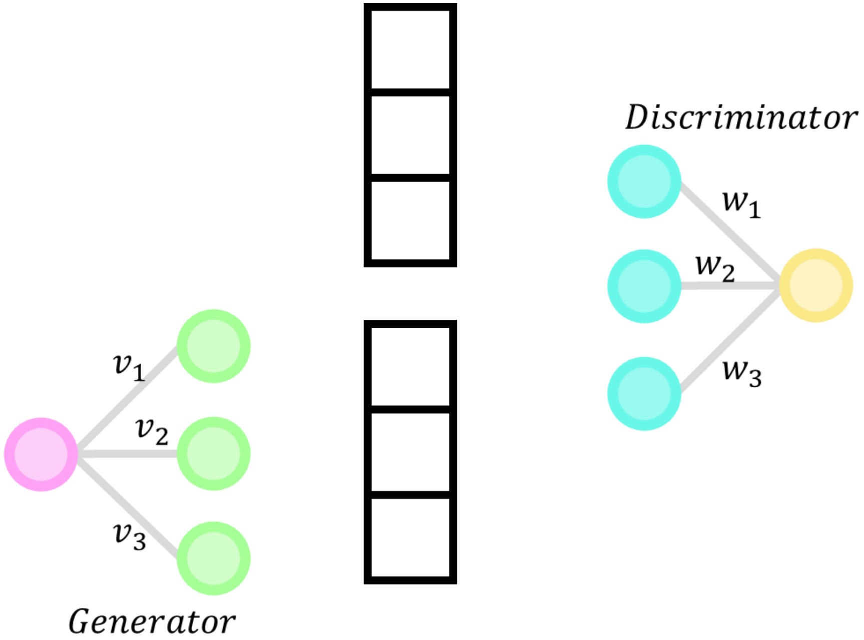
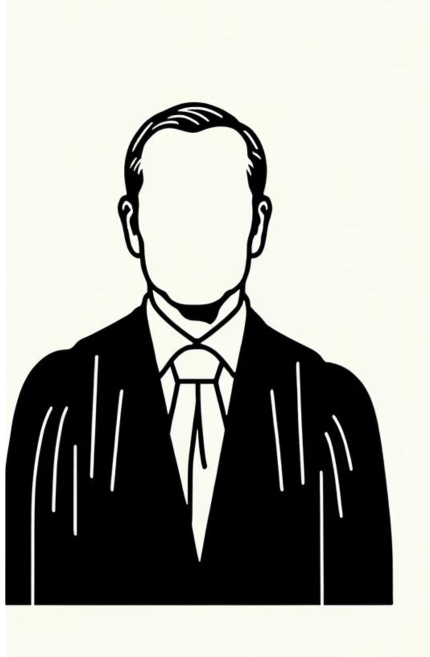
Deep Learning101

GAN

생성적 적대 신경망
Generative Adversarial Network





신박AI

안녕하세요 여러분 신박AI입니다

신박AI

이번 영상에서는 여러분께
GAN을 소개해 드리려고 합니다

Generative Adversarial Network

신박AI

GAN모델은 딥러닝의 역사에서 가장 중요한 혁신 중의 하나입니다

" GAN은 지난 10년간의 머신러닝 분야에서
가장 흥미로운 아이디어다"
Yann LeCun, Director, Facebook AI

오늘은 GAN은 무엇인지,

오늘은 GAN은 무엇인지,

모델은 어떤 구조로 생겼으며

오늘은 GAN은 무엇인지,

모델은 어떤 구조로 생겼으며

어떻게 작동하는지에 대해 알아보도록 하겠습니다.

GAN이 무엇인지 알아보기 전에

GAN이 할 수 있는 일에 대해 간략하게 소개해드리고자 합니다.

신박AI

GAN이 할 수 있는 일에 대해 간략하게 소개해드리고자 합니다.

https://thispersondoesnotexist.com/

신박AI

지금 보시는 모든 이미지들은 사실 실제 존재하는 사람들의 얼굴이 아닙니다.

https://thispersondoesnotexist.com/

신박AI

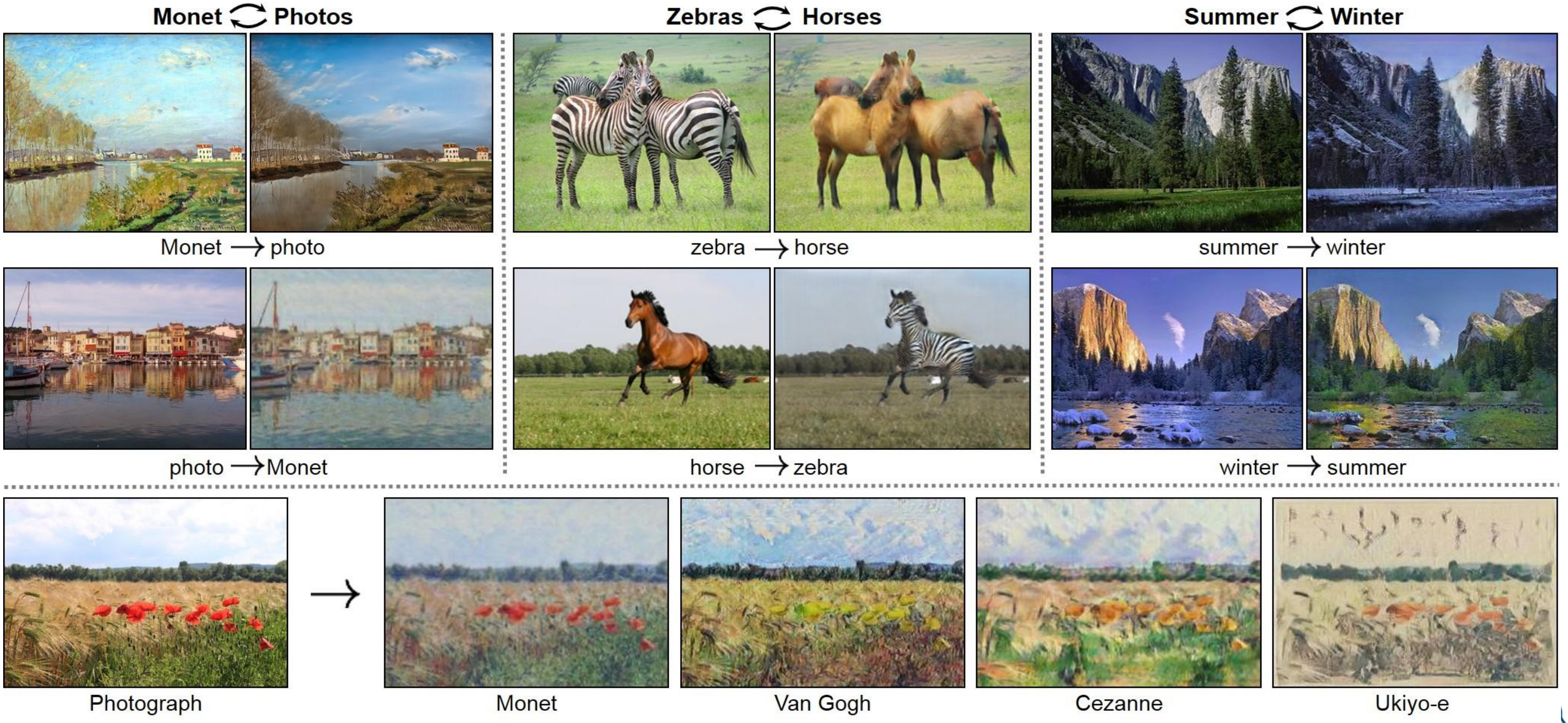
모두 다 GAN (StyleGAN2)이 만들어낸 가상의 생성 얼굴들입니다

https://thispersondoesnotexist.com/

또, GAN은 이미지의 스타일도 바꿀 수 있고,

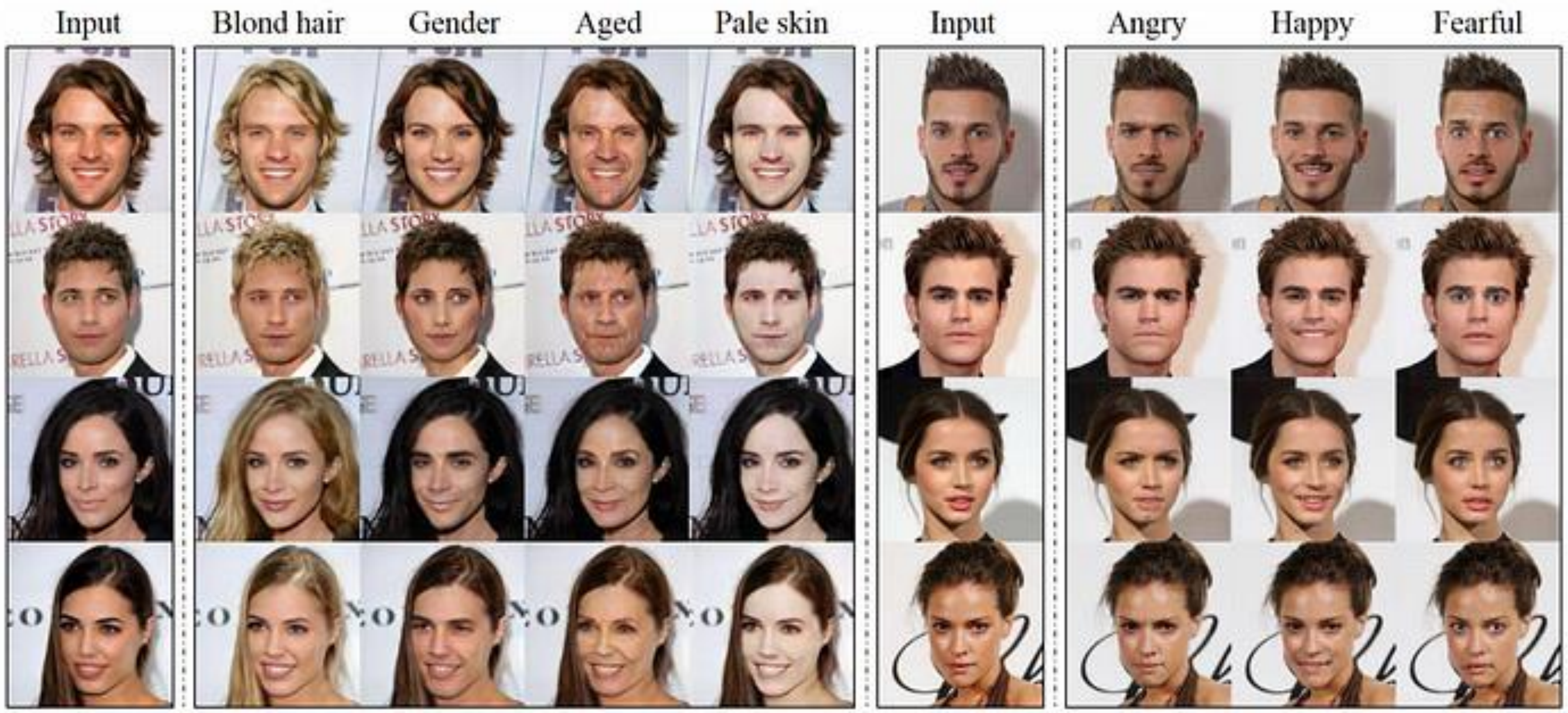
https://github.com/junyanz/CycleGAN/blob/master/imgs/horse2zebra.gif

또, GAN은 이미지의 스타일도 바꿀 수 있고,



https://github.com/junyanz/CycleGAN/tree/master

이렇게 한 이미지의 특정 특징만을 변화시킬 수도 있고,



Y. Choi, et al., "StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018 pp. 8789-8797.

저화질의 이미지나 동영상을 고화질로도 바꿀 수도 있습니다.

https://www.youtube.com/watch?v=sUhbIdSd6dc&ab_channel=SagarVinodababu

이처럼 GAN모델들은 다양한 일을 할 수 있기 때문에 GAN모델의 종류는 이와 같이 다양합니다.

StyleGAN

BigGAN

Progressive GAN

CycleGAN

StackGAN

Context Encoders

WGAN

Pix2Pix

SRGAN

AC-GAN

FCGAN

VGAN

cGAN

InfoGAN

LapGAN

DCGAN

GAN

오늘은 이런 모든 다양한 GAN모델의 아버지가 되는 기본 GAN모델을 살펴보도록 하겠습니다

StyleGAN

BigGAN

Progressive GAN

CycleGAN

StackGAN

Context Encoders

WGAN

Pix2Pix

SRGAN

AC-GAN

FCGAN

VGAN

cGAN

InfoGAN

LapGAN

DCGAN

GAN

신박AI

GAN모델은 그 이름에서 아실 수 있듯이 생성모델입니다.

Generative **A**dversarial **N**etwork

신박AI

우리가 GAN모델의 중요성을 이해하기 위해서 잠간 생성모델이 딥러닝에서 갖는 위치에 대해 먼저 알아보도록 하겠습니다.

Generative **A**dversarial **N**etwork

신박AI

딥러닝은 크게 판별모델 discriminative model과 생성모델 generative model 로 나눌 수 있습니다.

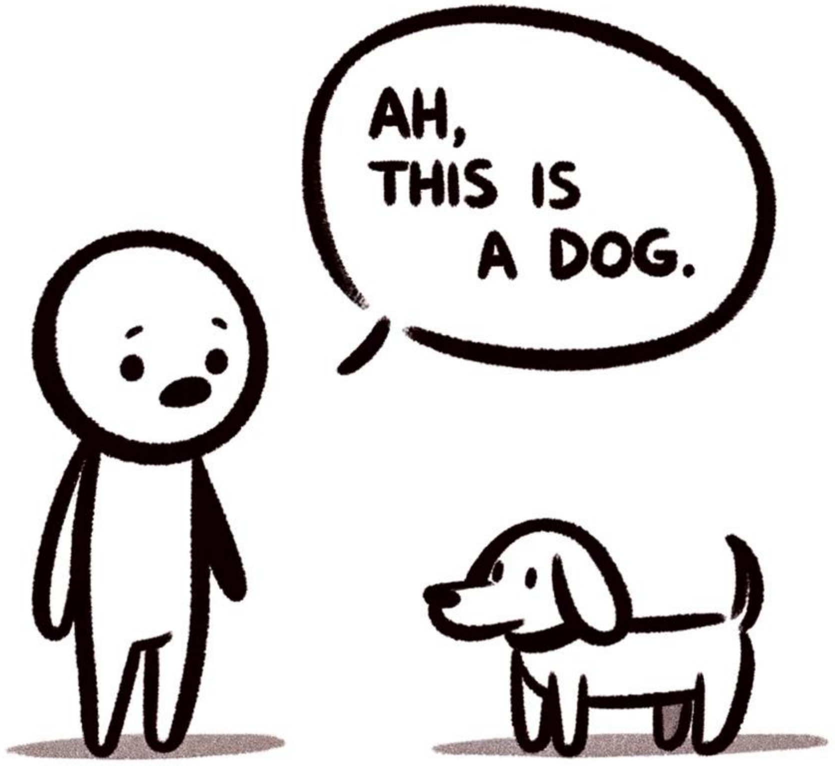
분류모델:
퍼셉트론, CNN, 로지스틱 회귀분석 모델 등등..

생성모델:
GAN, Transformer 등등..

인공지능이 인간의 지능을 모방한다는 관점에서 볼 때, 인간의 지능은 상당히 다양한 작업들을 수행할 수 있습니다.

신박AI

예를들면, 우리 인간은 학습을 통해 ‘개’라는 범주가 머리속에 그려지면, 생전 처음 보는 개라 할지라도, 이것을 ‘개’라고 범주화 할 수 있습니다.



신박AI

뿐만 아니라, 추상적 개념도 범주화 할 수 있는데요, 예를들면 우리가 게임이라는 단어를 들으면, 컴퓨터나 스맛폰에서 하는 게임을 떠올리기도 하지만,

Photo by Sam Pak on Unsplash

신박AI

실내에서 할 수 있는 보드게임, 카드게임도 게임이며,

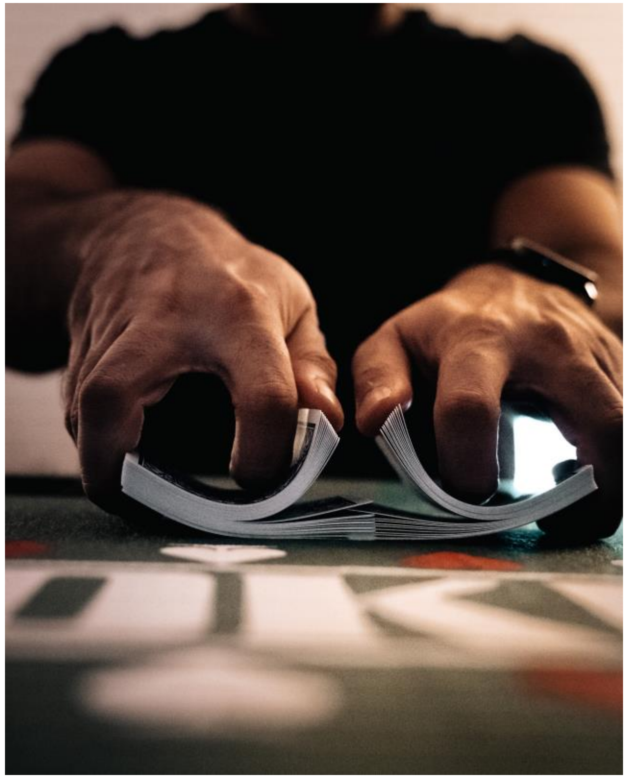


Photo by Amol Tyagi on Unsplash

신박AI

Photo by Dave Photoz on Unsplash

학교나 동네에서 친구들과 하는 놀이도 게임이며,

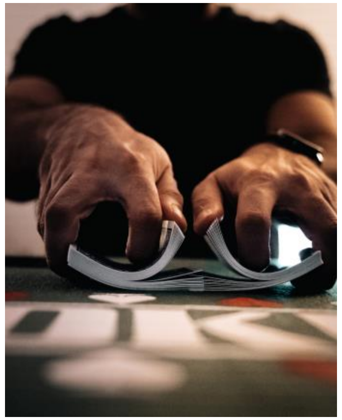
Photo by Artem Kniaz on Unsplash

올림픽이나 월드컵 같은 것도 우리는 게임이라 볼 수 있습니다.

신박AI

Photo by Vishal Butolia on Unsplash

이처럼 우리의 뇌는, 세상에 존재하는 유무형의 수많은 개념들을 특정한 기준에 따라 자유자재로 분류할 수 있는 능력이 있습니다.



신박AI

이런 관점에서 볼 때, 지능적인 모델의 중요한 특징은 주어진 분포를 얼마나 잘 분류하는지에 달려 있다고 할 수 있습니다.

분류모델:
퍼셉트론, CNN, 로지스틱 회귀분석 모델 등등..

생성모델:
GAN, Transformer 등등..

이러한 배경에서 퍼셉트론, CNN, 로지스틱 회귀분석 모델, 등등 많은 분류 모델들이 나오게 된 것입니다.

분류모델:
퍼셉트론, CNN, 로지스틱 회귀분석 모델 등등..

생성모델:
GAN, Transformer 등등..

또한 인간의 뇌는 이런 분류기능만 있는 것은 아닙니다.

분류모델:
퍼셉트론, CNN, 로지스틱 회귀분석 모델 등등..

생성모델:
GAN, Transformer 등등..

신박AI

우리 뇌의 중요한 기능 중 하나는 바로 생성기능입니다.

분류모델:
퍼셉트론, CNN, 로지스틱 회귀분석 모델 등등..

생성모델:
GAN, Transformer 등등..

우리 머리속에 ‘개’라는 이미지가 존재한다면, 우리는 다양한 형태의 ‘개’의 그림을 그려낼 수 있습니다.

신박AI

이러한 인간 지능의 생성적인 측면을 딥러닝으로 구현하려는 모델이 생성모델입니다.

신박AI

현재 딥러닝의 세계는 이런 생성모델의 전성기라 할 수 있을 정도로 많은 생성모델이 세상에 나왔습니다

Stable Diffusion

CHATGPT

Midjourney

오늘 우리가 살펴볼 GAN모델은 이러한 생성모델 시대의 포문을 연 중요한 모델로 여겨지고 있습니다

Generative **A**dversarial **N**etwork

신박AI

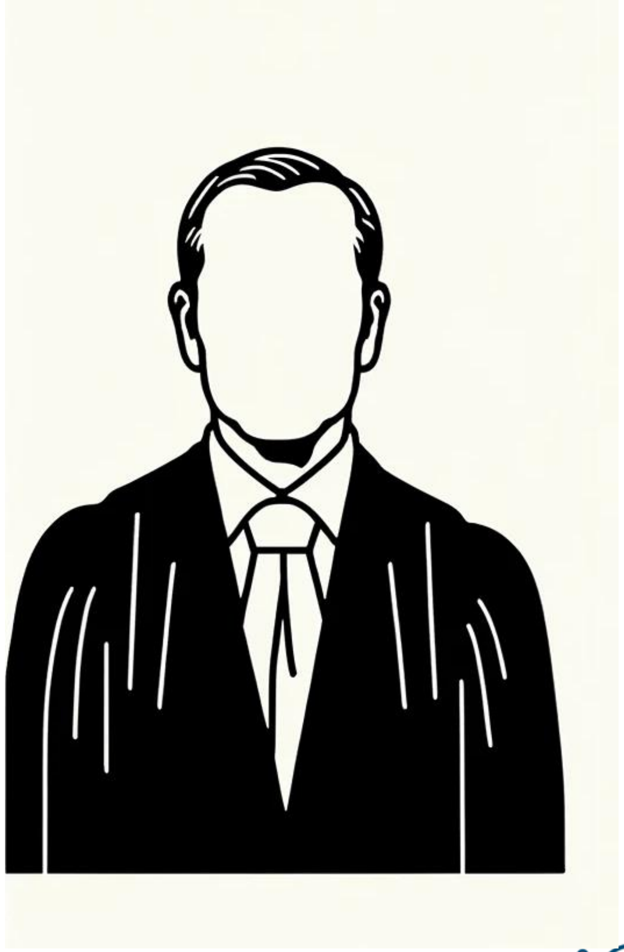
그러면 본격적으로 GAN모델에 대해 알아보도록 하겠습니다.

Generative **A**dversarial **N**etwork

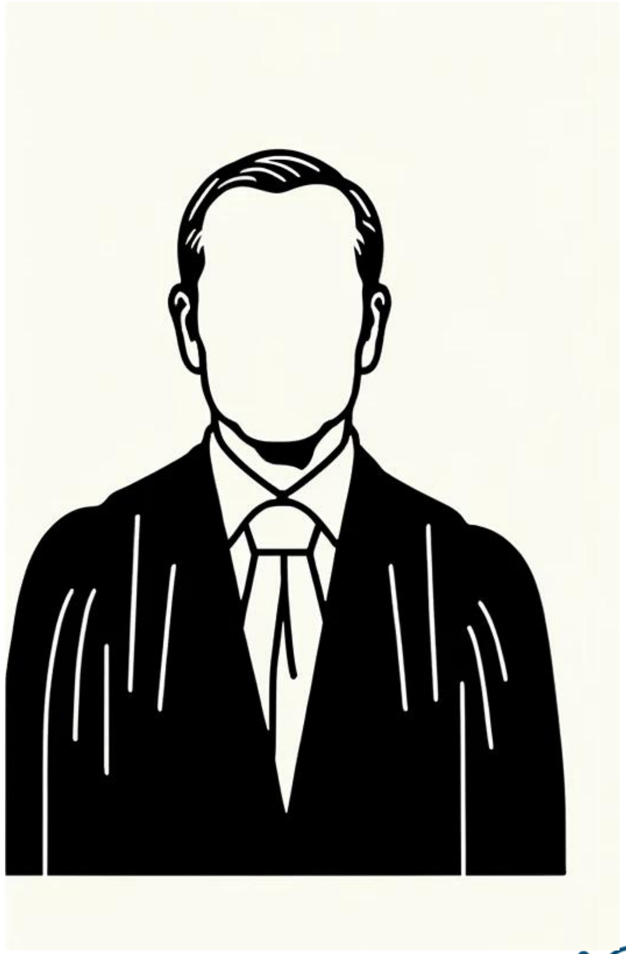
신박AI

GAN 모델이 어떻게 작동하는지 알기 위해서 전체적인 개념을 알기 위해서,

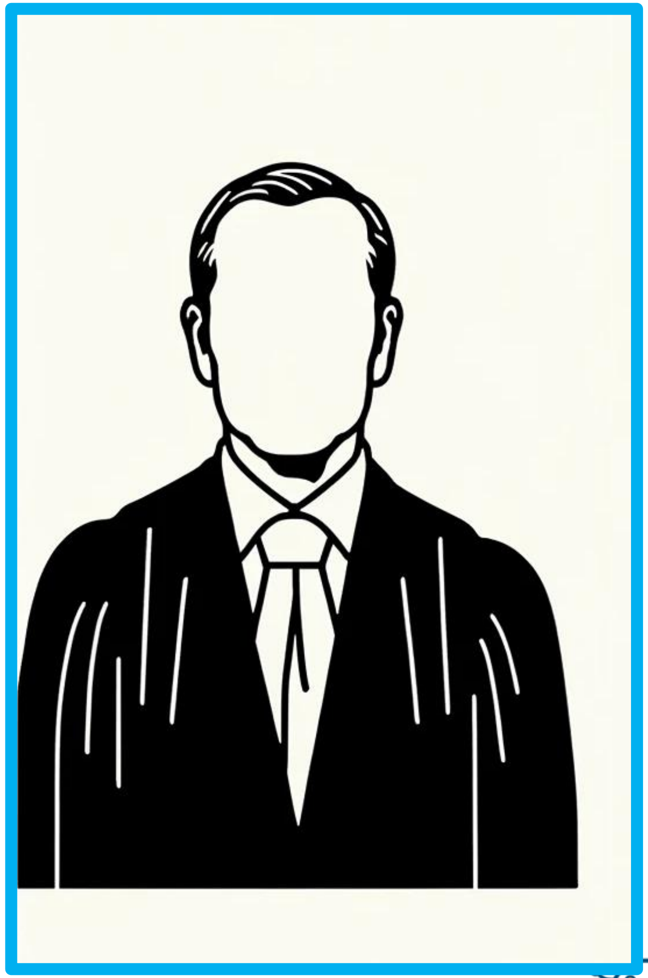
다음과 같은 두 사람의 경우로 비유를 들어보겠습니다.



한 사람은 모조작품을 만드는 위조작가이고,

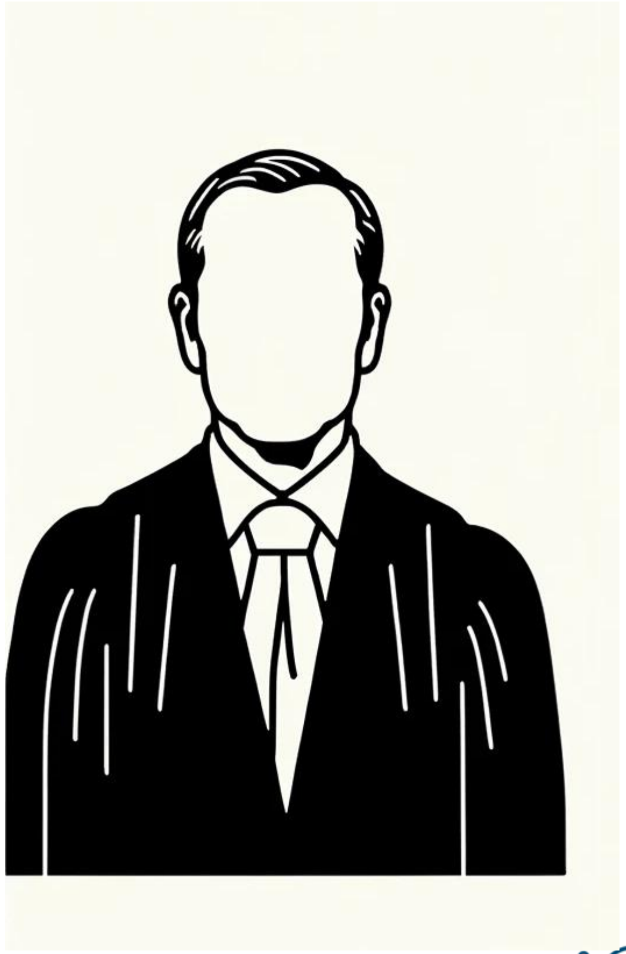


또 한 사람은 모조작품을 판별해 내는 감별사라고 가정해 봅시다.



