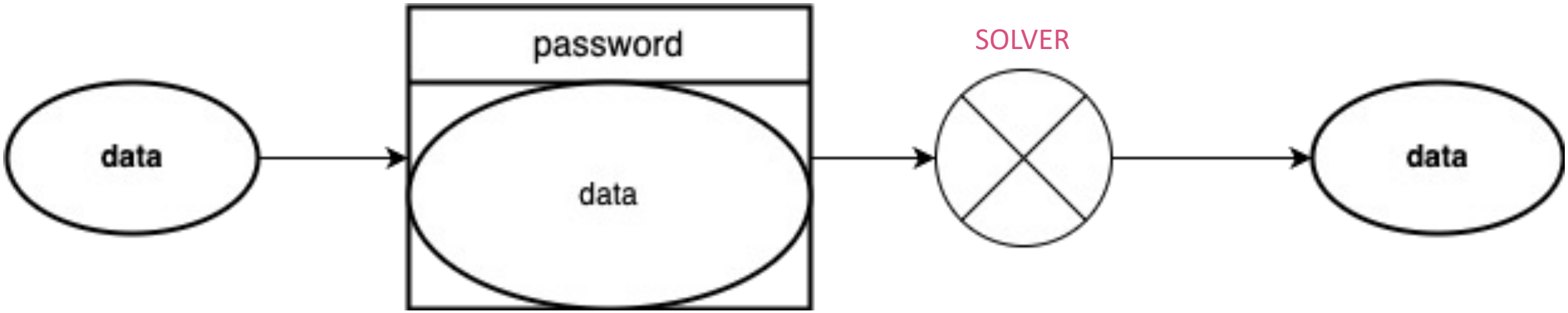
GAN을 사용한 **2**중 암호화 방안

LOCK GAN 아이디어

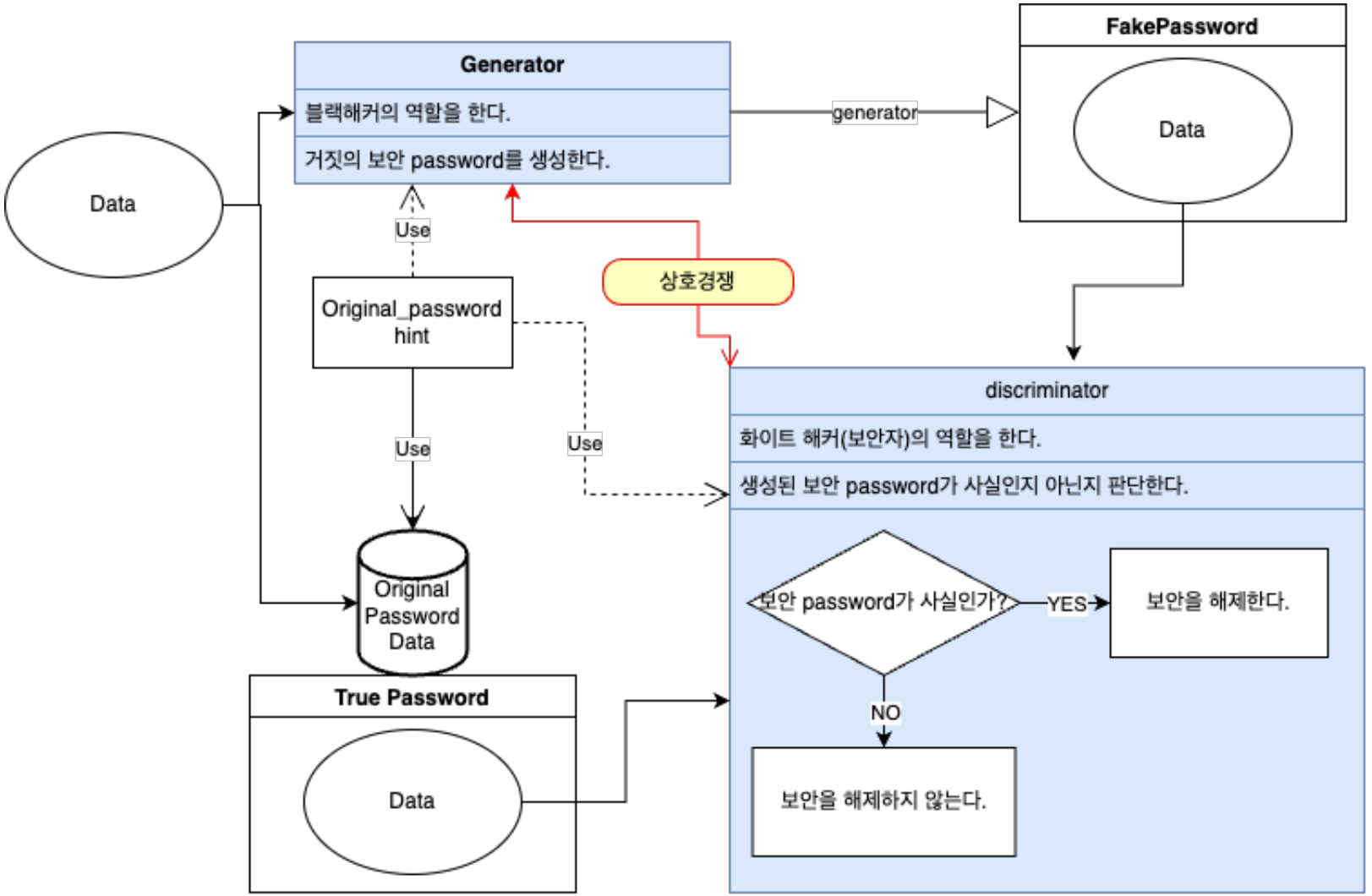
Q. GAN의 경쟁 시스템을 암호화 이론에 적용시켜볼 수 있을까?



