Generative Model
GAN + DCGAN

Yunjey Choi
DAVIAN Lab Study

2017.1.5

Table of Contents

Table of Contents

 Code에 대한 분석을 하고 싶다면 여기로 (p.88)

1. Generative Model

Generative Model

"What I cannot create, I do not understand."

—Richard Feynman

Image Modeling

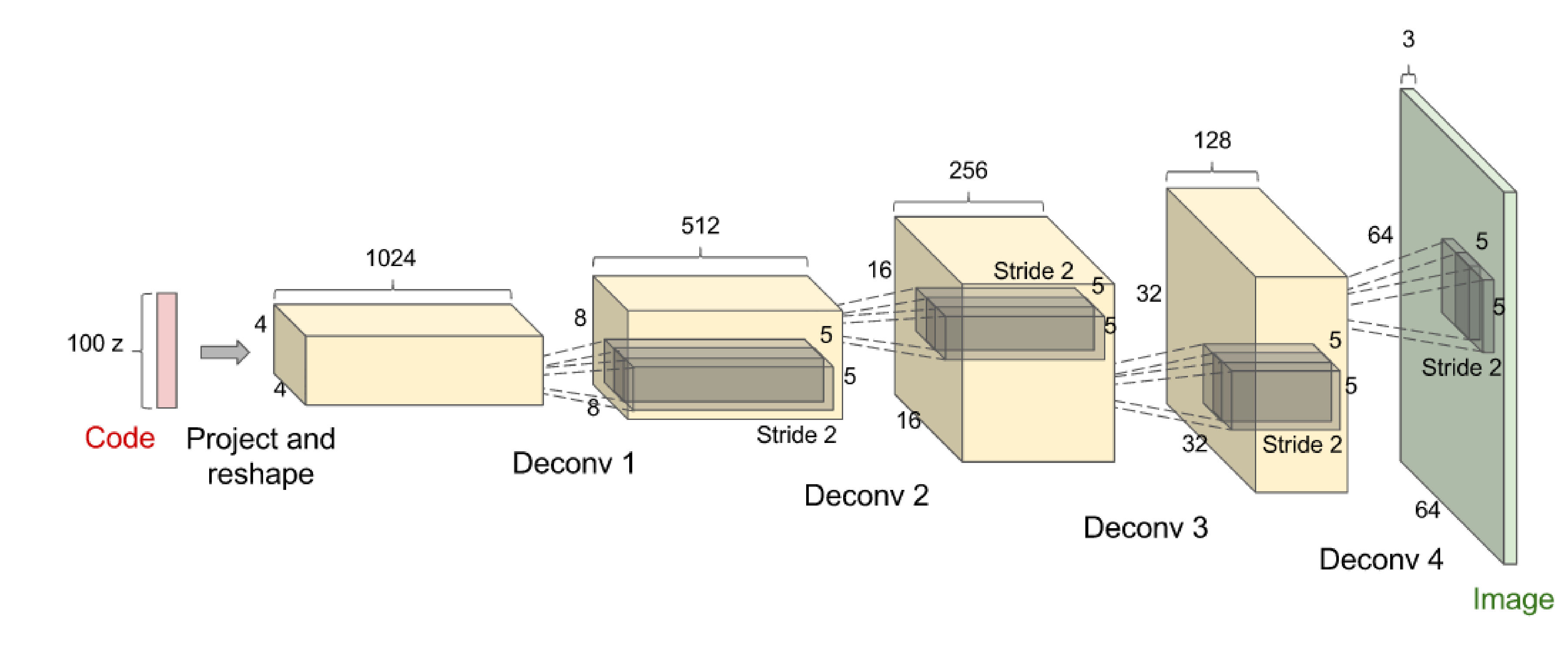
Language Modeling이 있듯이 Image Modeling도 존재한다

적은 수의 parameter로 image data의 distribution 함수를 만드는게 목표

120만 개의 imagenet dataset
1.2M x 256 x 256 x 3 (약 200GB의 pixel data)

Image Modeling

Code가 주어지면 Imagenet 데이터셋을 생성하는 모델



100MB of weights < 200GB of pixels

Generative Model

Generative Adversarial Networks

생성자와 식별자 두 모델 간의 적대적 학습 (Adversarial Learning)

Variational Auto-Encoders

문제를 PGM형태로 치환 후 data의 log likelihood lower bound를 maximize하는 쪽으로 학습

Pixel-RNN/CNN

모든 pixel들의 conditional distribution을 modeling (left-to-right, top-to-bottom)

Generative Model

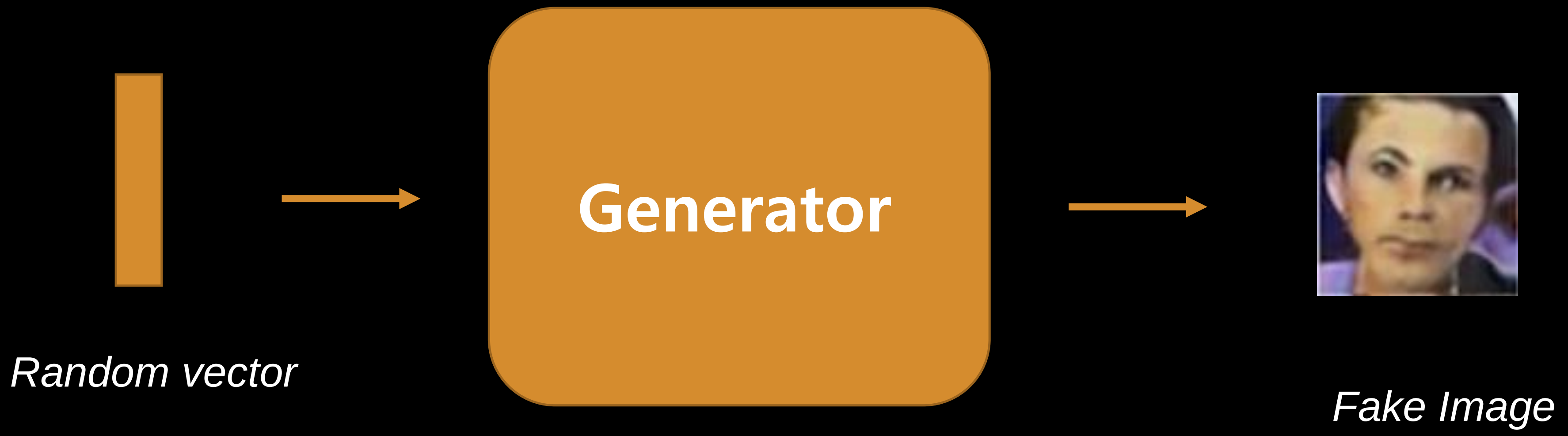
	Generative Adversarial Networks	Variational Auto-Encoders	Pixel-RNN/CNN
2013			
		VAE (2013.12)	
2014			
	GAN (2014.6)		
2015			
	DCGAN (2015.11)		
2016			Pixel-RNN (2016.1)
	InfoGAN (2016.6)	Improved VAE (2016.6)	Pixel-CNN (2016.6)
			Pixel-CNN++ (2016.11)
	EBGAN (2016.9)		Pyramid Pixel-CNN(2016.12)

Generative Model

	Generative Adversarial Networks	Variational Auto-Encoders	Pixel-RNN/CNN
2013			
	오늘 다룰 주제	VAE (2013.12)	
2014			
	GAN (2014.6)		
2015			
	DCGAN (2015.11)		
2016			Pixel-RNN (2016.1)
	InfoGAN (2016.6)	Improved VAE (2016.6)	Pixel-CNN (2016.6)
			Pixel-CNN++ (2016.11)
	EBGAN (2016.9)		Pyramid Pixel-CNN(2016.12)

2. GAN

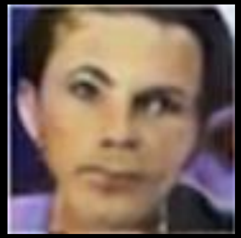
Generative Adversarial Network (GAN)



Generative Adversarial Network (GAN)

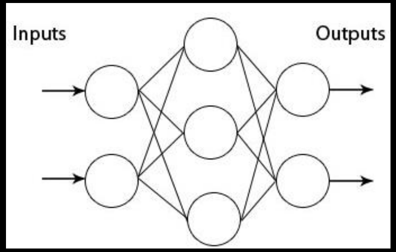
Random vector

Generator



Fake Image

Neural Network



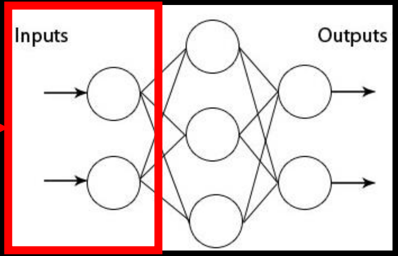
Generative Adversarial Network (GAN)

Random vector

Generator

Fake Image

Neural Network



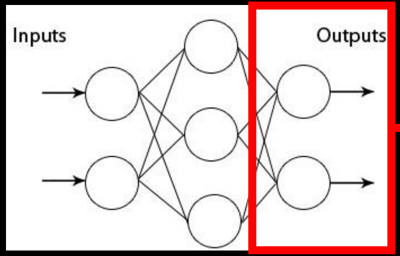
Generative Adversarial Network (GAN)

Random vector

Generator

Fake Image

Neural Network



Reshape

Generative Adversarial Network (GAN)

Generative Adversarial Network (GAN)

진짜 이미지인가?

Real Image

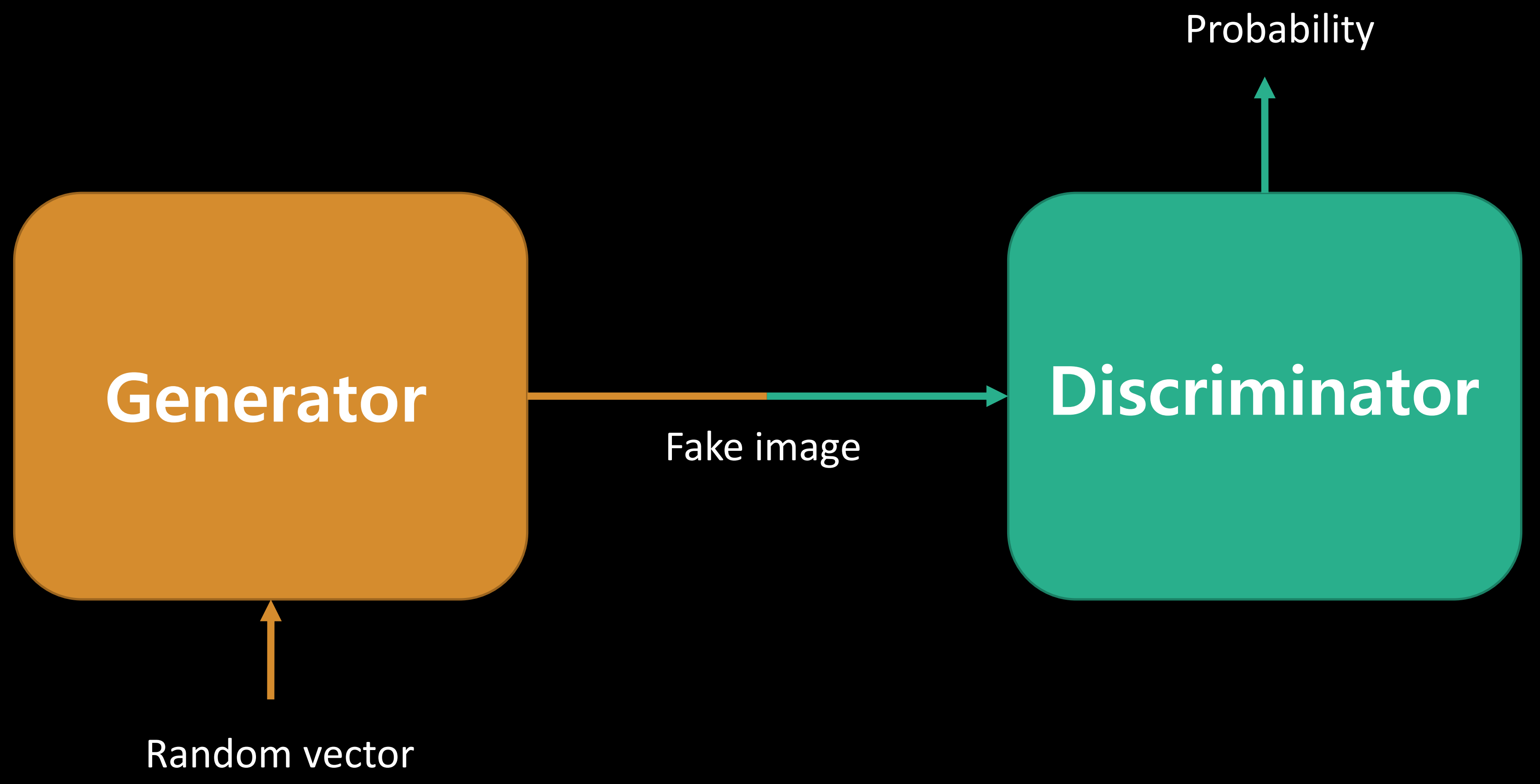
Discriminator

1 (Yes)

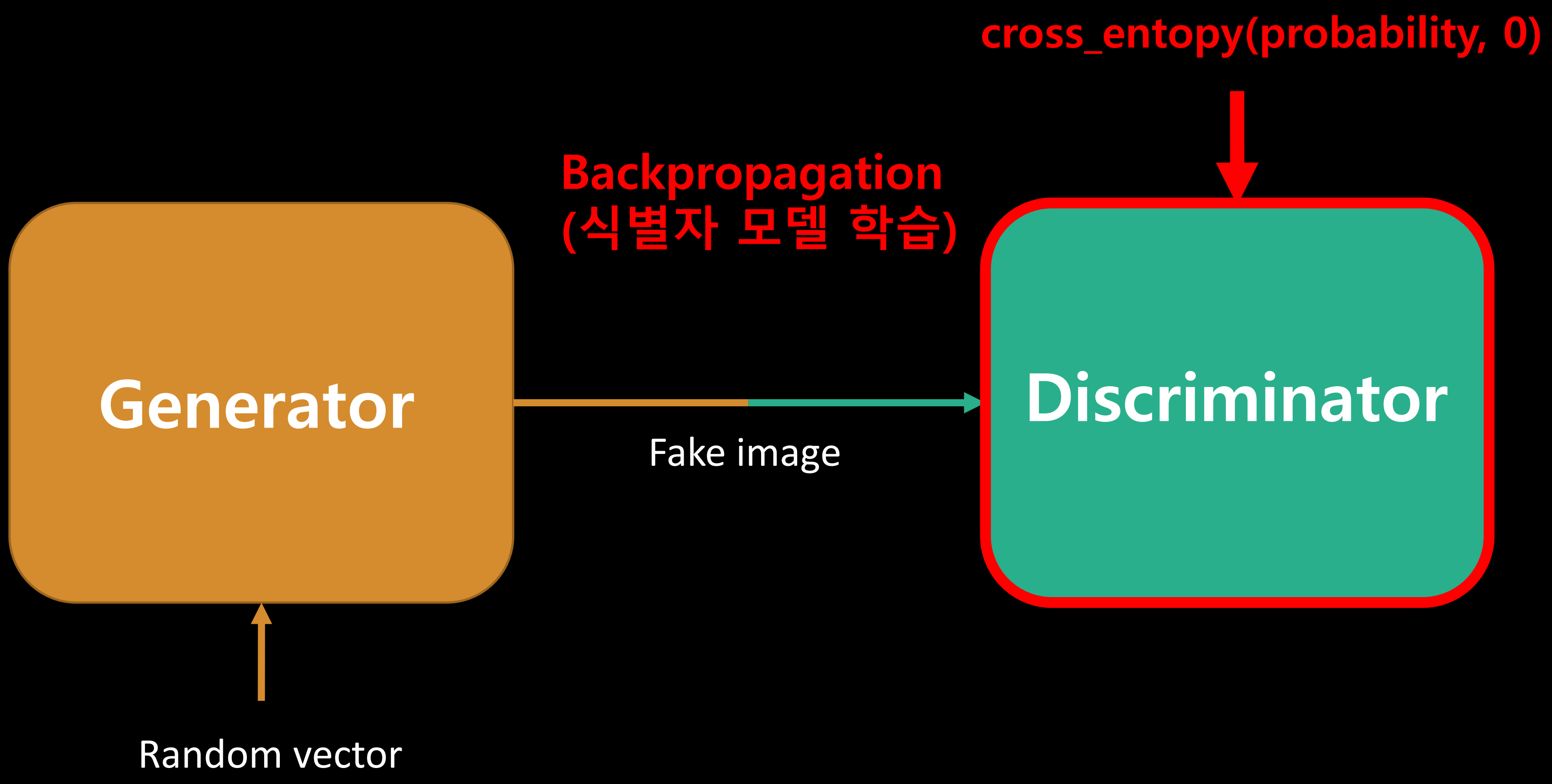
식별자는 진짜 이미지를
'진짜'로 판별하도록 학습

Generative Adversarial Network (GAN)

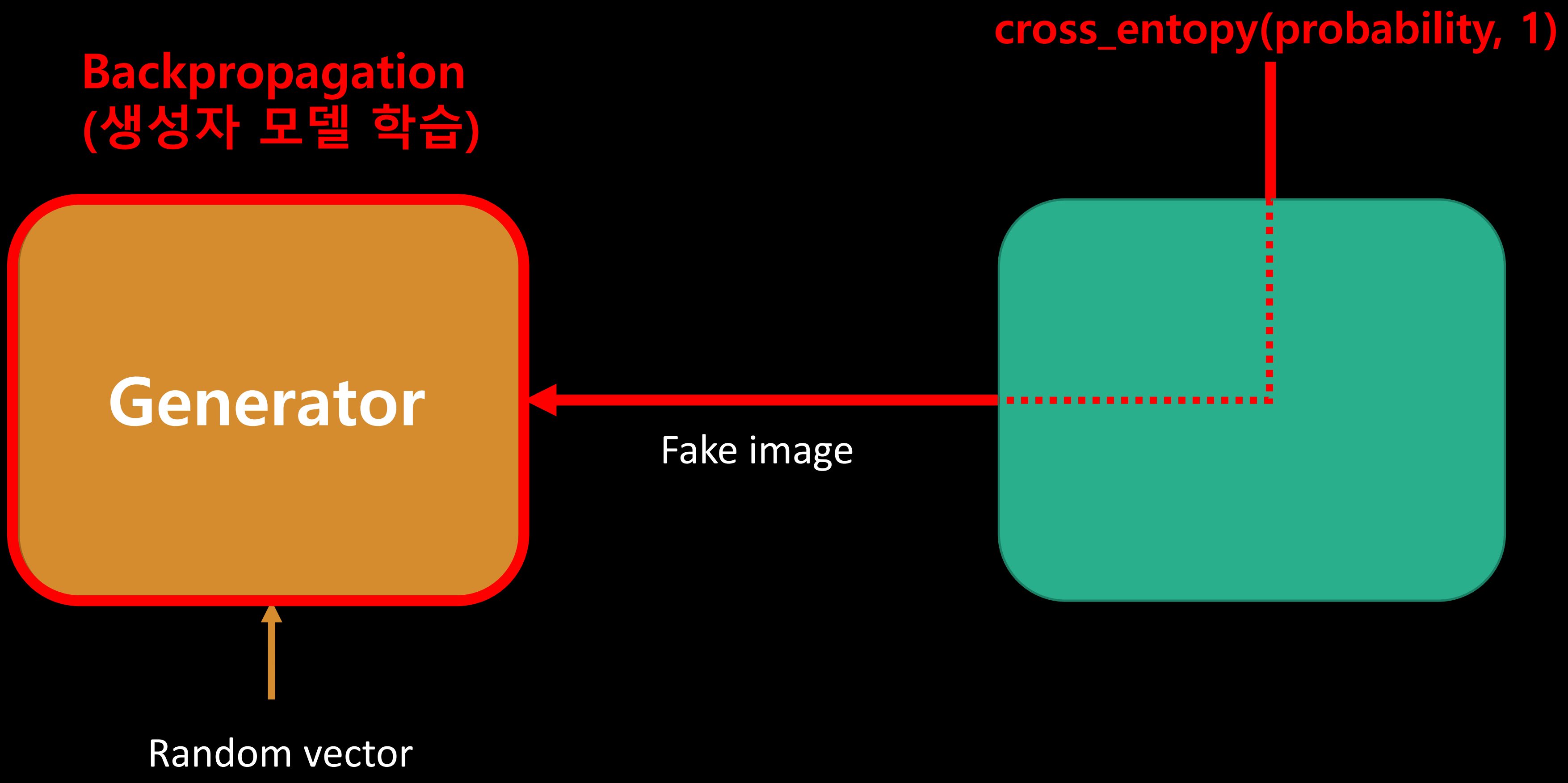
Generative Adversarial Network (GAN)



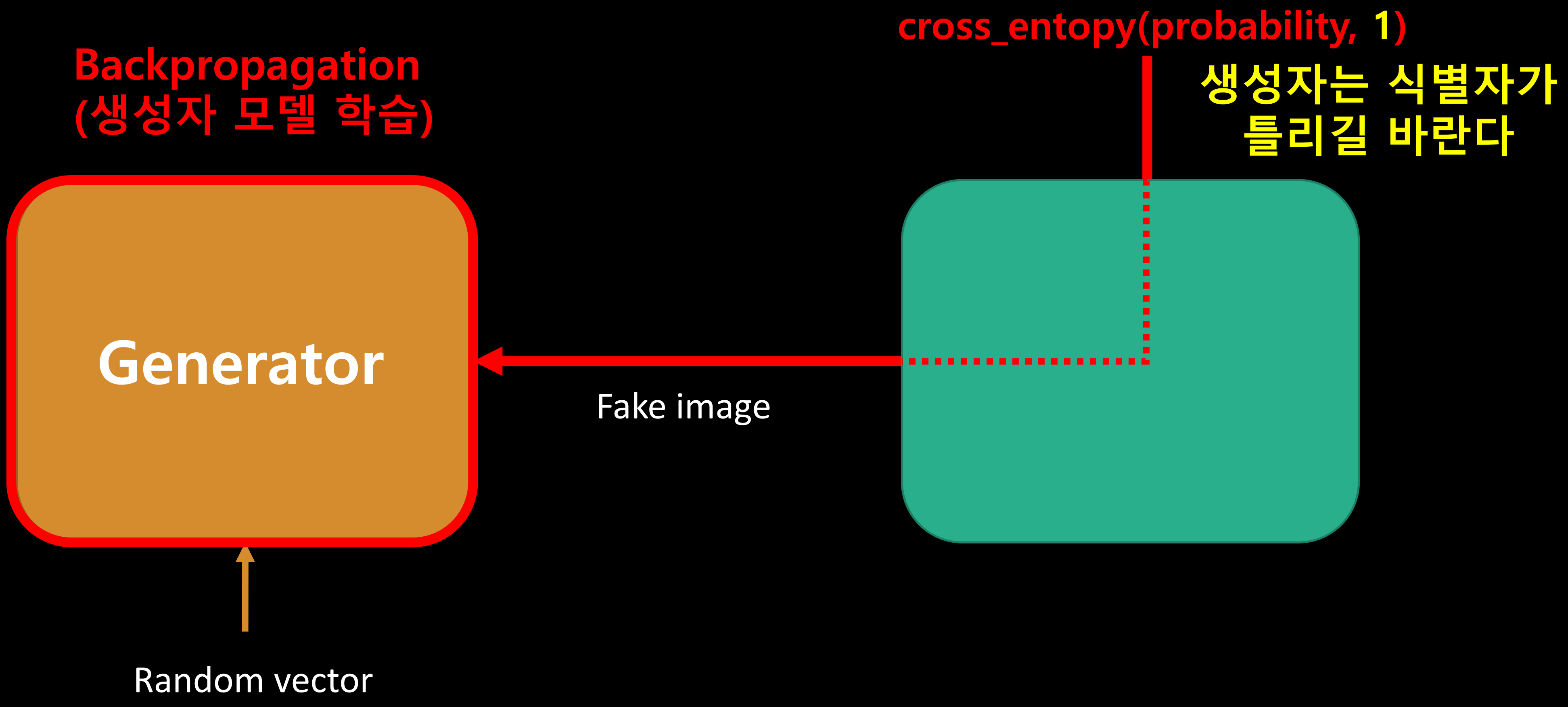
Generative Adversarial Network (GAN)