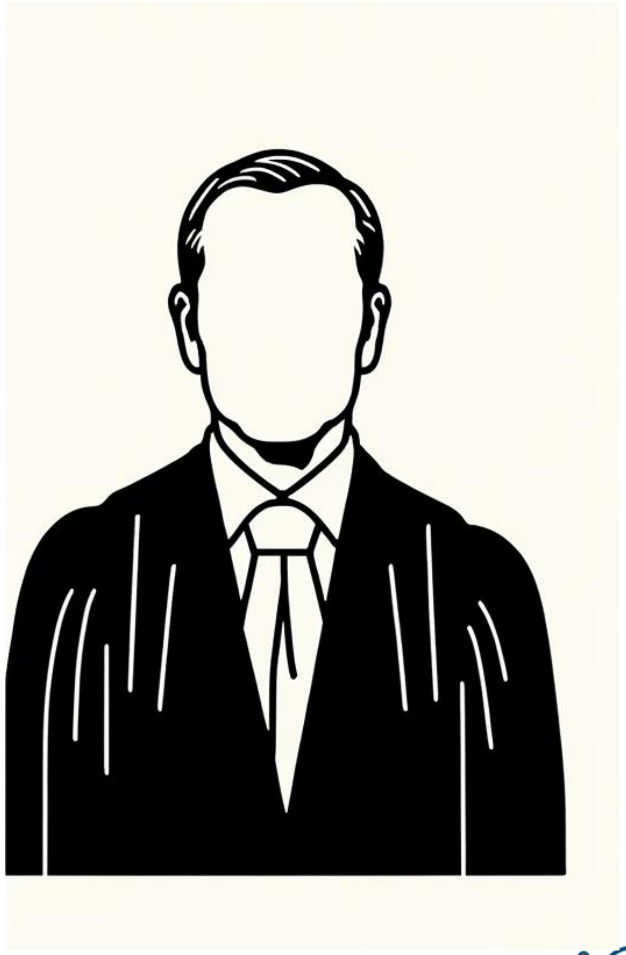
신박AI

위조작가가 해야하는 일은, 어떻게든지 그림을 그려내는데 진품과 유사한 그림을 그려내는 일입니다.



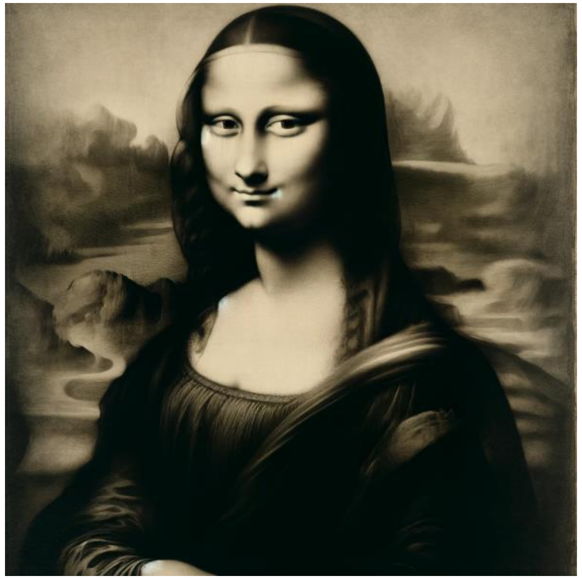
신박AI

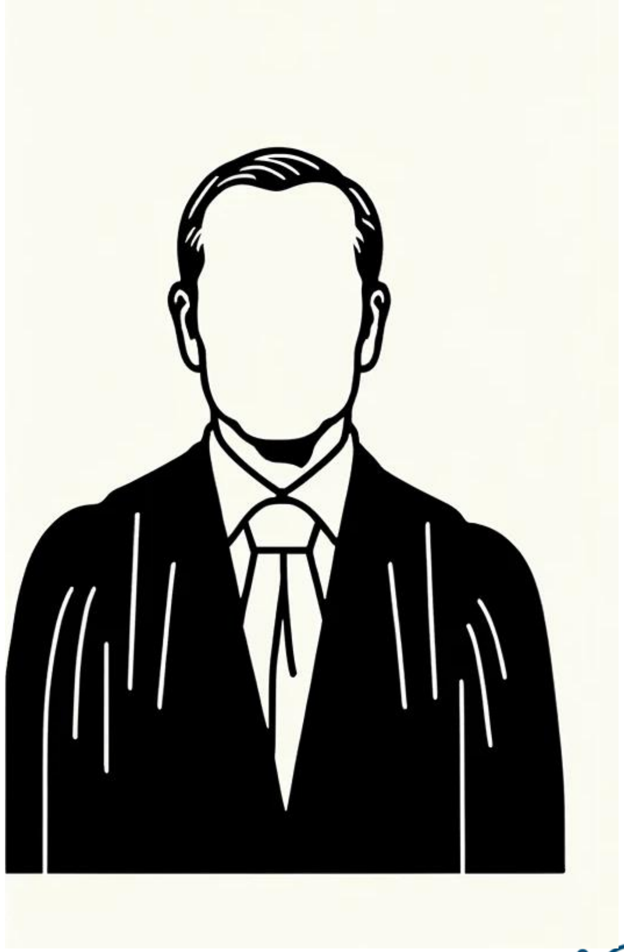
아마 위조작가도 처음에는 누가봐도 위작이라고 말할 정도로 형편없이 그리겠지만,



신박AI

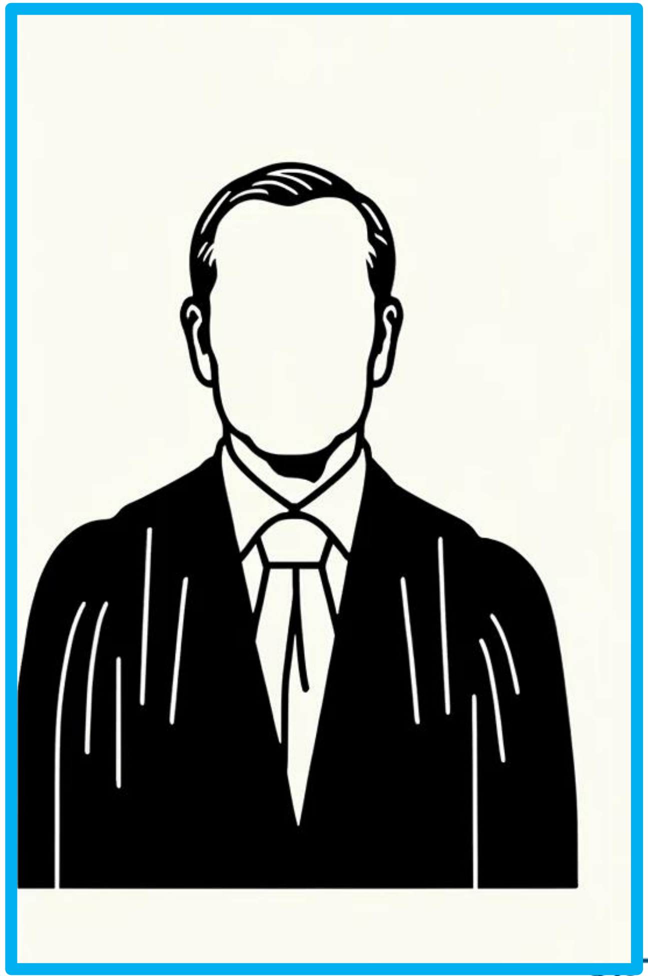
계속되는 연습을 통해서 위작과 진품의 차이가 거의 없도록 최대한 노력할 것입니다.





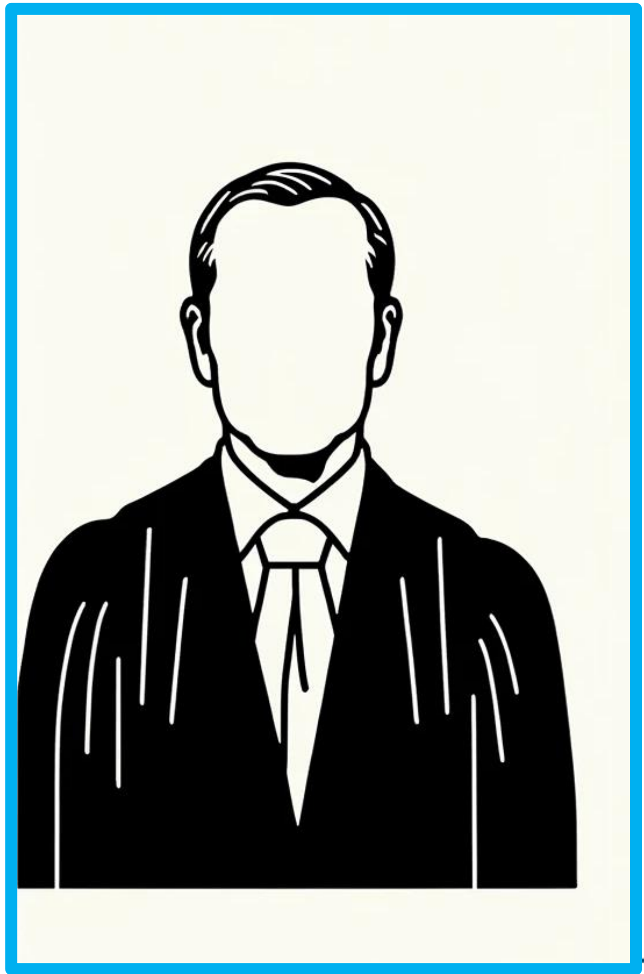
신박AI

그에 반해 감별사가 해야하는 일은, 어떻게든지, 주어진 그림이 진품인지 모조품인지 구별해내야 합니다.



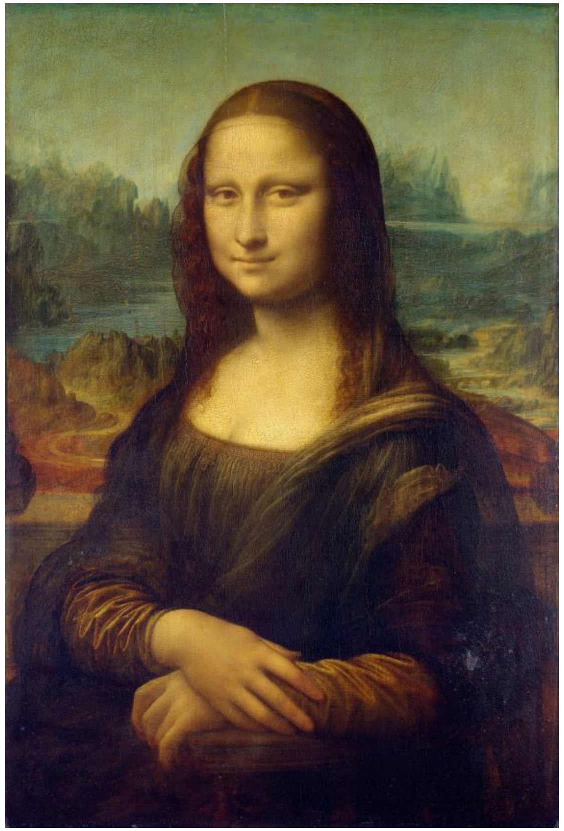
감별사도 처음에 일을 시작할 때는, 진품을 잘 구별해내지 못할 것입니다.

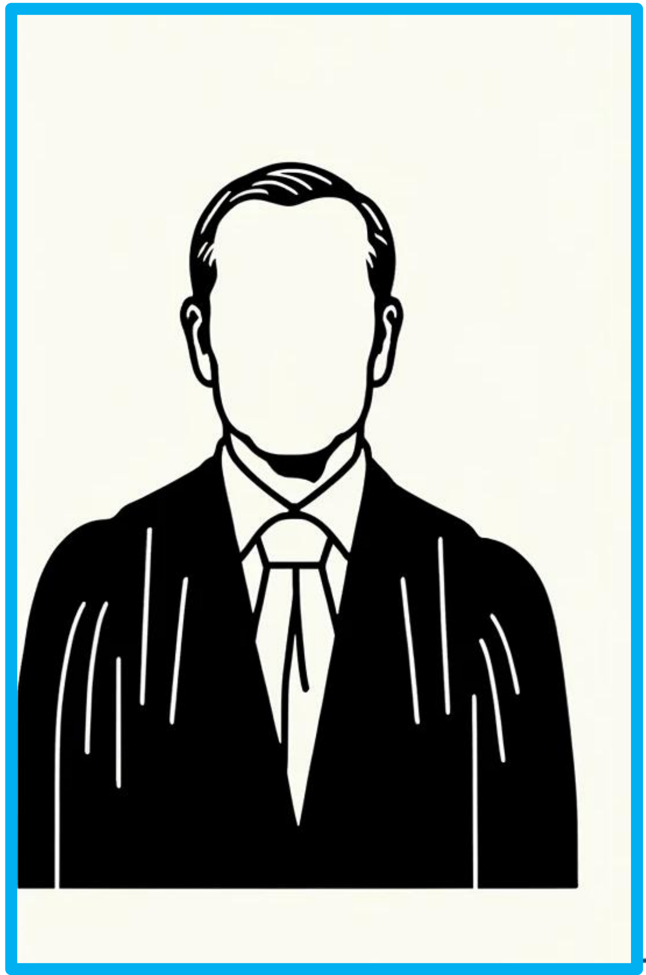
진품명품일세..



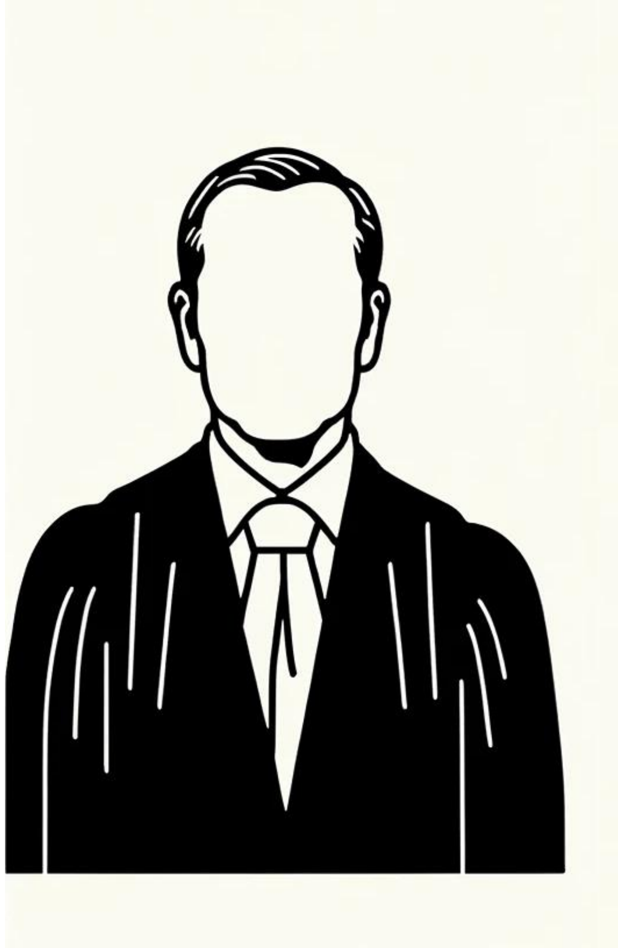
신박AI

그러나 계속 연습을 통해 진품과 모조품의 미세한 차이도 극명하게 구별해내는 훌륭한 감별사가 될 것입니다.

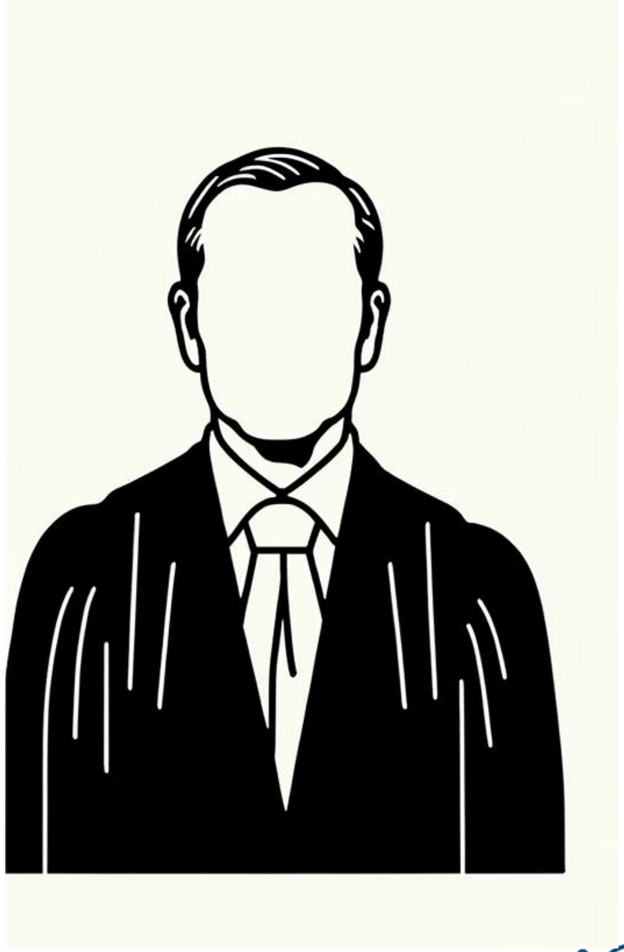




즉, 위조작가가 모조품을 정교하게 그리면 그릴수록 감별사도 더 잘 구별하도록 성장하고,

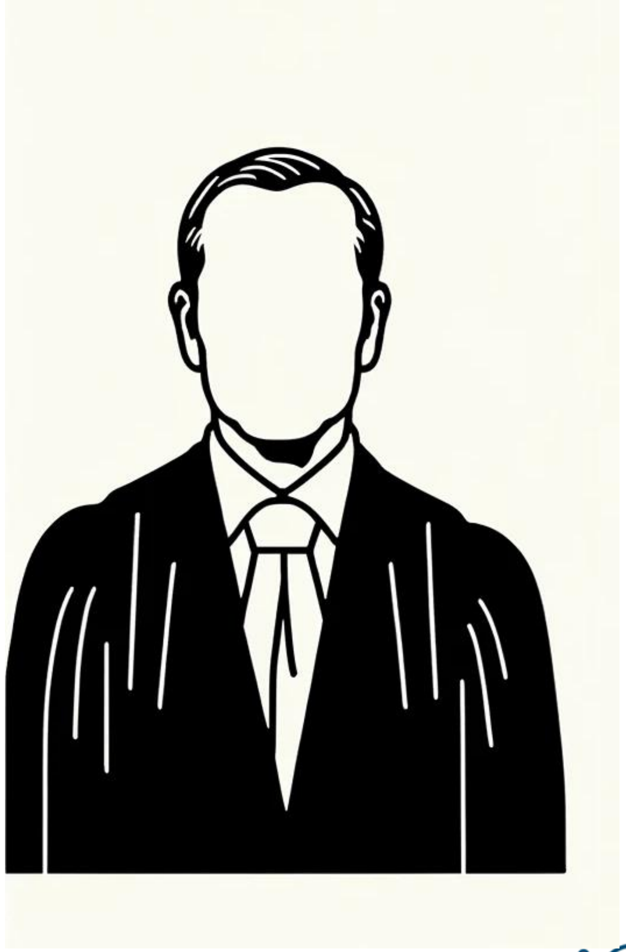


감별사의 눈이 더욱 정확 해질수록, 위조작가도 더 진품 같은 그림을 그리도록 성장합니다.



이처럼 GAN 모델도 위조작가 역할을 하는 generator가 있고,

Generator

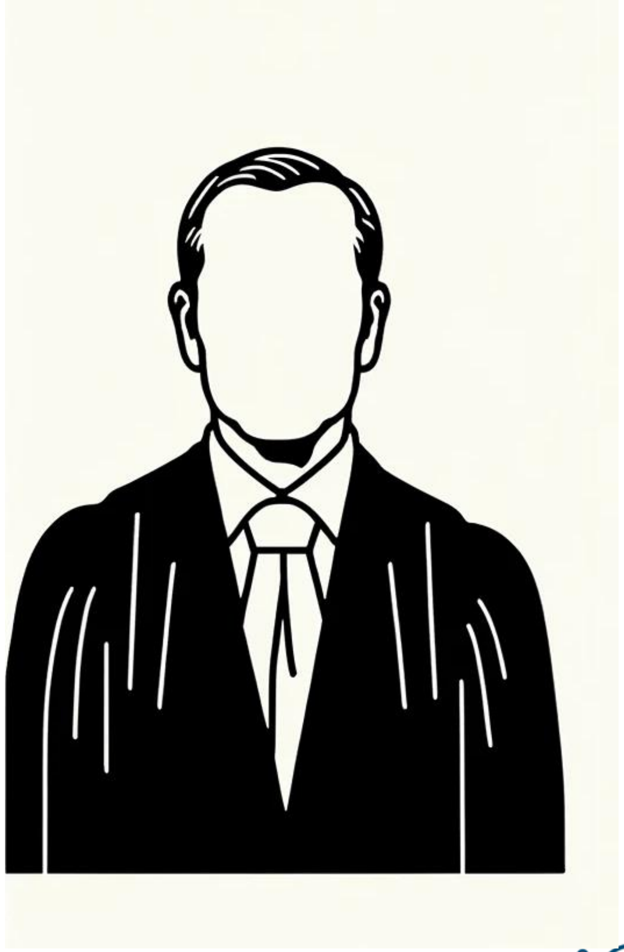


신박AI

감별사 역할을 하는 discriminator가 있어서 서로 경쟁하며 발전하는 것이 GAN모델의 개념입니다.

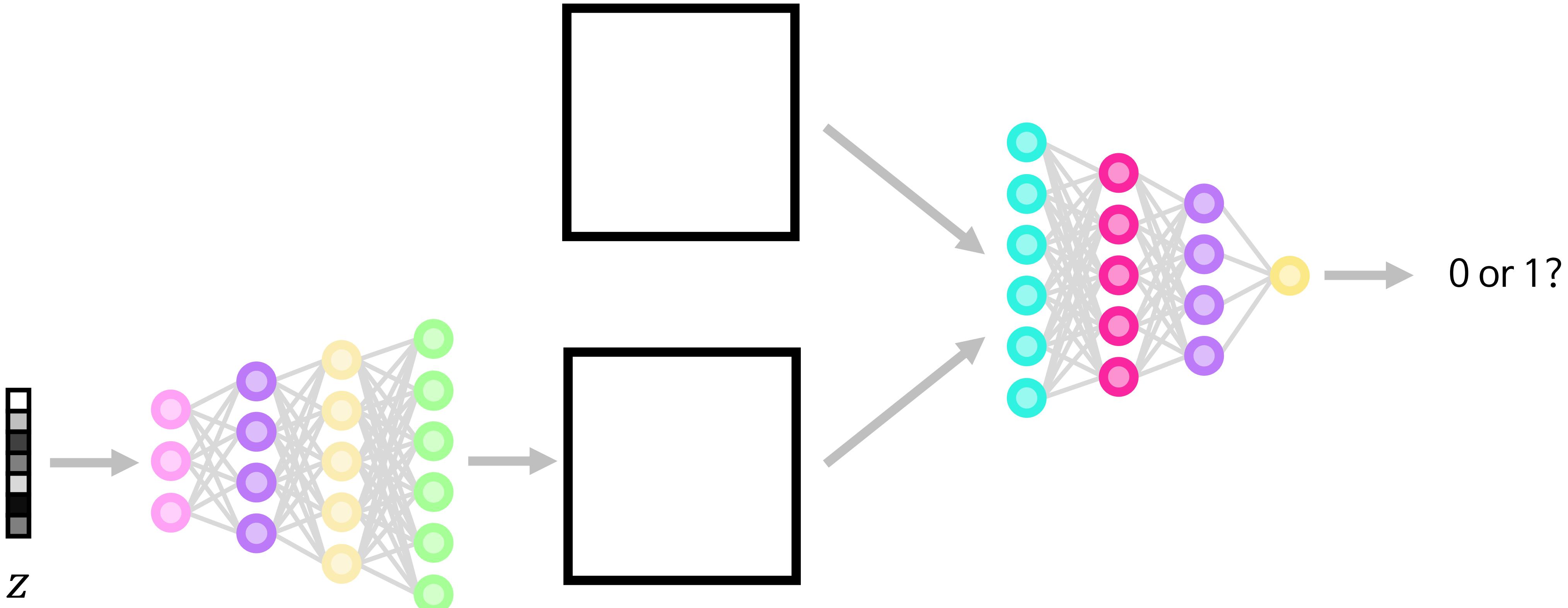
Generator

Discriminator

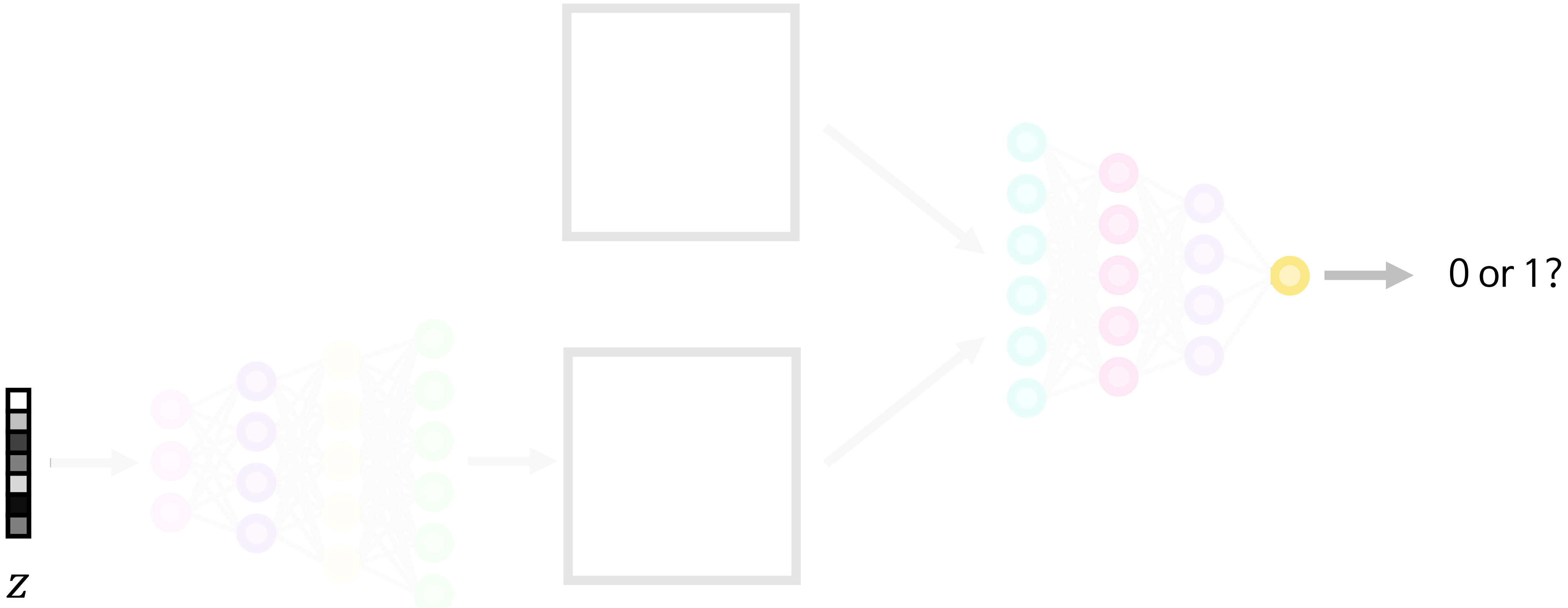


자 그러면 이제는 이러한 개념을 바탕으로 실제 신경망을 어떻게 구현했는지 구조를 살펴보도록 하겠습니다.