LOCK GAN 개요 – 1차 암호화

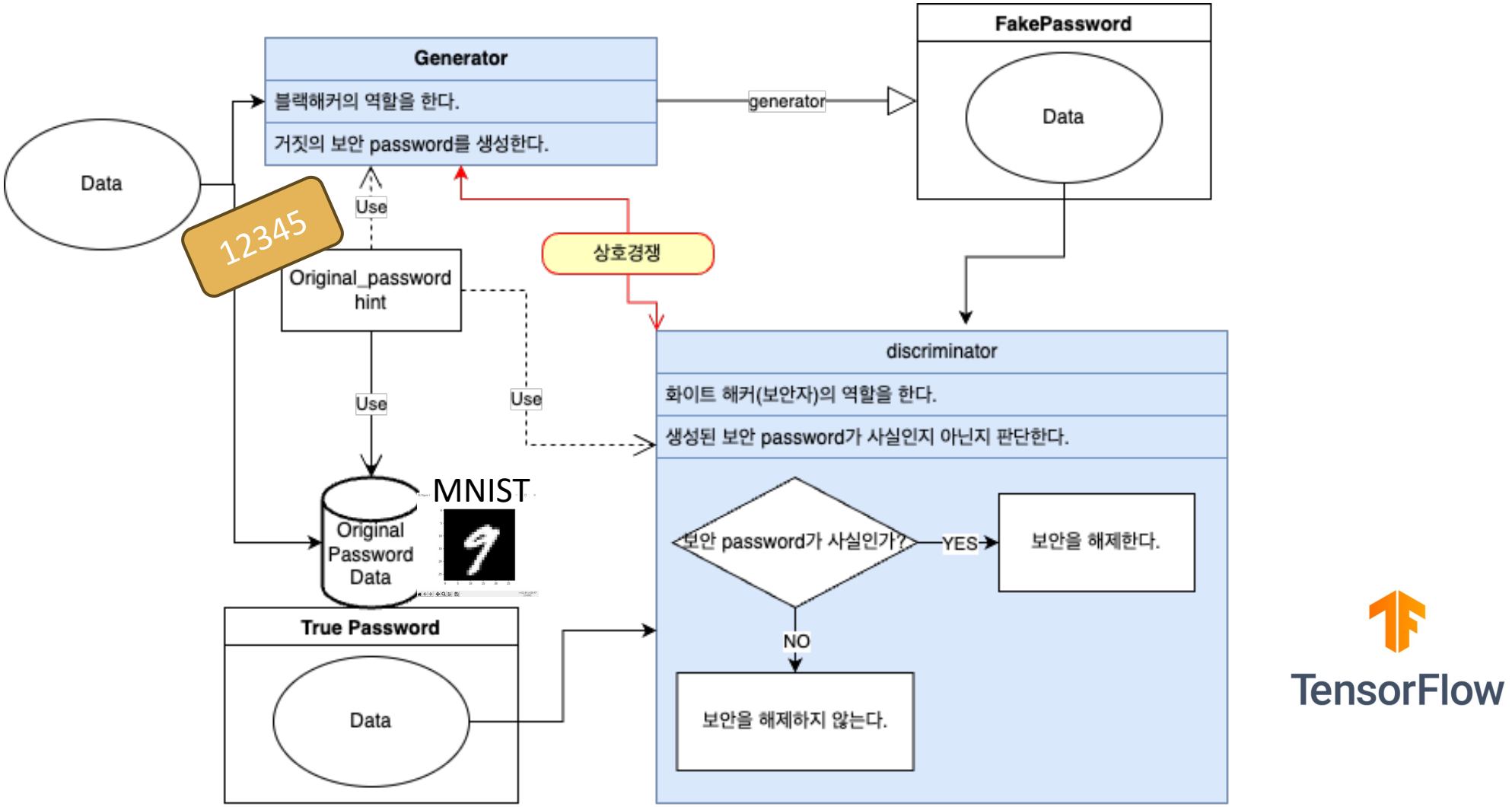


LOCK GAN 개요 – 학습구성



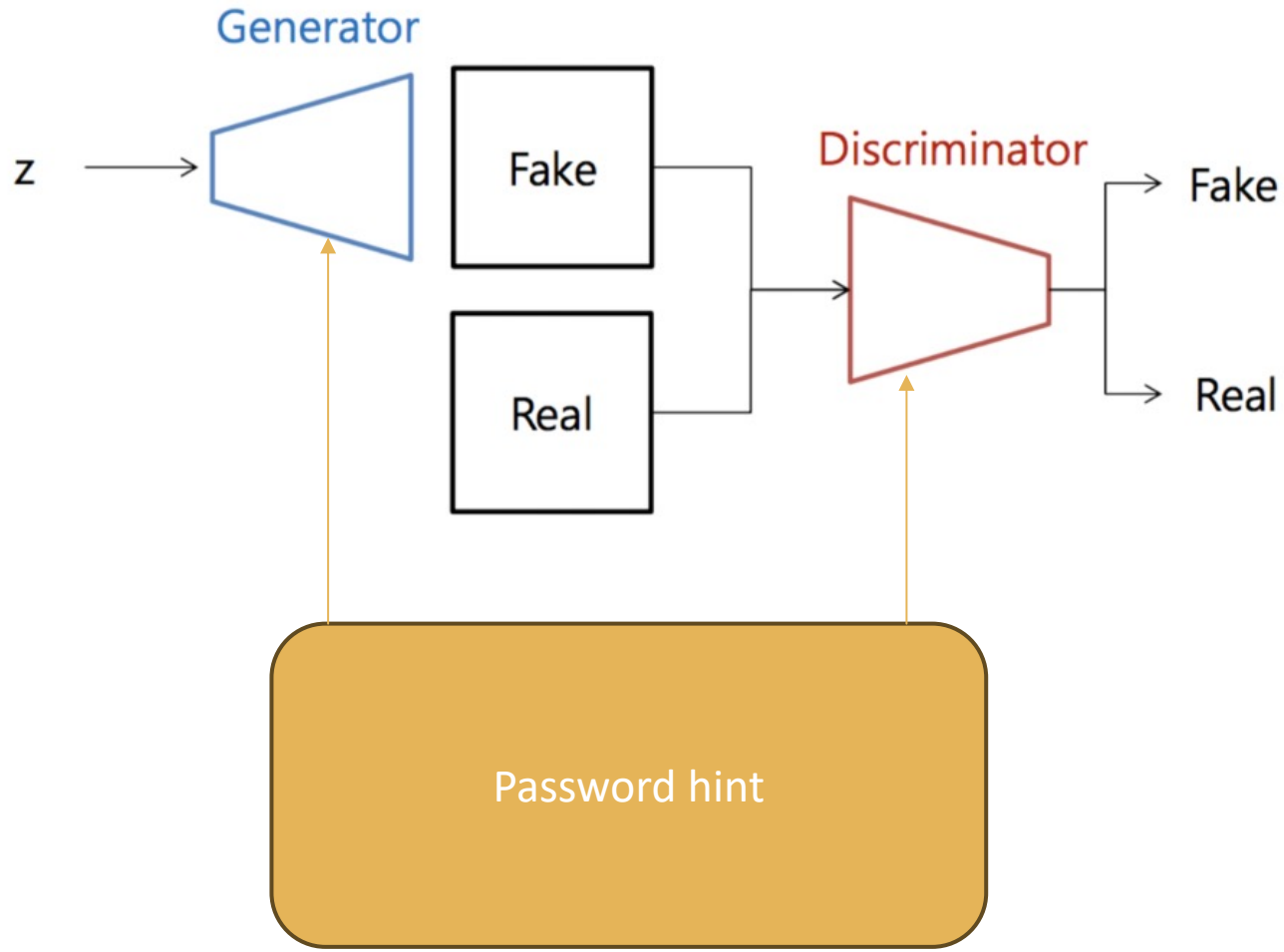
LOCK GAN 구현 - MNIST

LOCK GAN 구현 – 구성

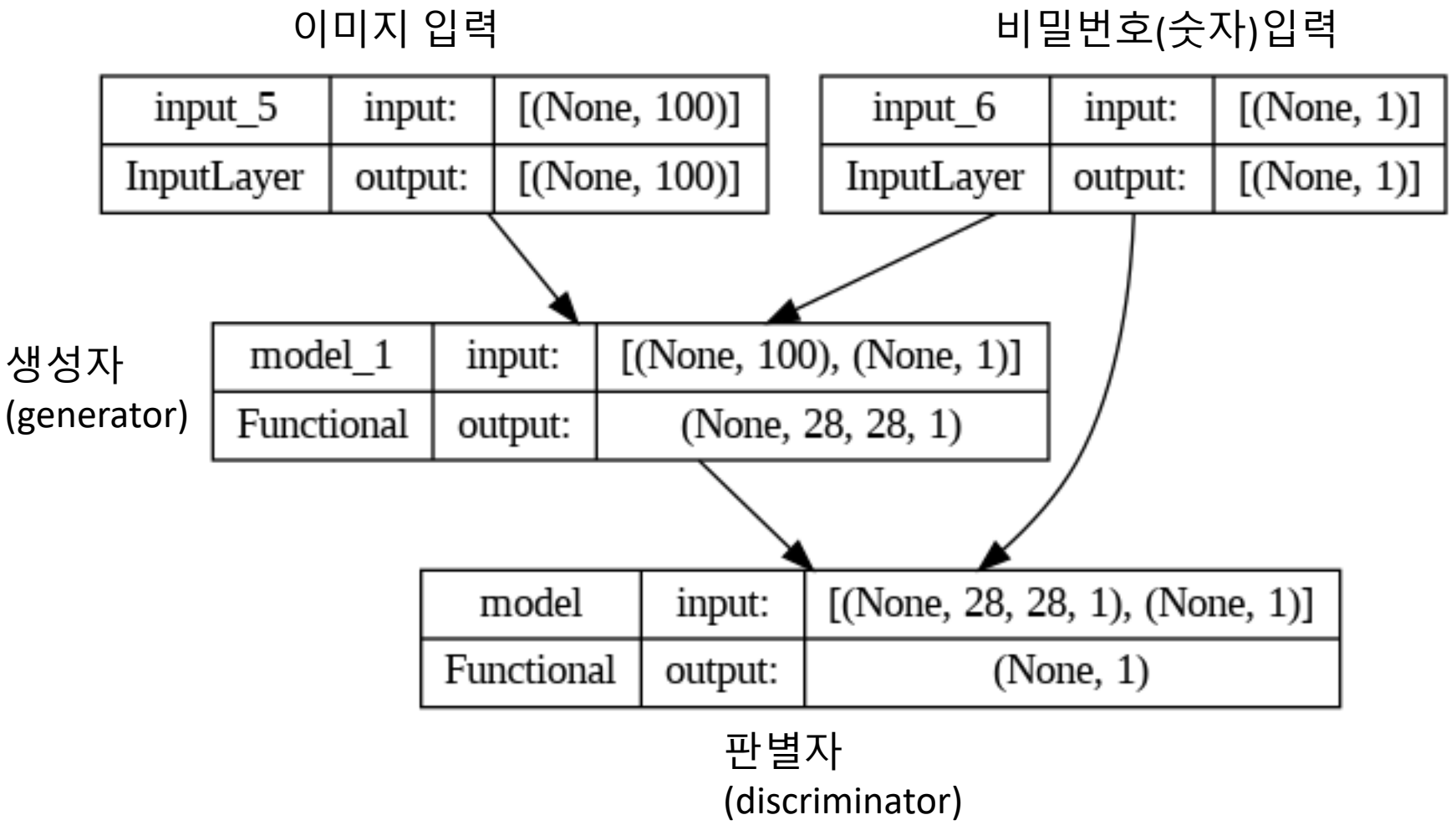


TensorFlow

LOCK GAN 구현

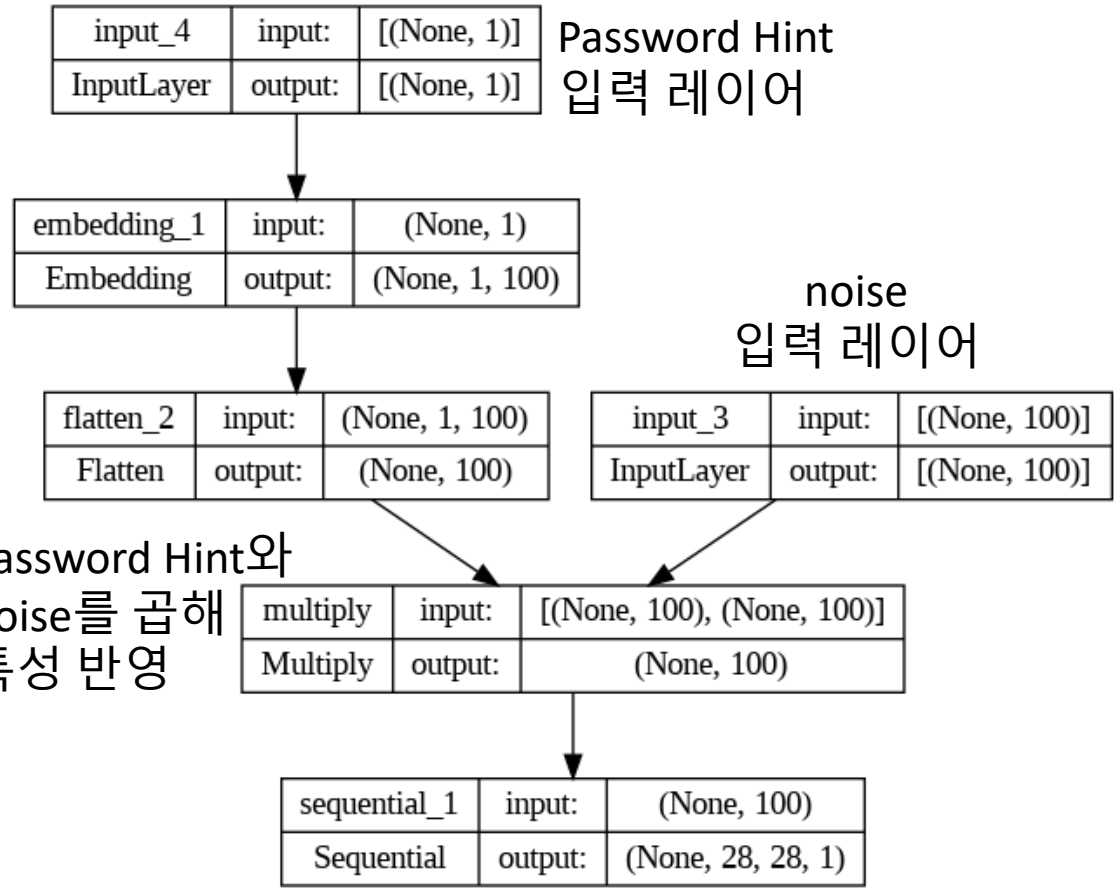

Generator
z
Fake
Real
Discriminator
Fake
Real
Password hint