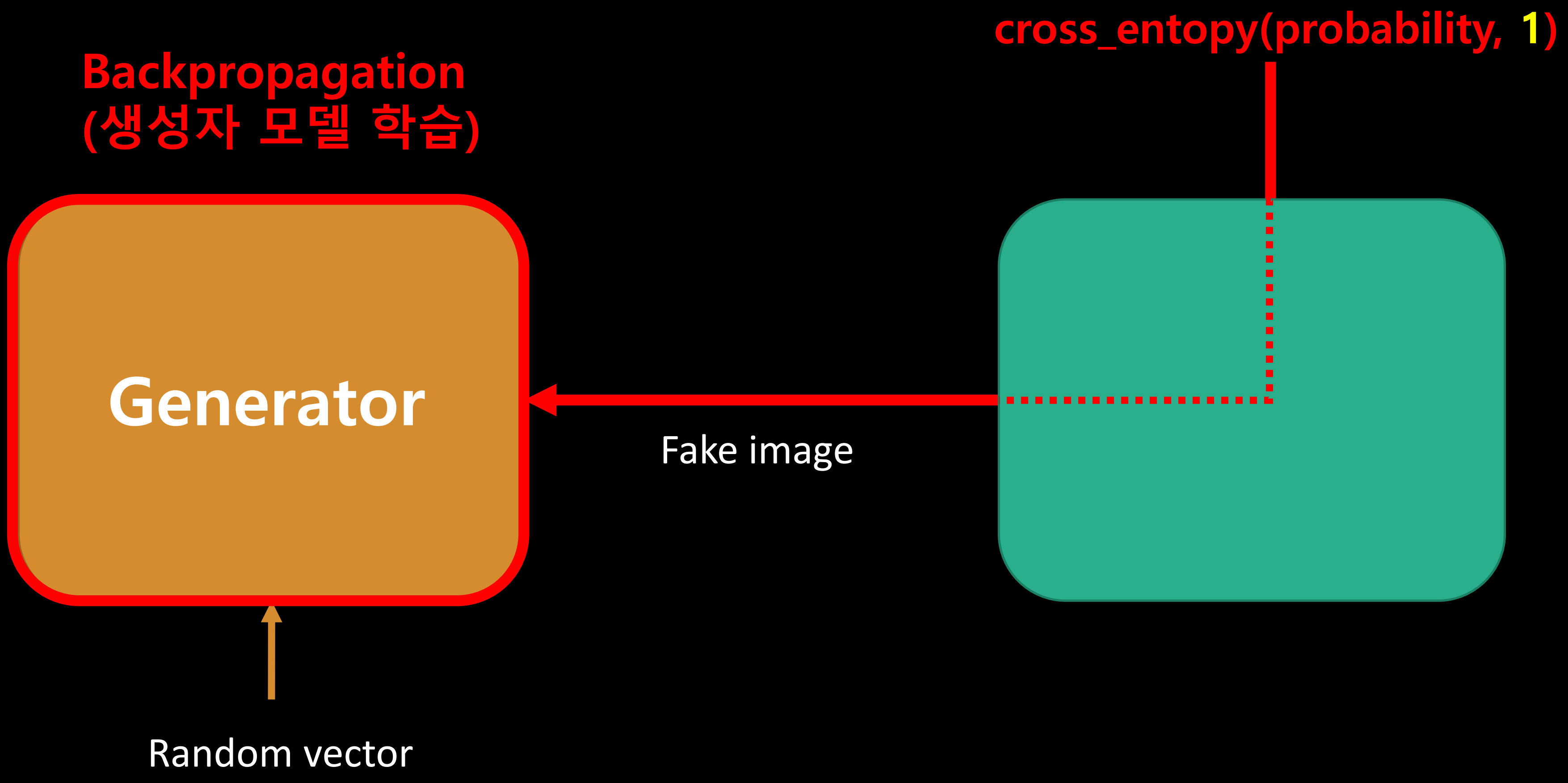
Generative Adversarial Network (GAN)



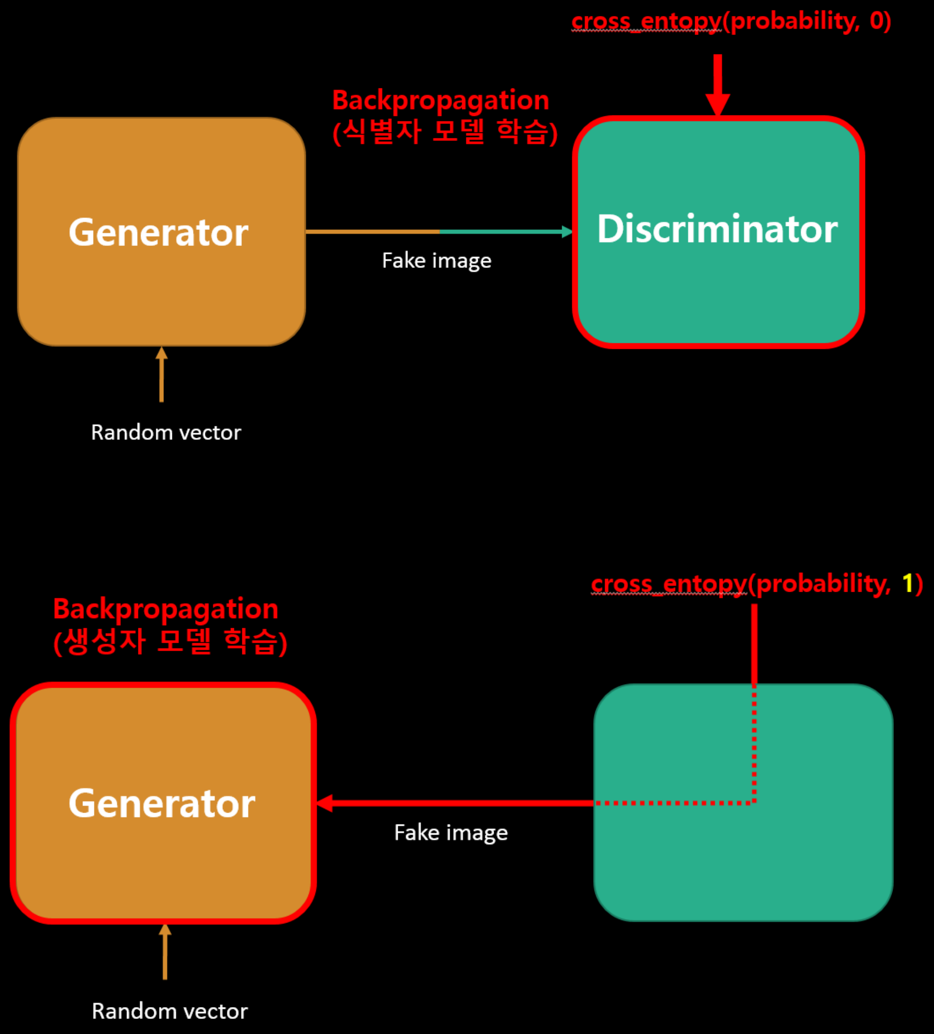
Generative Adversarial Network (GAN)



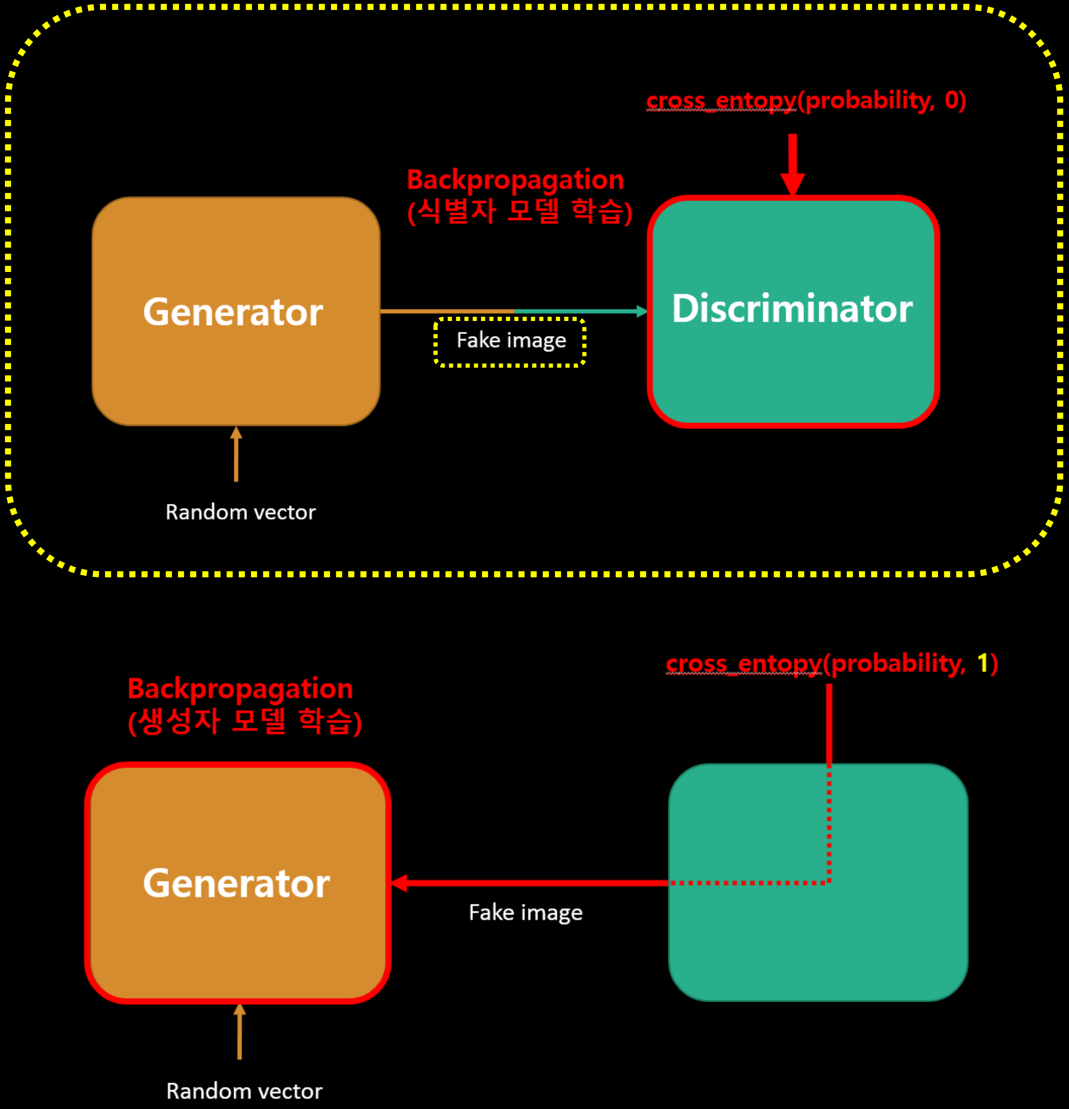
Generative Adversarial Network (GAN)



Generative Adversarial Network (GAN)

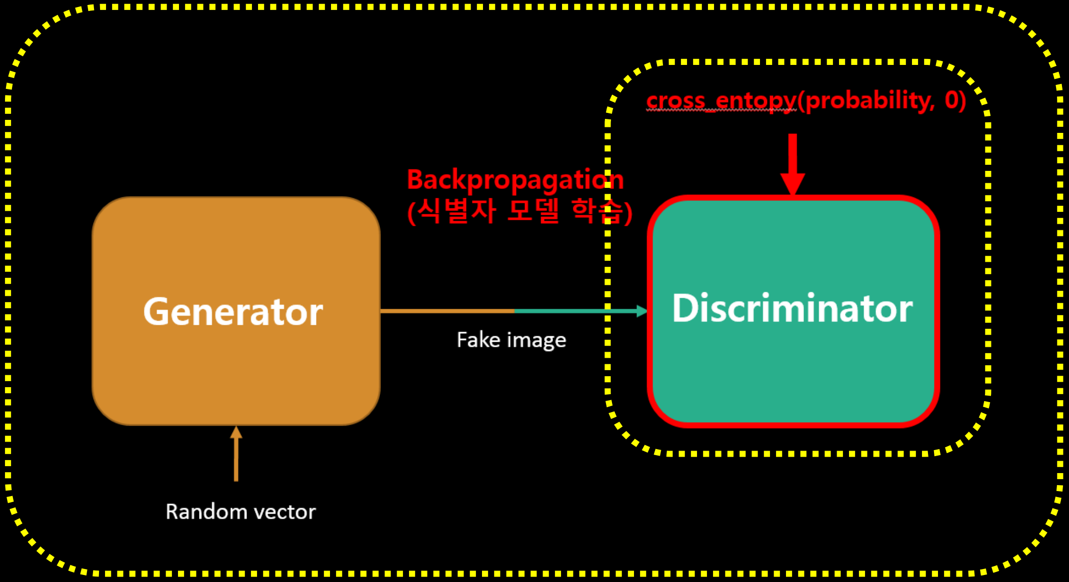


Generative Adversarial Network (GAN)

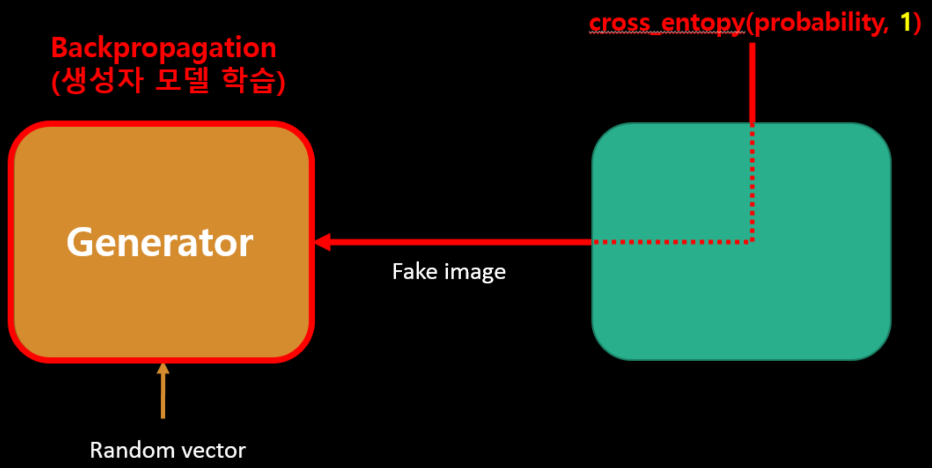


생성자는 가짜
이미지를 생성하고

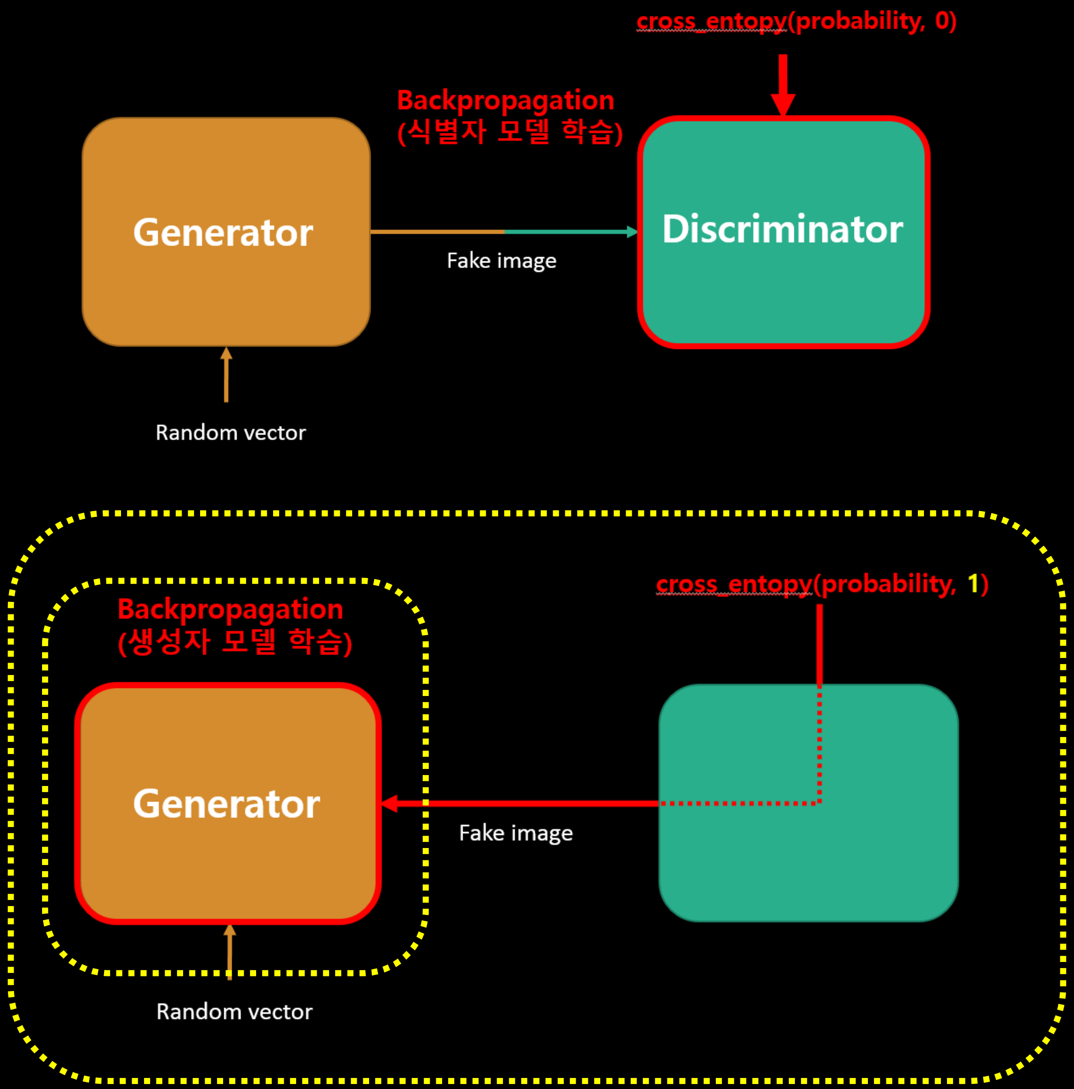
Generative Adversarial Network (GAN)



식별자는 가짜이미지를
'가짜'라고 판별하도록 학습

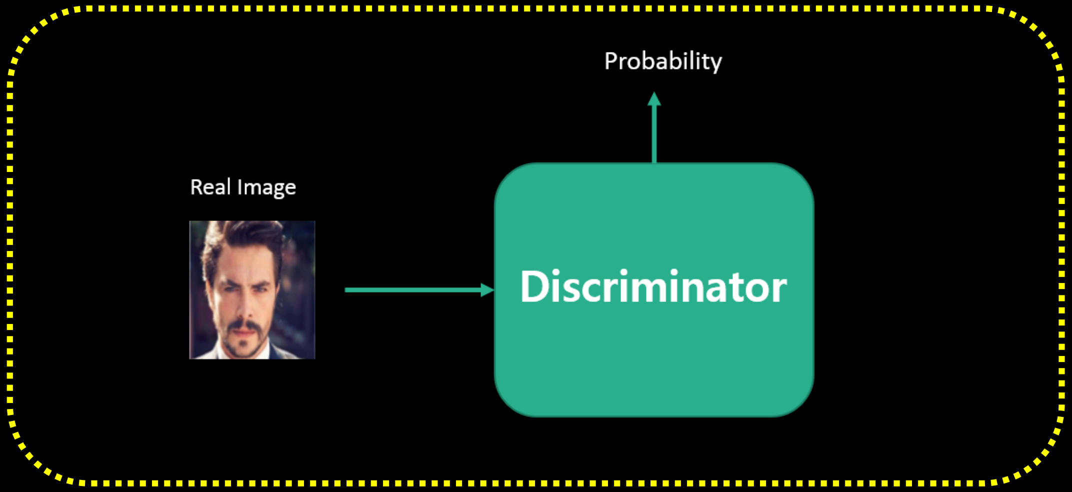


Generative Adversarial Network (GAN)

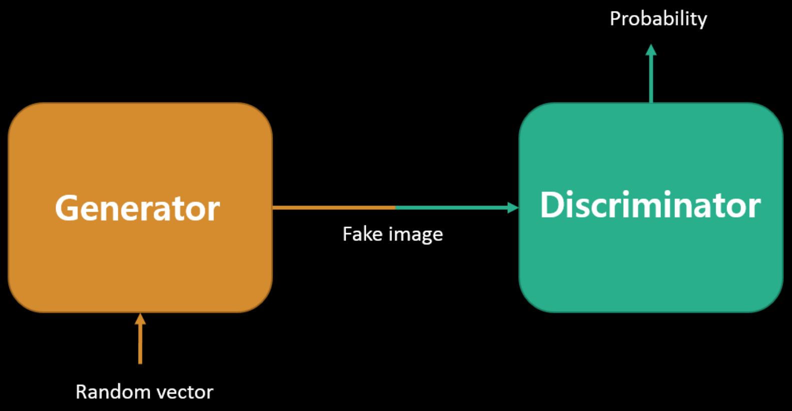


생성자는 식별자가 '가짜'를
'진짜'로 판별하도록 학습

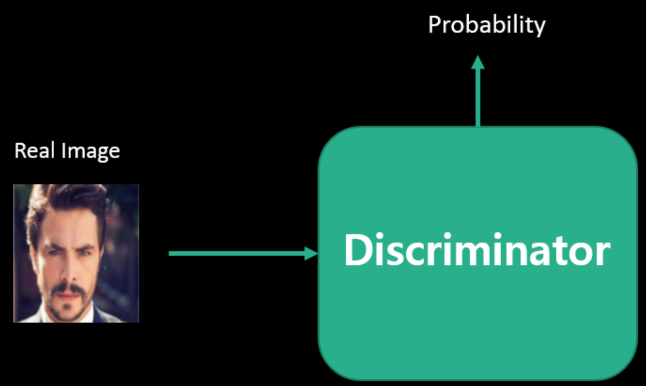
Generative Adversarial Network (GAN)