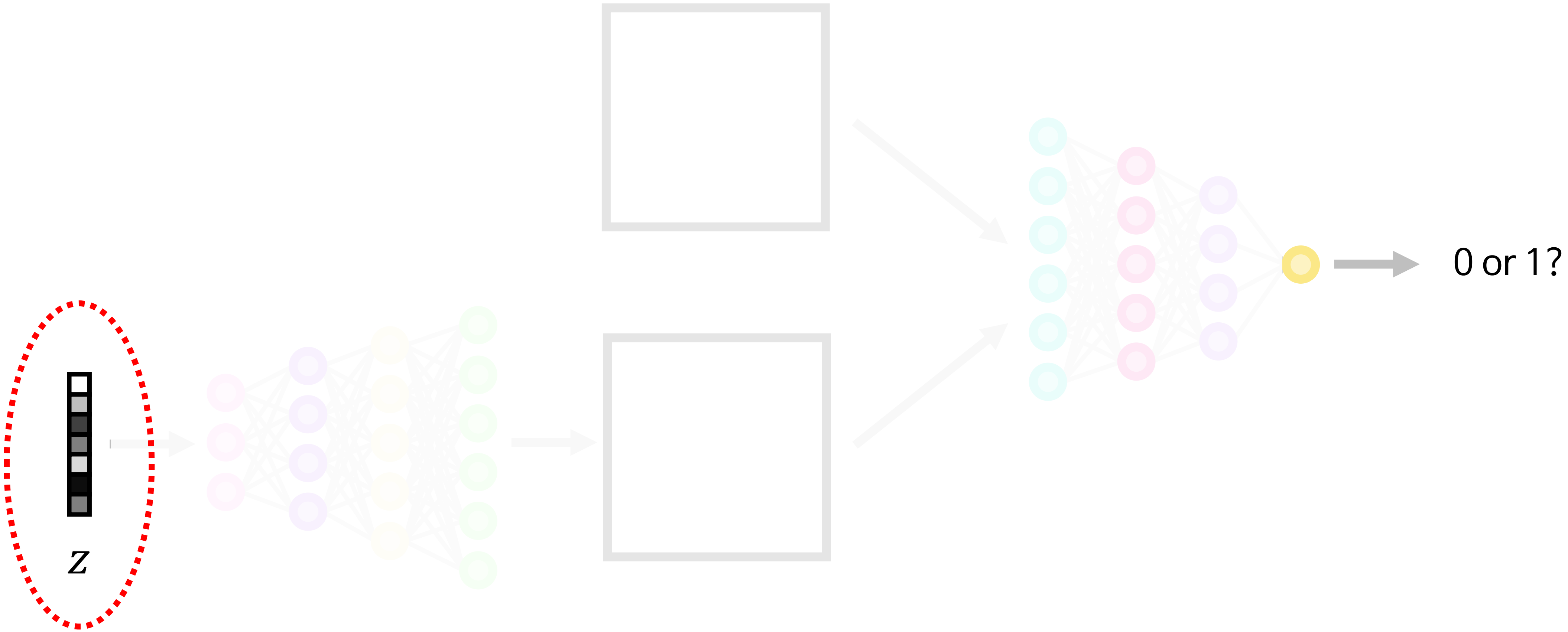
GAN모델의 구조는 다음과 같이 그려볼 수 있습니다.



이해를 돕기 위해서 먼저 입력과 출력이 무엇인지를 알아보겠습니다.

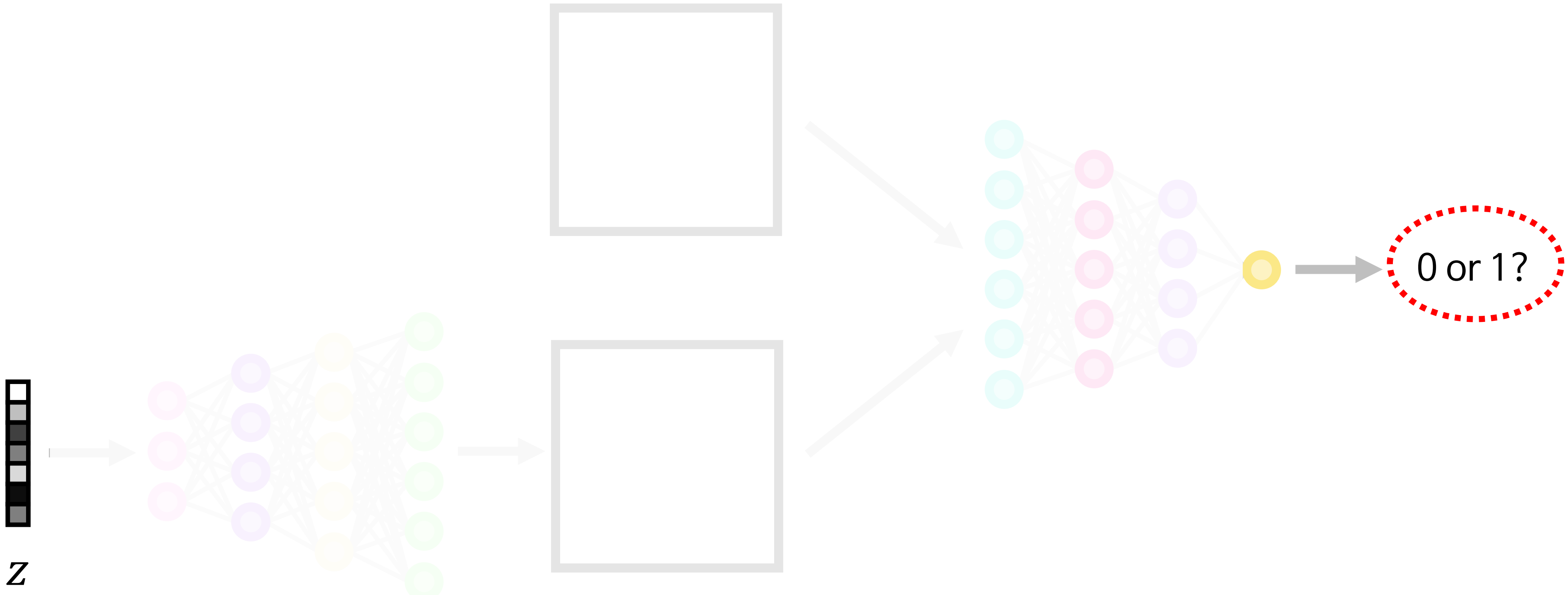


여기 보시는 것 처럼 입력은 1차원 벡터이고,

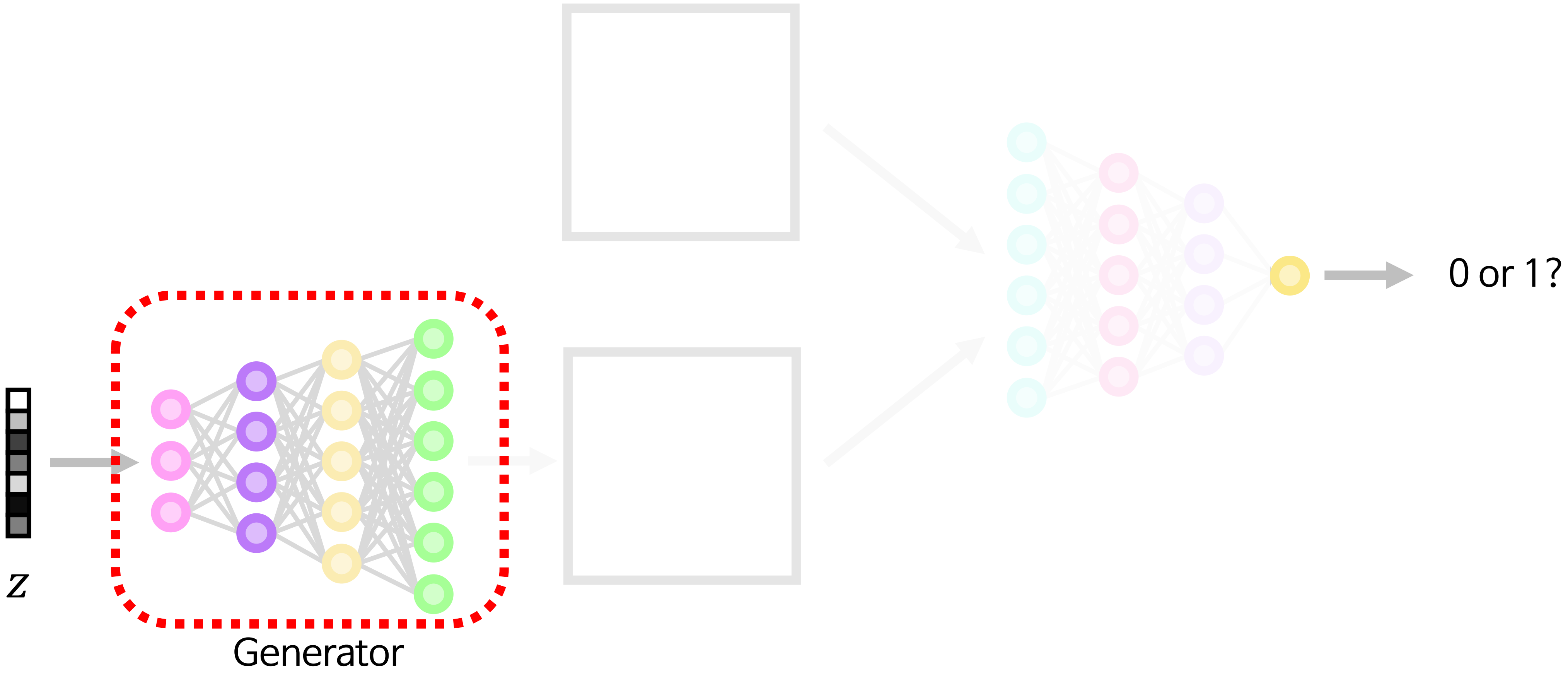


신박AI

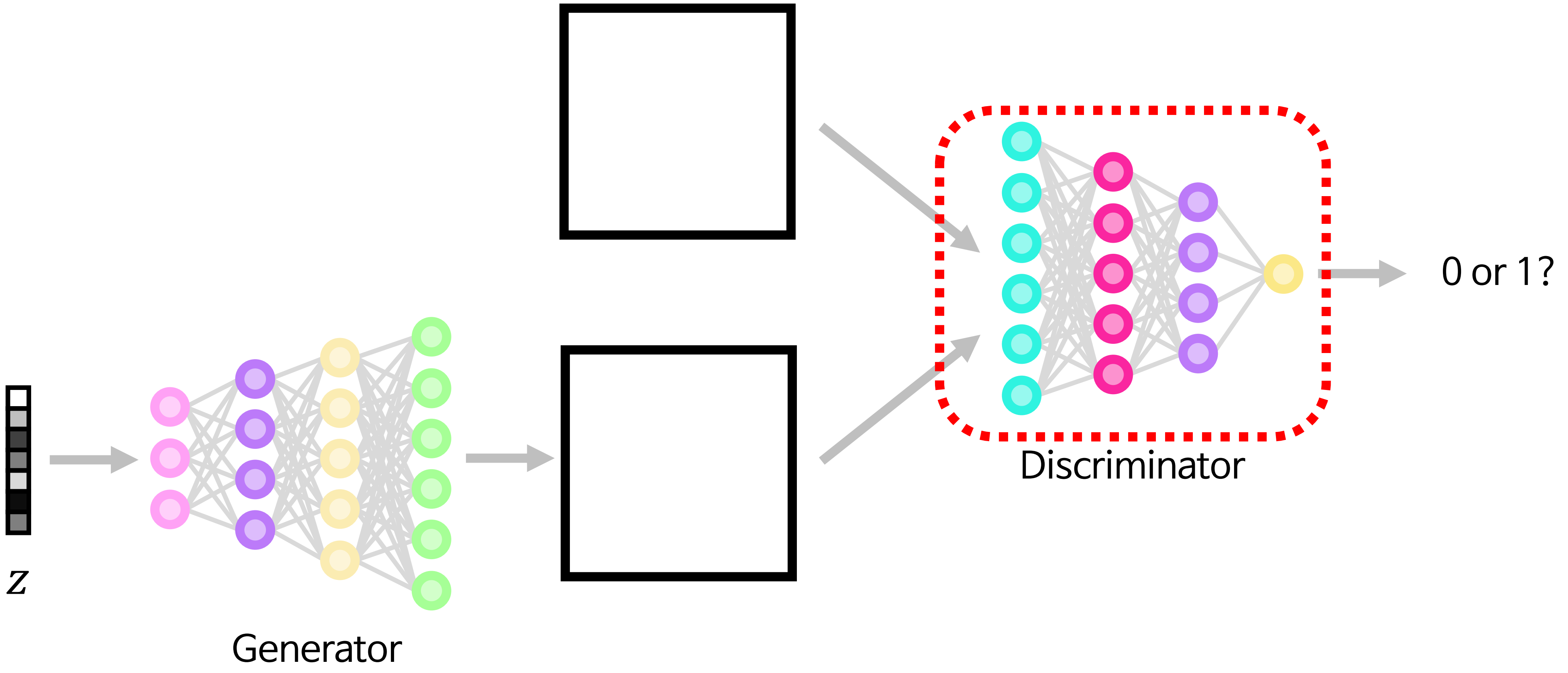
출력은 단순히 0과1 사이의 숫자 하나입니다



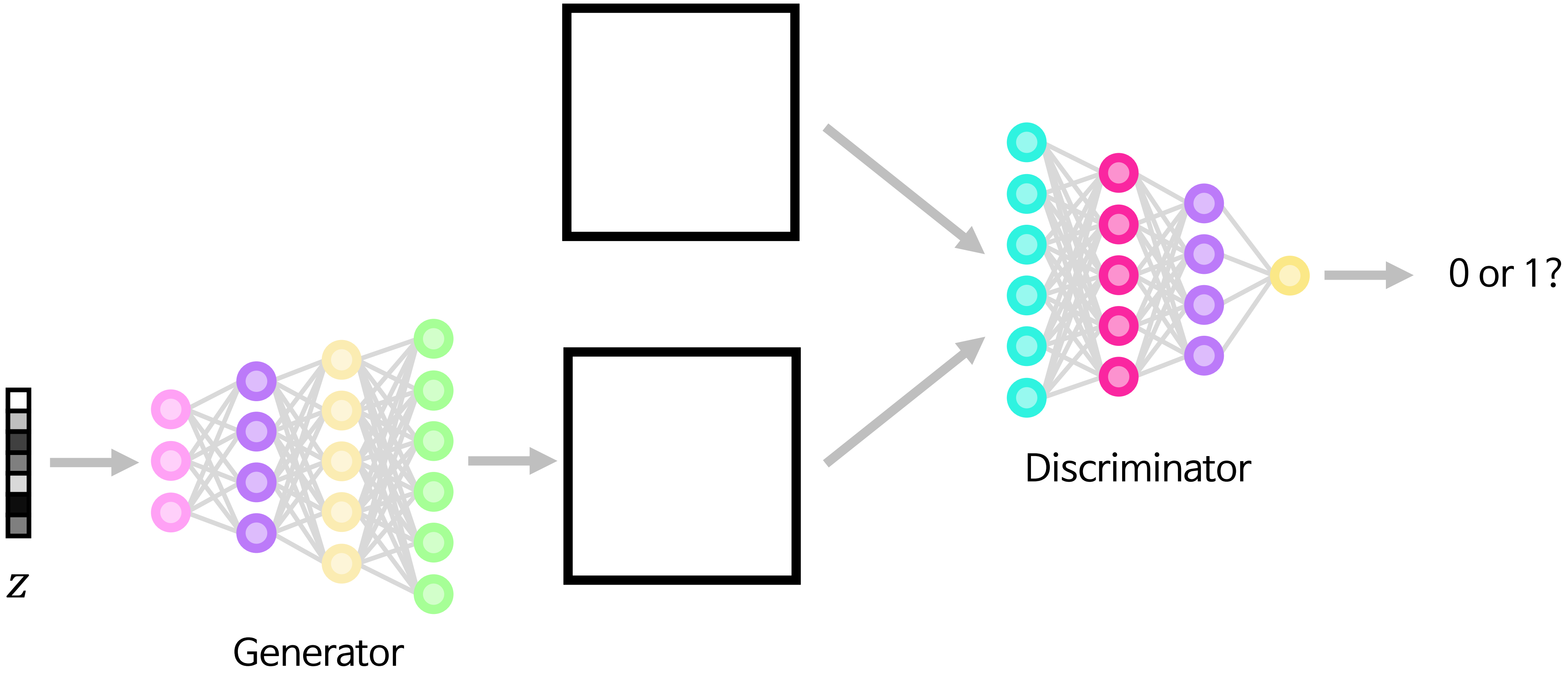
1차원벡터를 입력으로 받는 부분이 바로 위조작가 역할을 하는 generator입니다.



그리고 0 또는 1을 출력하는 부분은 위조품을 판별하는 감별사 역할을 하는, Discriminator라고 합니다.

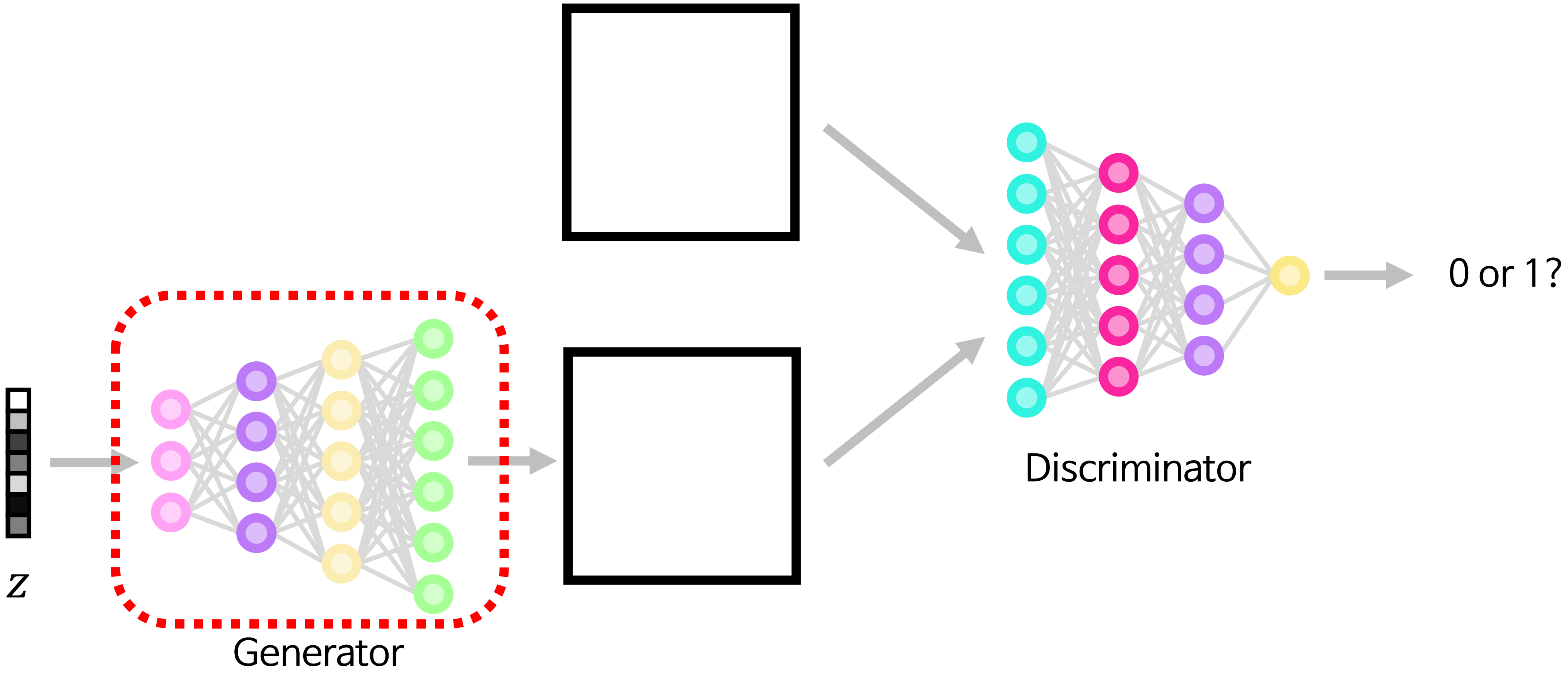


위조 작가가 모조품을 그렸던 것 처럼, 여기서도 랜덤 1차원 벡터를 받아서 사람의 얼굴을 만들어 내는 신경망이라고 가정해 봅시다.