LOCK GAN 구현 – 전체구성

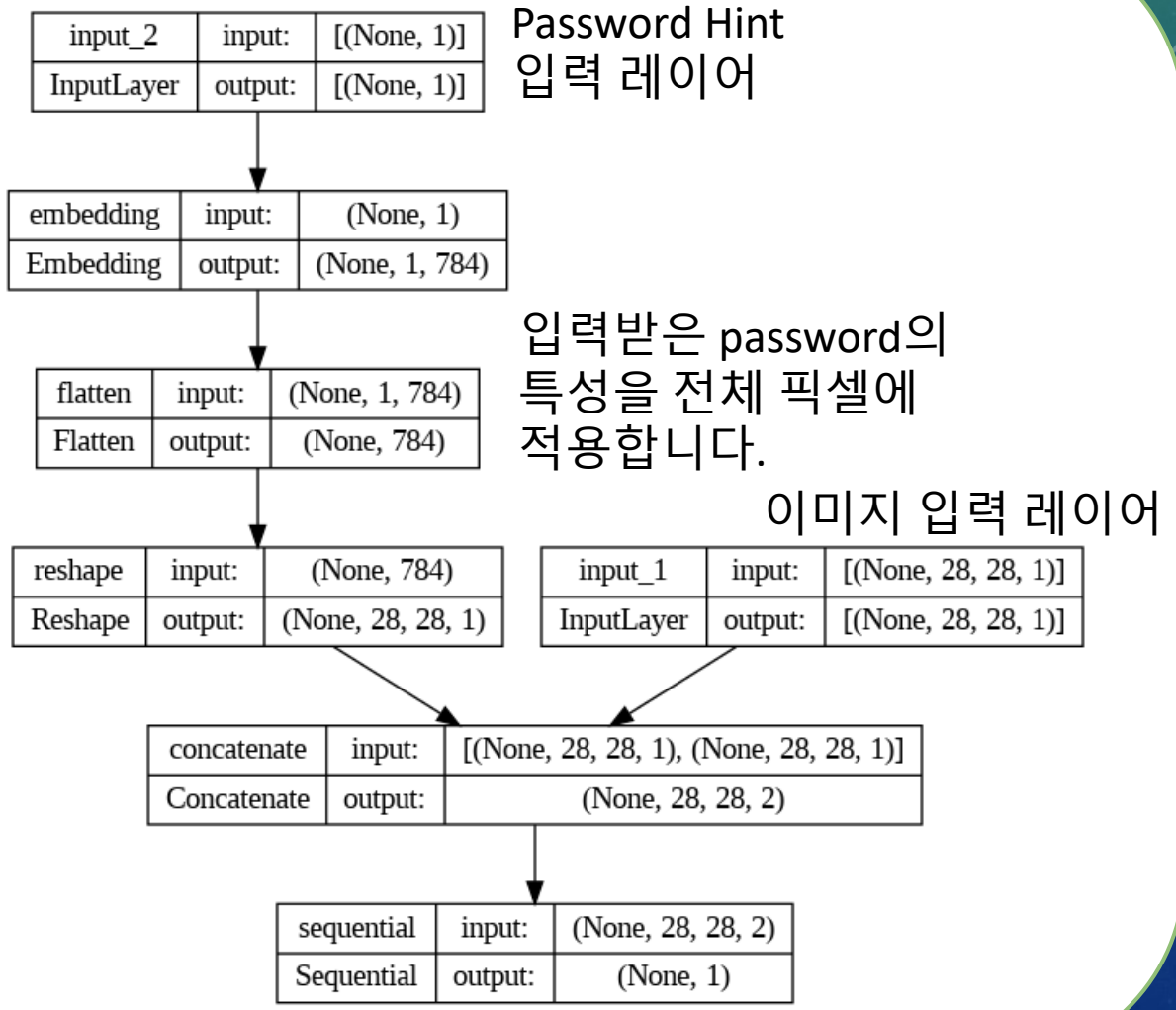


LOCK GAN 구현 – 생성자, 판별자 구현

생성자(Generator)



판별자(discriminator)



LOCK GAN 구현 – 생성자의 SEQUENTIAL

dense_input	input:	[(None, 100)]
InputLayer	output:	[(None, 100)]

dense	input:	(None, 100)
Dense	output:	(None, 12544)

reshape	input:	(None, 12544)
Reshape	output:	(None, 7, 7, 256)