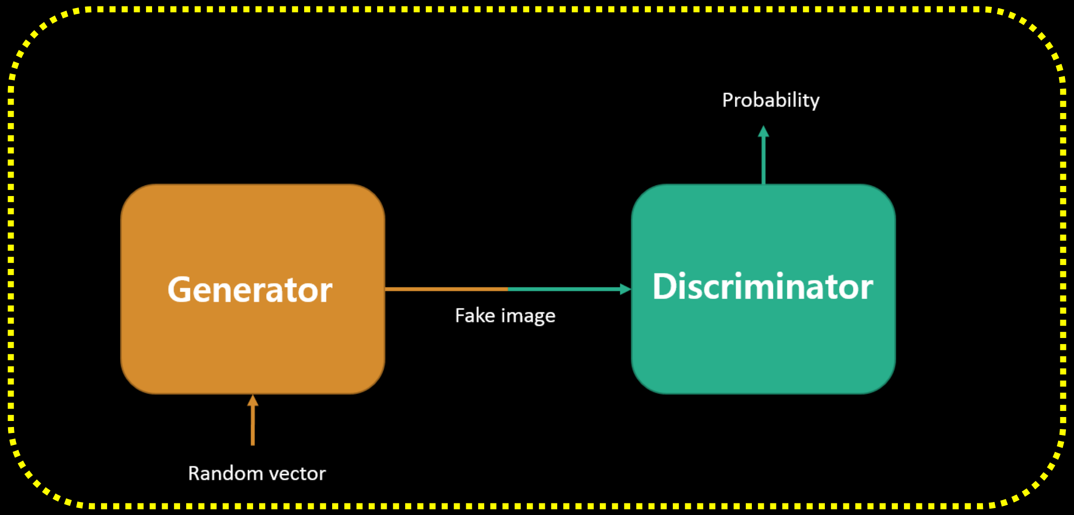


진짜 이미지를 가지고 학습할 때 구조
(식별자만 학습)



Generative Adversarial Network (GAN)

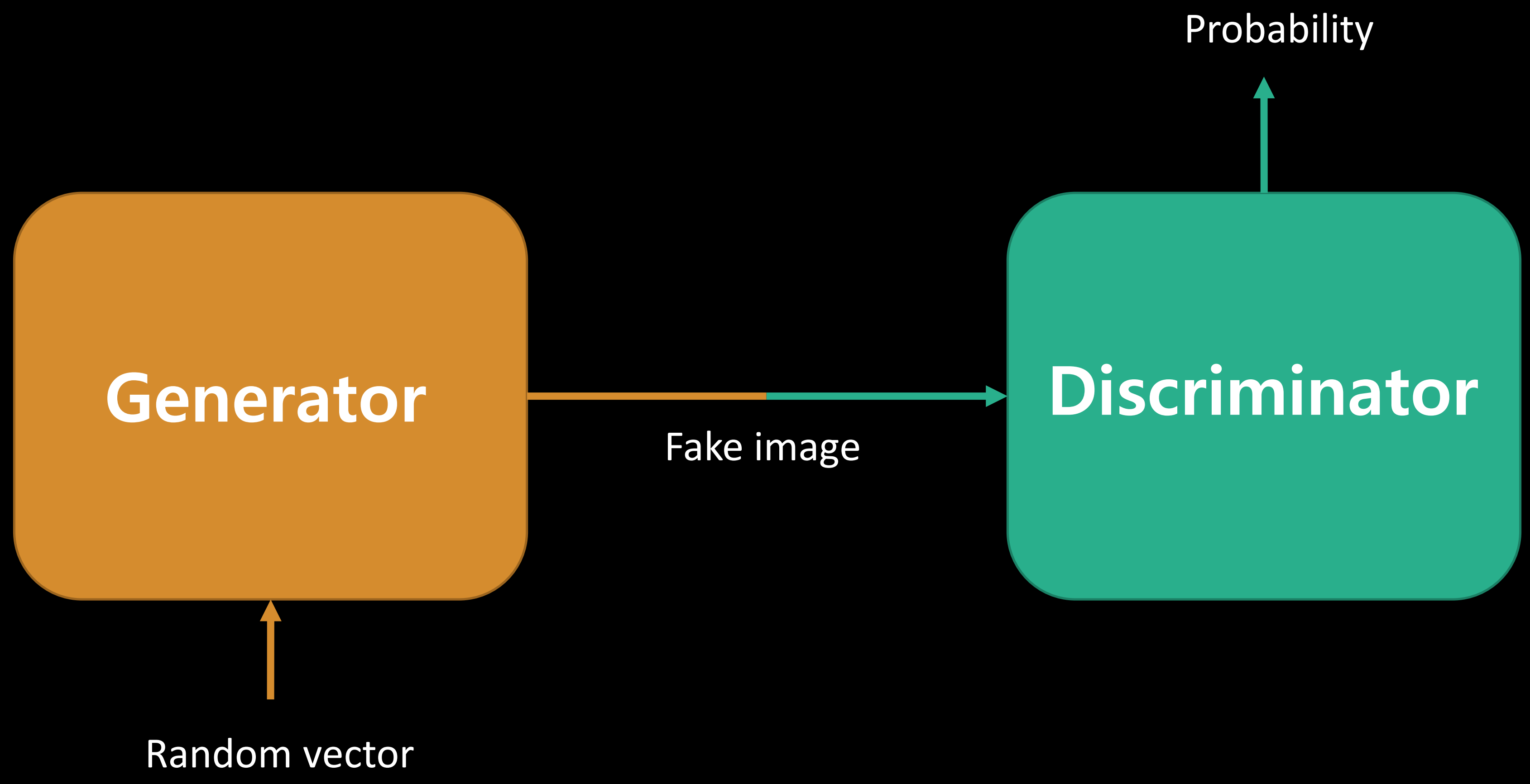




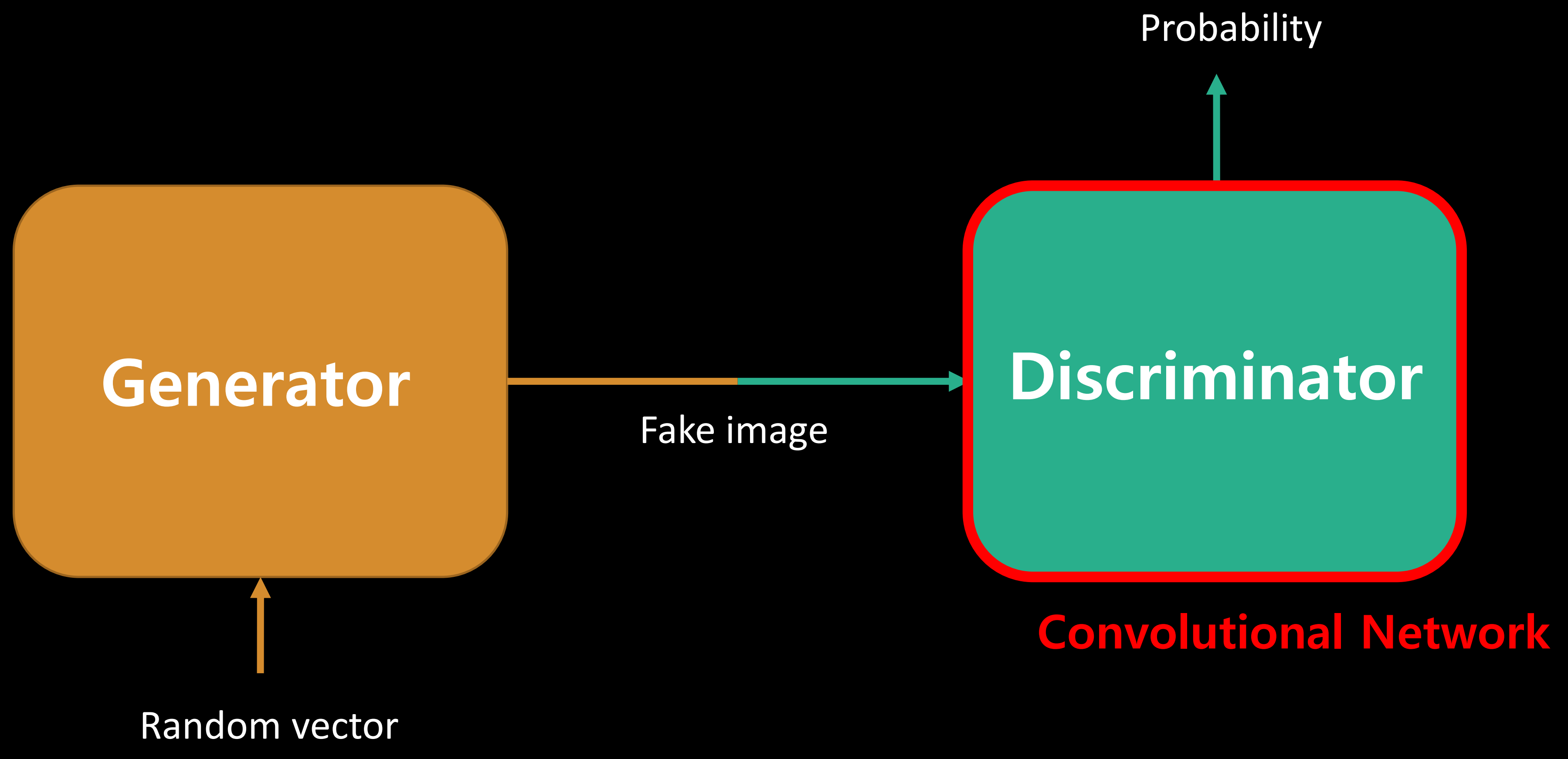
가짜 이미지를 가지고 학습할 때 구조
(식별자와 생성자 모두 학습)

3. DCGAN

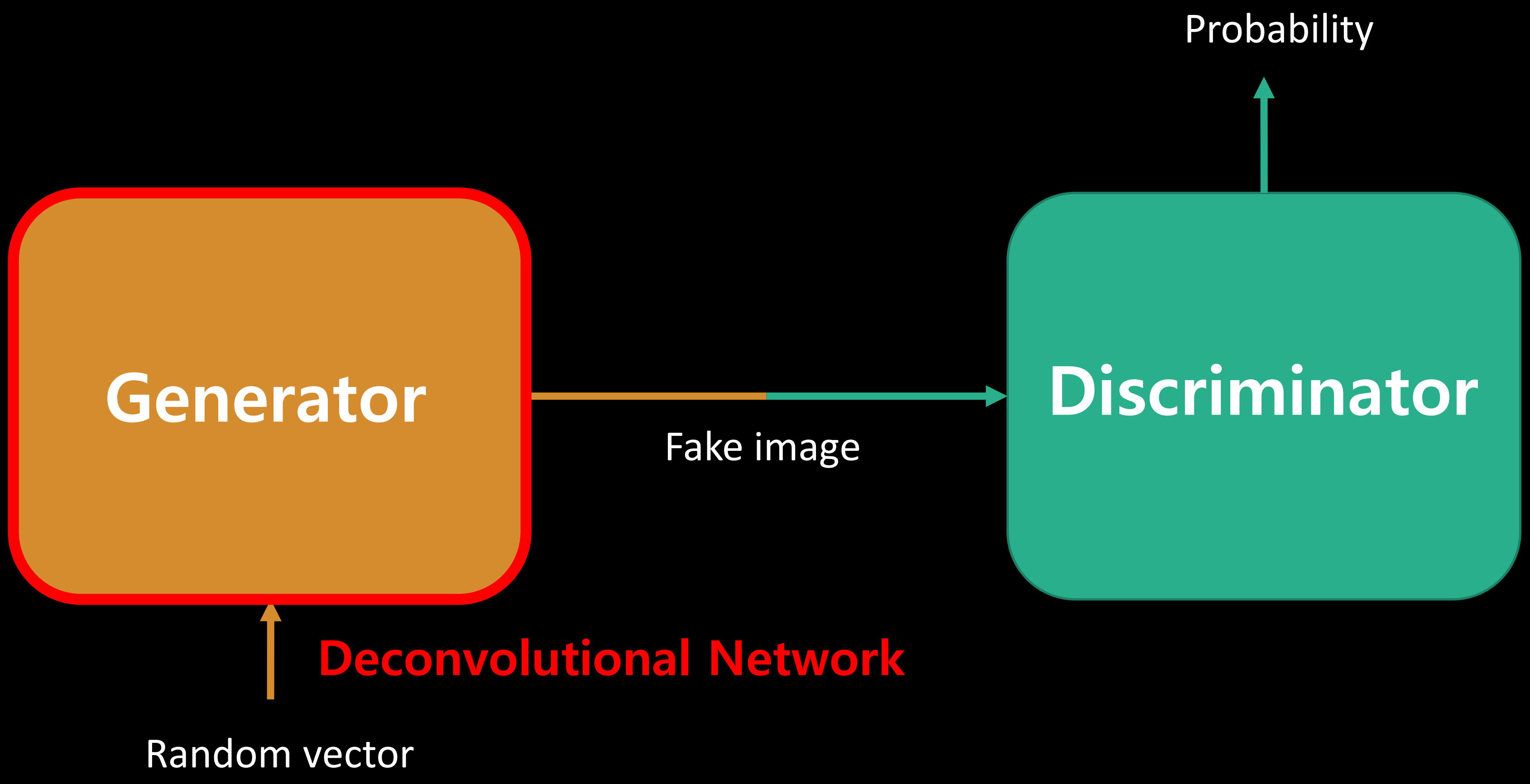
Deep Convolutional GAN (DCGAN)



Deep Convolutional GAN (DCGAN)

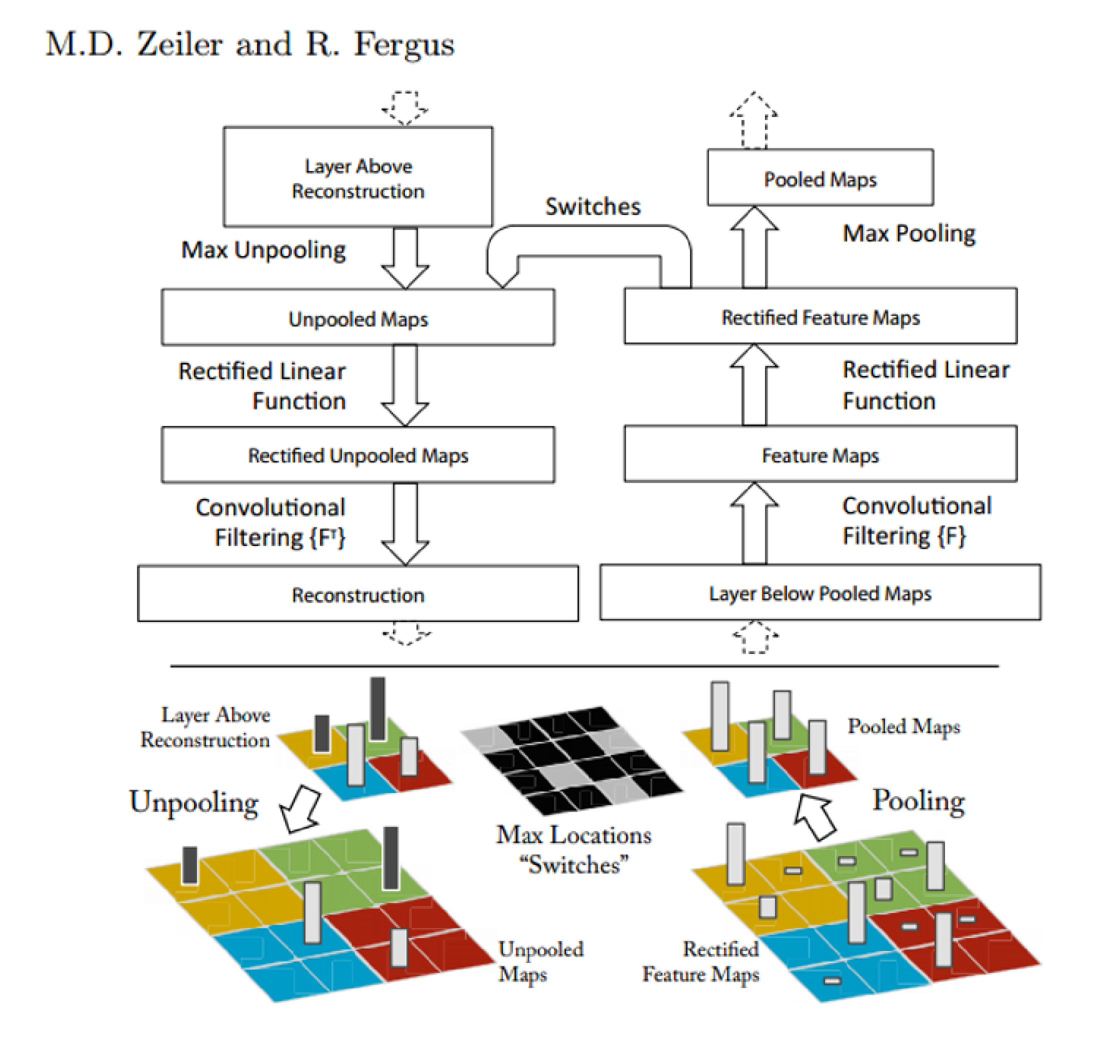


Deep Convolutional GAN (DCGAN)



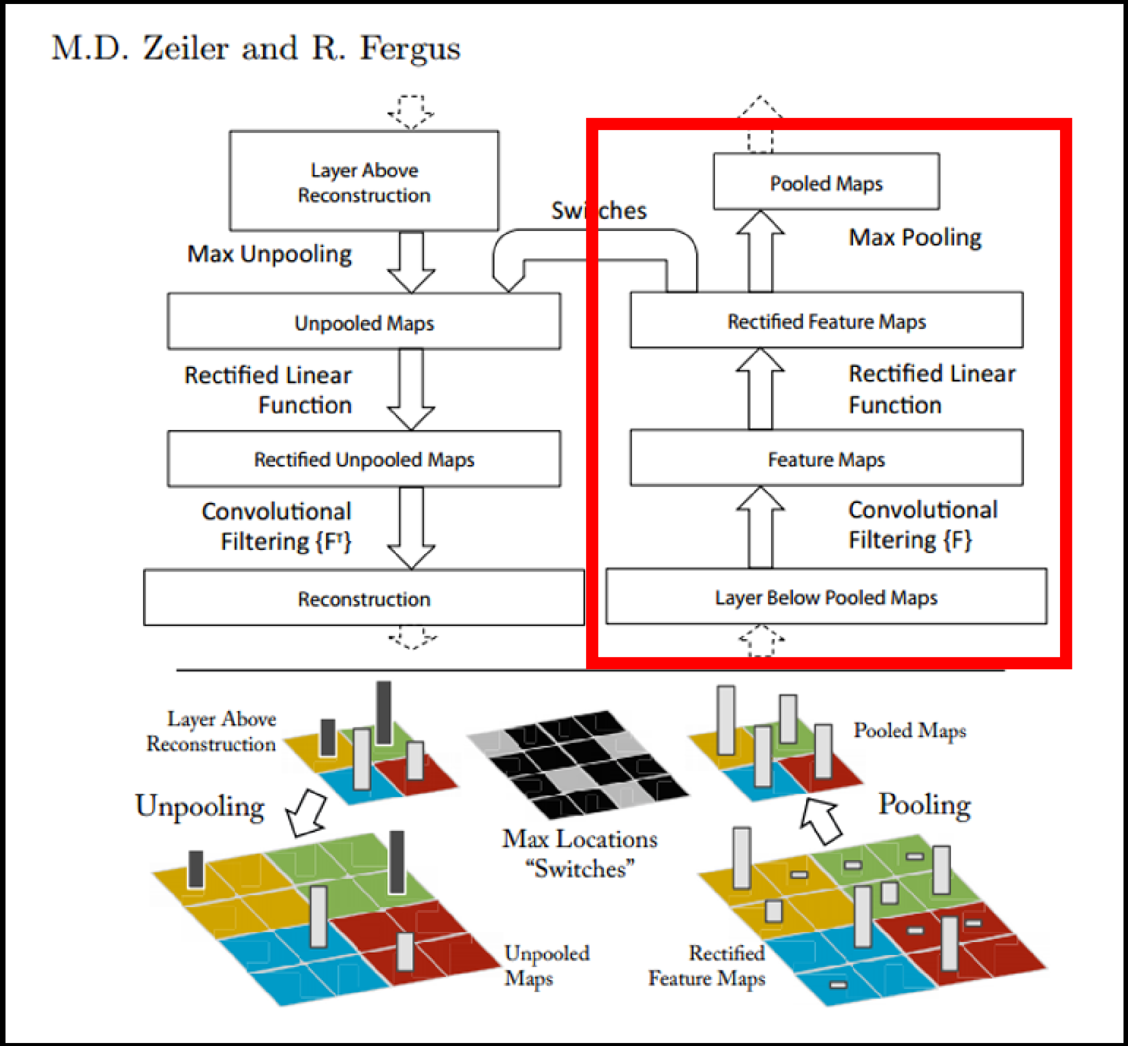
Deconvolution?