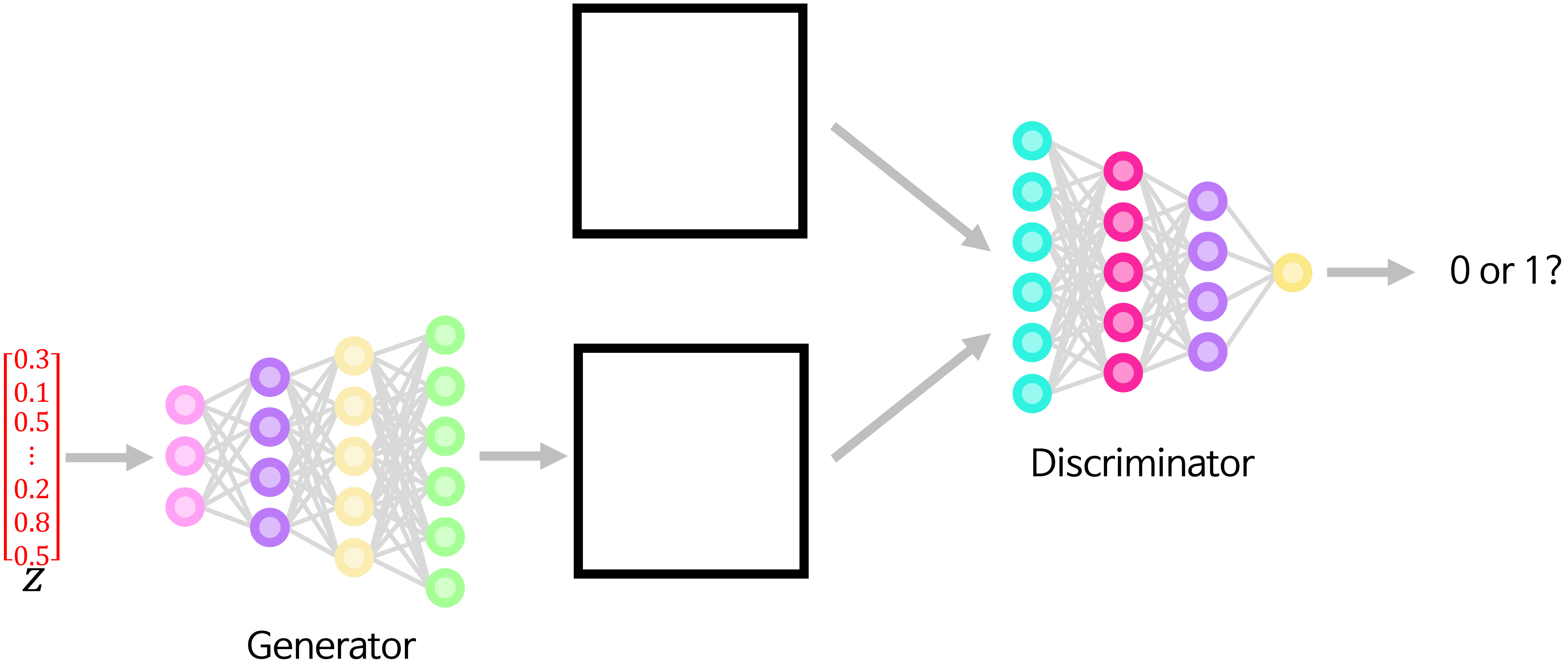


신박AI

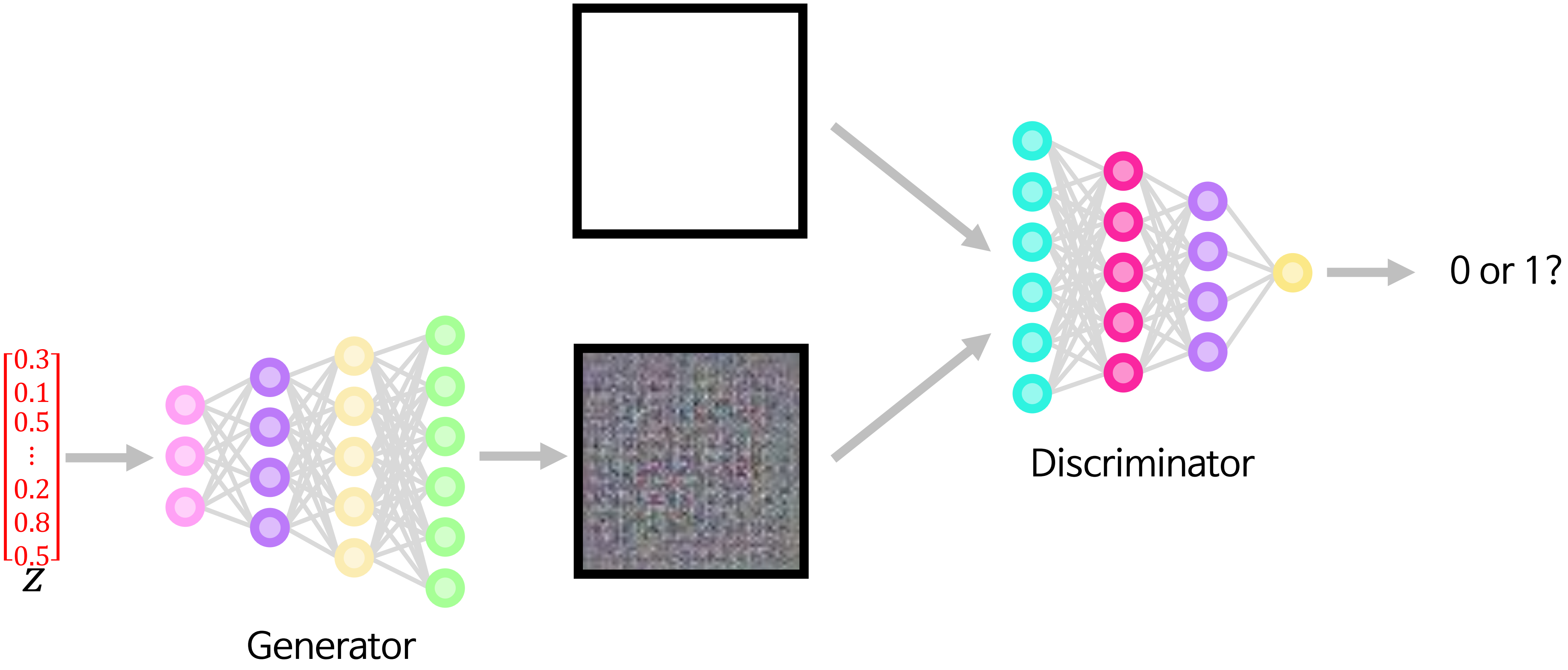
Generator 신경망의 가중치도 처음에는 랜덤이고



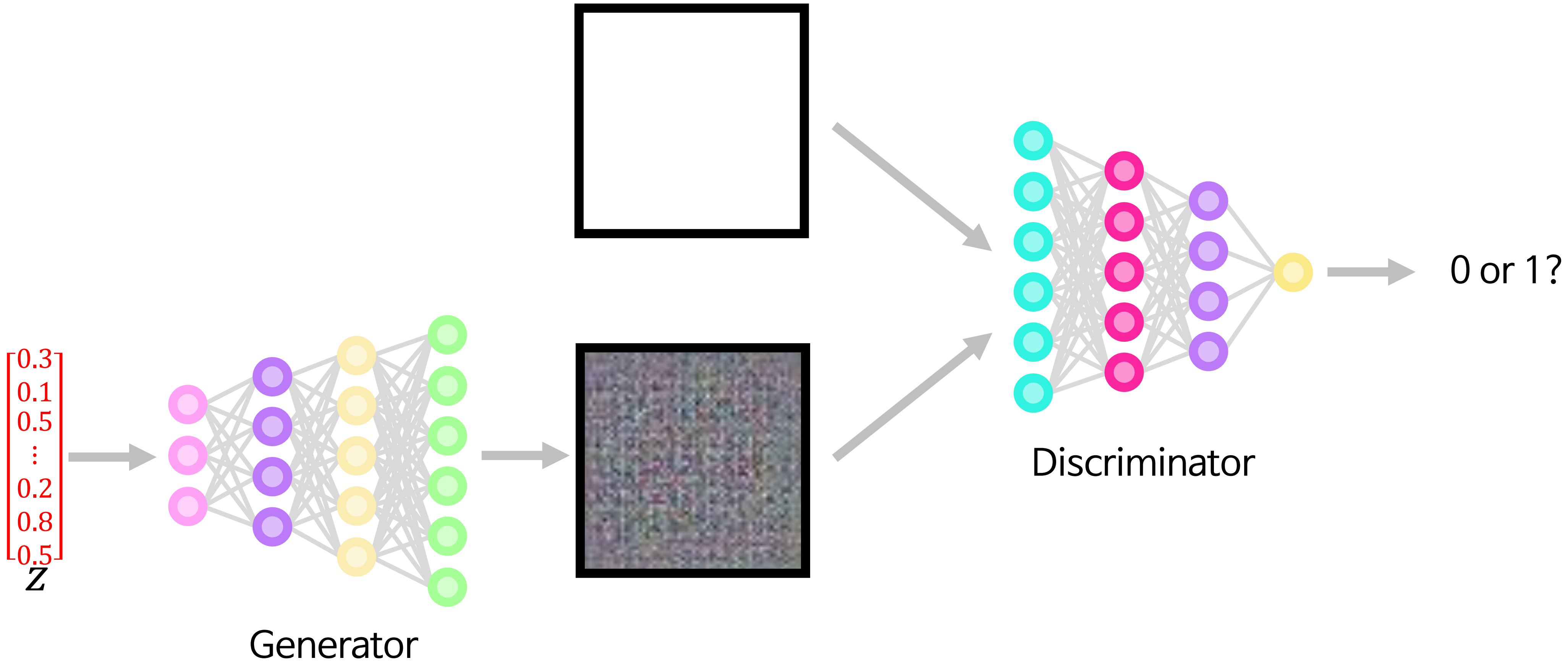
입력은 항상 랜덤이기 때문에



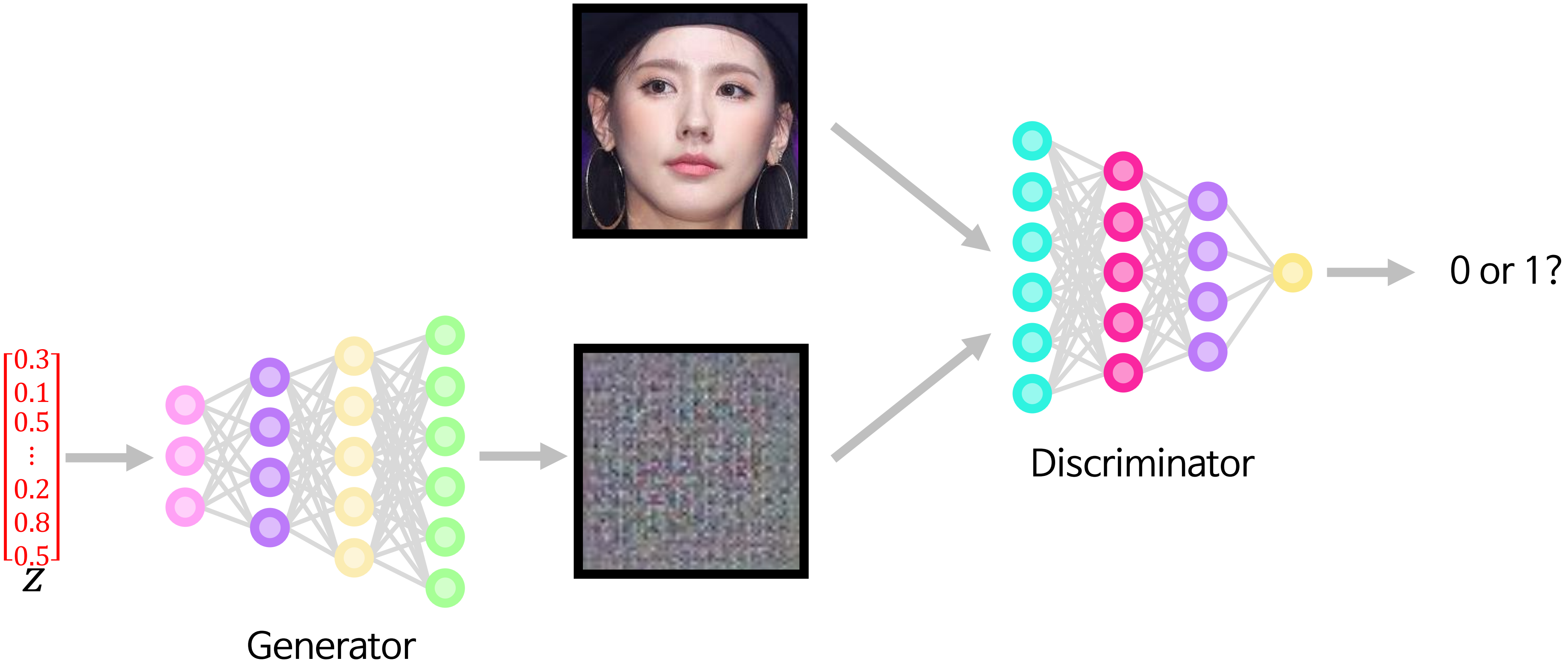
처음에는 알아보기 힘든 아웃풋을 만들어 낼 것입니다.



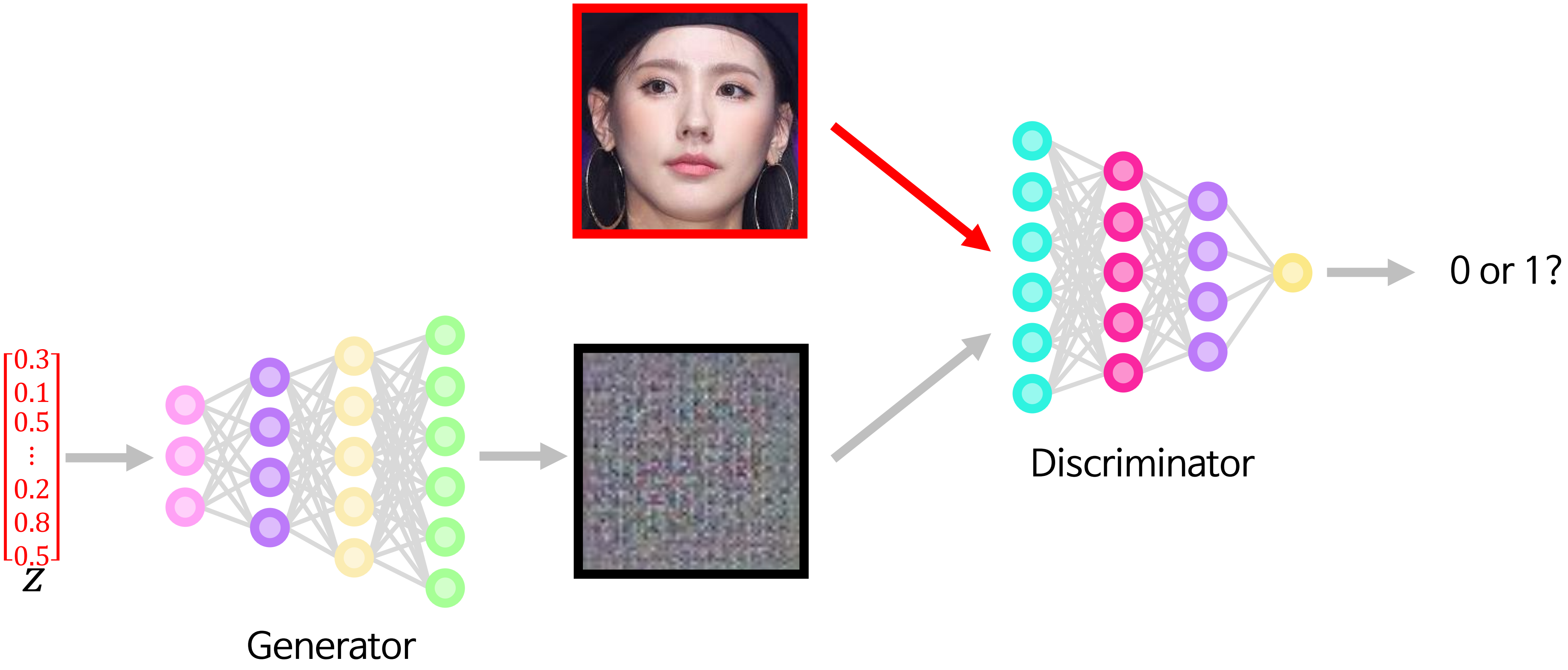
이제는 Discriminator의 차례입니다.



Discriminator를 훈련시키기 위해 실제 데이터를 준비합니다

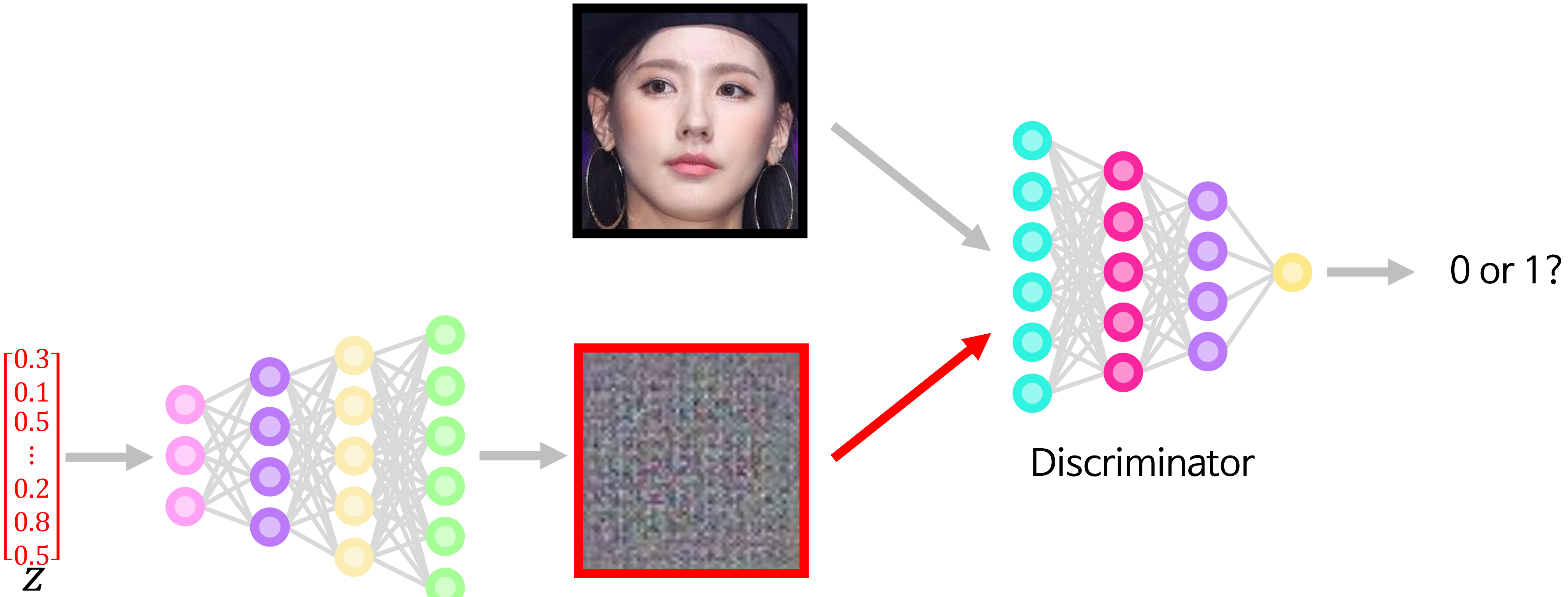


그리고 이 둘 중 하나를 Discriminator의 입력으로 사용합니다.



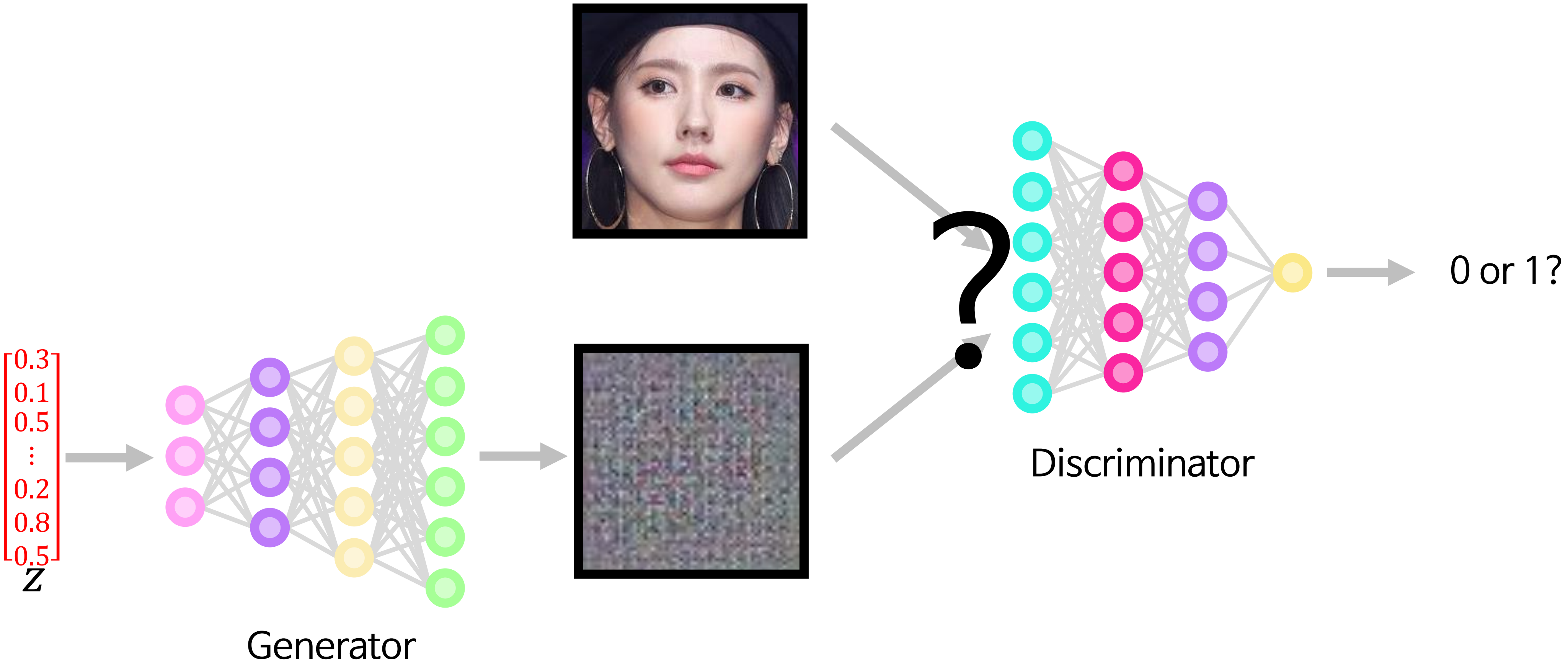
신박AI

여기서 핵심은 이 둘 중 하나의 입력이 랜덤하게 선택이 된다는 것이고

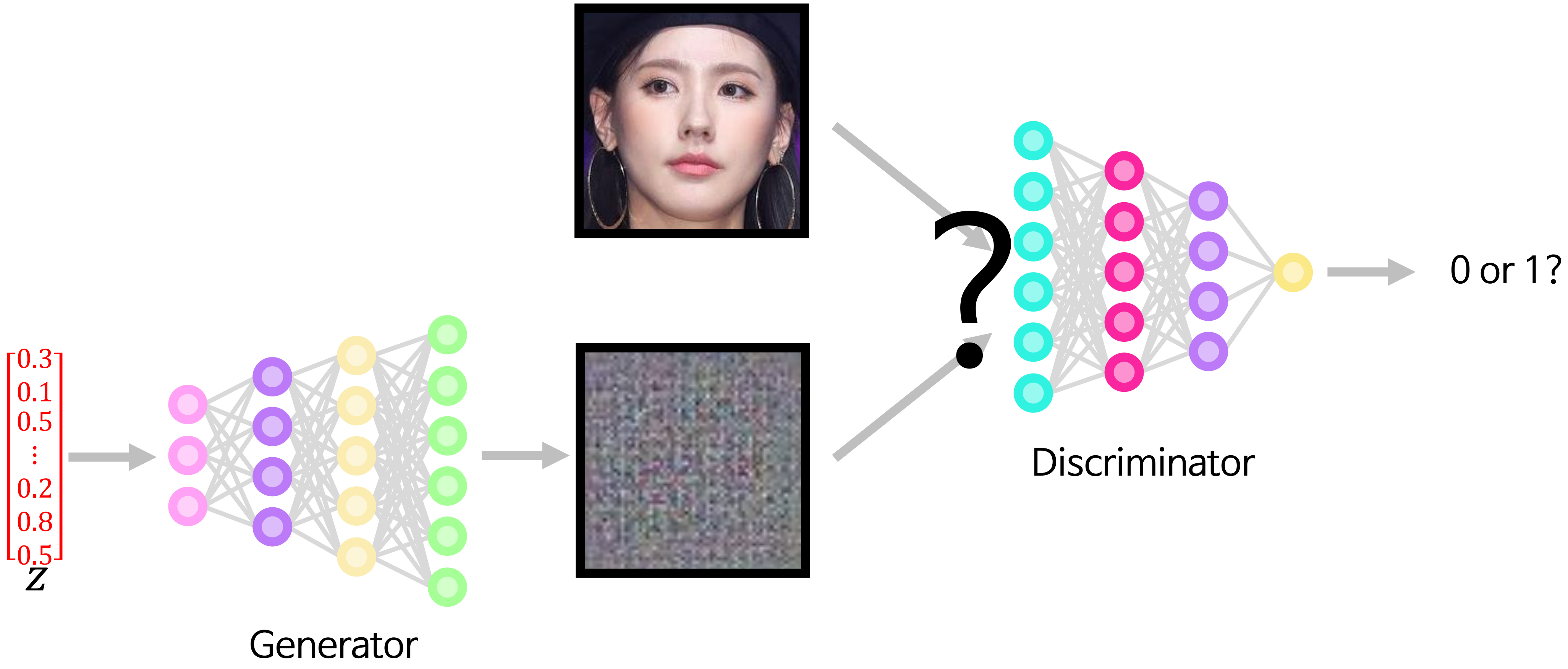


신박AI

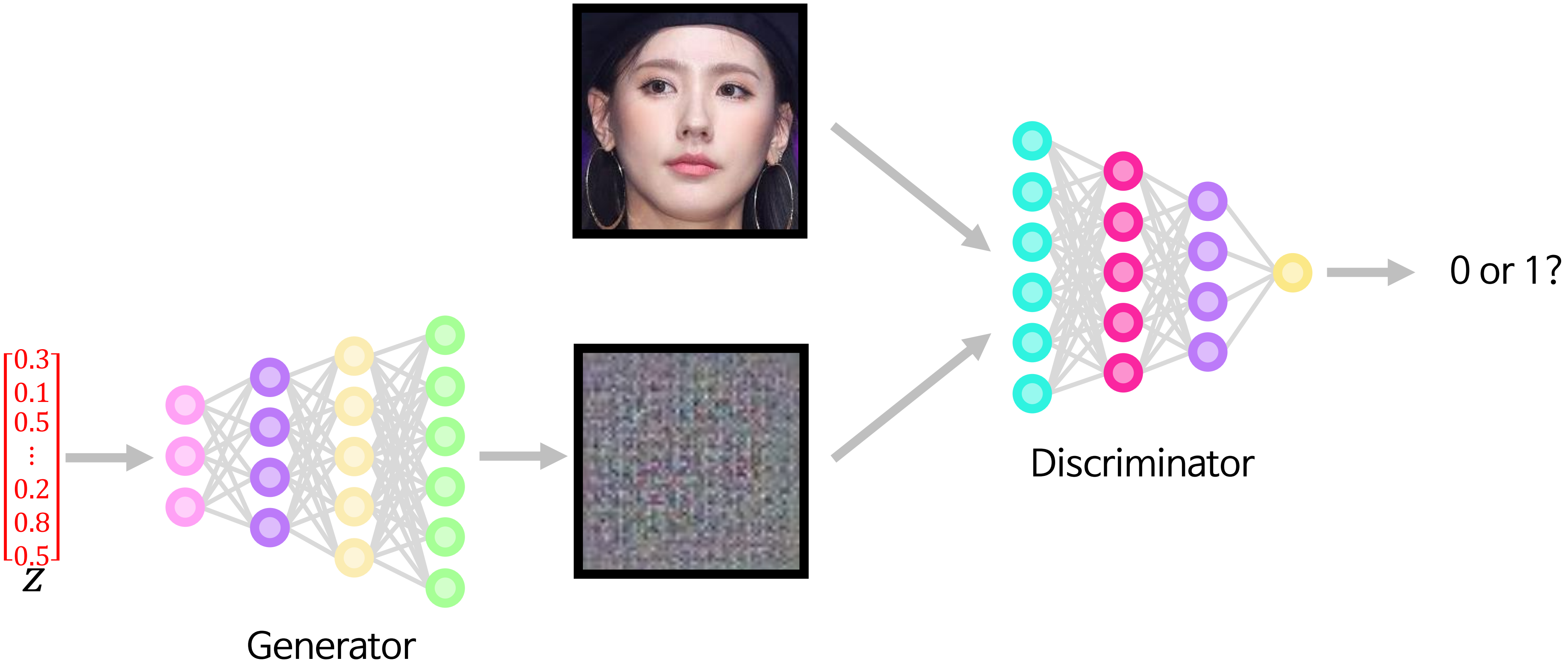
discriminator입장에서는 들어오는 입력이 generator가 생성한 사진인지



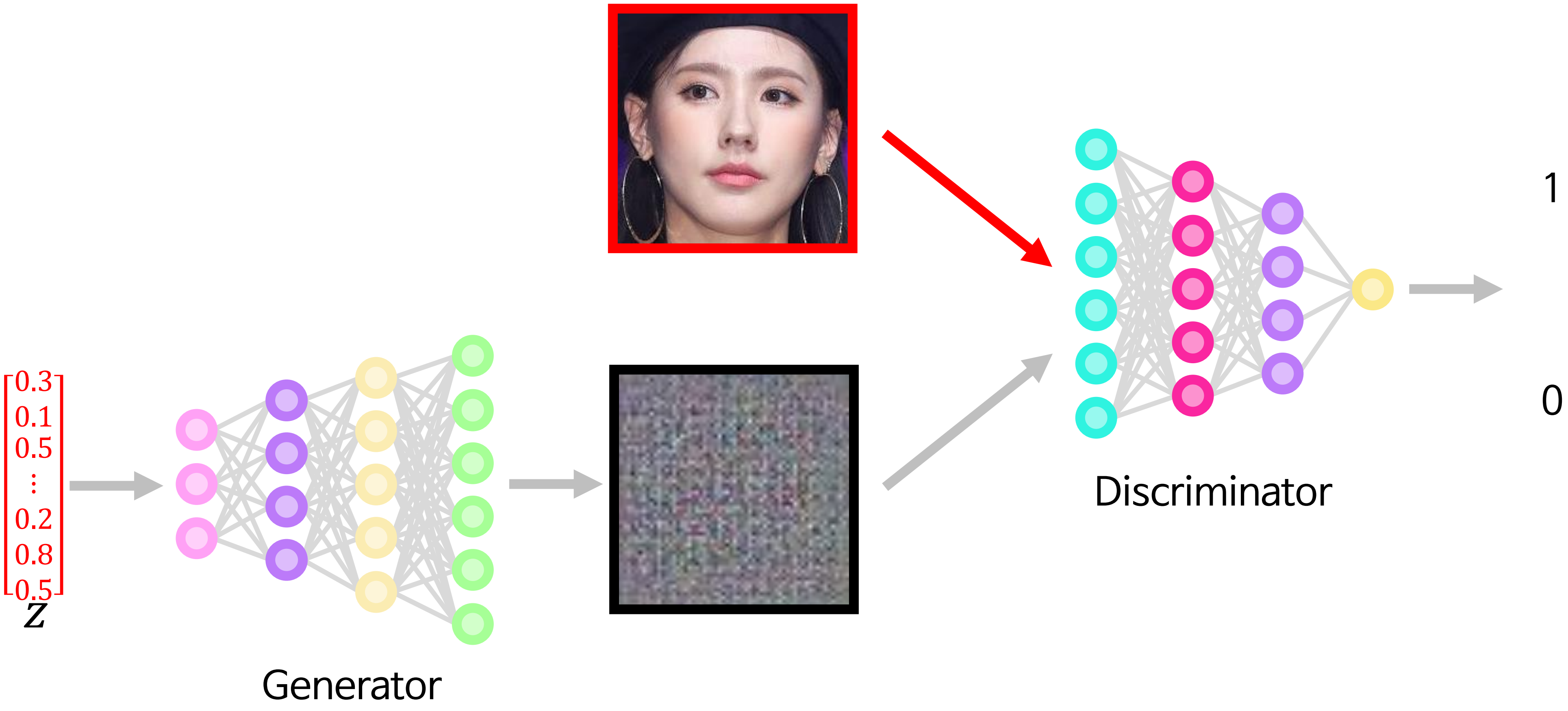
실제 사진인지 모르는 상태에서



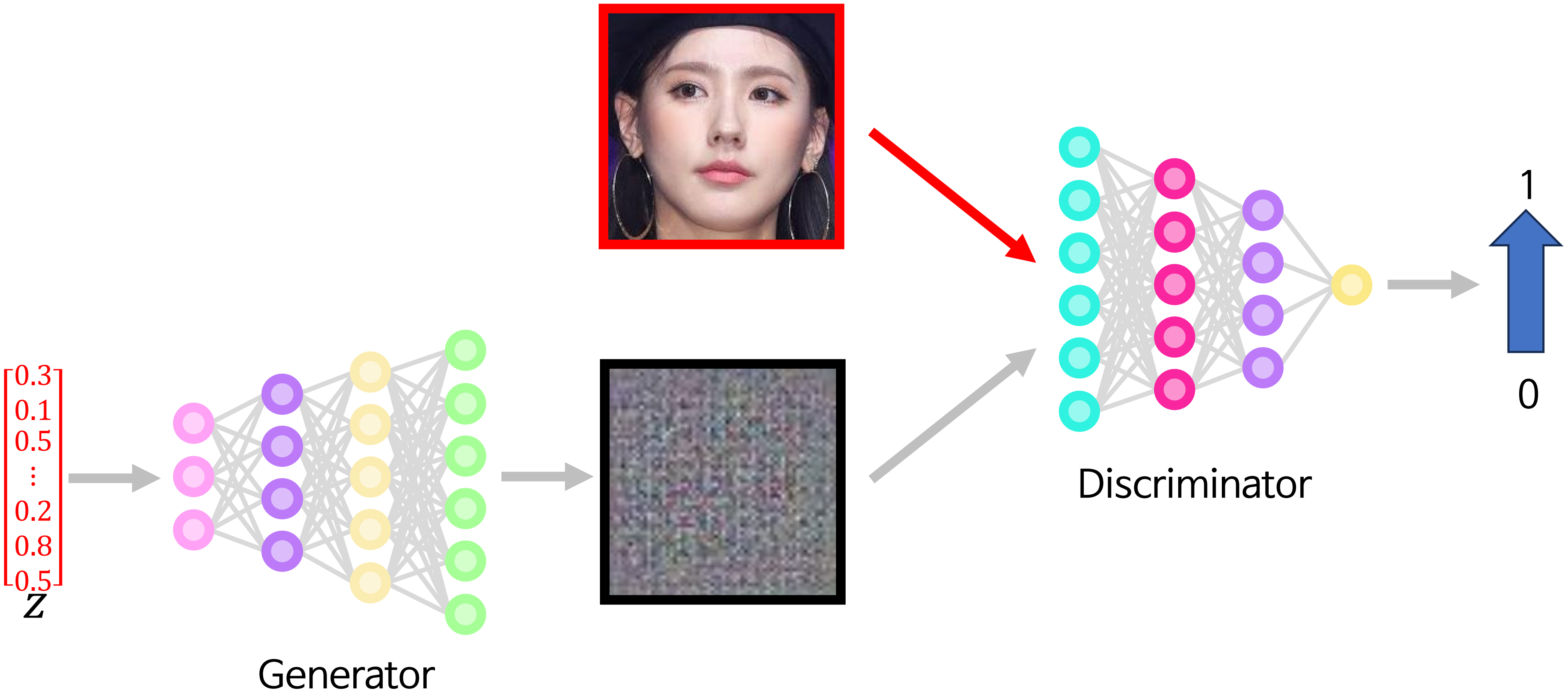
오직 데이터만 보고, 마치 감별사가 하는 것 처럼, 실제 데이터인지 아닌지를 구분해 내야 합니다.