conv2d_transpose	input:	(None, 7, 7, 256)
Conv2DTranspose	output:	(None, 14, 14, 128)

batch_normalization	input:	(None, 14, 14, 128)
BatchNormalization	output:	(None, 14, 14, 128)

leaky_re_lu	input:	(None, 14, 14, 128)
LeakyReLU	output:	(None, 14, 14, 128)

conv2d_transpose_1	input:	(None, 14, 14, 128)
Conv2DTranspose	output:	(None, 14, 14, 64)

batch_normalization_1	input:	(None, 14, 14, 64)
BatchNormalization	output:	(None, 14, 14, 64)

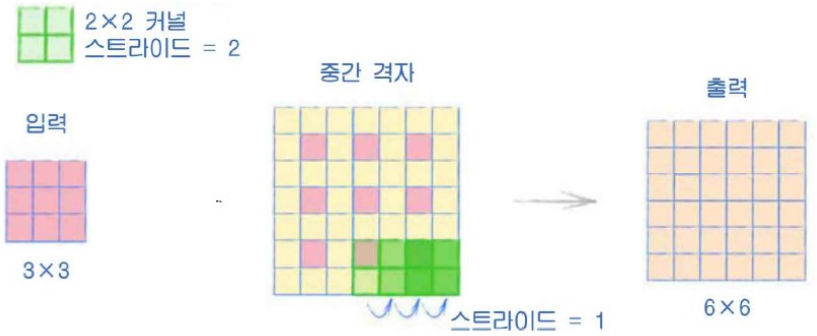
leaky_re_lu_1	input:	(None, 14, 14, 64)
LeakyReLU	output:	(None, 14, 14, 64)

conv2d_transpose_2	input:	(None, 14, 14, 64)
Conv2DTranspose	output:	(None, 28, 28, 1)