Deconvolutional Neural Network



visualizing and understanding convolutional networks (2013)

Deconvolutional Neural Network

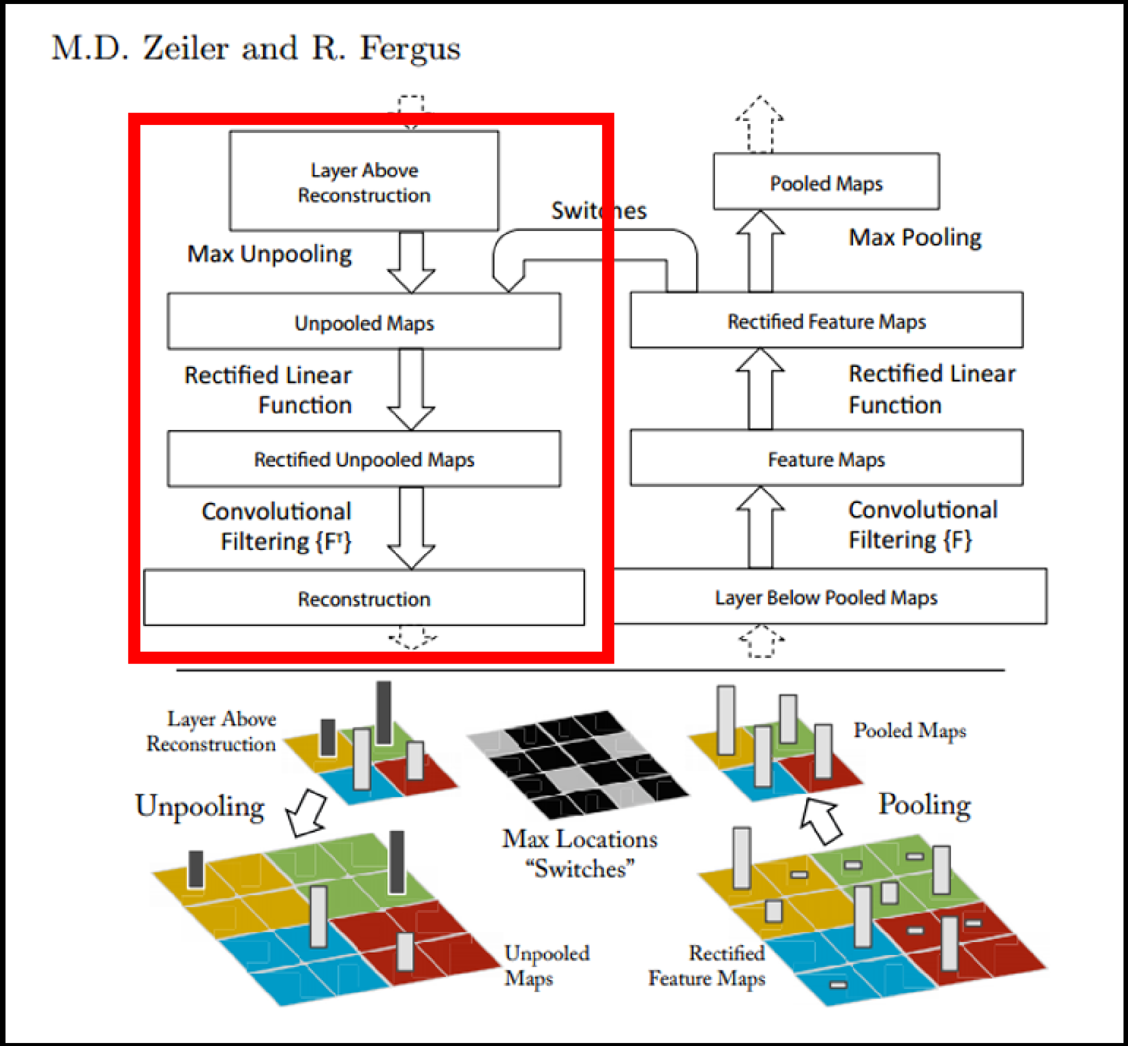


Convolutional Network

visualizing and understanding convolutional networks (2013)

Deconvolutional Neural Network

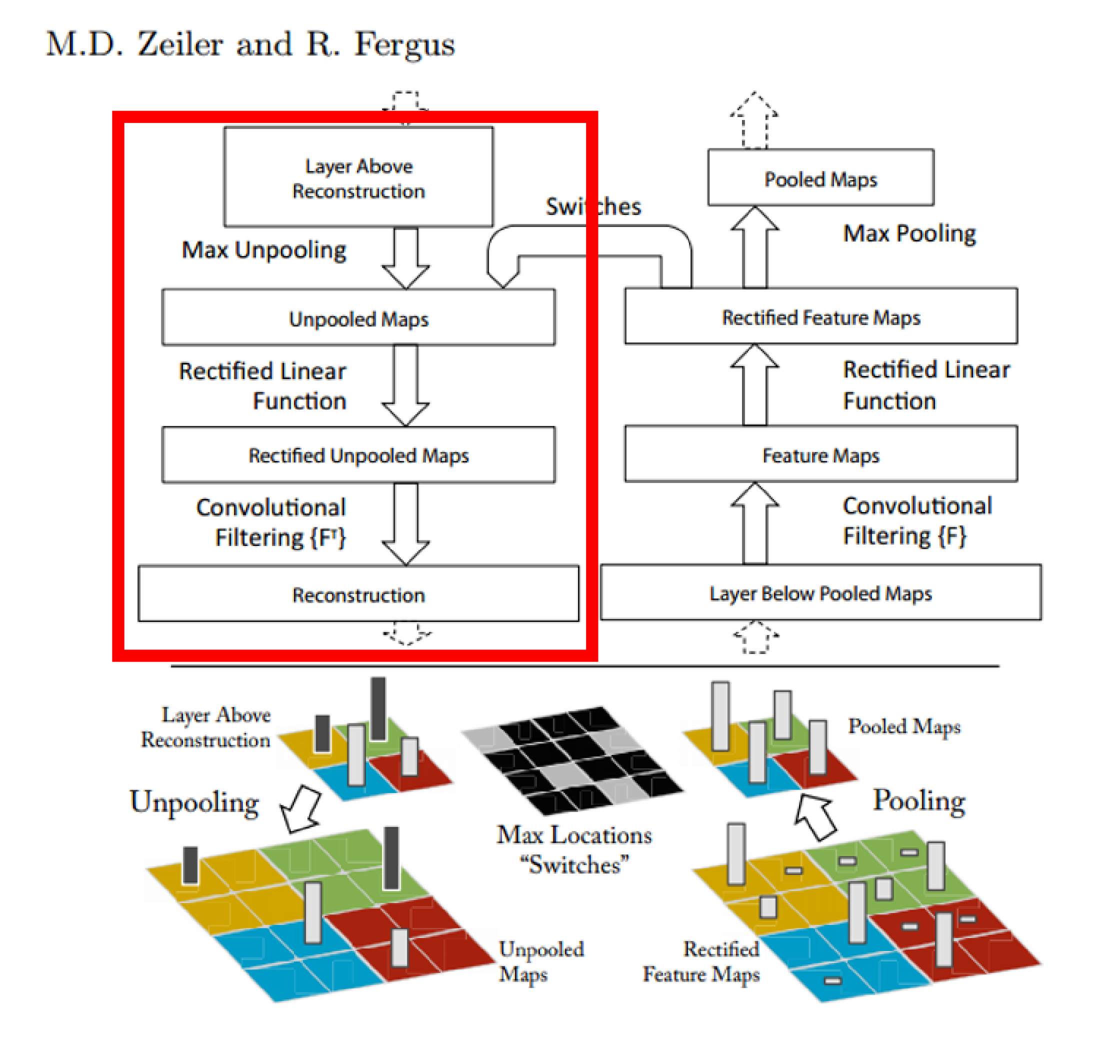
Deconvolutional Network



visualizing and understanding convolutional networks (2013)

Deconvolutional Neural Network

Deconvolutional Network
= CNN Backpropagation



visualizing and understanding convolutional networks (2013)

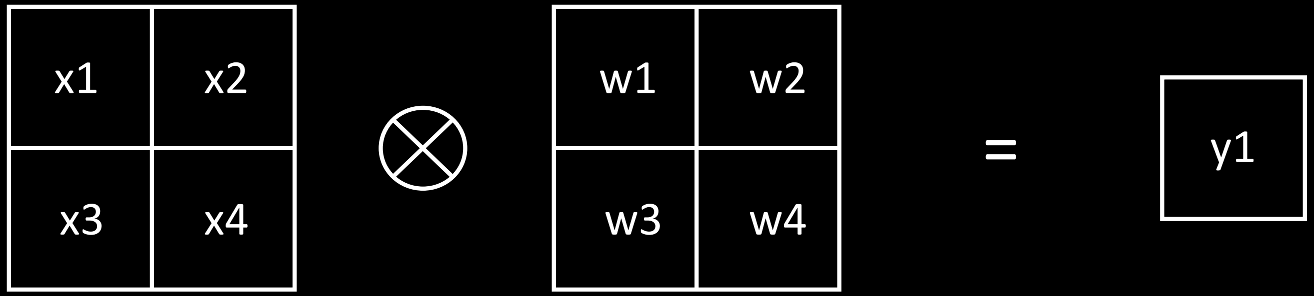
Deconvolution = Convolution Backprop

convolution forward

$$\begin{bmatrix} x1 & x2 \\ x3 & x4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \begin{bmatrix} y1 \end{bmatrix}$$

Deconvolution = Convolution Backprop

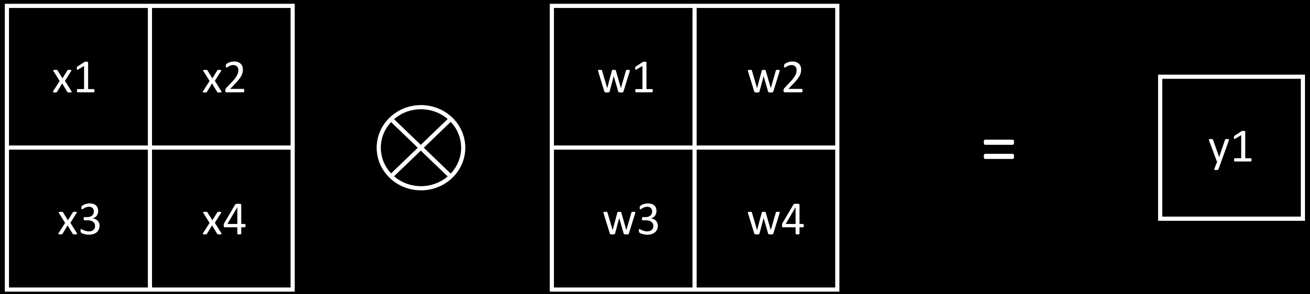
convolution forward



$$y1 = w1 \cdot x1 + w2 \cdot x2 + w3 \cdot x3 + w4 \cdot x4$$

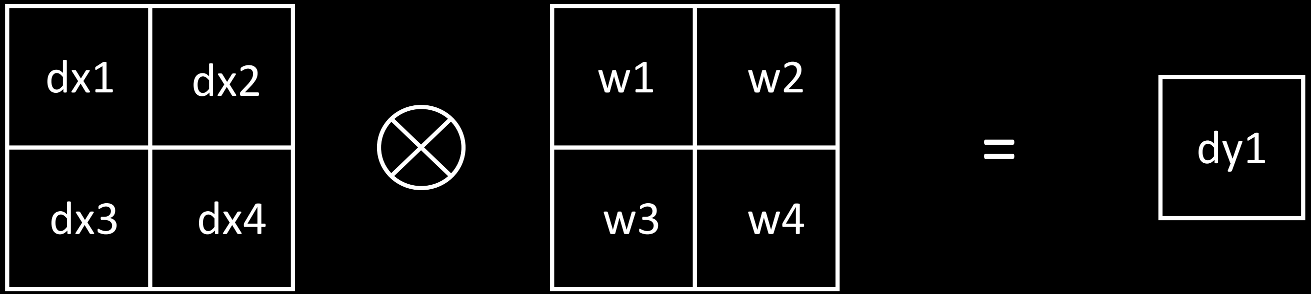
Deconvolution = Convolution Backprop

convolution forward



$$y1 = w1 \cdot x1 + w2 \cdot x2 + w3 \cdot x3 + w4 \cdot x4$$

convolution backward



Deconvolution = Convolution Backprop

convolution forward

$$\begin{bmatrix} x1 & x2 \\ x3 & x4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{y1}$$

$$y1 = w1 \cdot x1 + w2 \cdot x2 + w3 \cdot x3 + w4 \cdot x4$$

convolution backward

$$\begin{bmatrix} dx1 & dx2 \\ dx3 & dx4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{dy1}$$

$dy1 = \frac{\partial L}{\partial y1}$

(L은 loss값)

Deconvolution = Convolution Backprop

convolution forward

$$\begin{bmatrix} x1 & x2 \\ x3 & x4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{y1}$$

$$y1 = w1 \cdot x1 + w2 \cdot x2 + w3 \cdot x3 + w4 \cdot x4$$

convolution backward

$$\begin{bmatrix} dx1 & dx2 \\ dx3 & dx4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{dy1} \qquad dy1 = \frac{\partial L}{\partial y1}$$

$dx1 = ?$

Deconvolution = Convolution Backprop

convolution forward

$$\begin{bmatrix} x1 & x2 \\ x3 & x4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{y1}$$

$$y1 = w1 \cdot x1 + w2 \cdot x2 + w3 \cdot x3 + w4 \cdot x4$$

convolution backward

$$\begin{bmatrix} dx1 & dx2 \\ dx3 & dx4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{dy1} \qquad dy1 = \frac{\partial L}{\partial y1}$$

$$dx1 = \frac{\partial L}{\partial x1} = ?$$

Deconvolution = Convolution Backprop

convolution forward

$$\begin{bmatrix} x1 & x2 \\ x3 & x4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{y1}$$

y1 = w1 · x1 + w2 · x2 + w3 · x3 + w4 · x4

convolution backward

$$\begin{bmatrix} dx1 & dx2 \\ dx3 & dx4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{dy1}$$

$$dy1 = \frac{\partial L}{\partial y1}$$

$$dx1 = \frac{\partial L}{\partial x1} = \frac{\partial L}{\partial y1}\frac{\partial y1}{\partial x1} = ?$$

Deconvolution = Convolution Backprop

convolution forward

$$\begin{bmatrix} x1 & x2 \\ x3 & x4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{y1}$$

y1 = w1 · x1 + w2 · x2 + w3 · x3 + w4 · x4

convolution backward

$$\begin{bmatrix} dx1 & dx2 \\ dx3 & dx4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{dy1} \qquad dy1 = \frac{\partial L}{\partial y1}$$

$$dx1 = \frac{\partial L}{\partial x1} = \frac{\partial L}{\partial y1}\frac{\partial y1}{\partial x1} = dy1 \cdot w1$$

Deconvolution = Convolution Backprop

convolution forward

$$\begin{bmatrix} x1 & x2 \\ x3 & x4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{y1}$$

y1 = w1 · x1 + w2 · x2 + w3 · x3 + w4 · x4

convolution backward

$$\begin{bmatrix} dx1 & dx2 \\ dx3 & dx4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{dy1} \qquad dy1 = \frac{\partial L}{\partial y1}$$

$$dx1 = \frac{\partial L}{\partial x1} = \frac{\partial L}{\partial y1}\frac{\partial y1}{\partial x1} = dy1 \cdot w1 \longrightarrow dxi = dy1 \cdot wi \ \ (1 \leq i \leq 4)$$

Deconvolution = Convolution Backprop

숫자로 예를 들어보자

convolution backward

$$\begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 2 \end{bmatrix} \qquad dy1 = \frac{\partial L}{\partial y1}$$

$dxi = dy1 \cdot wi \quad (1 \leq i \leq 4)$

Deconvolution = Convolution Backprop

위에 있는 layer로 부터 받은 미분 값

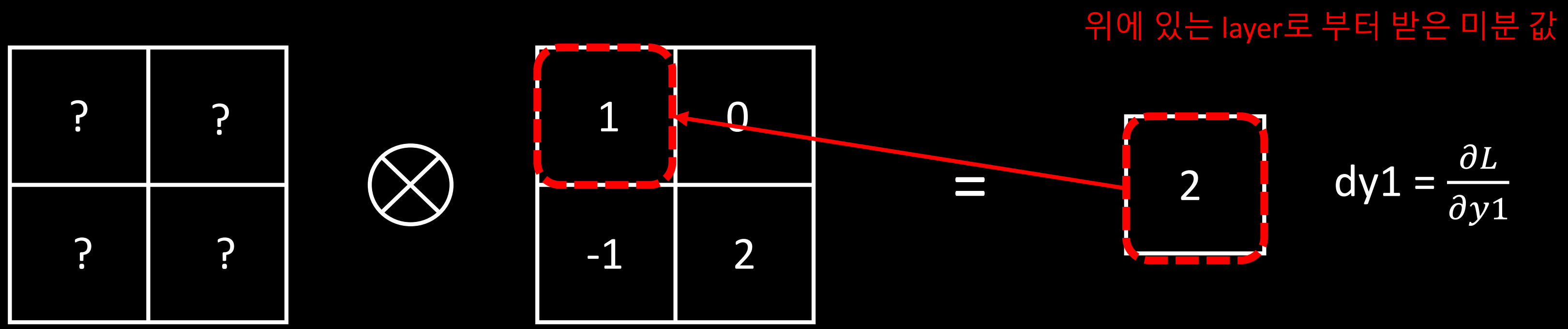
convolution backward

$$\begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ -1 & 2 \end{bmatrix} = \boxed{2} \quad dy1 = \frac{\partial L}{\partial y1}$$

dxi = dy1 · wi (1 ≤ i ≤4)

Deconvolution = Convolution Backprop

convolution backward



dxi = dy1 · wi (1 ≤ i ≤4)

Deconvolution = Convolution Backprop

convolution backward

$$\begin{bmatrix} 2 & ? \\ ? & ? \end{bmatrix} \otimes \begin{bmatrix} \boxed{1} & 0 \\ -1 & 2 \end{bmatrix} = \boxed{2}$$

위에 있는 layer로 부터 받은 미분 값

$dy1 = \frac{\partial L}{\partial y1}$

$dxi = dy1 \cdot wi \ \ (1 \leq i \leq 4)$ $\quad i = 1$

Deconvolution = Convolution Backprop

convolution backward

위에 있는 layer로 부터 받은 미분 값

$$\begin{bmatrix} 2 & 0 \\ ? & ? \end{bmatrix} \otimes \begin{bmatrix} 1 & \boxed{0} \\ -1 & 2 \end{bmatrix} = \boxed{2} \qquad dy1 = \frac{\partial L}{\partial y1}$$

dxi = dy1 · wi (1 ≤ i ≤4) i = 2

Deconvolution = Convolution Backprop

convolution backward

위에 있는 layer로 부터 받은 미분 값

$$\begin{bmatrix} 2 & 0 \\ -2 & ? \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ \boxed{-1} & 2 \end{bmatrix} = \boxed{2} \qquad dy1 = \frac{\partial L}{\partial y1}$$

$dxi = dy1 \cdot wi \;\; (1 \leq i \leq 4)$ $\quad i = 3$

Deconvolution = Convolution Backprop

convolution backward

위에 있는 layer로 부터 받은 미분 값

$$\begin{bmatrix} 2 & 0 \\ -2 & 4 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ -1 & \boxed{2} \end{bmatrix} = \boxed{2} \qquad dy1 = \frac{\partial L}{\partial y1}$$

dxi = dy1 · wi (1 ≤ i ≤4)   i = 4

Deconvolution = Convolution Backprop

deconvolution forward

$$\begin{bmatrix} x1 & x2 \\ x3 & x4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \begin{bmatrix} y1 \end{bmatrix}$$

$xi = y1 \cdot wi \;\; (1 \leq i \leq 4)$

convolution backward

$$\begin{bmatrix} dx1 & dx2 \\ dx3 & dx4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \begin{bmatrix} dy1 \end{bmatrix}$$

$dxi = dy1 \cdot wi \;\; (1 \leq i \leq 4)$

Deconvolution = Convolution Backprop

deconvolution forward

$$\begin{bmatrix} x1 & x2 \\ x3 & x4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{y1}$$

$xi = y1 \cdot wi \ \ (1 \leq i \leq 4)$

convolution backward

위에 있는 layer로 부터 받은 미분 값

$$\begin{bmatrix} dx1 & dx2 \\ dx3 & dx4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{dy1}$$

$dxi = dy1 \cdot wi \ \ (1 \leq i \leq 4)$

Deconvolution = Convolution Backprop

deconvolution forward

아래에 있는 layer로 부터 받은 activation 값

$$\begin{bmatrix} x1 & x2 \\ x3 & x4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{y1}$$

$xi = y1 \cdot wi \ \ (1 \leq i \leq 4)$

convolution backward

위에 있는 layer로 부터 받은 미분 값

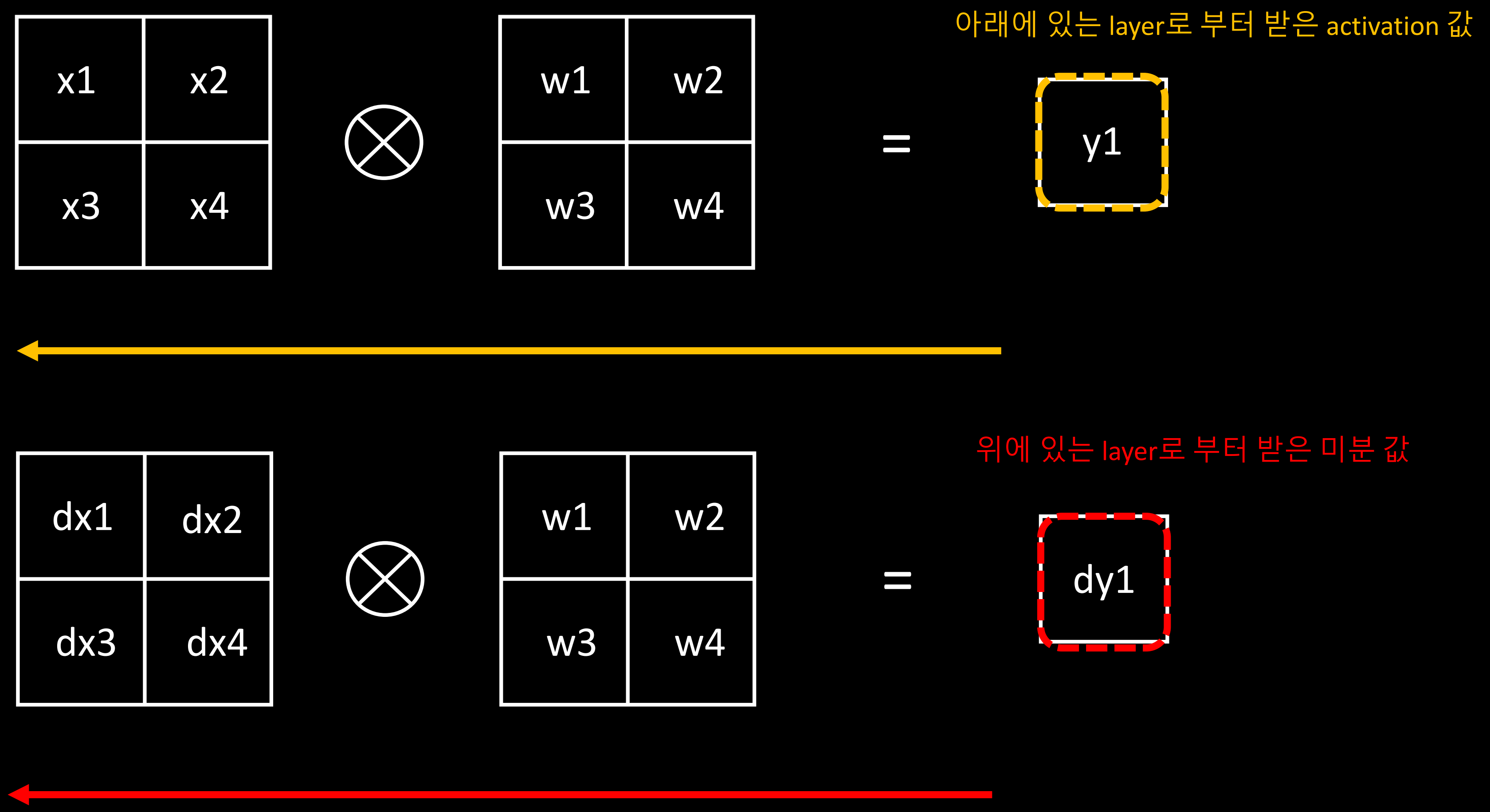
$$\begin{bmatrix} dx1 & dx2 \\ dx3 & dx4 \end{bmatrix} \otimes \begin{bmatrix} w1 & w2 \\ w3 & w4 \end{bmatrix} = \boxed{dy1}$$