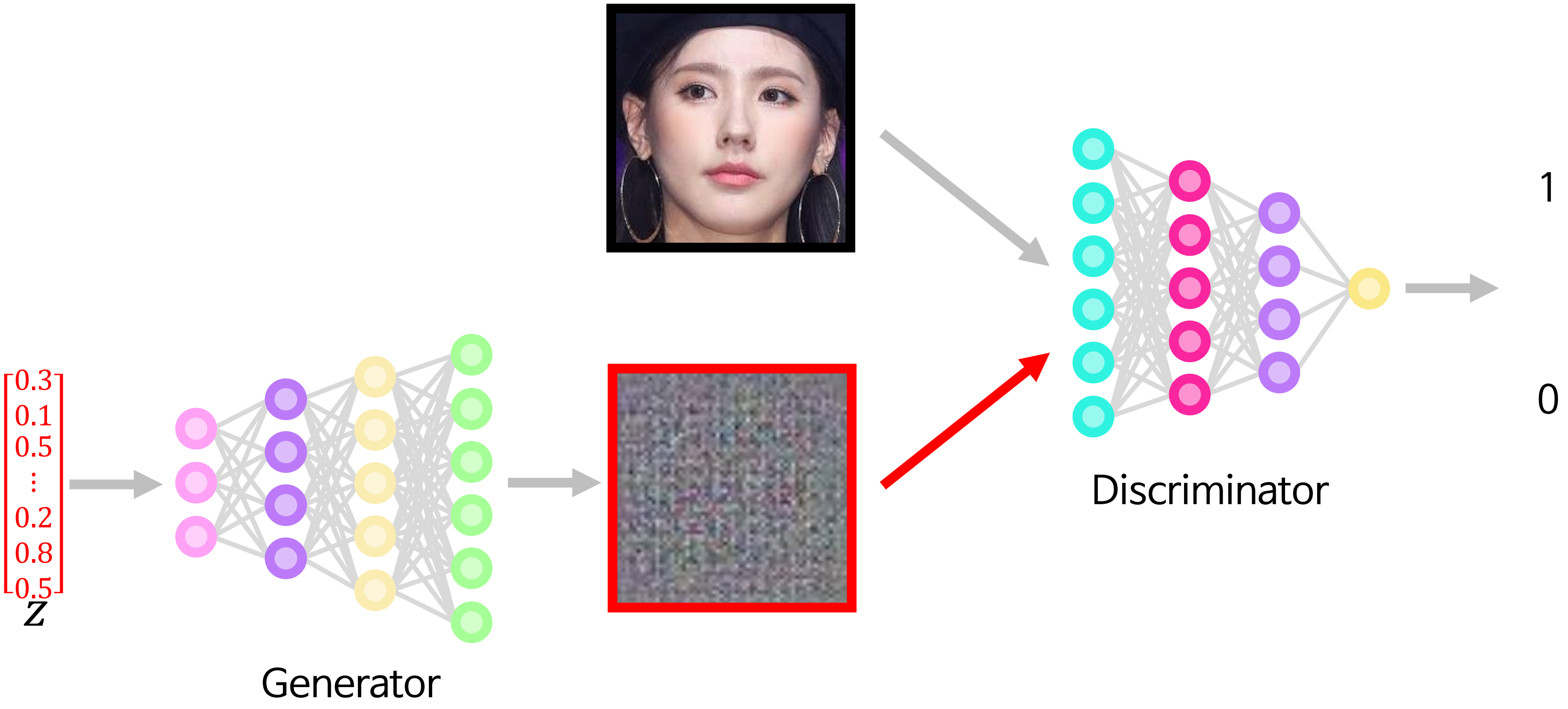
그래서 Discriminator의 경우는, 실제 이미지가 들어왔을 때에는



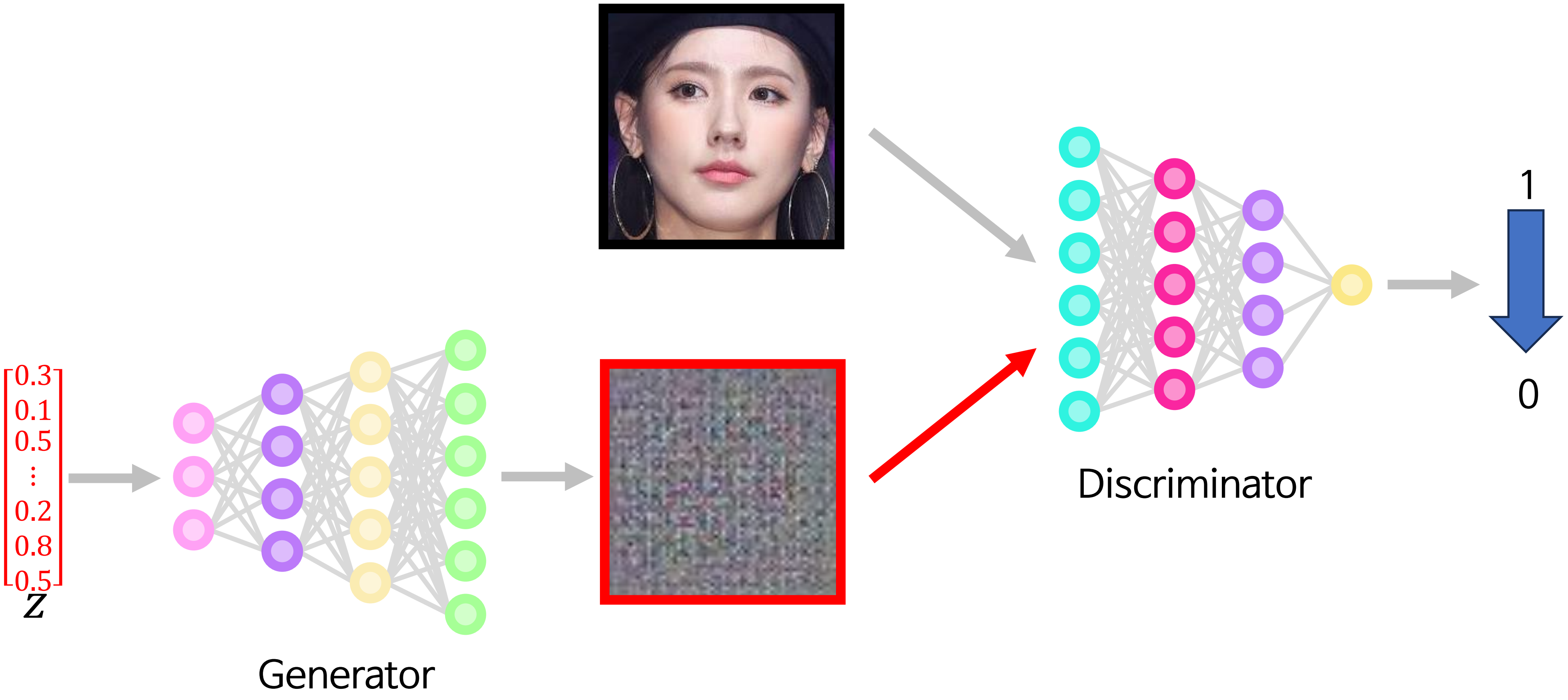
최대한 1에 가까운 값을 내도록 학습되어야 할 것이고,



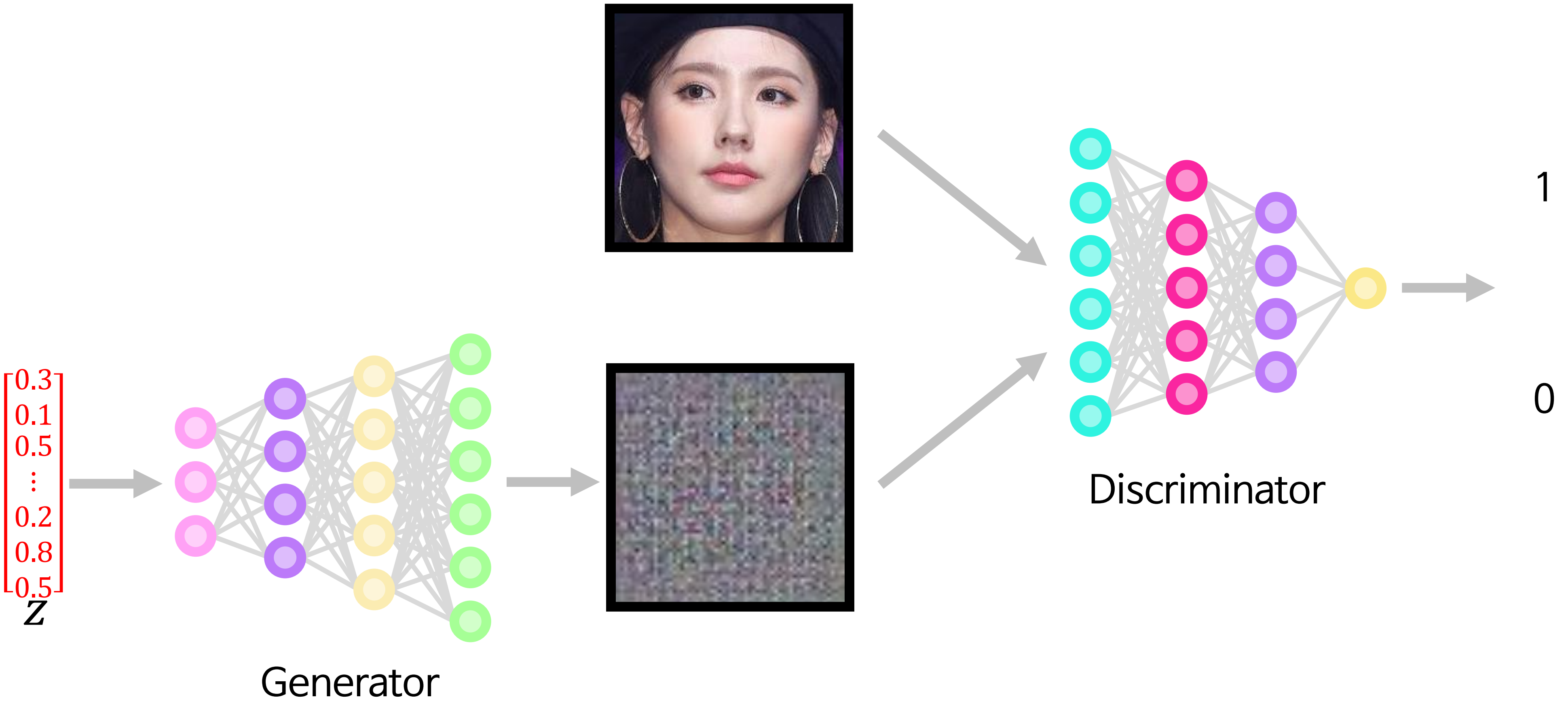
가짜 이미지가 들어왔을 경우엔,



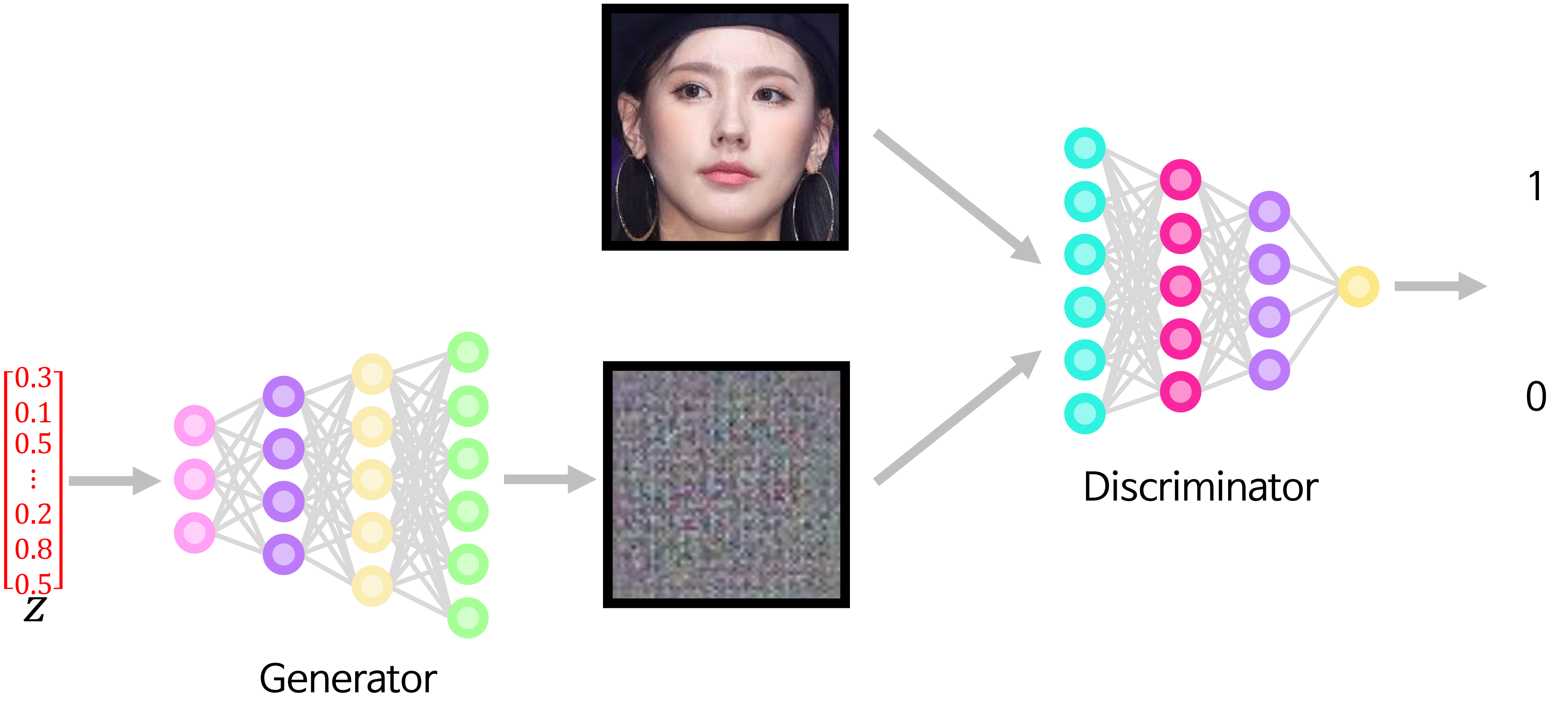
어떻게든 출력값이 0에 가깝도록 훈련되어야 할 것입니다.



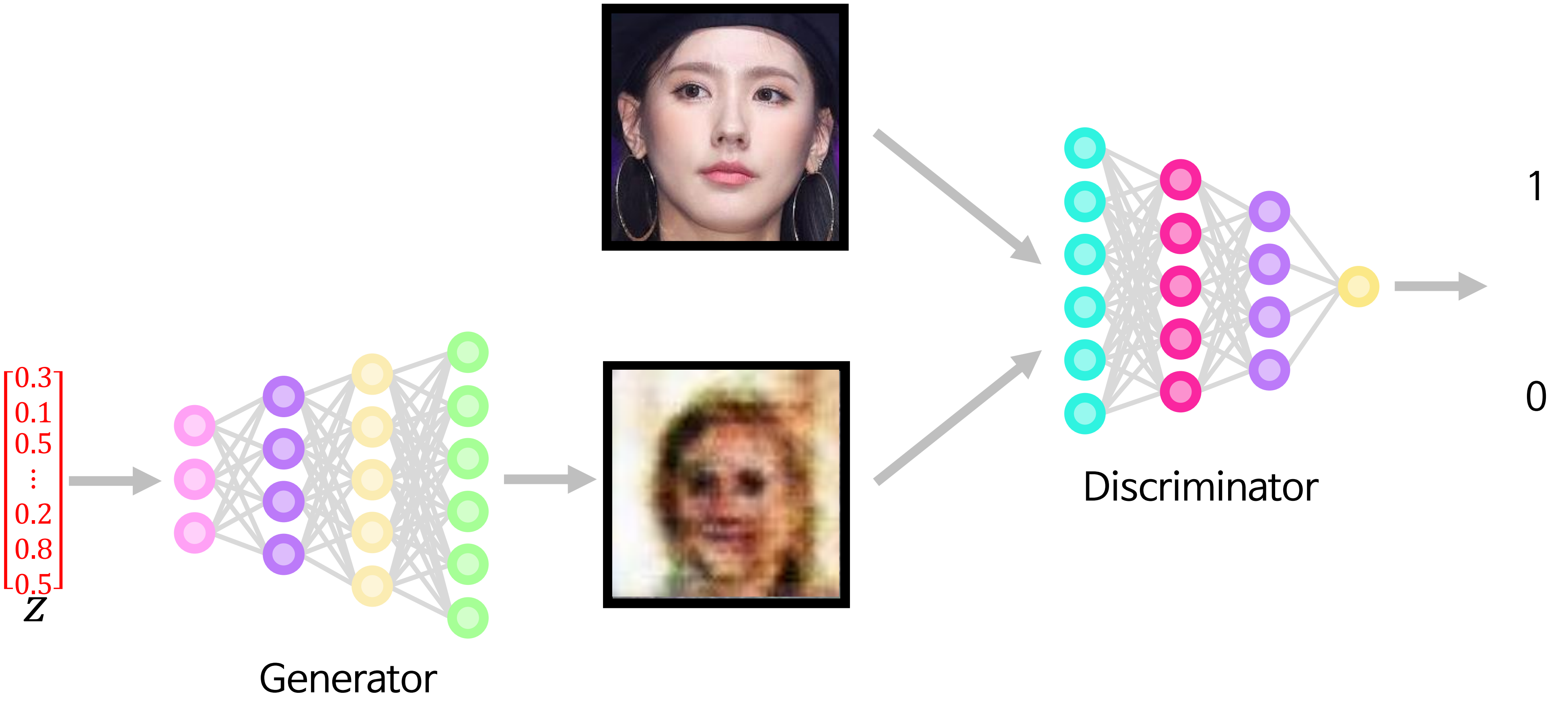
이렇게 Discriminator가 학습을 통해 진짜 이미지와 가짜 이미지를 잘 구분해 내는 감별사가 되어 갑니다.



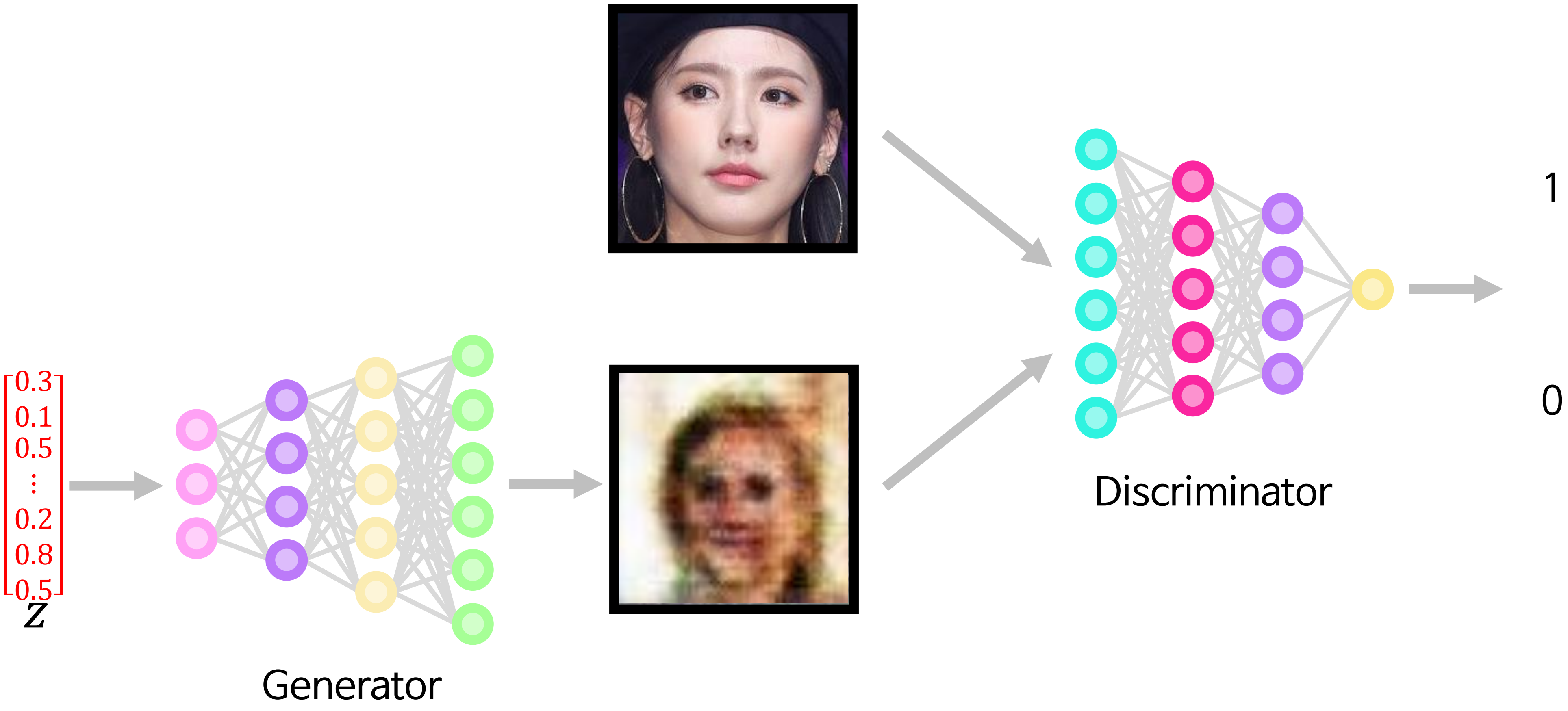
이렇게 Discriminator가 성장하면 성장할 수록,



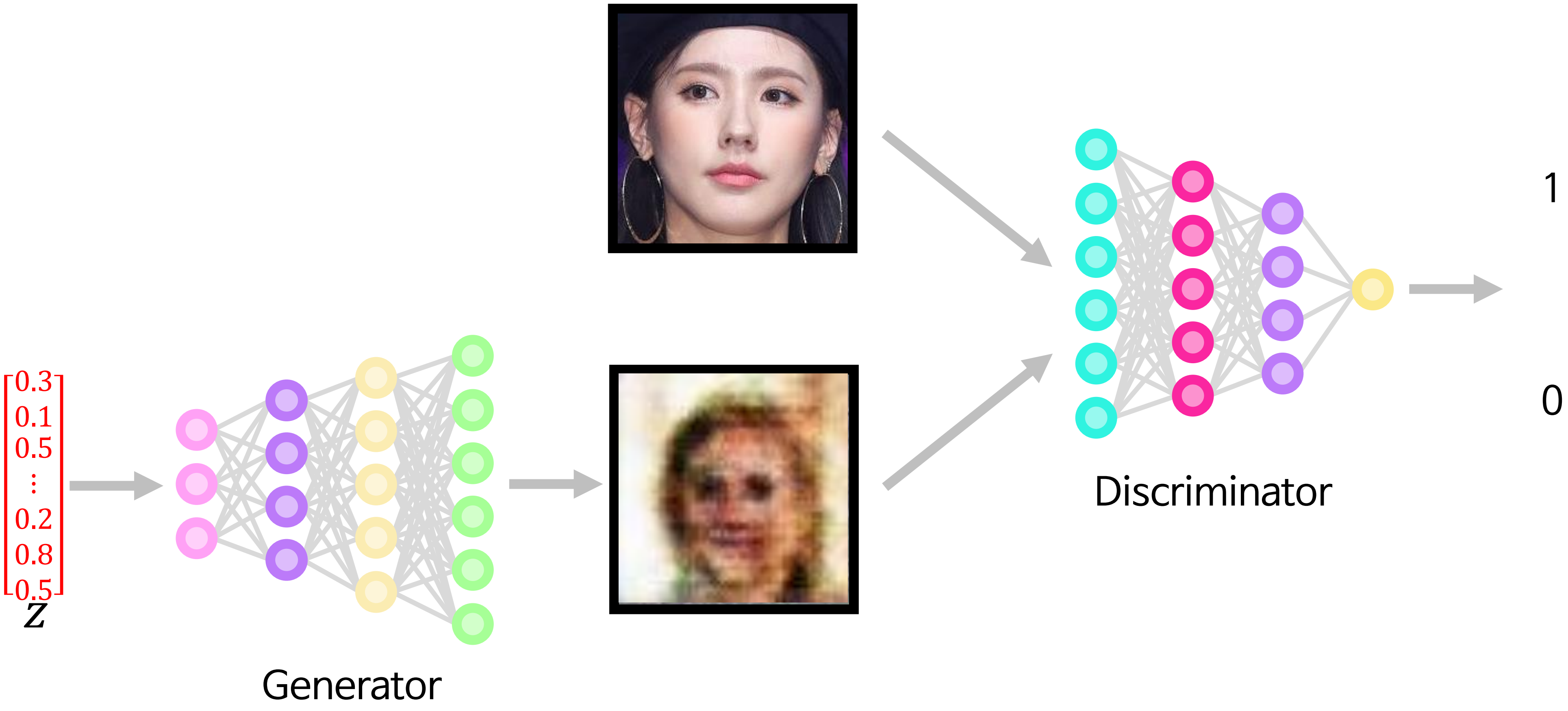
Generator 또한, 실제 이미지와 가까운 이미지를 생성해 내도록 학습이 되어 갑니다.