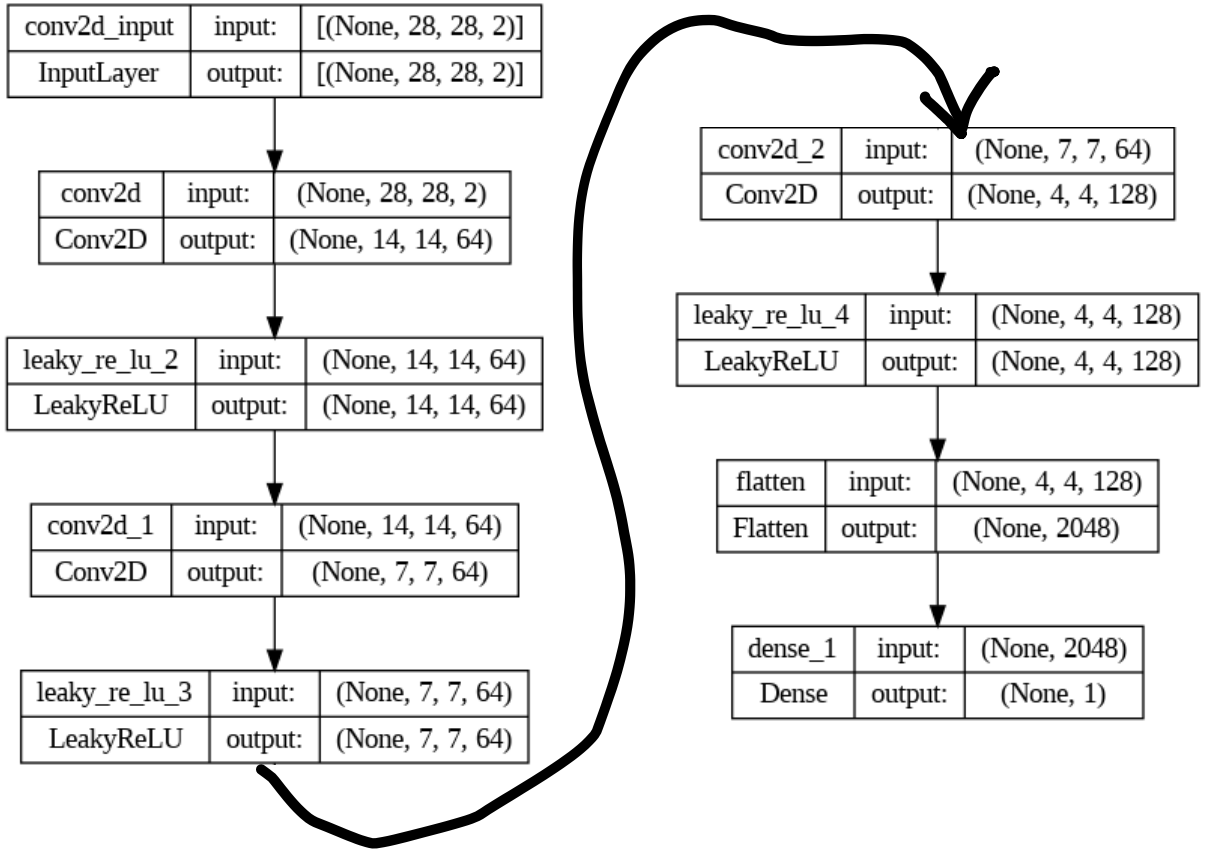
activation	input:	(None, 28, 28, 1)
Activation	output:	(None, 28, 28, 1)

tanh

- 총 4개의 레이어.
- 1개의 Dense
- 3개의 2D 전치 합성곱 신경망

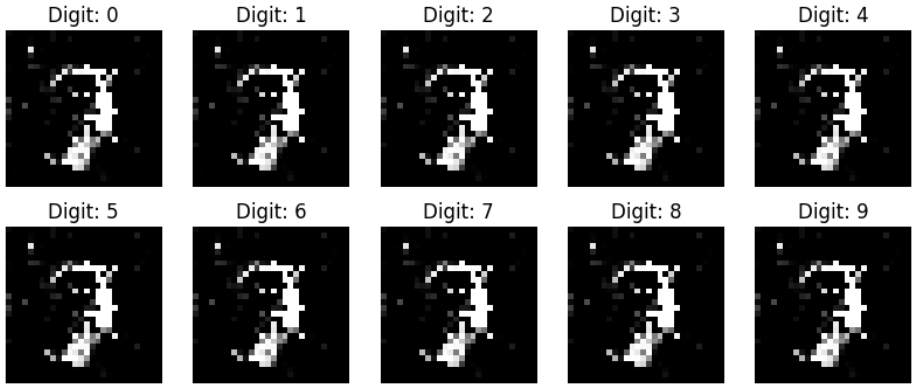


LOCK GAN 구현 – 판별자의 SEQUENTIAL

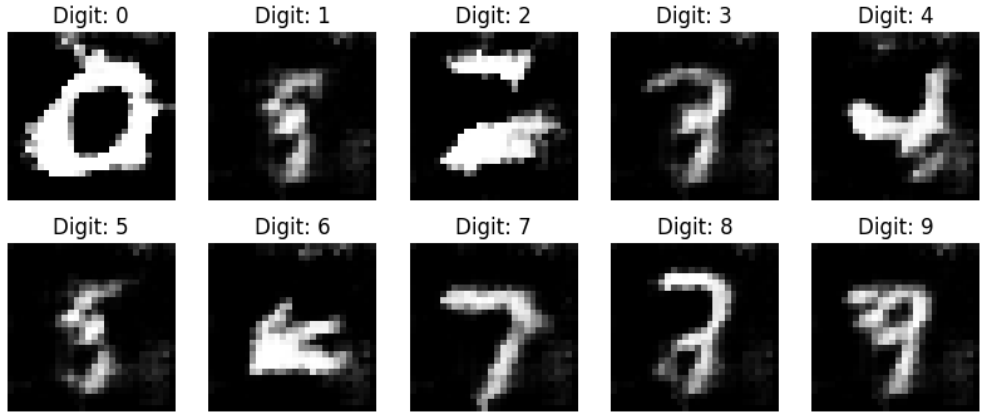


- 3개의 컨볼루션 레이어.