$dxi = dy1 \cdot wi \ \ (1 \leq i \leq 4)$

Deconvolution = Convolution Backprop

deconvolution forward

convolution backward



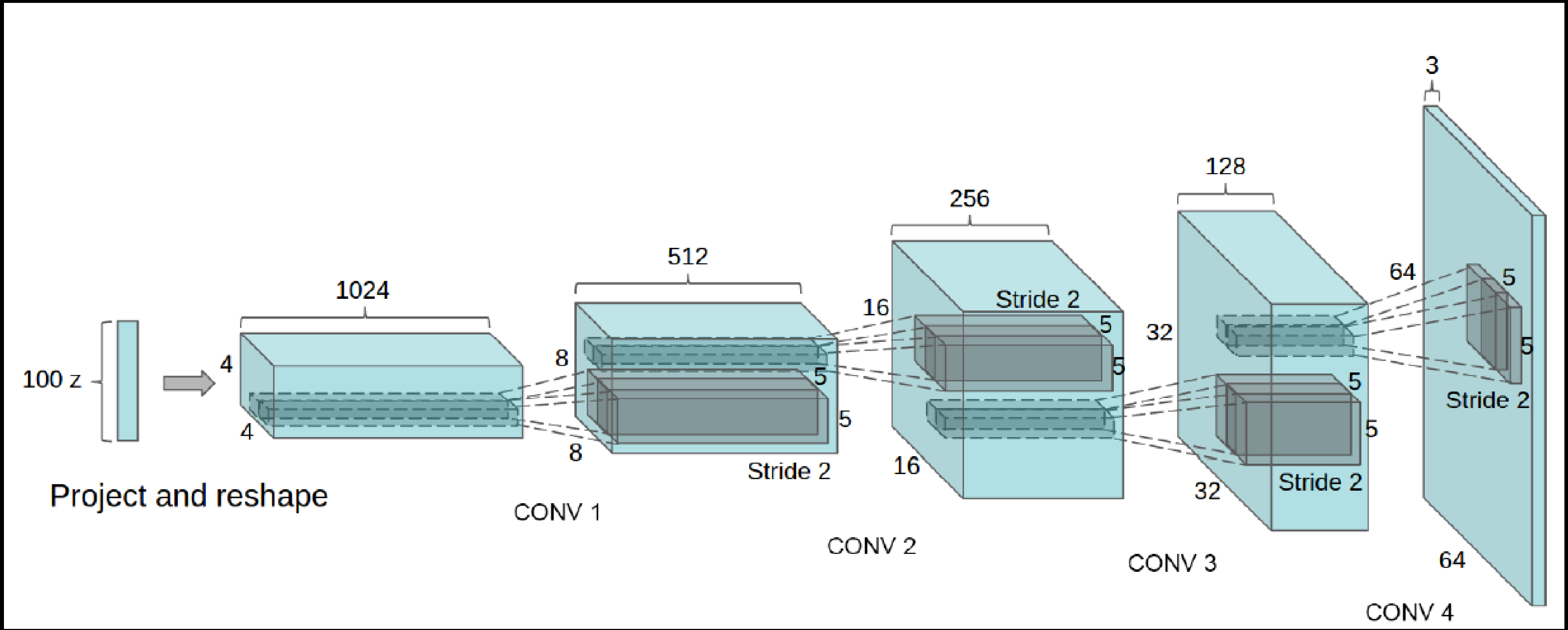
결론은

Convolution Backward = Deconvolution Forward

Convolution Forward = Deconvolution Backward

다시 DCGAN으로 돌아와서

Deep Convolutional GAN (DCGAN)



Deep Convolutional GAN (DCGAN)

Discriminator = Convolutional Network

Generator= Deconvolutional Network

Deep Convolutional GAN (DCGAN)

이전에도 CNN과 DCNN을 사용한 연구는 많았지만 DCGAN에서 다른 점이 있다

1. **Pooling Layer를 사용하지 않음 -> All Strided Convolution**

Pooling Layer를 사용하게 되면 Blocky한 이미지들이 생성되는데 이를 방지

Striving for simplicity: The all convolutional net (2014) 에서 아이디어를 얻음

Deep Convolutional GAN (DCGAN)

이전에도 CNN과 DCNN을 사용한 연구는 많았지만
DCGAN에서 다른 점이 있다

1. Pooling Layer를 사용하지 않음

2. Batch Normalization

CNN과 DCNN에 Batch Normalization Layer를 추가

Deep Convolutional GAN (DCGAN)

이전에도 CNN과 DCNN을 사용한 연구는 많았지만
DCGAN에서 다른 점이 있다

1. Pooling Layer를 사용하지 않음

2. Batch Normalization 사용

3. Fully Connected Layer 최소화 -> All Convolution Layer

Generator의 첫 번째 layer와 Discriminator의 마지막 layer를
제외하고는 모두 Convolution Layer를 사용

Going deeper into neural networks (2015)에서 아이디어를 얻음

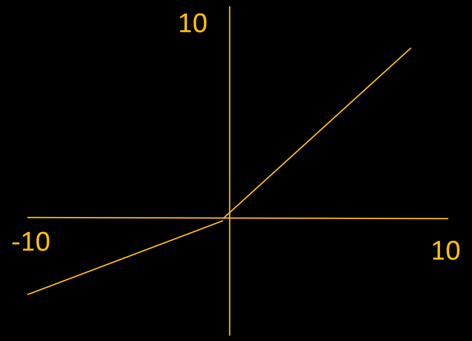
Deep Convolutional GAN (DCGAN)

이전에도 CNN과 DCNN을 사용한 연구는 많았지만
DCGAN에서 다른 점이 있다

1. Pooling Layer를 사용하지 않음

2. Batch Normalization 사용

3. Fully Connected Layer 최소화



4. ReLU와 Leaky ReLU 사용
Generator는 ReLU를 사용, Discriminator는 Leaky ReLu를 사용
실험적으로 더 높은 퀄리티의 이미지를 생성

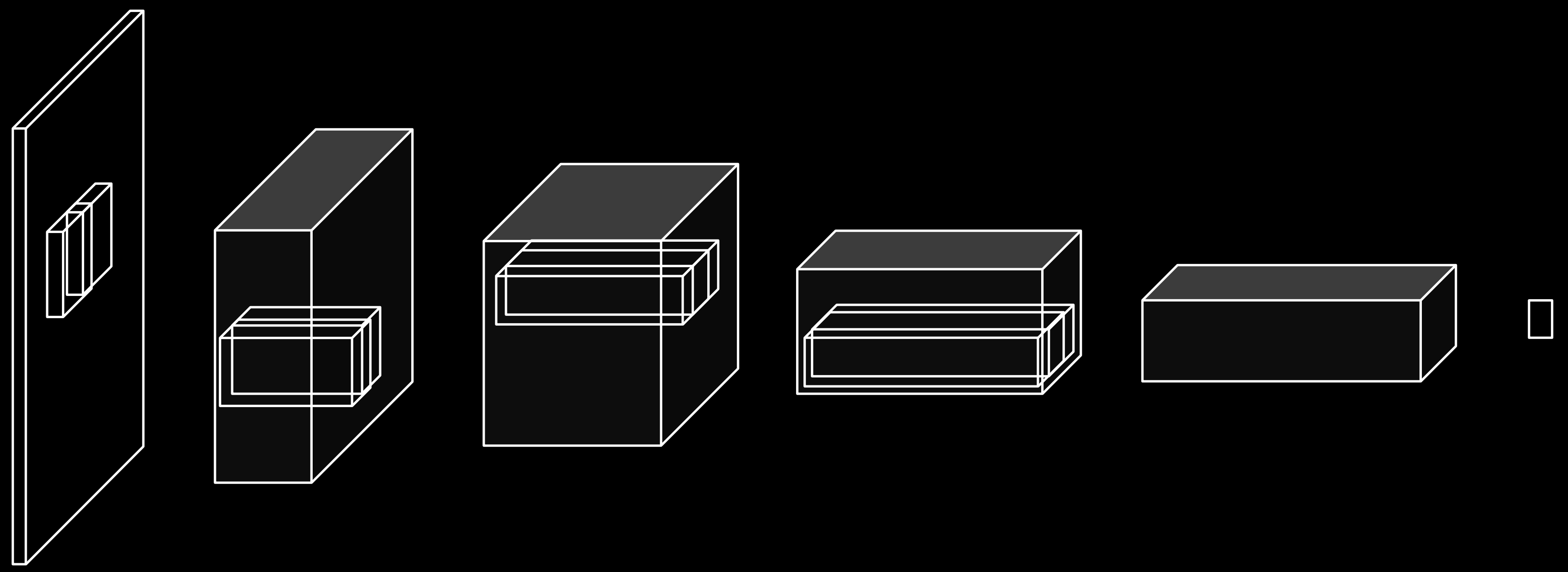
Deep Convolutional GAN (DCGAN)

이전에도 CNN과 DCNN을 사용한 연구는 많았지만 DCGAN에서 다른 점이 있다

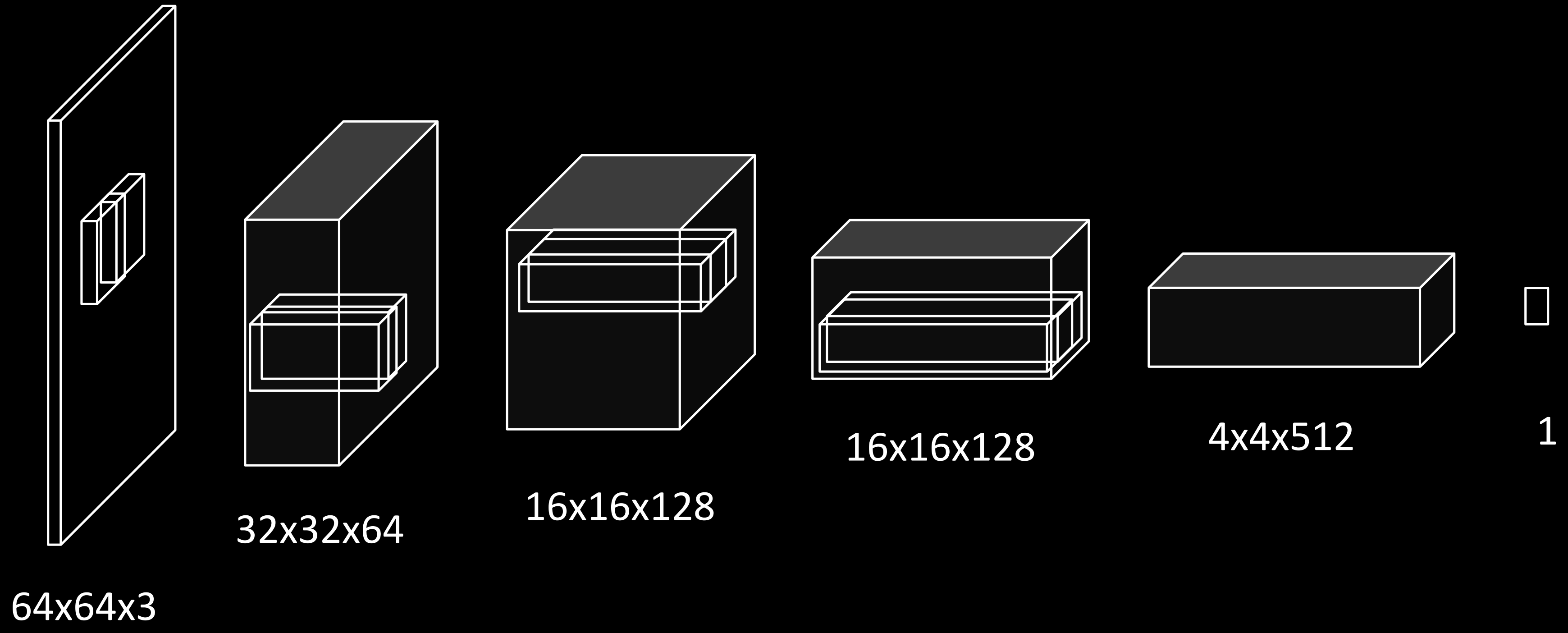
1. Pooling Layer를 사용하지 않음
2. Batch Normalization 사용
3. Fully Connected Layer 최소화
4. ReLU와 Leaky ReLU 사용

결과적으로 이전 연구보다 퀄리티가 더 높은 이미지를 생성

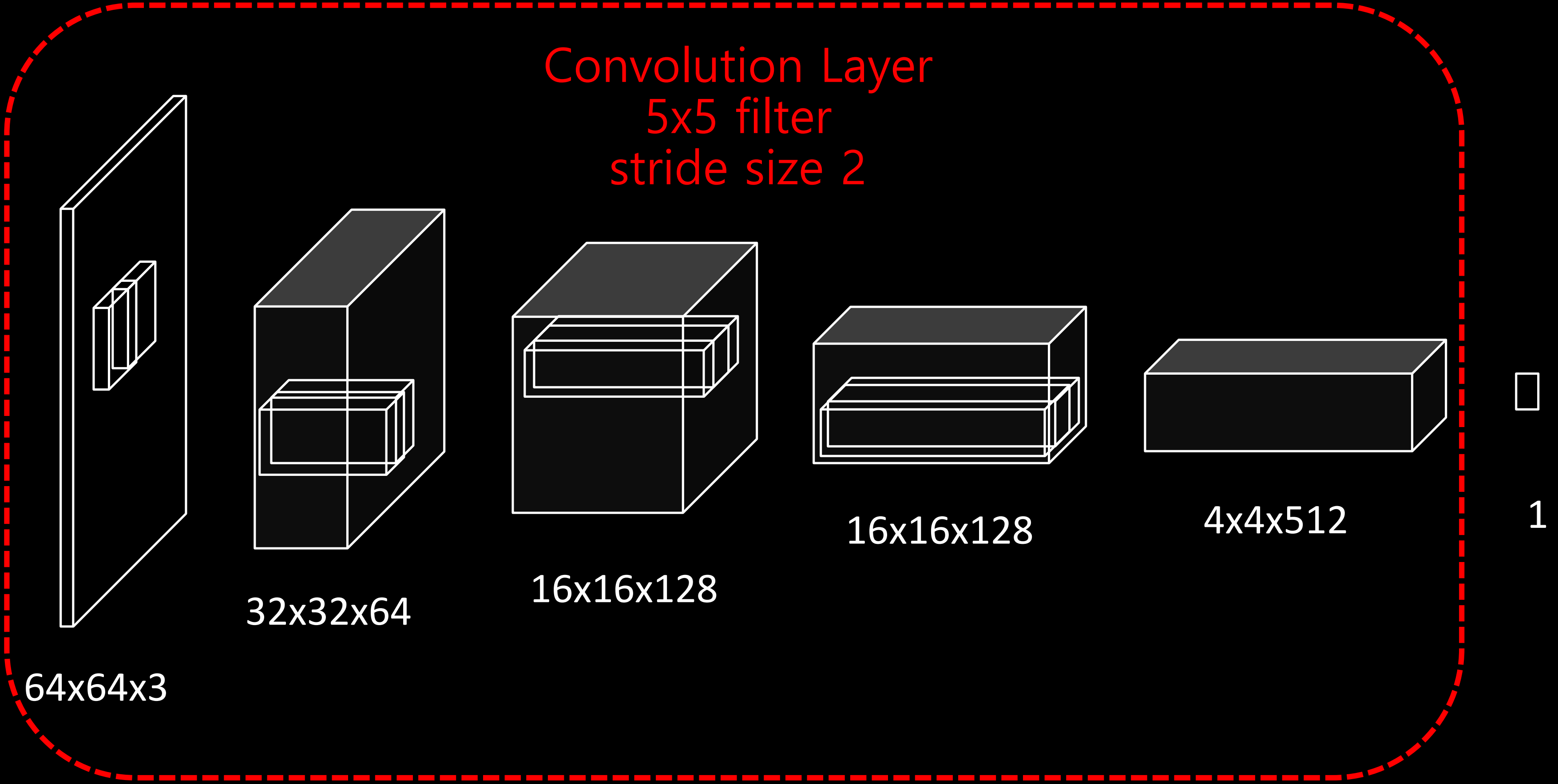
Discriminator(식별자)



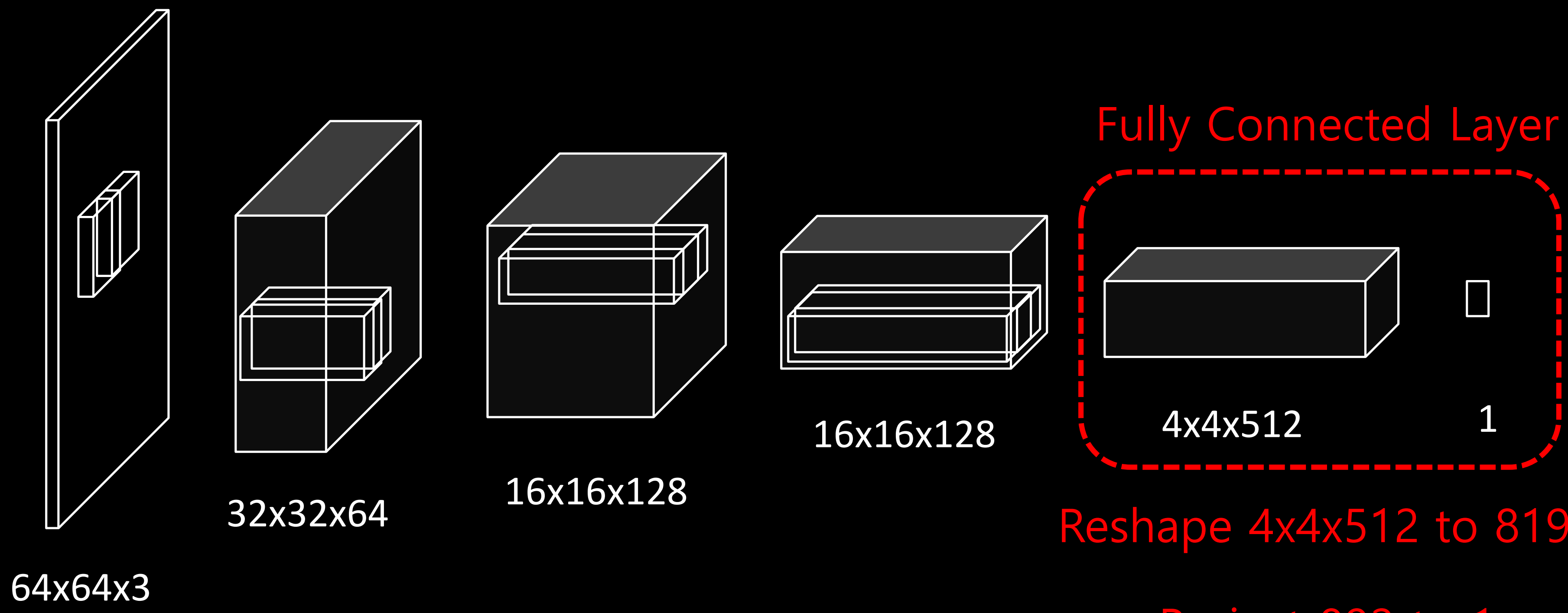
Discriminator(식별자)



Discriminator(식별자)



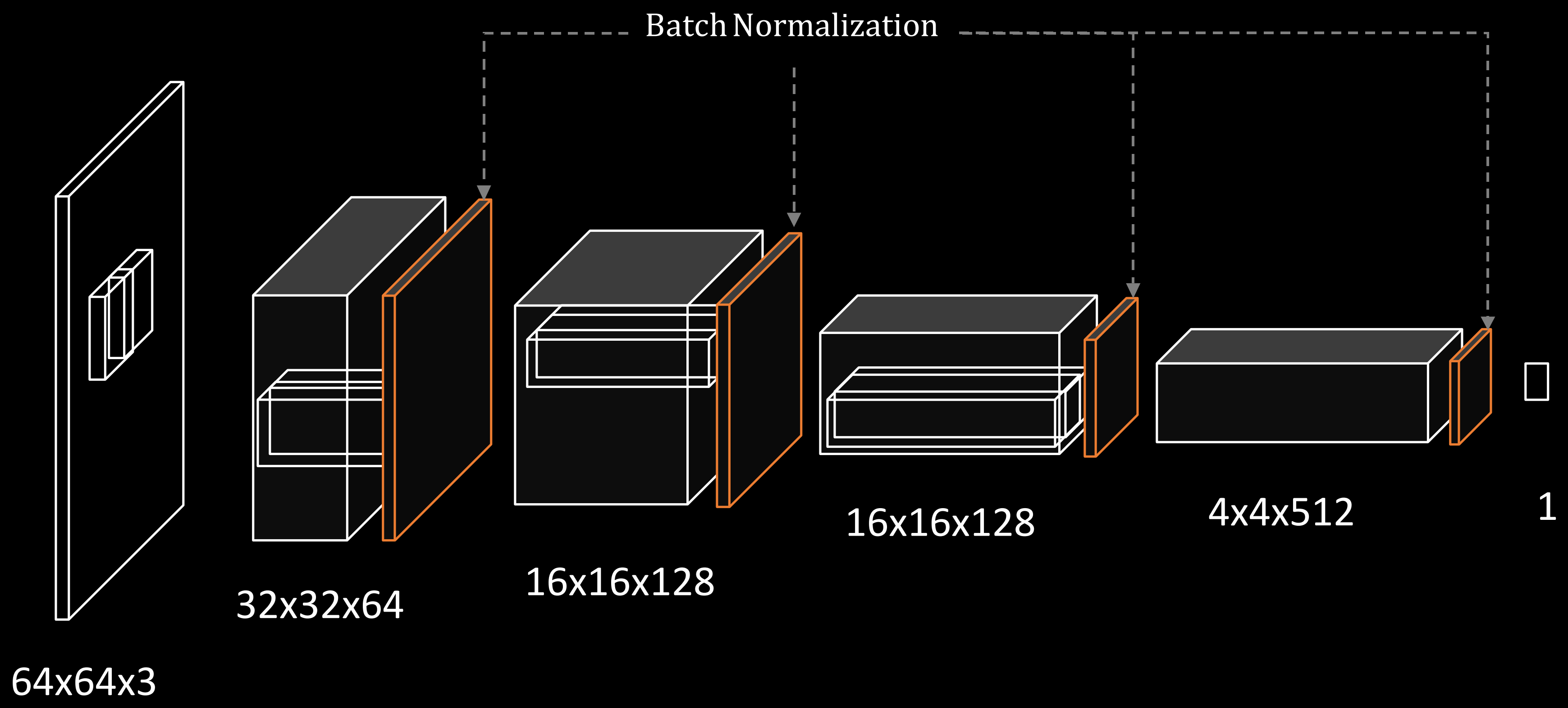
Discriminator(식별자)