이렇게 Discriminator와 Generator가 서로 경쟁하듯 학습하는 모델이라고 하여



생성적 적대모델 Generative Adversarial Network라고 불리는 것입니다

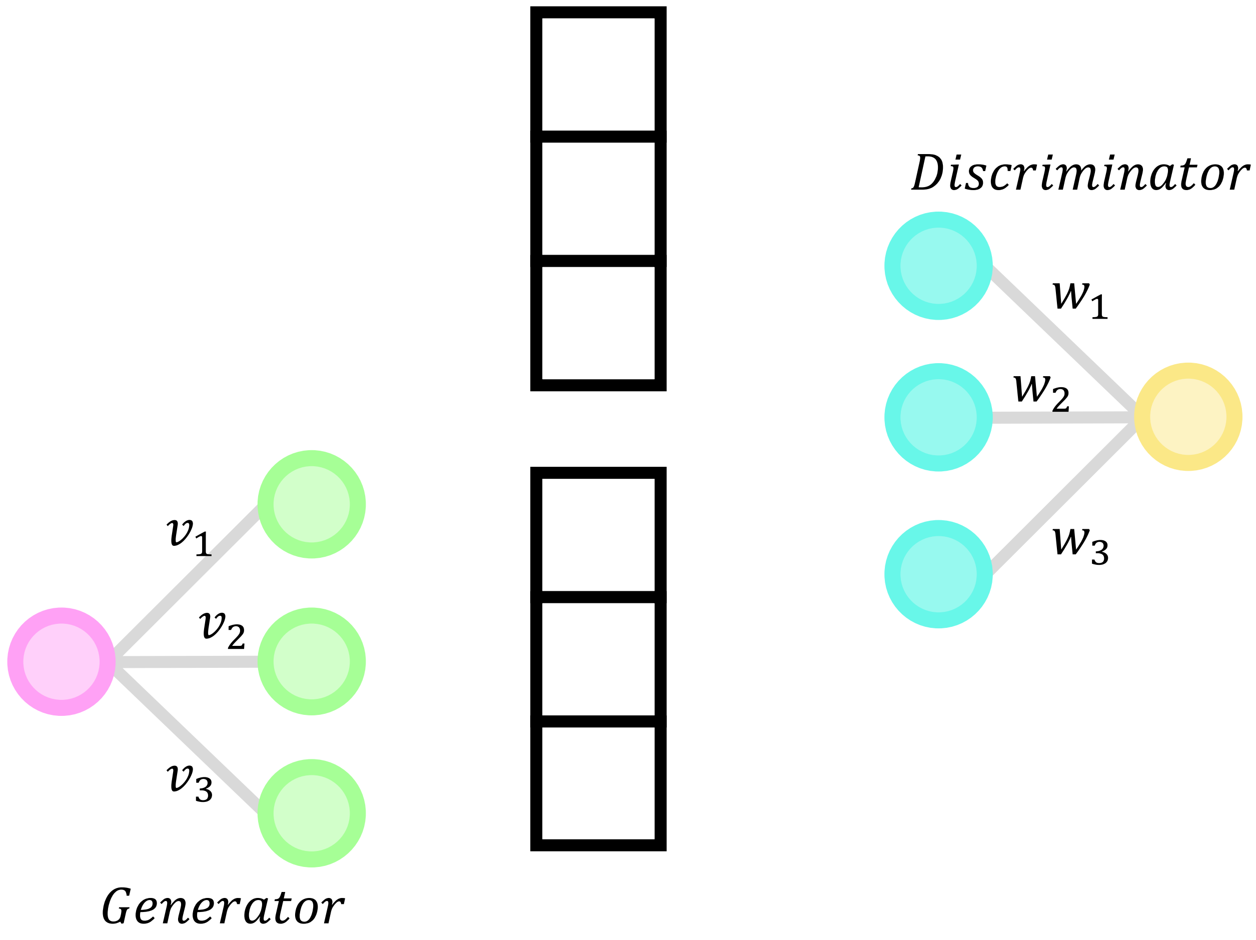


그러면, 이제는 실제 GAN 모델이 어떻게 정보를 처리하고 학습하는지

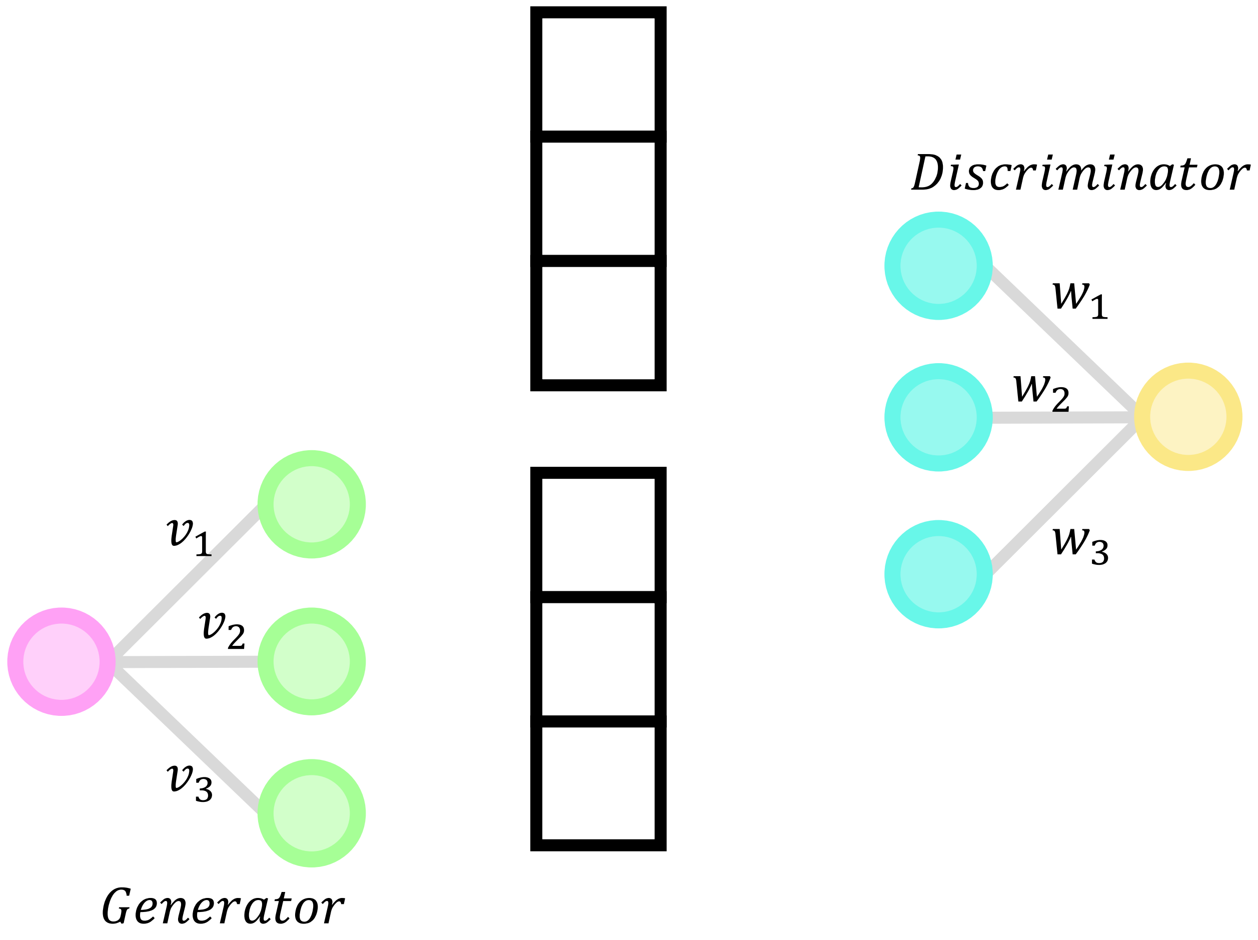
실제 숫자를 넣어서 간략하게 살펴보도록 하겠습니다

실제 숫자를 통해 계산하는 것이니 만큼, 최대한 간단한 모델을
사용하도록 하겠습니다

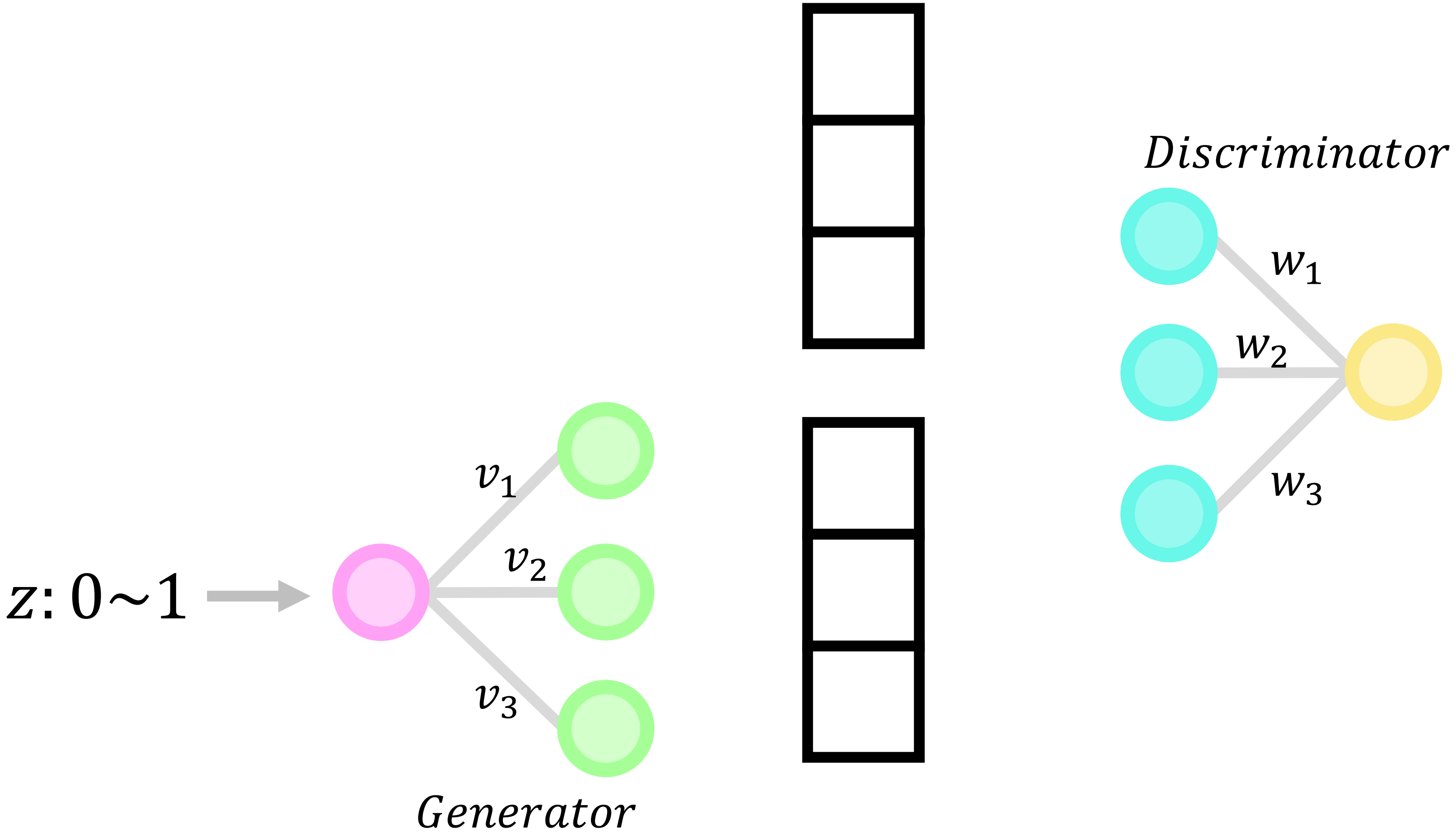
실제 숫자를 통해 계산하는 것이니 만큼, 최대한 간단한 모델을 사용하도록 하겠습니다



각각의 generator와 discriminator는 계산을 간단하게 하기 위해 single layer 신경망으로 구성해보겠습니다

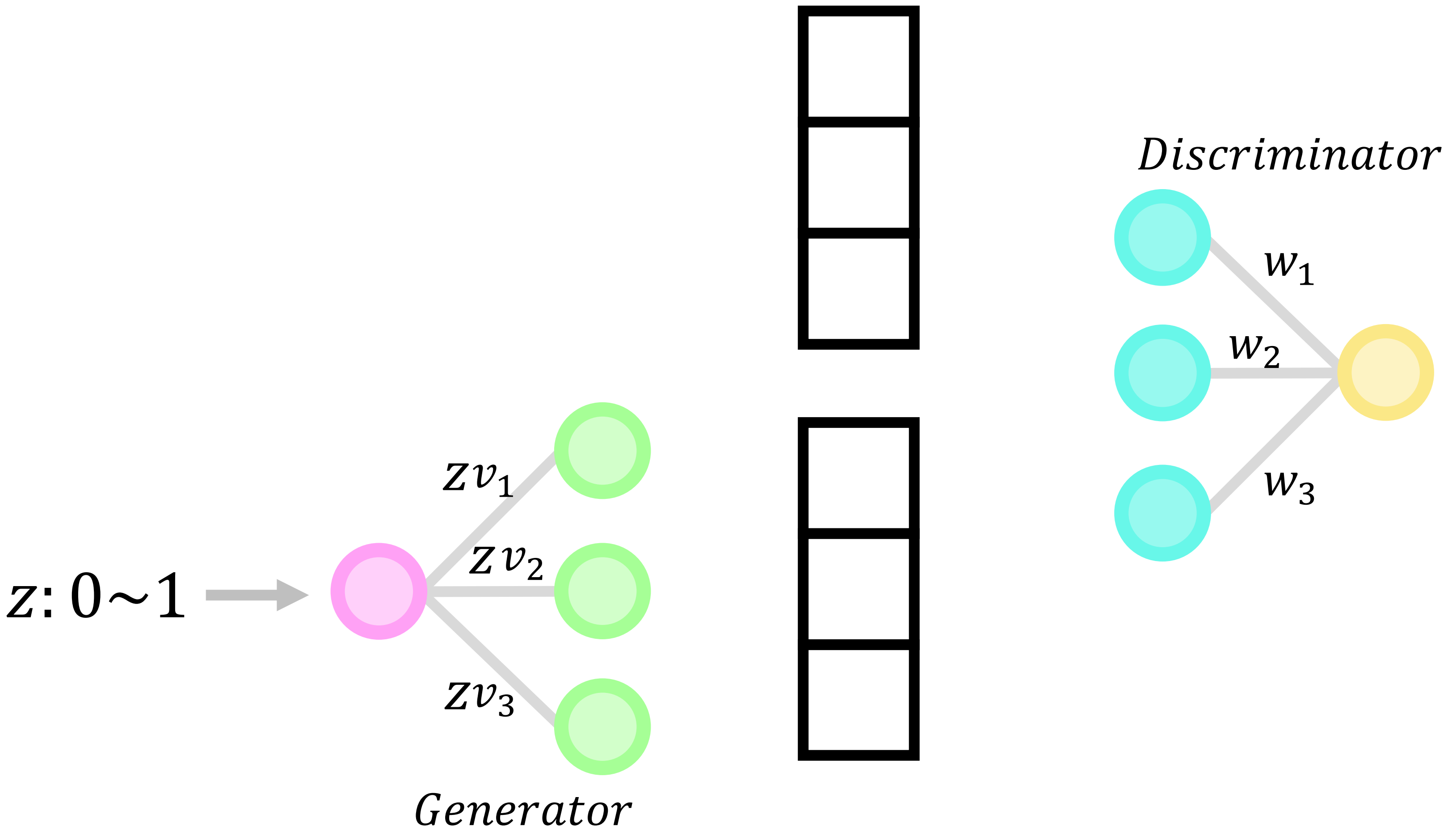


그리고 generator의 입력값은 크기가 1이고 0과 1사이의 값을 랜덤하게 입력하도록 하겠습니다.

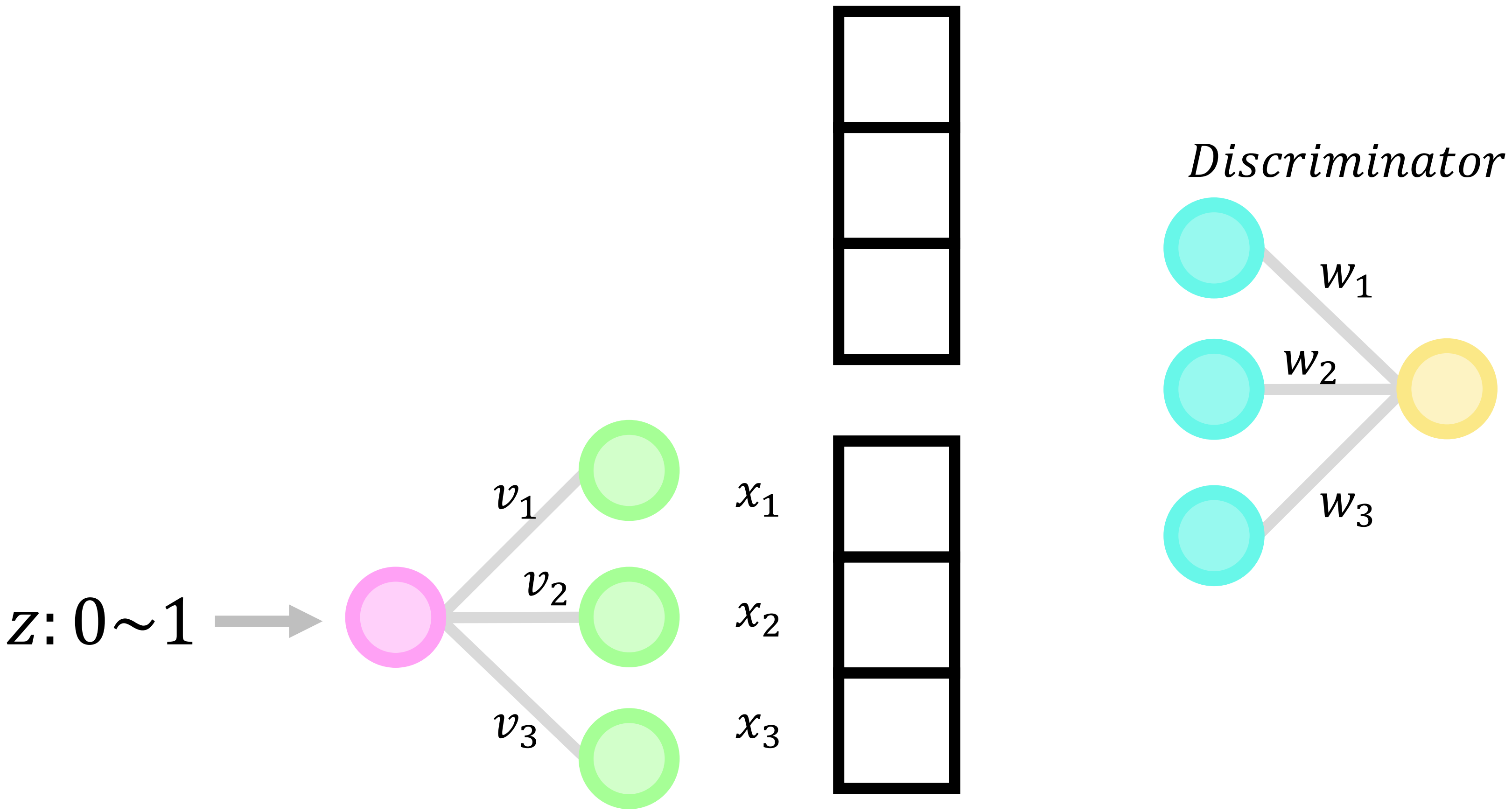


신박AI

그리고 그 입력값은 generator의 각각 가중치와 결합하여..

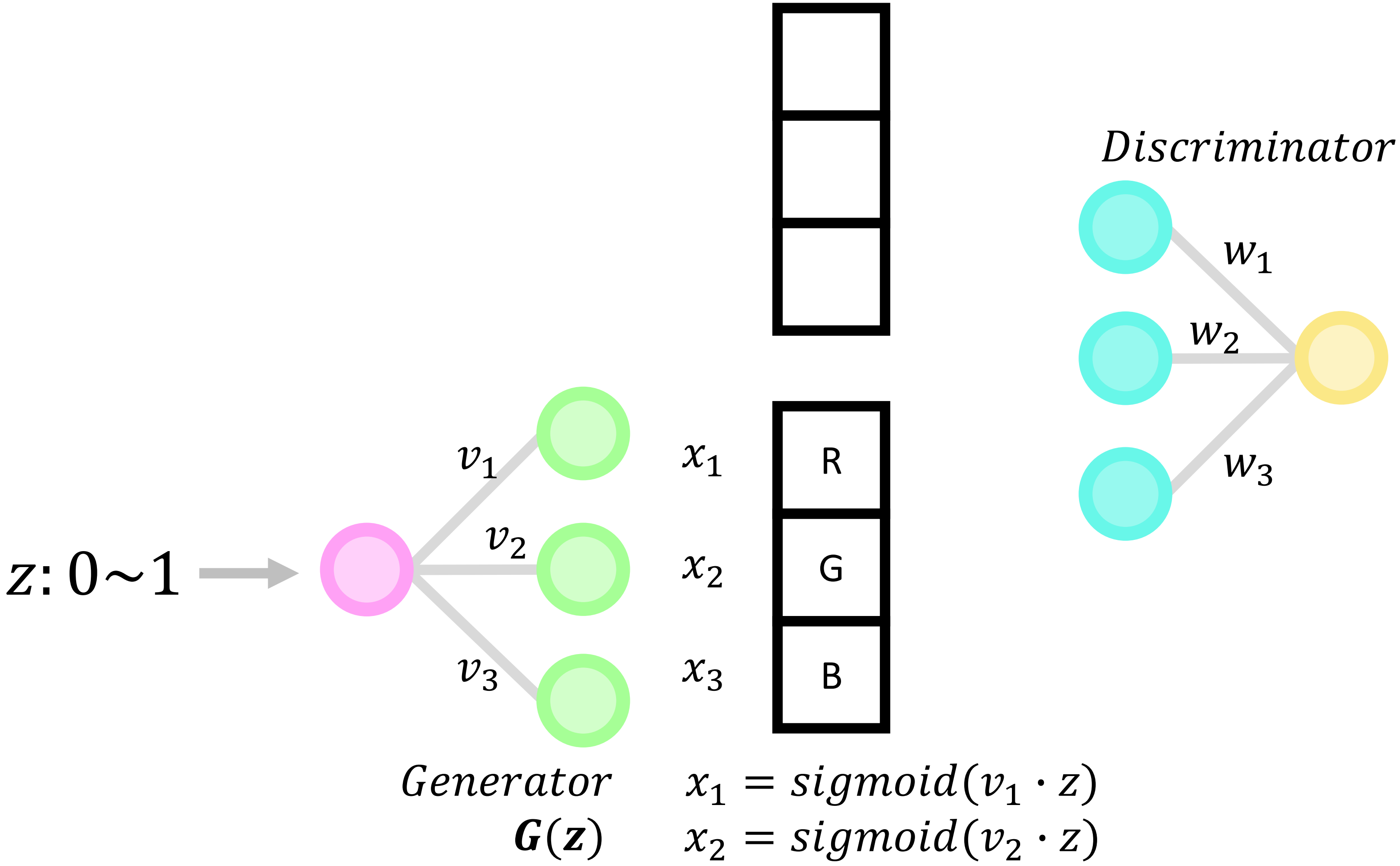


Generator의 아웃풋인 3개의 원소를 갖는 1차원 행렬을 생성하는 일을 한다고 가정해보겠습니다.



$$x_1 = sigmoid(v_1 \cdot z)$$
$$x_2 = sigmoid(v_2 \cdot z)$$
$$x_3 = sigmoid(v_3 \cdot z)$$

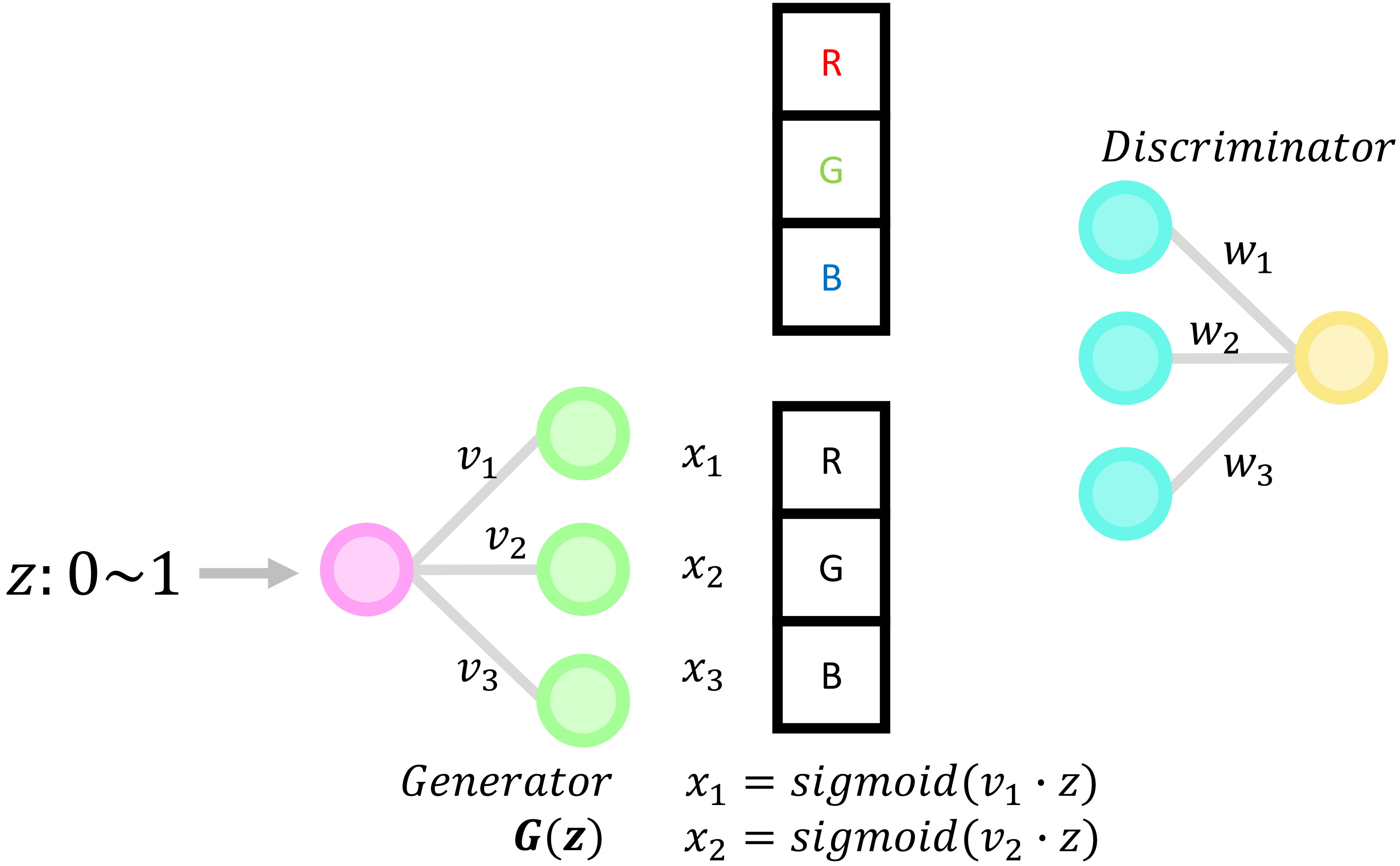
즉 generator는 일종의 rgb값을 갖는 컬러 행렬 (3x1)을 생성합니다



$$x_1 = sigmoid(v_1 \cdot z)$$
$$x_2 = sigmoid(v_2 \cdot z)$$
$$x_3 = sigmoid(v_3 \cdot z)$$