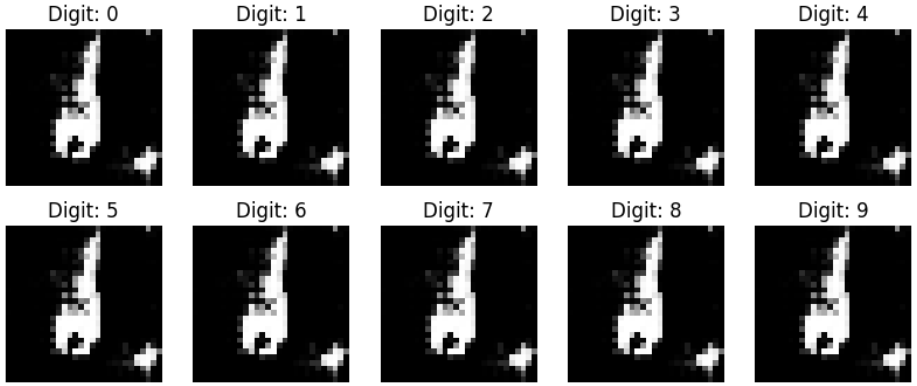
LOCK GAN 학습



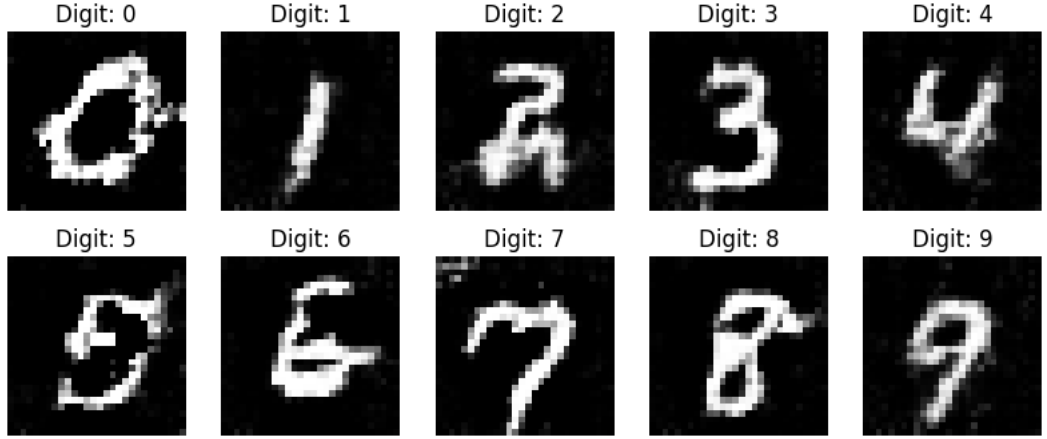
Epoch 1000



Epoch 17000

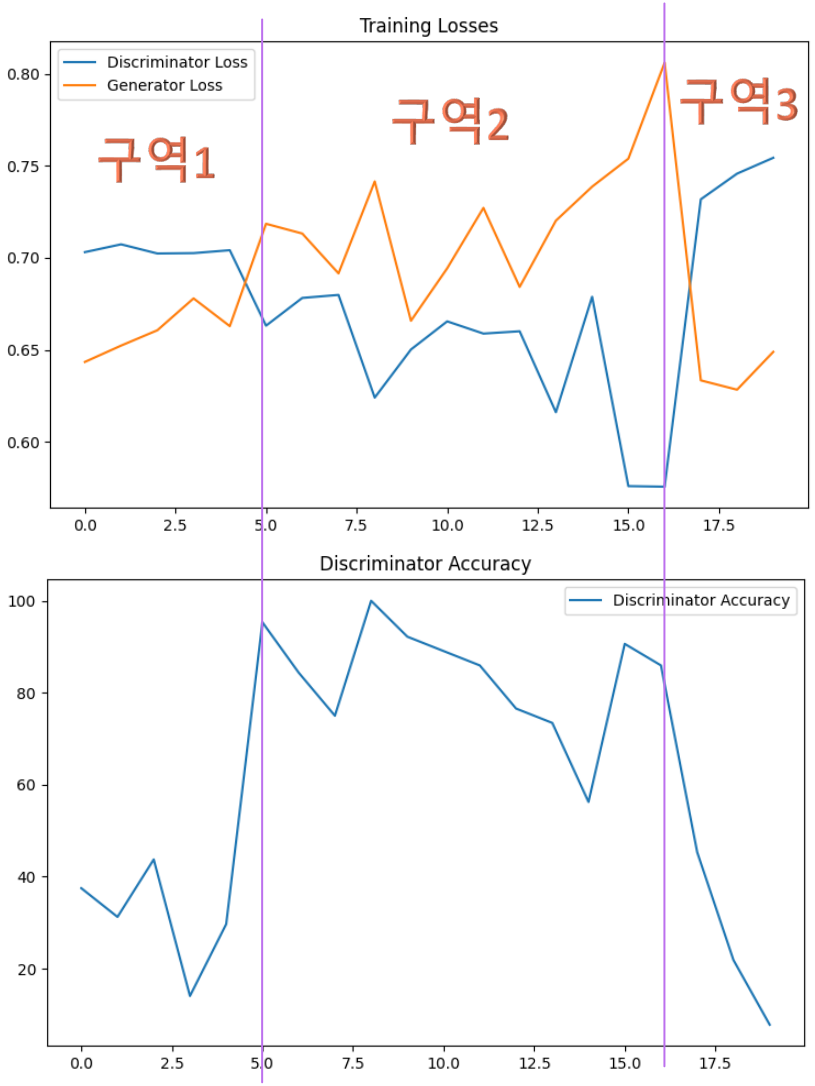


Epoch 10000



Epoch 20000

LOCK GAN 학습



구역1 : 비신뢰구간.

- 이 구간은 generator와 discriminator가 모두 초기상태임.
- Generator와 discriminator의 성능 모두 좋지 못함.

구역2 : 판별자 우세구간.

- 이 구간은 generator에 비해 discriminator가 더욱 우세함.
- 첨차적으로 해커의 역할을 하는 generator의 성능이 높아지기 시작함.