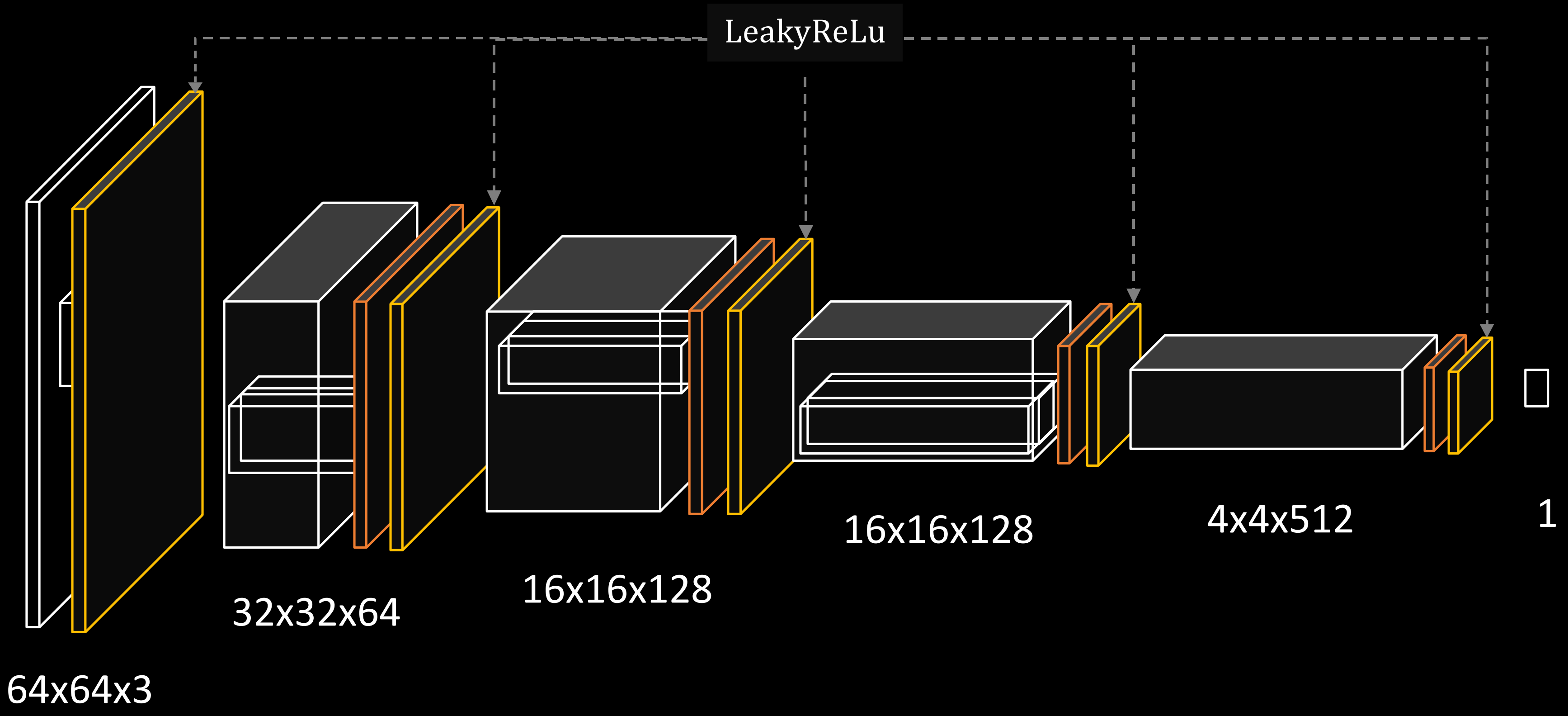
Reshape 4x4x512 to 8192

Project 892 to 1

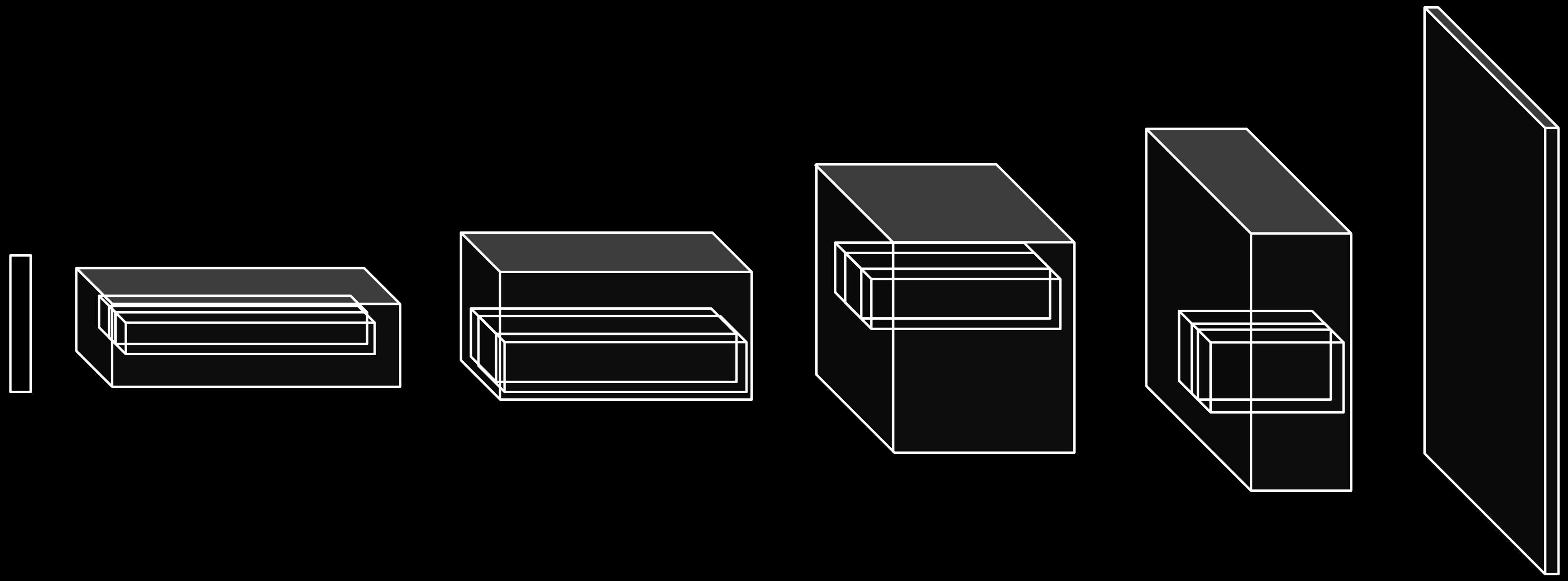
Discriminator(식별자)



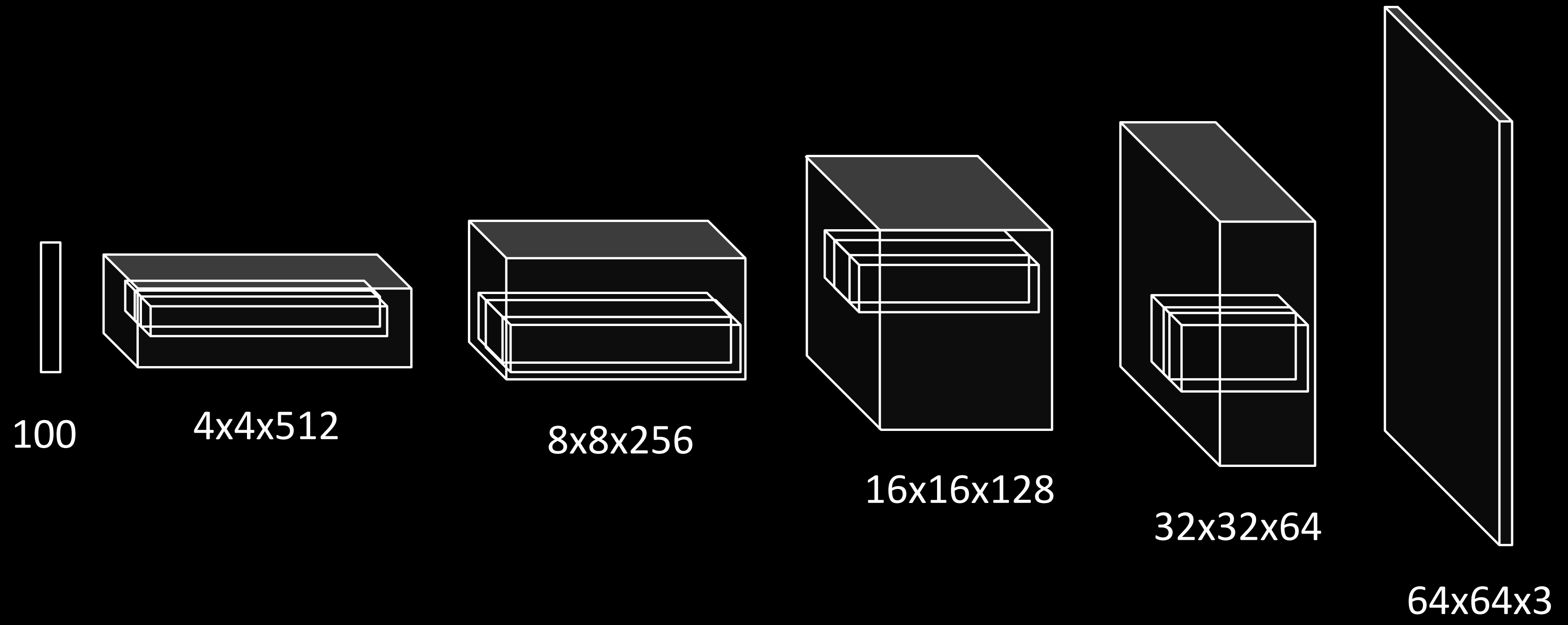
Discriminator(식별자)



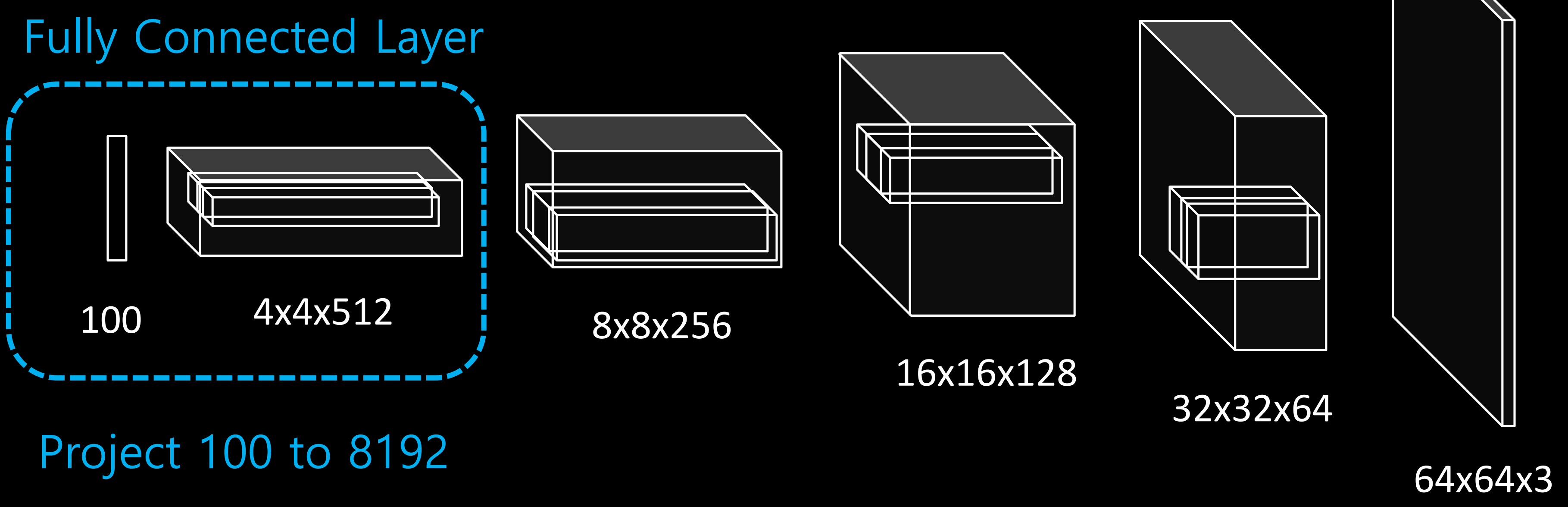
Generator(생성자)



Generator(생성자)



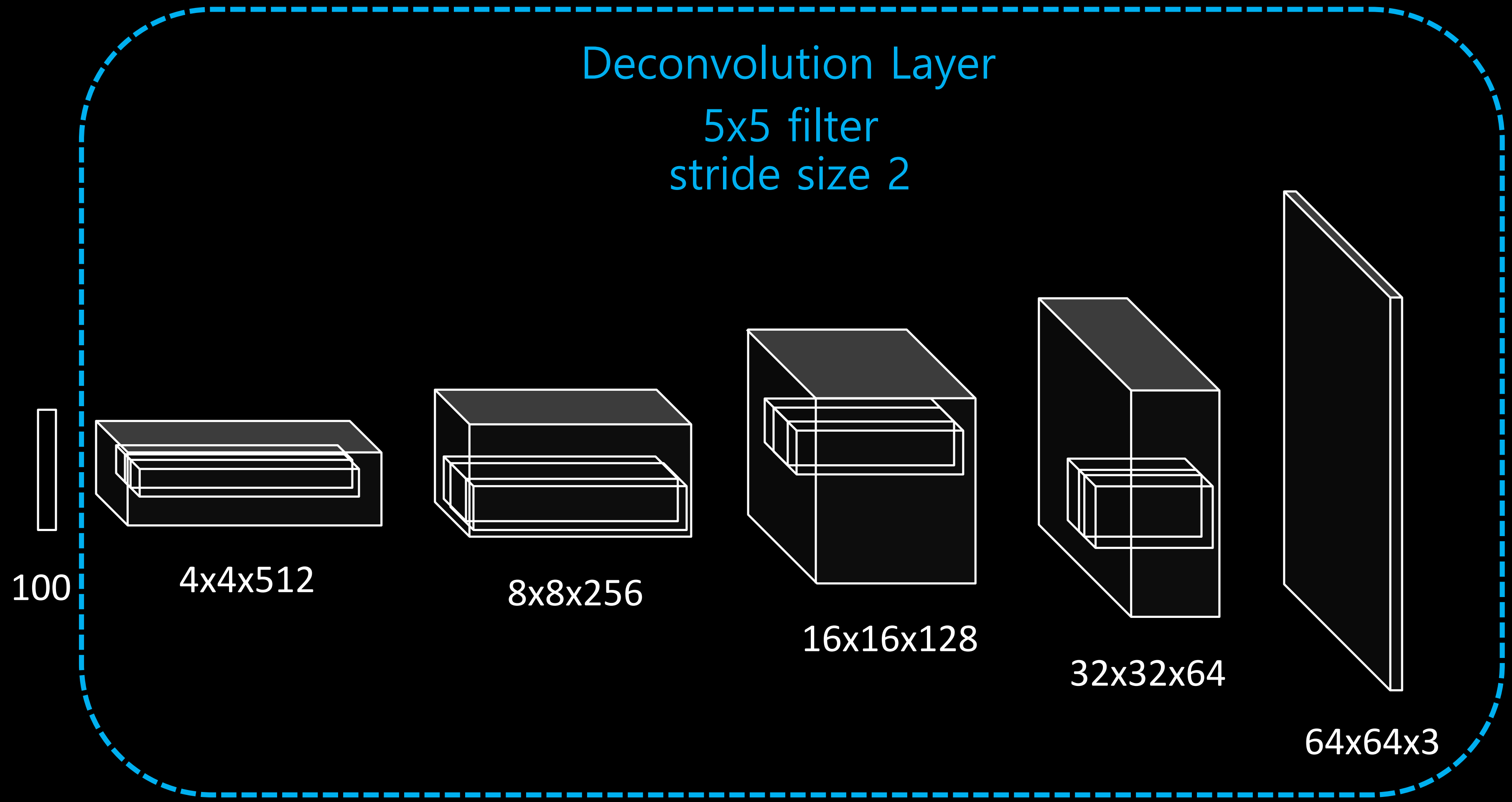
Generator(생성자)



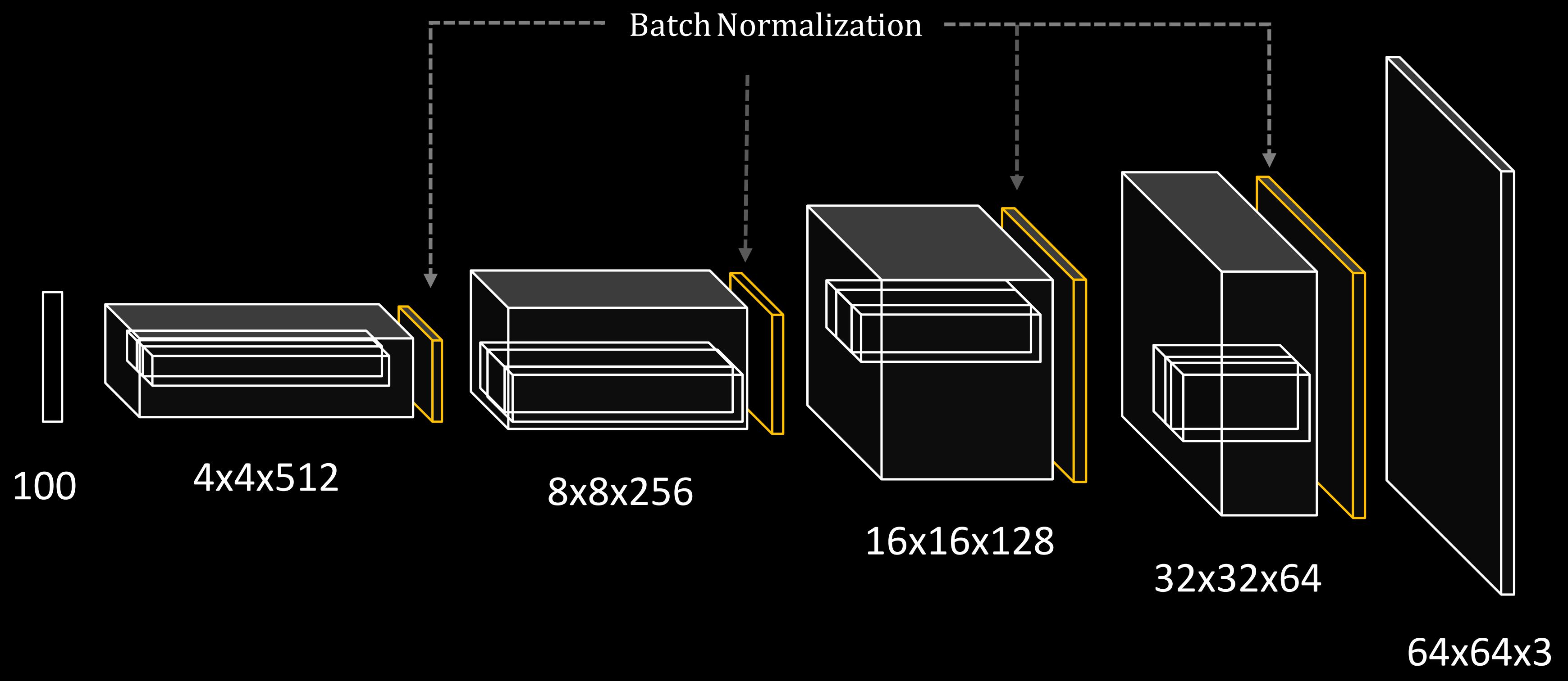
Project 100 to 8192

Reshape 8192 to 4x4x512

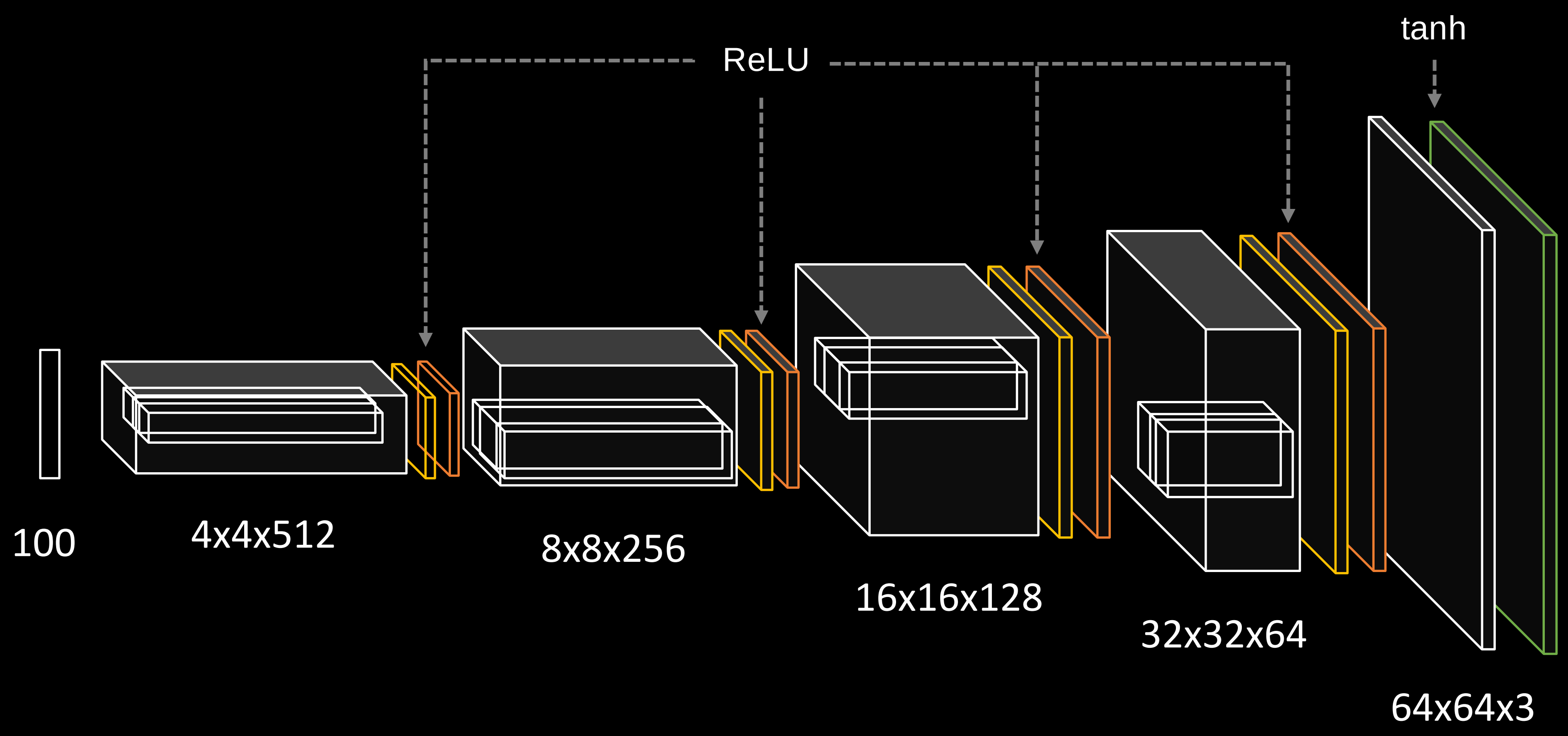
Generator(생성자)