신박AI

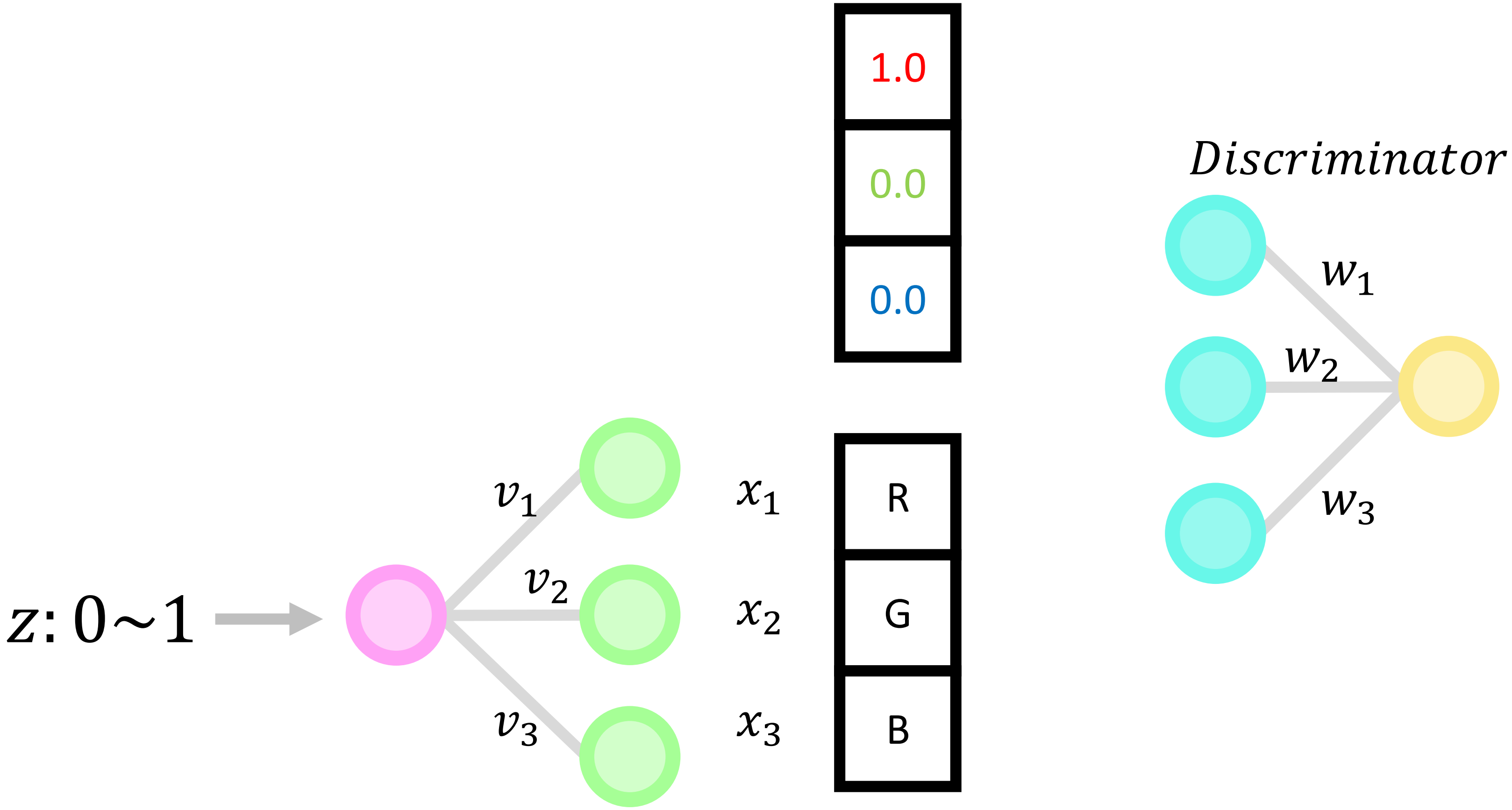
그리고 GAN모델은 아까의 설명에서 처럼 진짜 그림 혹은 진짜 사람 얼굴에 해당하는 real 데이터를 사용한 입력값도 있어야 합니다.



$$x_1 = sigmoid(v_1 \cdot z)$$
$$x_2 = sigmoid(v_2 \cdot z)$$
$$x_3 = sigmoid(v_3 \cdot z)$$

신박AI

지금의 GAN모델에서는 진짜 빨간색 (pure red)인 rgb값을 진짜에 해당하는 값으로 정의를 내리고,

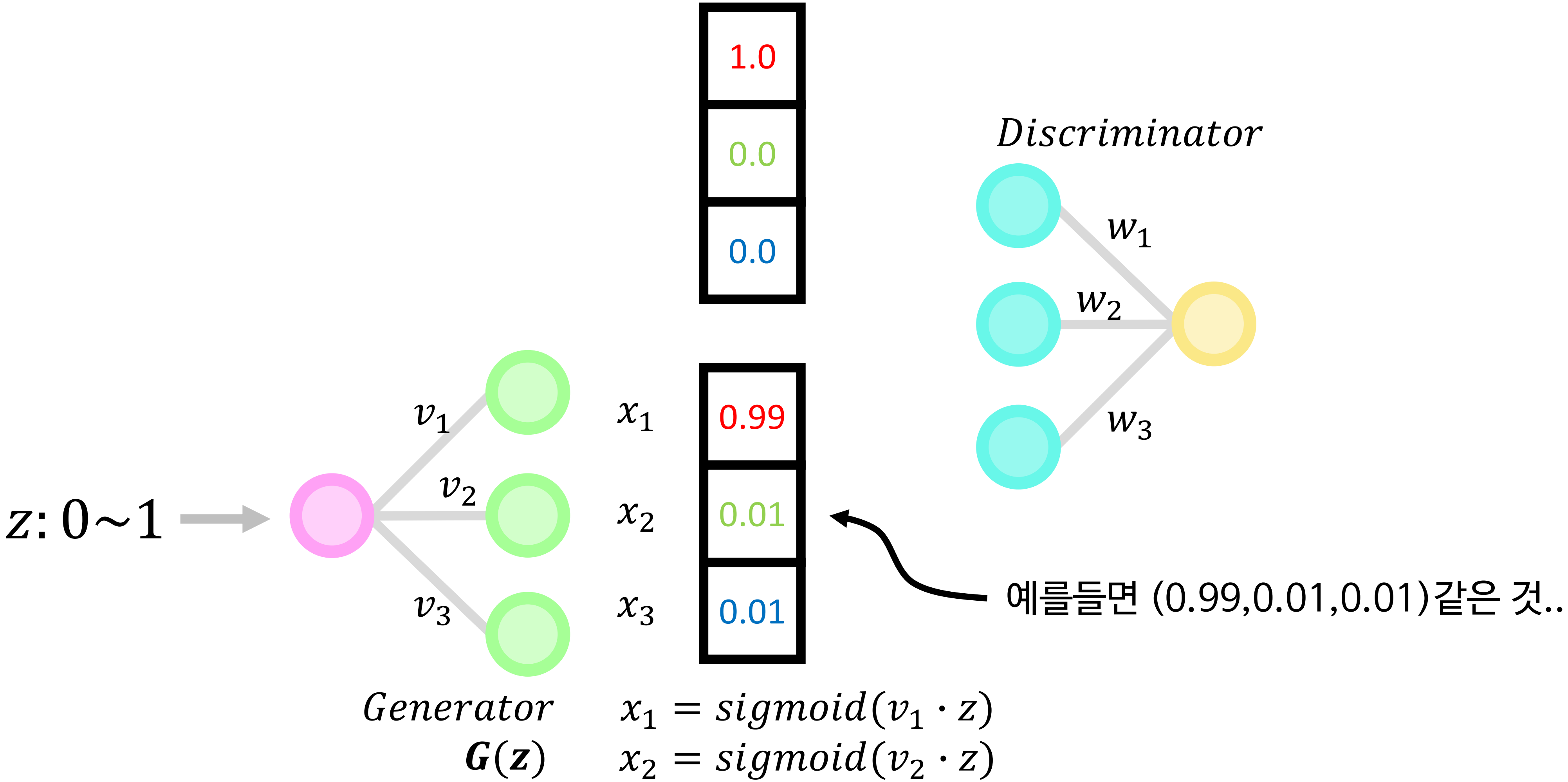


Generator
$\boldsymbol{G(z)}$

$$x_1 = sigmoid(v_1 \cdot z)$$
$$x_2 = sigmoid(v_2 \cdot z)$$
$$x_3 = sigmoid(v_3 \cdot z)$$

신박AI

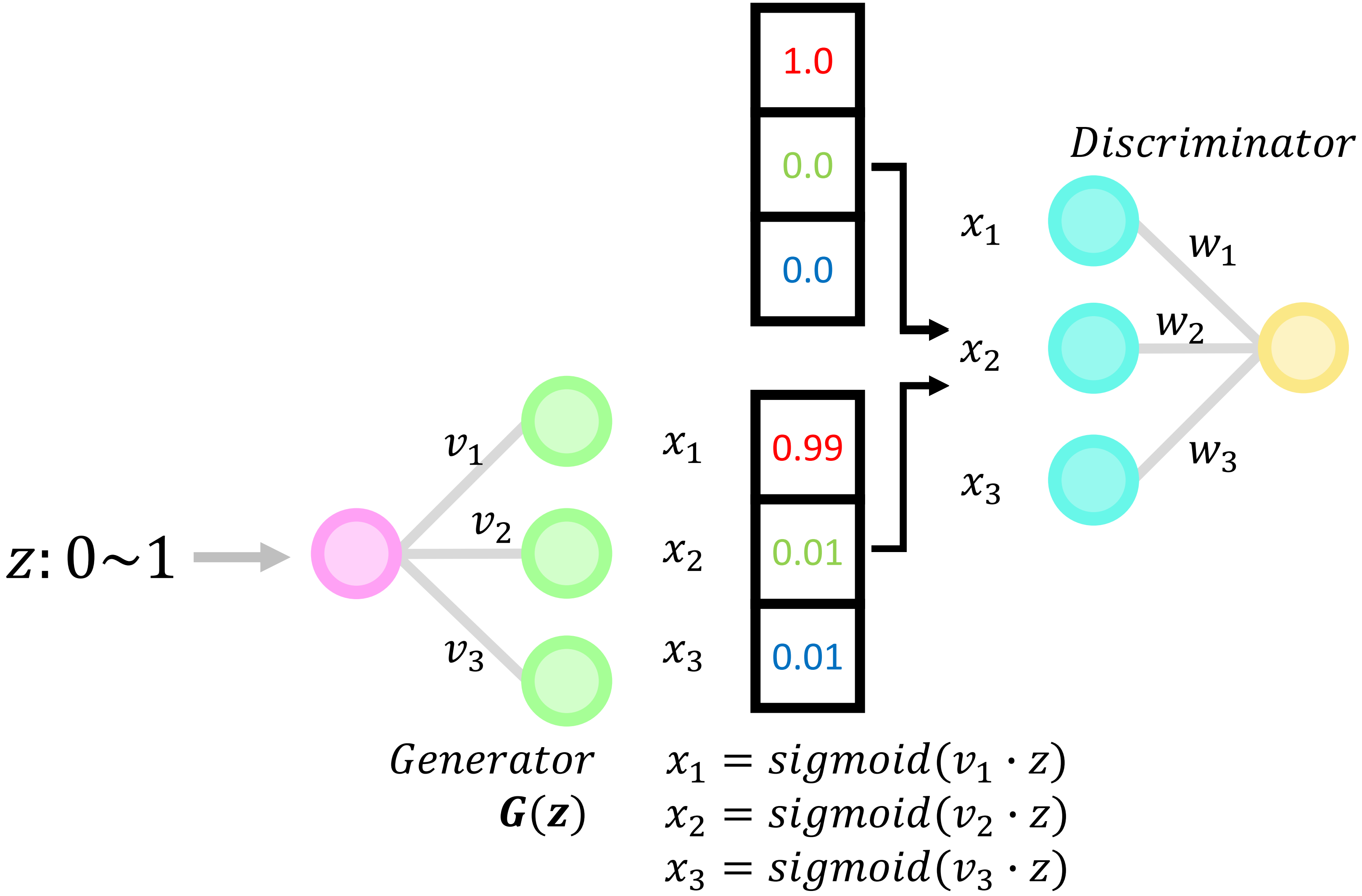
학습을 통해 generator가 진짜 빨간색에 가까운rgb값을 출력하도록 만드는 것이 학습의 목표라고 정의를 내려보겠습니다



$$x_1 = sigmoid(v_1 \cdot z)$$
$$x_2 = sigmoid(v_2 \cdot z)$$
$$x_3 = sigmoid(v_3 \cdot z)$$

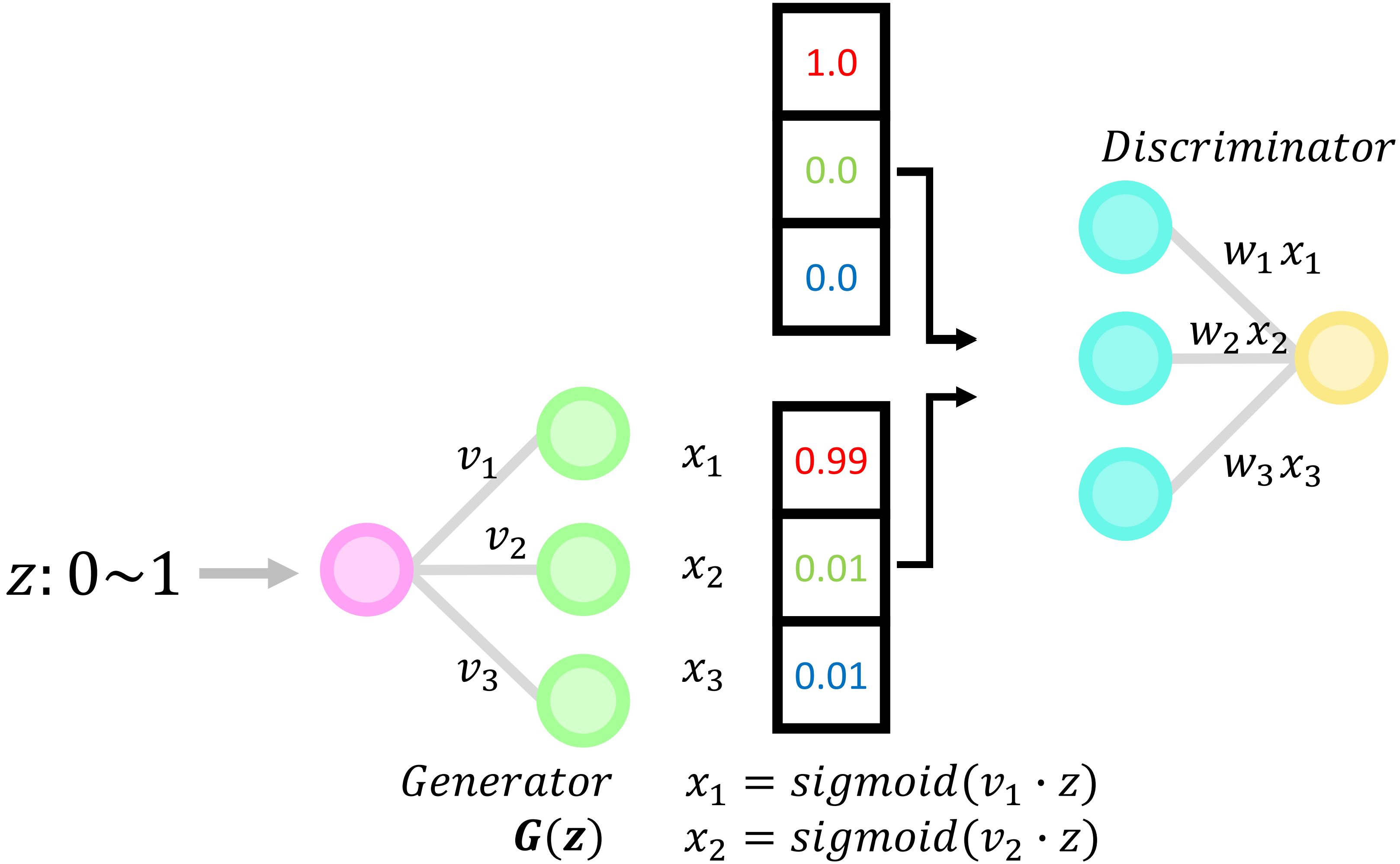
신박AI

그래서 실제 데이터 (pure red)든, 생성 데이터 (fake red)든 랜덤하게 discriminator에 입력으로 넣고,



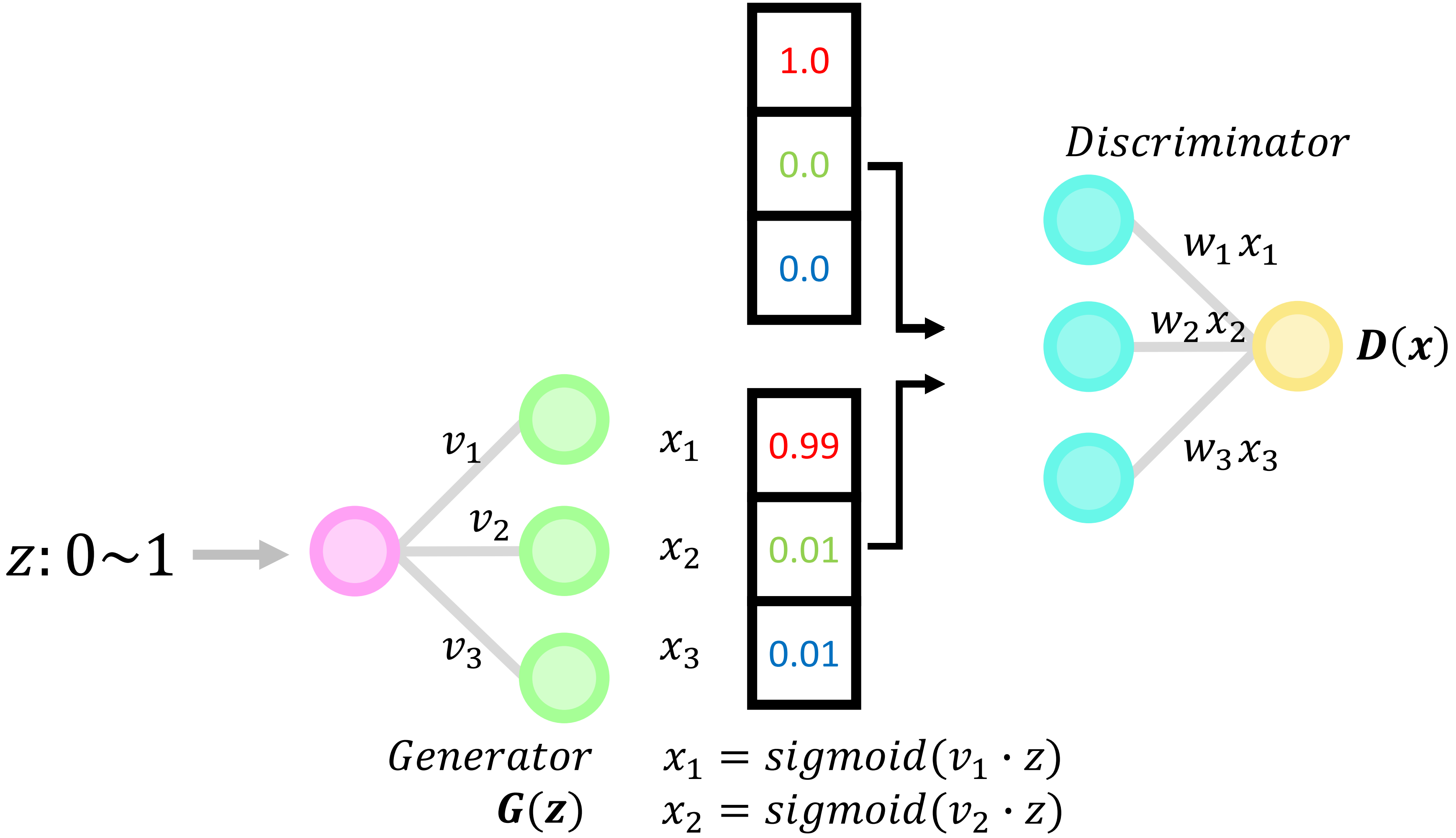
신박AI

그래서 실제 데이터 (pure red)든, 생성 데이터 (fake red)든 랜덤하게 discriminator에 입력으로 넣고,



$$x_1 = sigmoid(v_1 \cdot z)$$
$$x_2 = sigmoid(v_2 \cdot z)$$
$$x_3 = sigmoid(v_3 \cdot z)$$

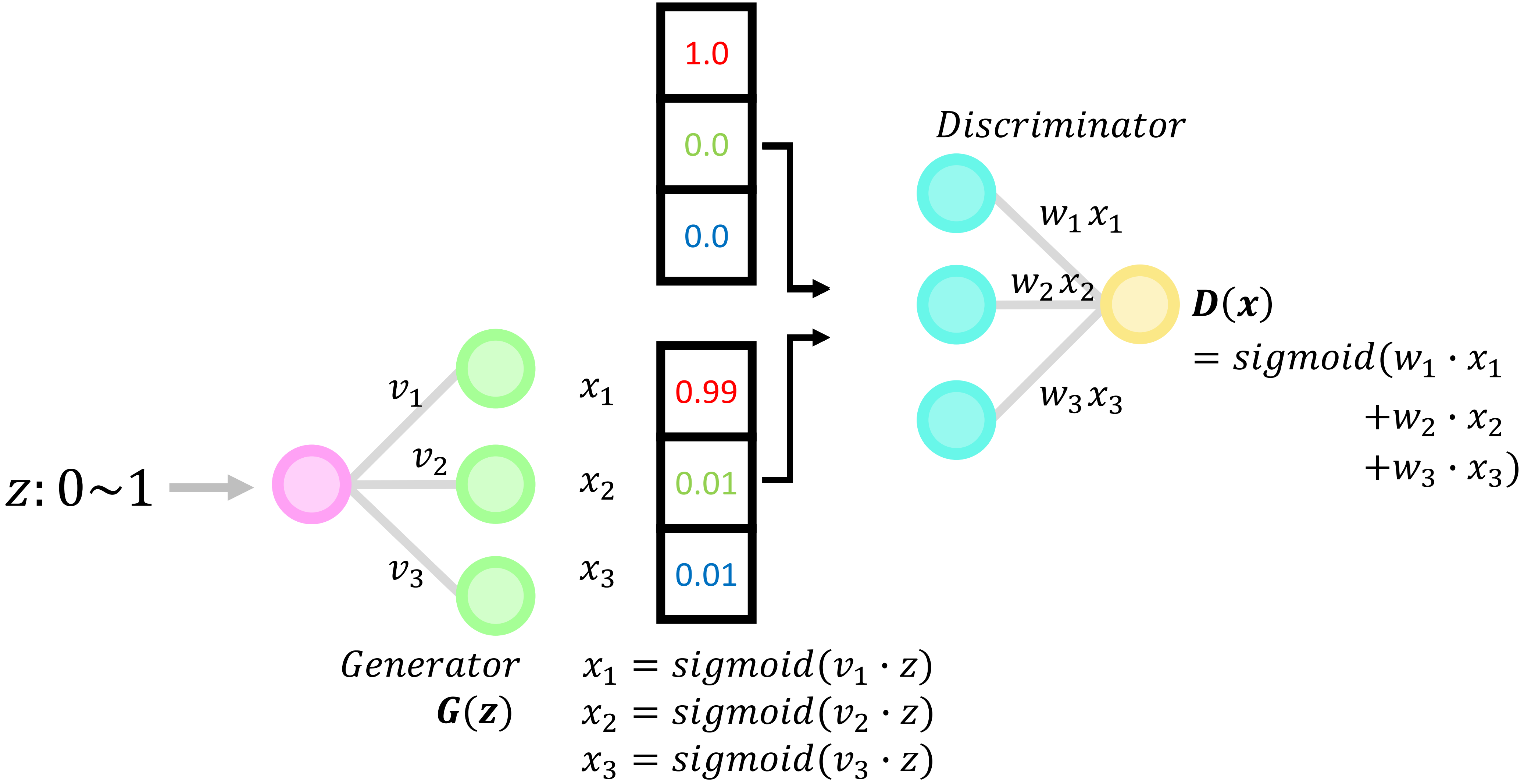
Discriminator는 들어온 입력값이 실제 데이터인지 아니면 generator가 만든 데이터인지를,



$$x_1 = sigmoid(v_1 \cdot z)$$
$$x_2 = sigmoid(v_2 \cdot z)$$
$$x_3 = sigmoid(v_3 \cdot z)$$