구역3 : 생성자 우세구간.(해킹완료구간)

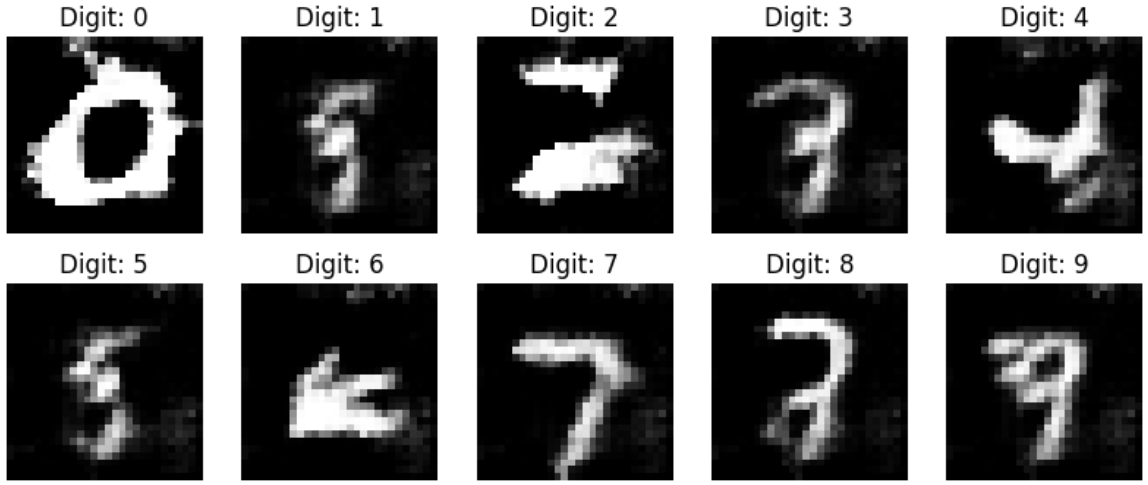
- 이 구간은 discriminator에 비해 generator가 더욱 우세함.
- 즉, 해커의 역할을 하는 generator가 해킹을 성공한 상태임.

LOCK GAN 학습

이 훈련 루프를 거쳐서 discriminator의 정확도가 기하급수적으로 증가합니다.

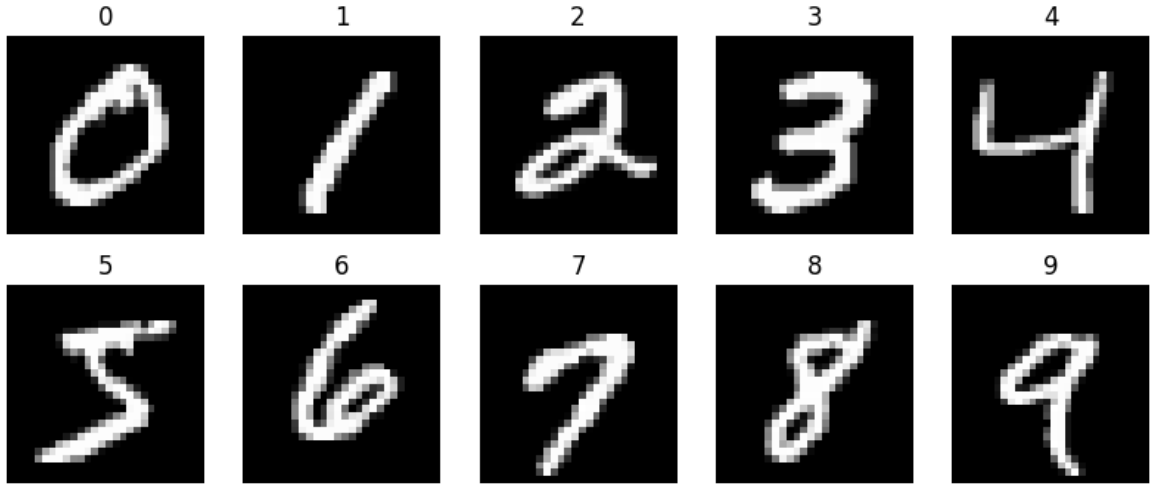
why? : generator가 fake이미지를 계속 만들면서 거짓 이미지를 구별하는 훈련을 집중적으로 진행하기 때문에!

LOCK GAN 학습



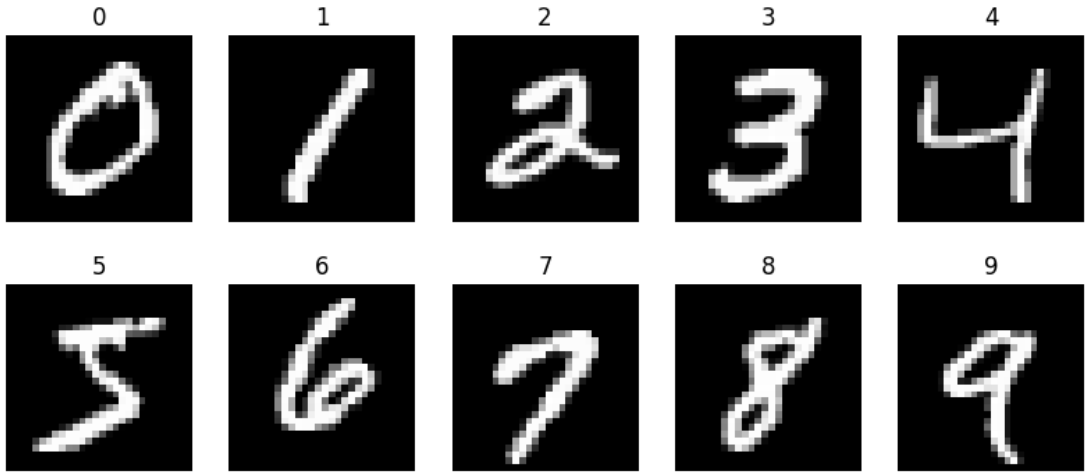
False

True



LOCK GAN 특징

1. password dataset(예. 이미지)가 더욱 복잡할수록 안전함.
   <---> 복잡할수록 학습시간이 증가.



<

LOCK GAN 특징

2. 딥러닝 기반이기 때문에 해킹하기 어려움.
   -> 해킹을 할려면 최소 몇억개의 파라미터를 모두 의도적으로 조작해야 함.

3. 계속해서 생성자가 해킹을 시도하기때문에 실전 상황에서 강한 장점이 있음.

BUT, 저의 머리속에서 나온 GAN 활용 방안이기 때문에 더욱 추가적인 연구 필요.

감사합니다.