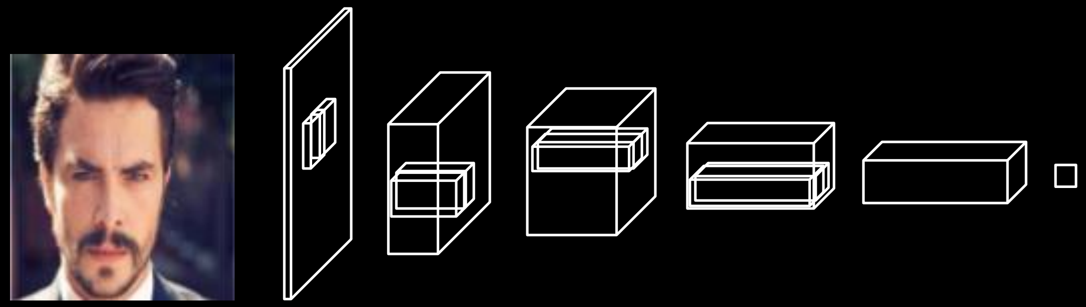
Generator(생성자)



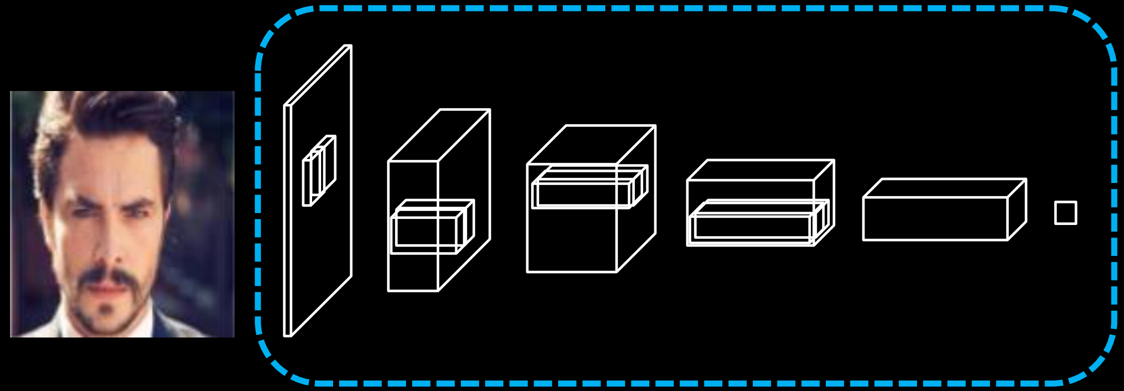
Generator(생성자)



진짜 이미지

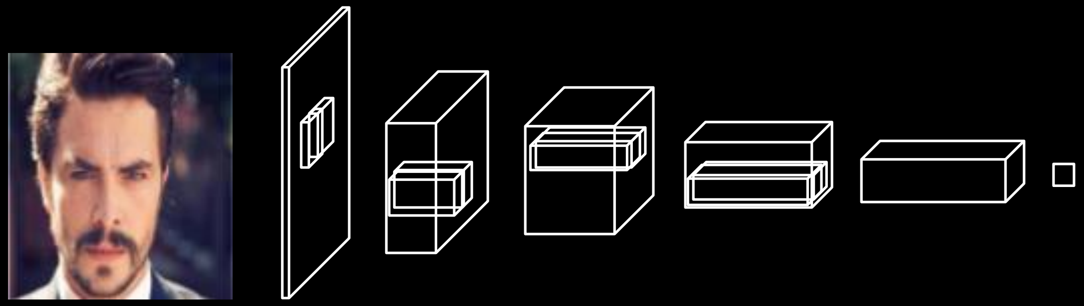


진짜 이미지

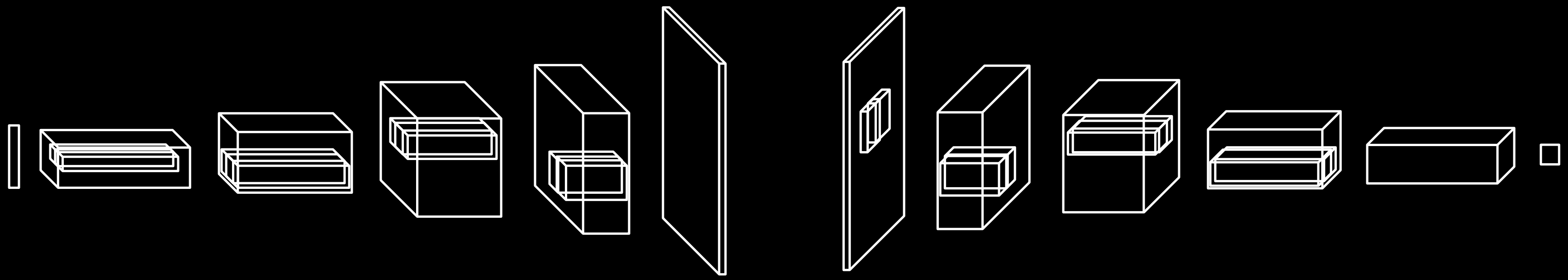


1이 나오도록
식별자만 학습

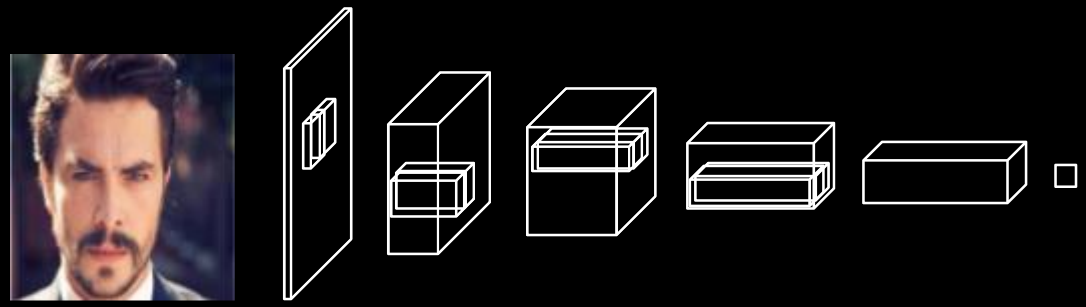
진짜 이미지



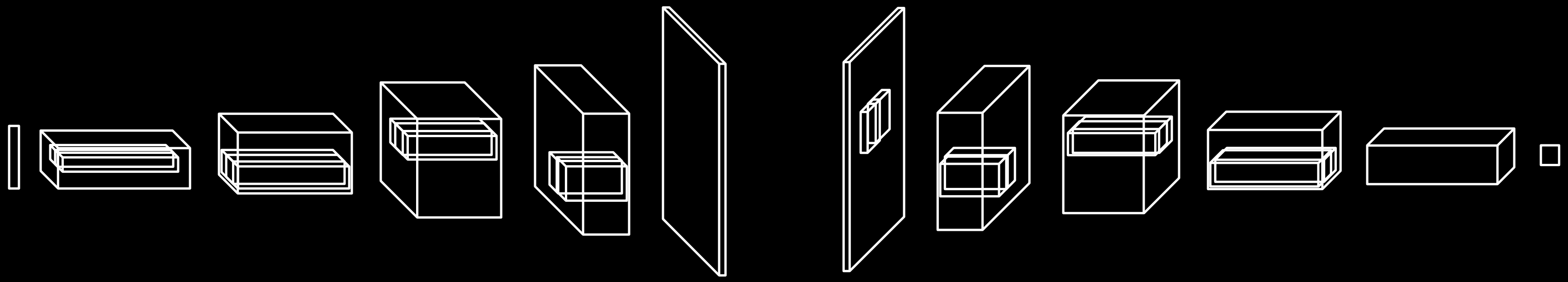
가짜 이미지



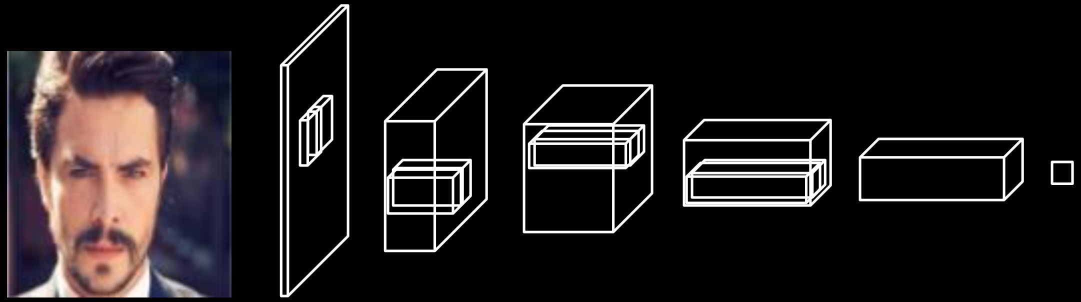
진짜 이미지



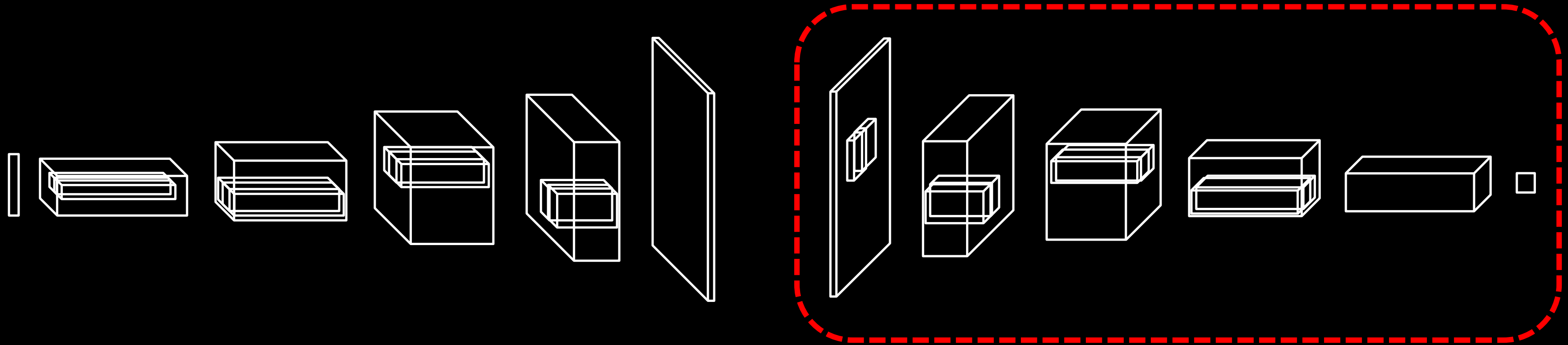
가짜 이미지



진짜 이미지

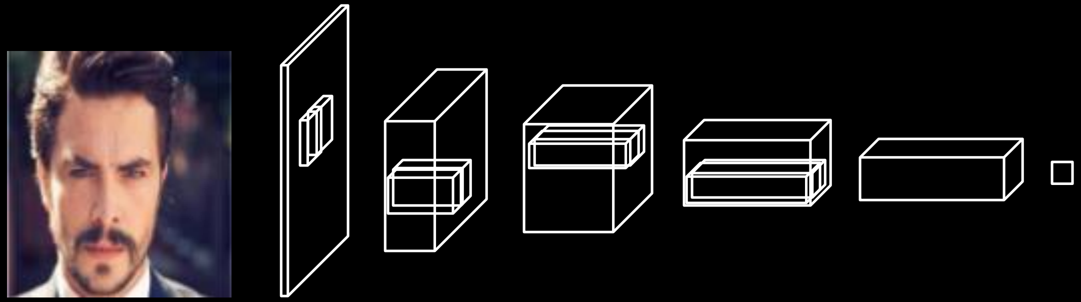


가짜 이미지

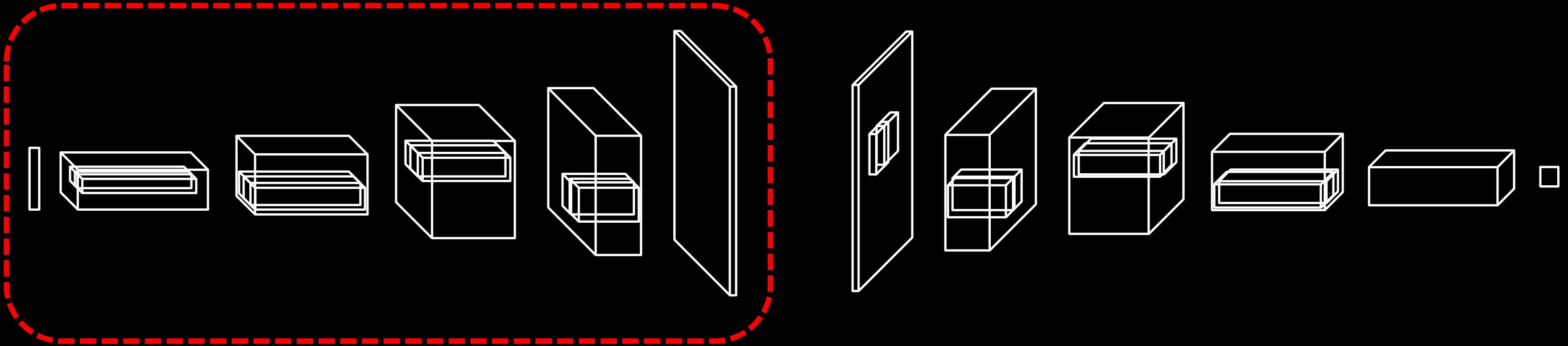


0이 나오도록
식별자 학습

진짜 이미지

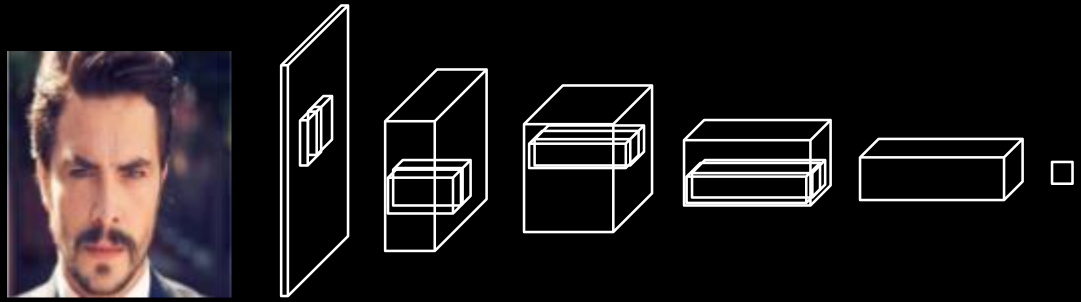


가짜 이미지

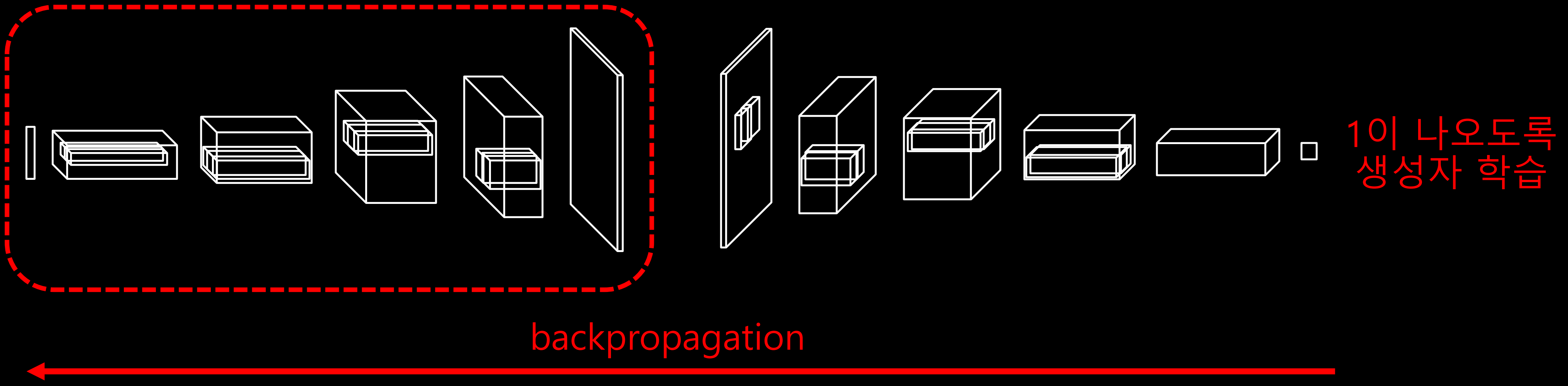


1이 나오도록
생성자 학습

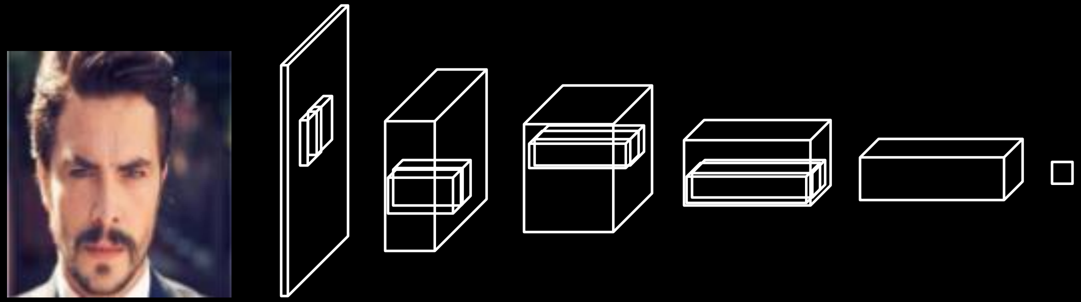
진짜 이미지



가짜 이미지

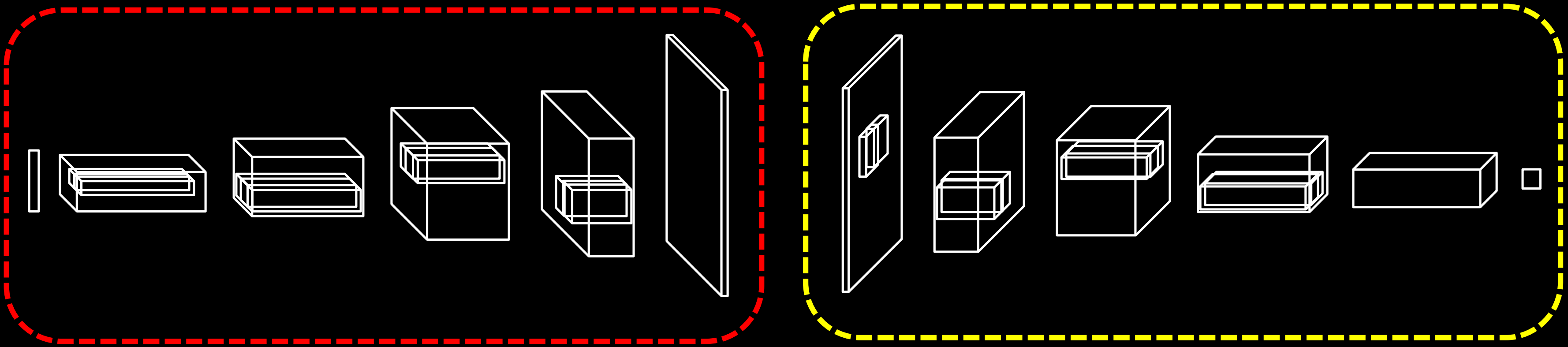


진짜 이미지



가짜 이미지

이쪽은 backprop만 되고
weight 학습 x

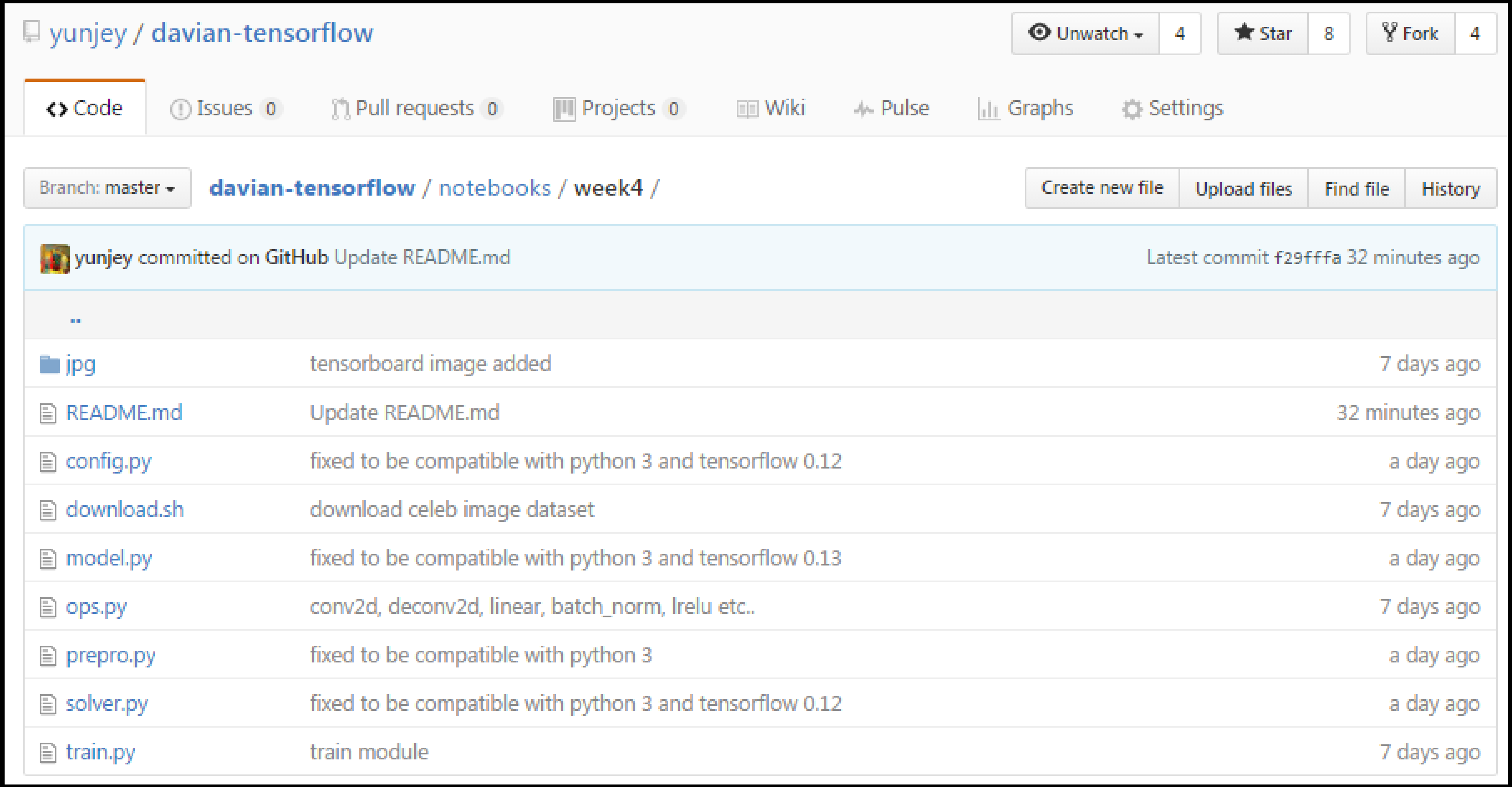


1이 나오도록
생성자 학습

backpropagation

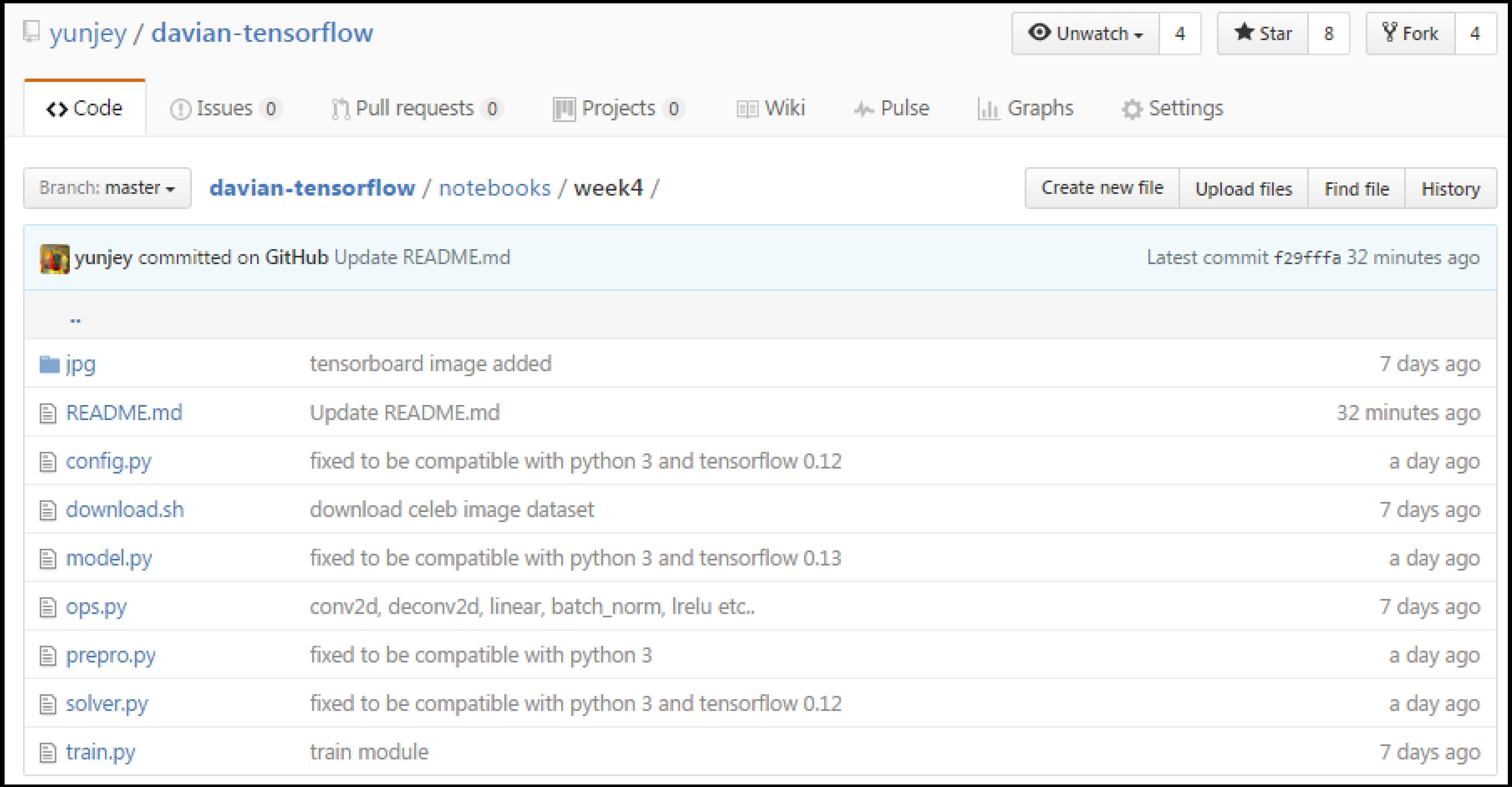
4. TensorFLow Implementation

TensorFlow Implementation



https://github.com/yunjey/davian-tensorflow/tree/master/notebooks/week4

TensorFlow Implementation



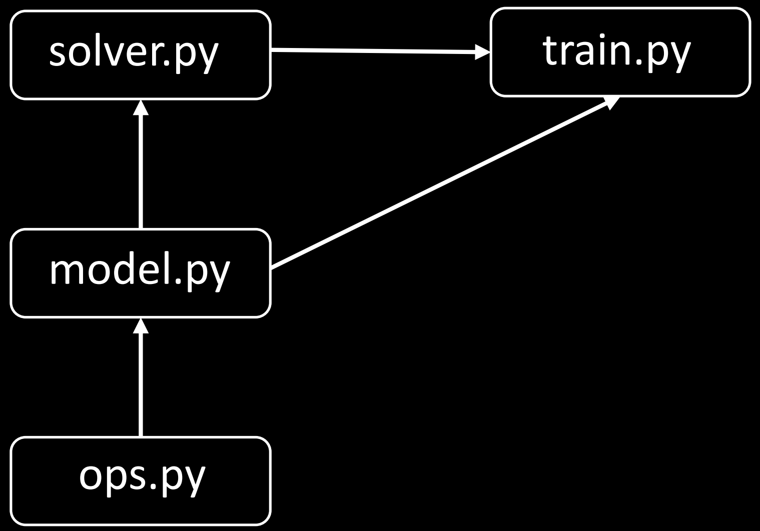
먼저 코드 구조부터 파악해보자

TensorFlow Implementation

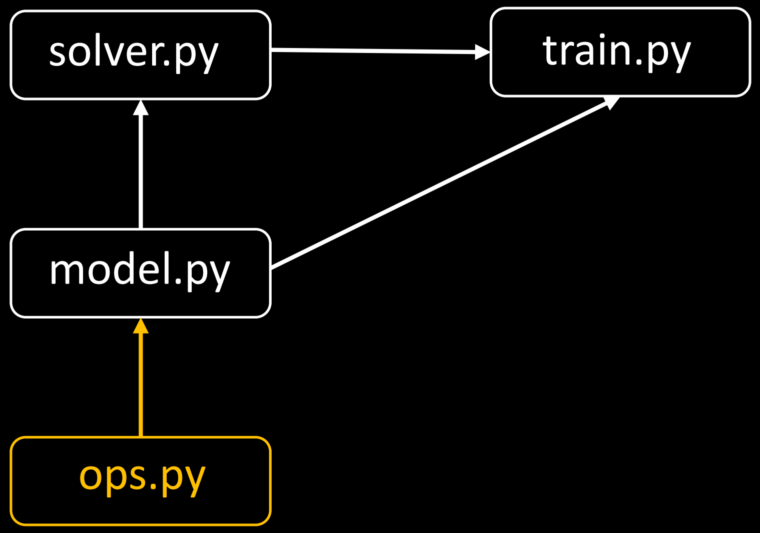
아래 3개가 핵심 코드!

config.py	fixed to be compatible with python 3 and tensorflow 0.12	a day ago
download.sh	download celeb image dataset	7 days ago
model.py	fixed to be compatible with python 3 and tensorflow 0.13	a day ago
ops.py	conv2d, deconv2d, linear, batch_norm, lrelu etc..	7 days ago
prepro.py	fixed to be compatible with python 3	a day ago
solver.py	fixed to be compatible with python 3 and tensorflow 0.12	a day ago
train.py	train module	7 days ago

TensorFlow Implementation



TensorFlow Implementation

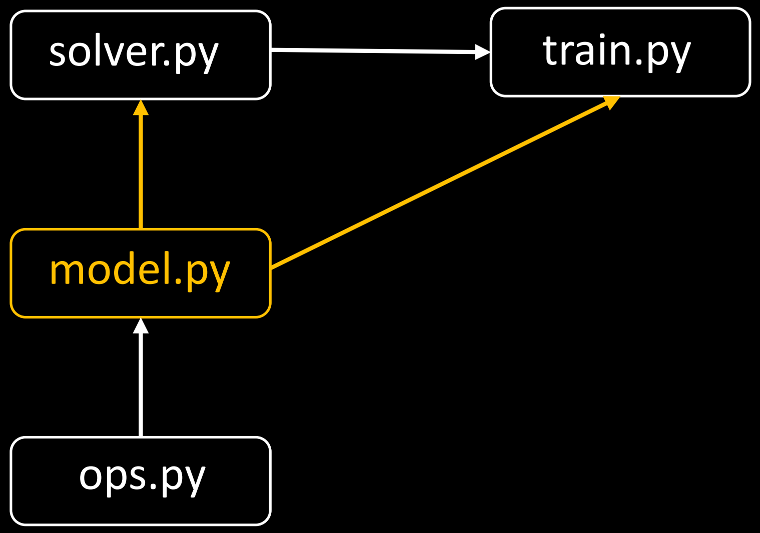


convolution, deconvolution과 같은
연산자들을 정의

TensorFlow Implementation

DCGAN 모델을 정의
discriminator와 generator 정의

build_model 함수호출 시
TensorFlow Graph를 생성

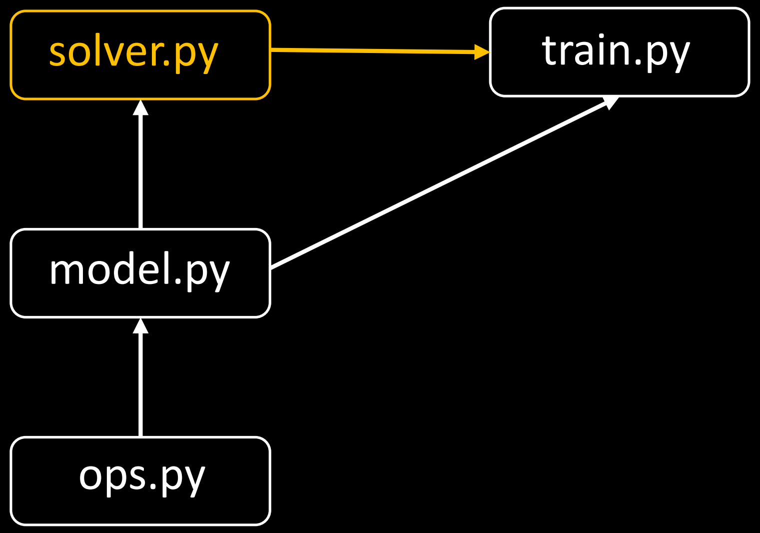


TensorFlow Implementation

학습 데이터를 불러온 뒤 DCGAN 모델을 학습

__init__ 함수에서 build_model 호출

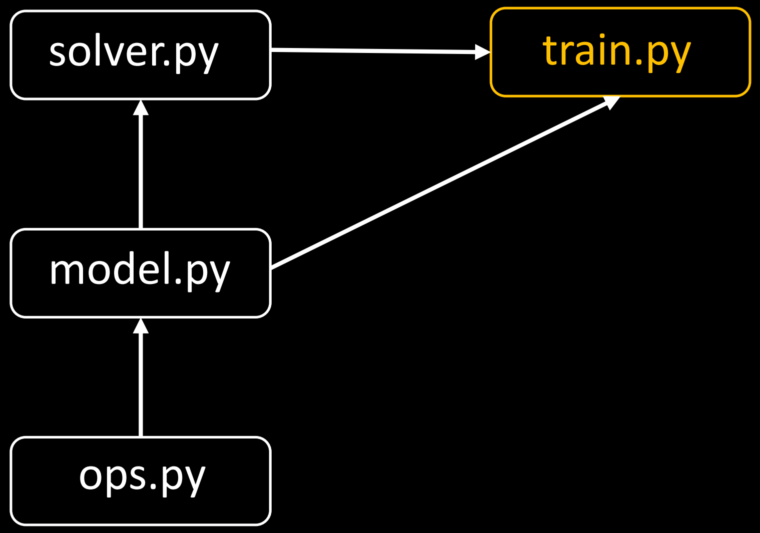
train 함수 호출 시 Session이 실행되고 학습시작



TensorFlow Implementation

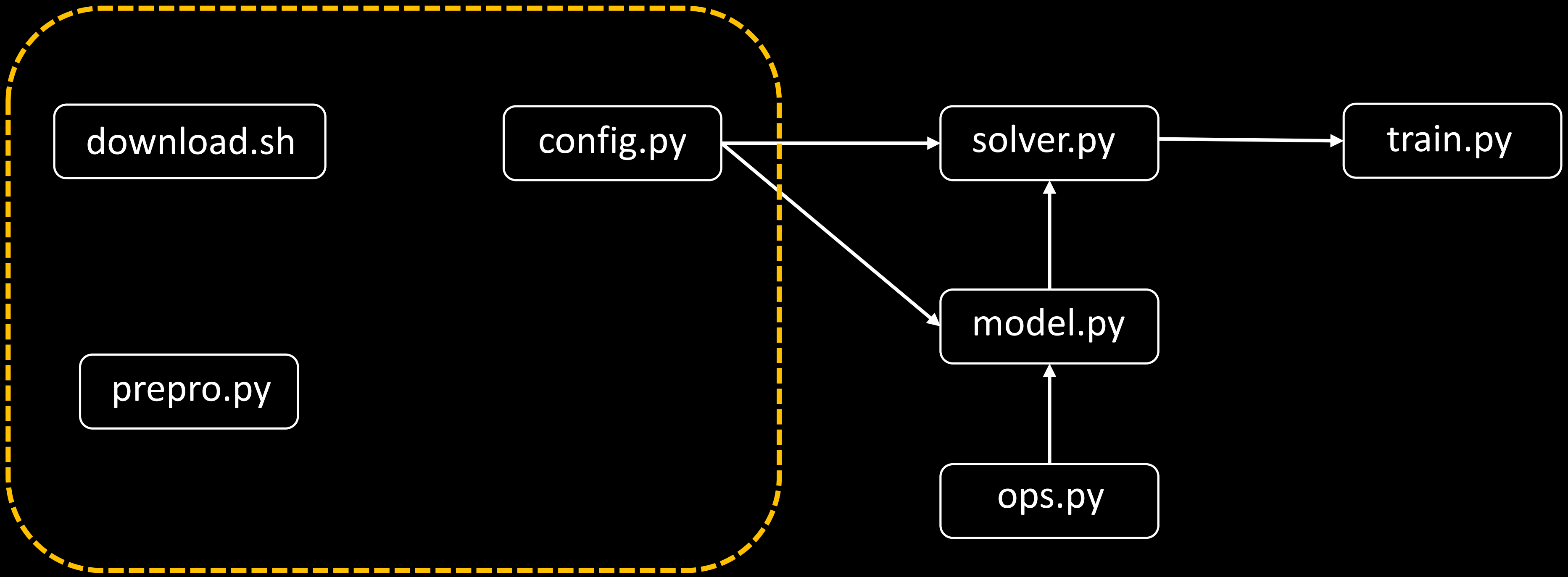
DCGAN model을 생성

Solver를 통해 DCGAN 모델을 학습



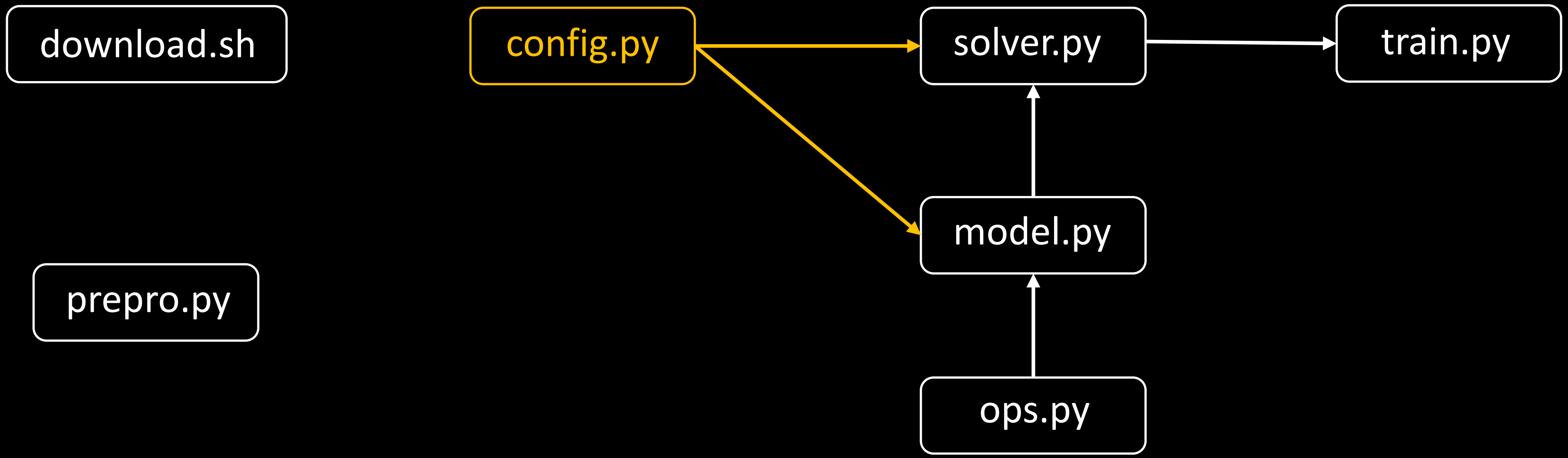
TensorFlow Implementation

부수적인 역할을 해주는 module



TensorFlow Implementation

TensorFlow 0.11과 0.12 버전 모두 호환이 되게 설정

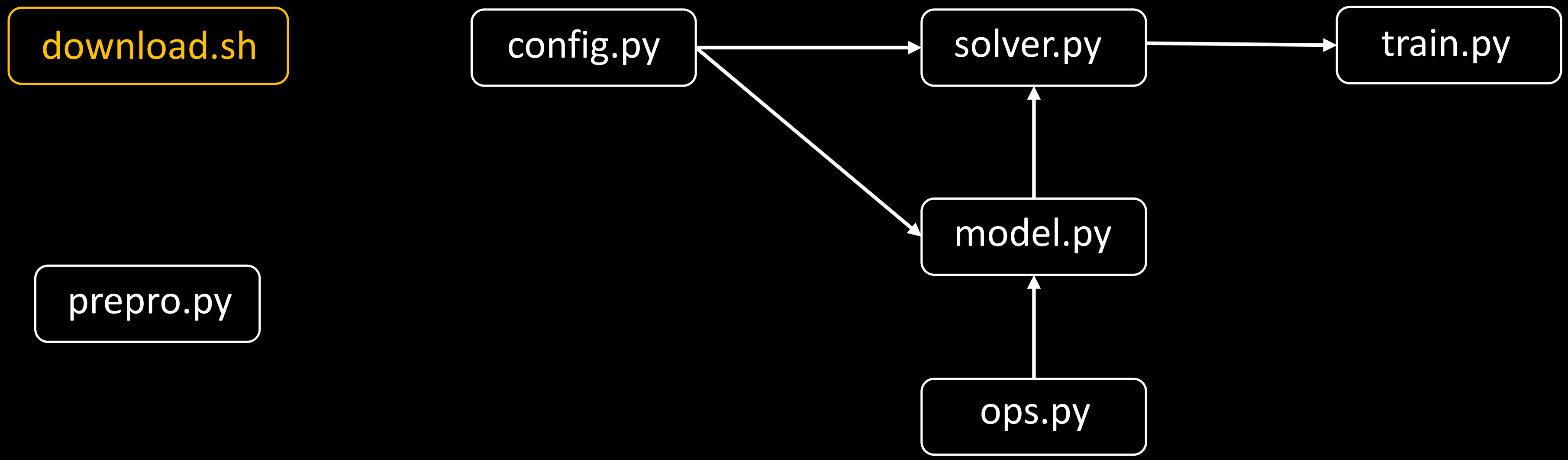


TensorFlow 0.11 -> 0.12 바뀐점

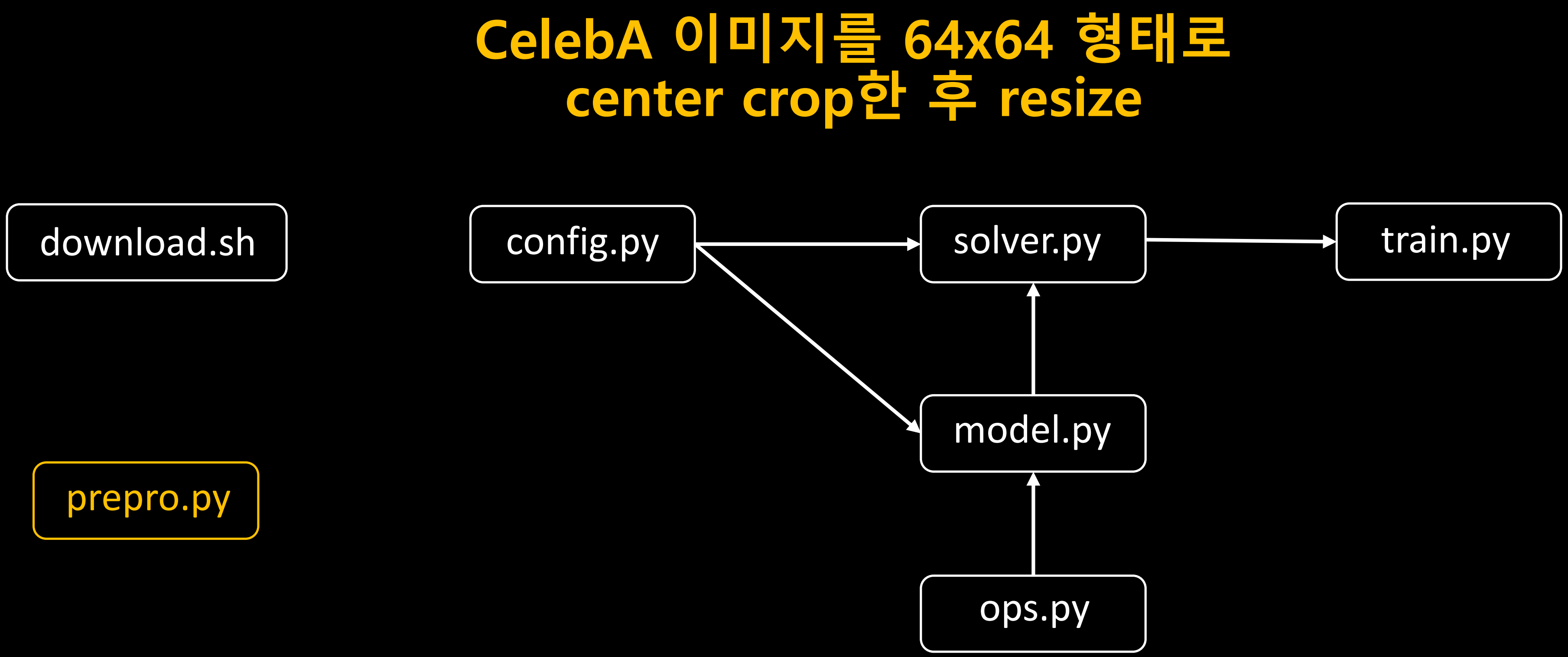
1. Variable -> Global Variable로 명칭이 바뀜
2. Tensorboard를 위한 함수들을 부르는 방식이 바뀜

TensorFlow Implementation

CelebA 이미지 데이터셋 다운로드



TensorFlow Implementation



TensorFlow Implementation

퀴즈

1. ops.py에서 batch_norm만 class로 구현한 이유는?

2. model.py에서 sampled images(57번째 줄)를 생성할 때 batch_norm 함수가 moving average와 variance를 사용하는가?

(사용하지 않는 다면 사용하게끔 코드를 수정하시오)

Thank You