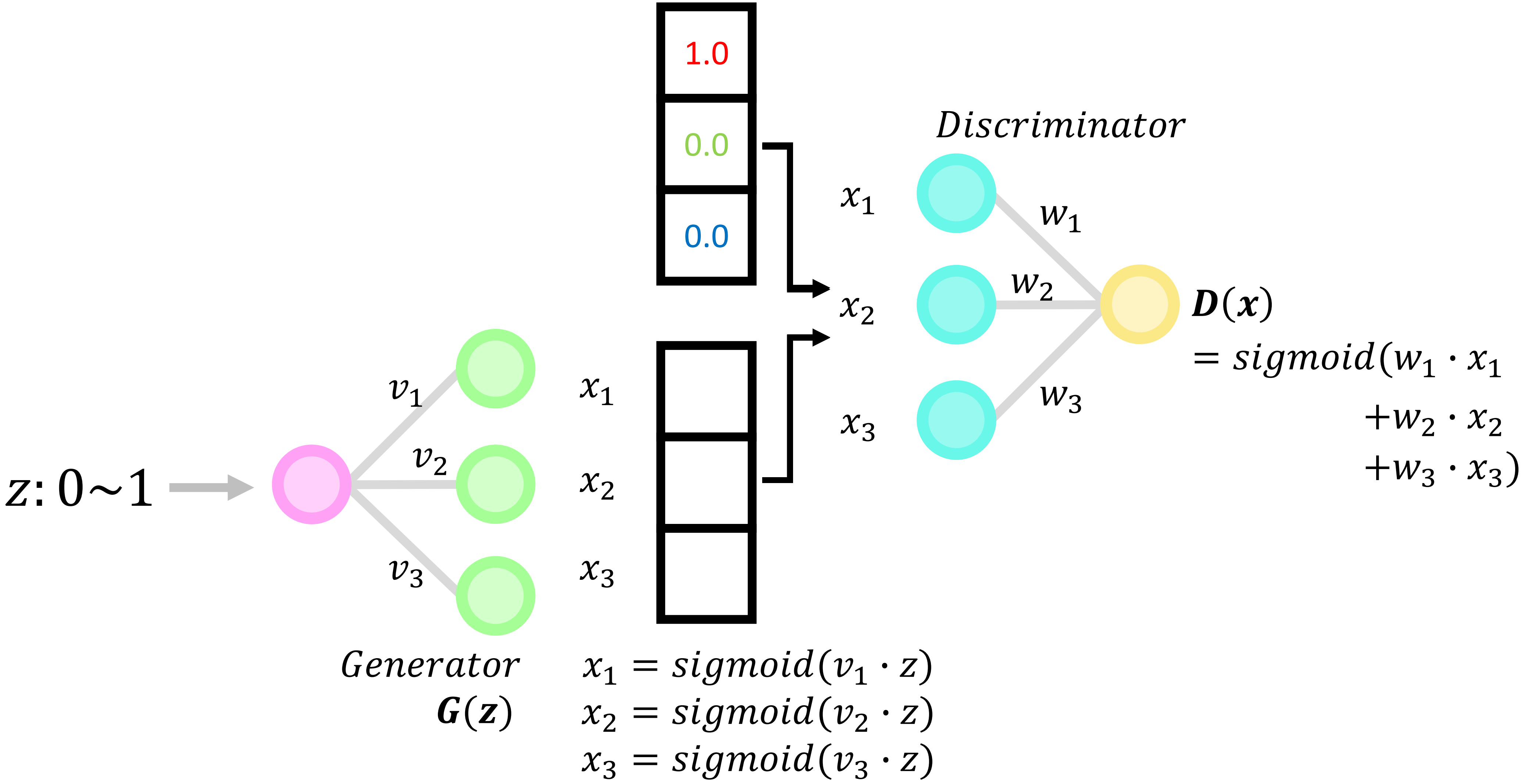
신박AI

다음 계산과 시그모이드 함수를 통해 0과 1 사이의 값으로 표현하게 됩니다



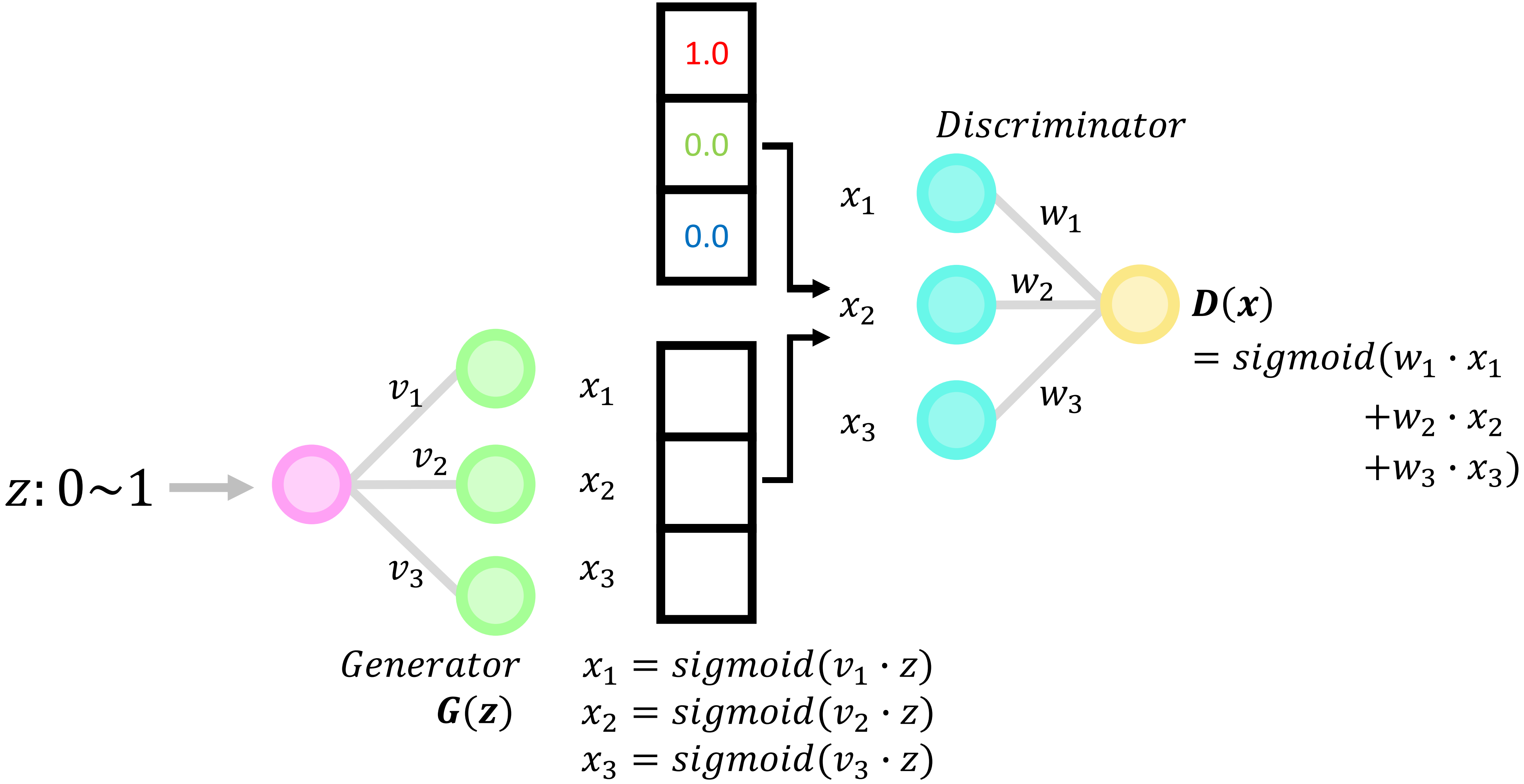
신박AI

이렇게 아주 간단한 형태의 GAN모델을 만들어 보았습니다

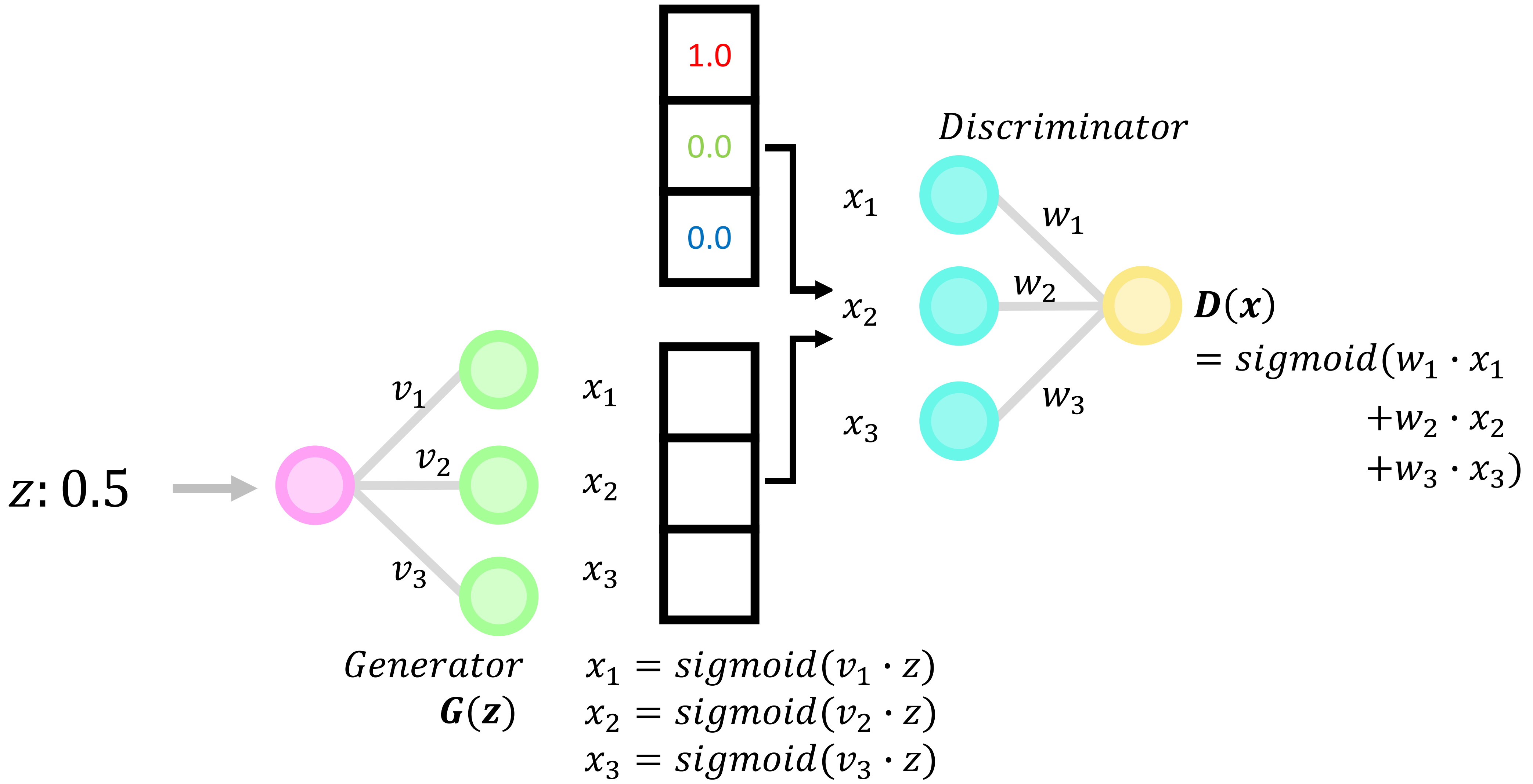


신박AI

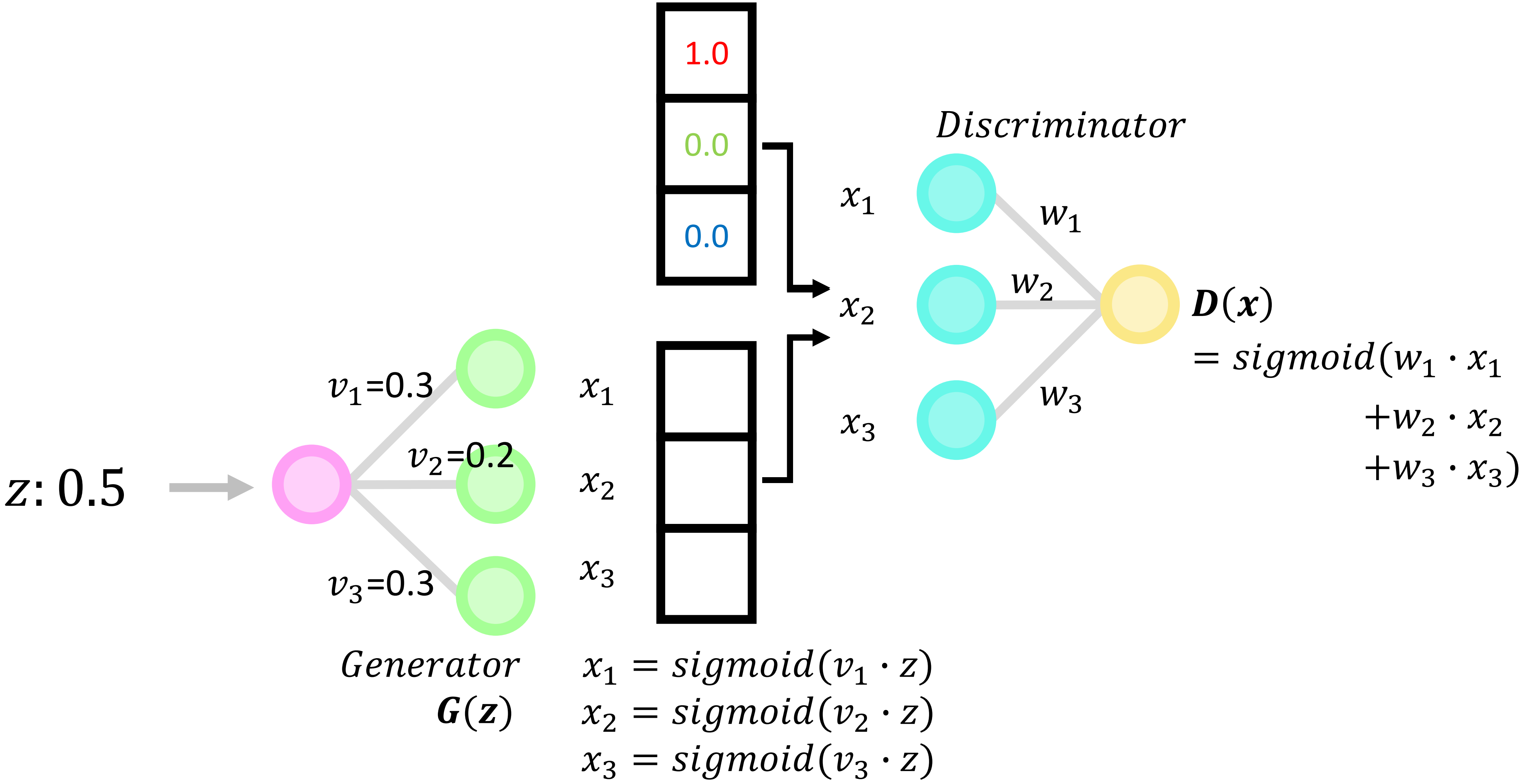
그러면 순전파부터 한번 계산해보도록 하겠습니다.



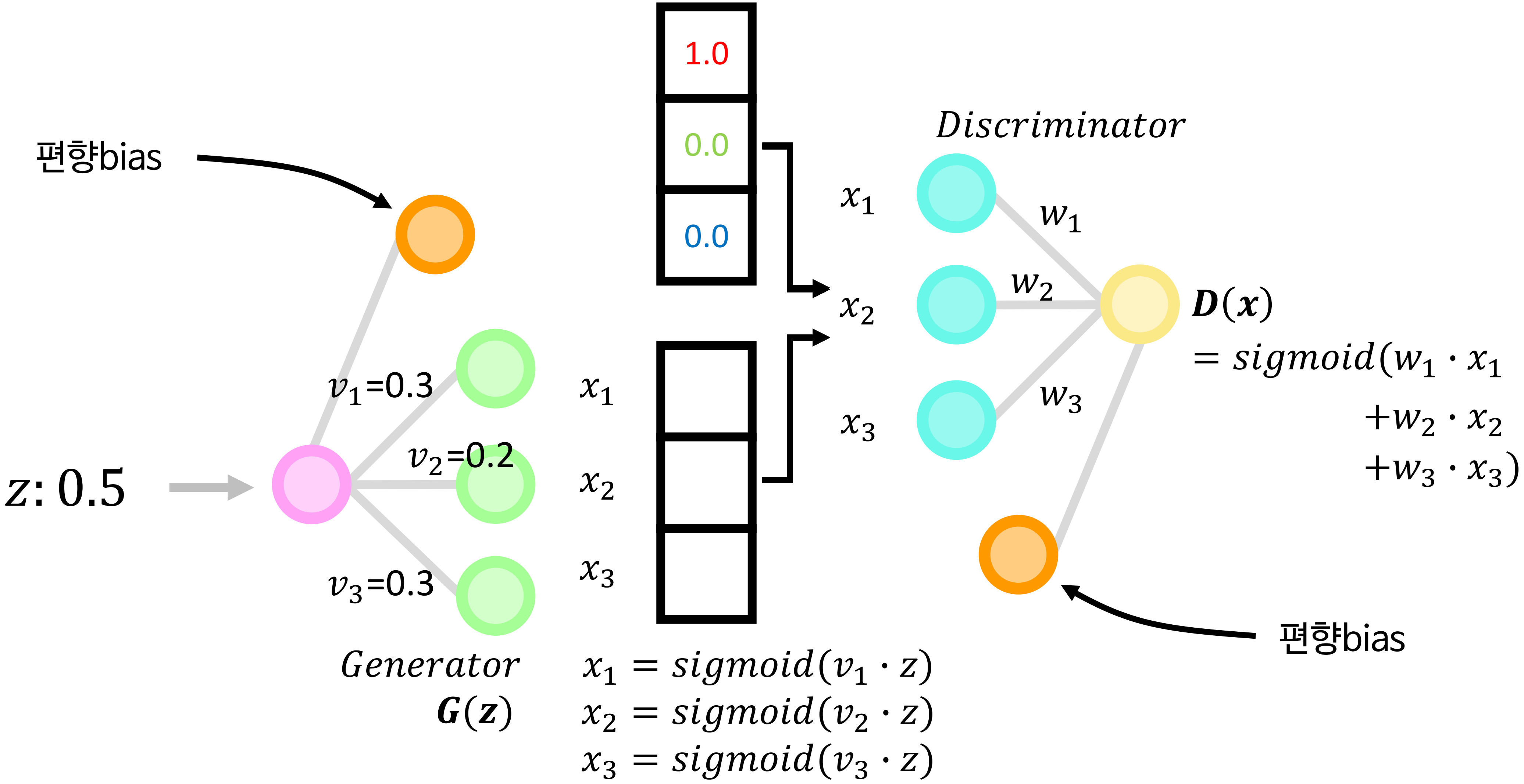
우선 z값은 0.5로 하도록 하겠습니다.



그리고 generator의 가중치 값은 다음과 같이 랜덤하게 정하도록 하겠습니다.

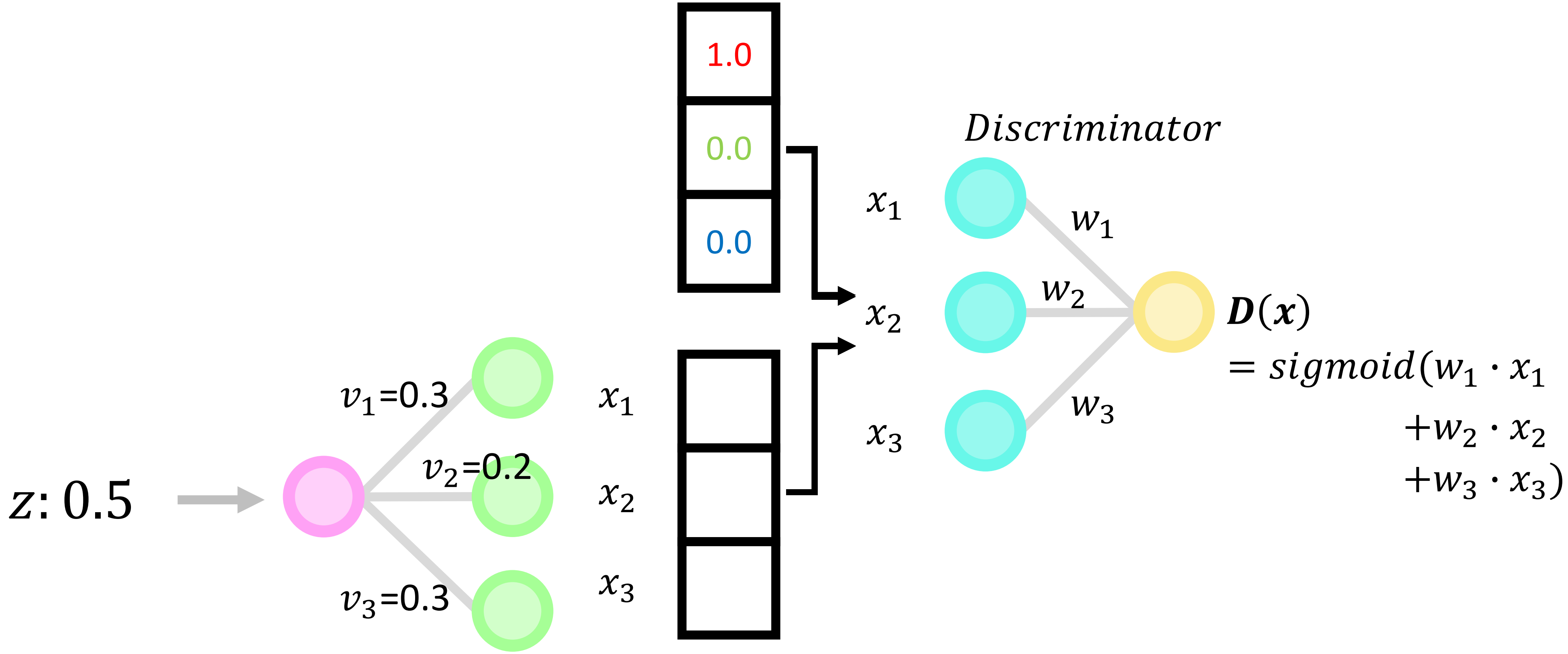


아 물론 여기서도 편향을 넣어주어야 완전한 generator와 discriminator가 되지만, 간단한 계산을 위해 생략하도록 하겠습니다.



신박AI

그러면 generator의 순전파는

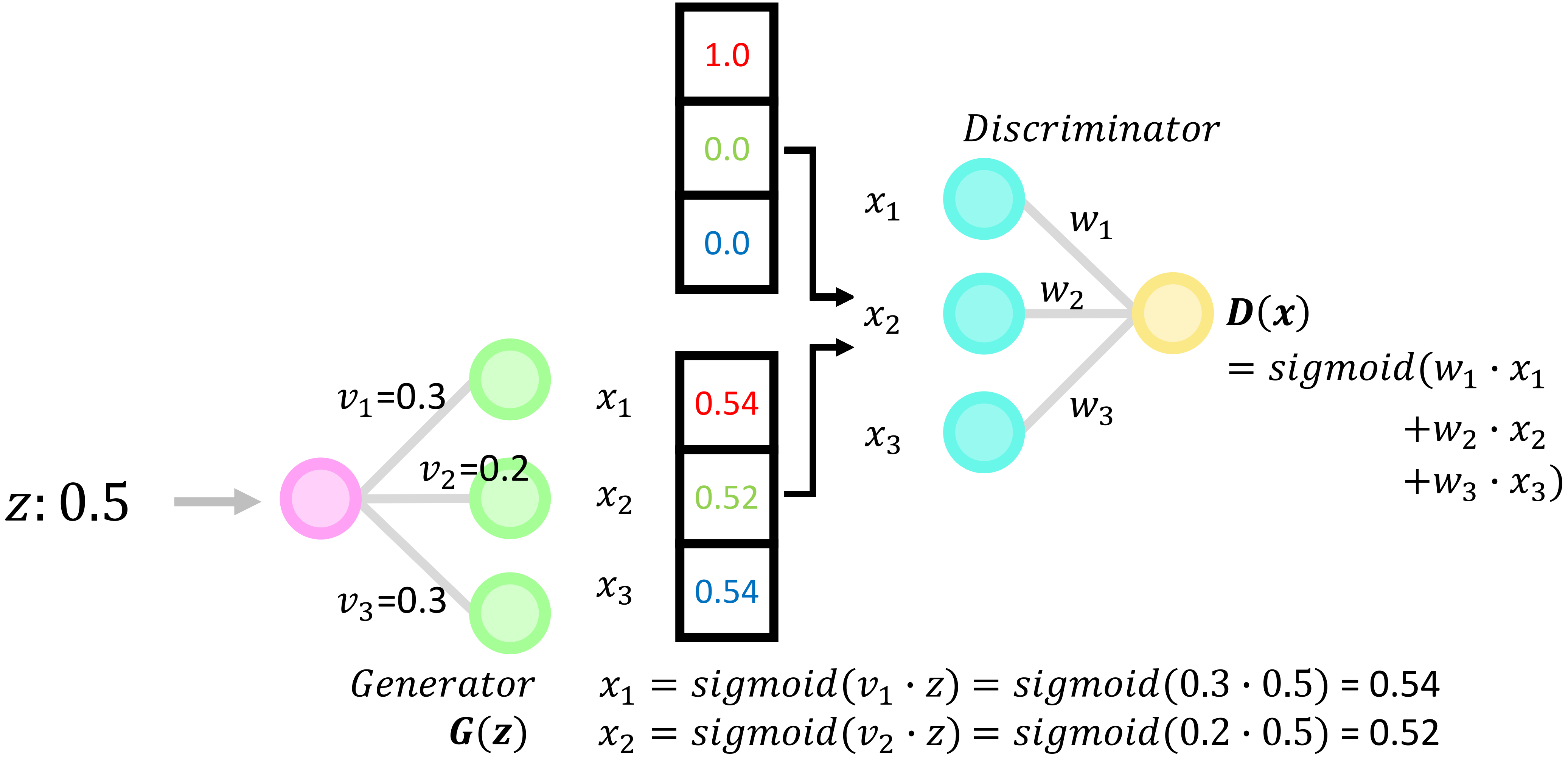


Generator
$\boldsymbol{G}(\boldsymbol{z})$

$$x_1 = sigmoid(v_1 \cdot z)$$
$$x_2 = sigmoid(v_2 \cdot z)$$
$$x_3 = sigmoid(v_3 \cdot z)$$

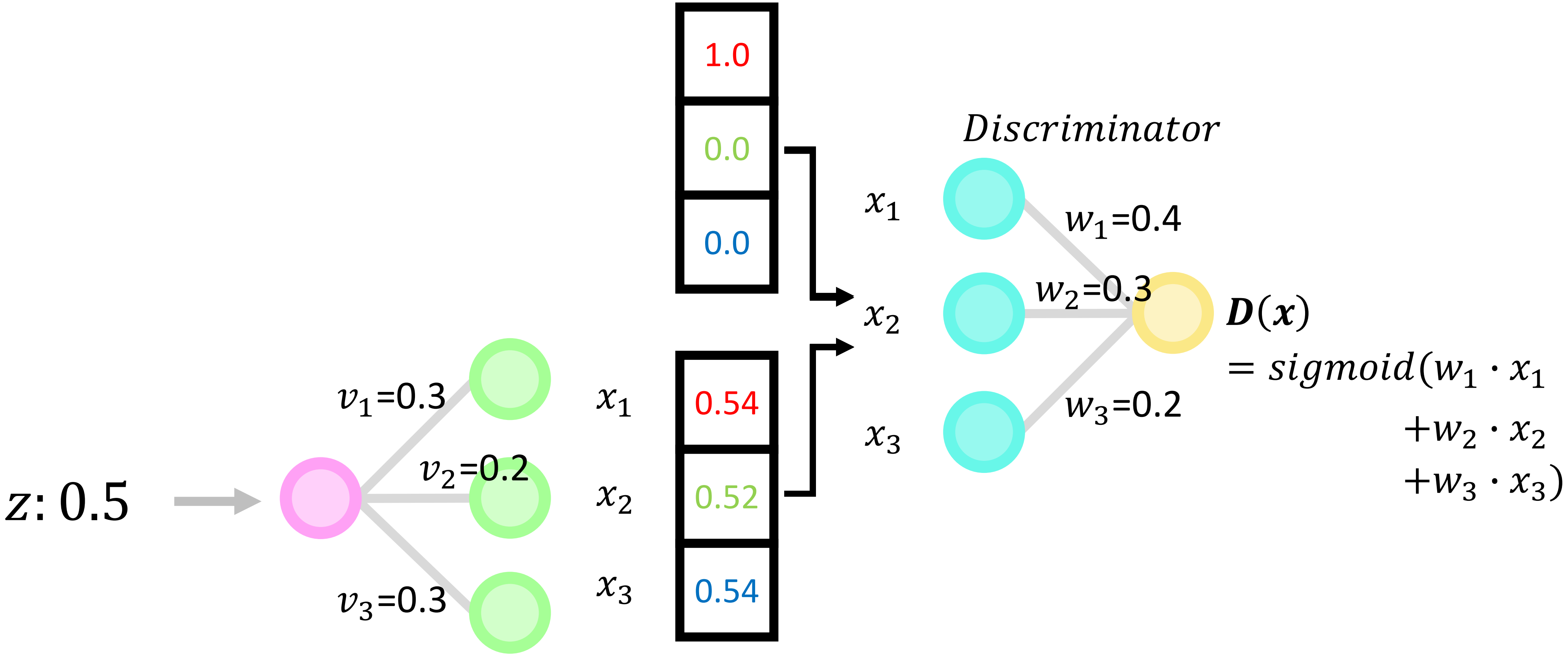
신박AI

다음과 같이 계산이 됩니다.



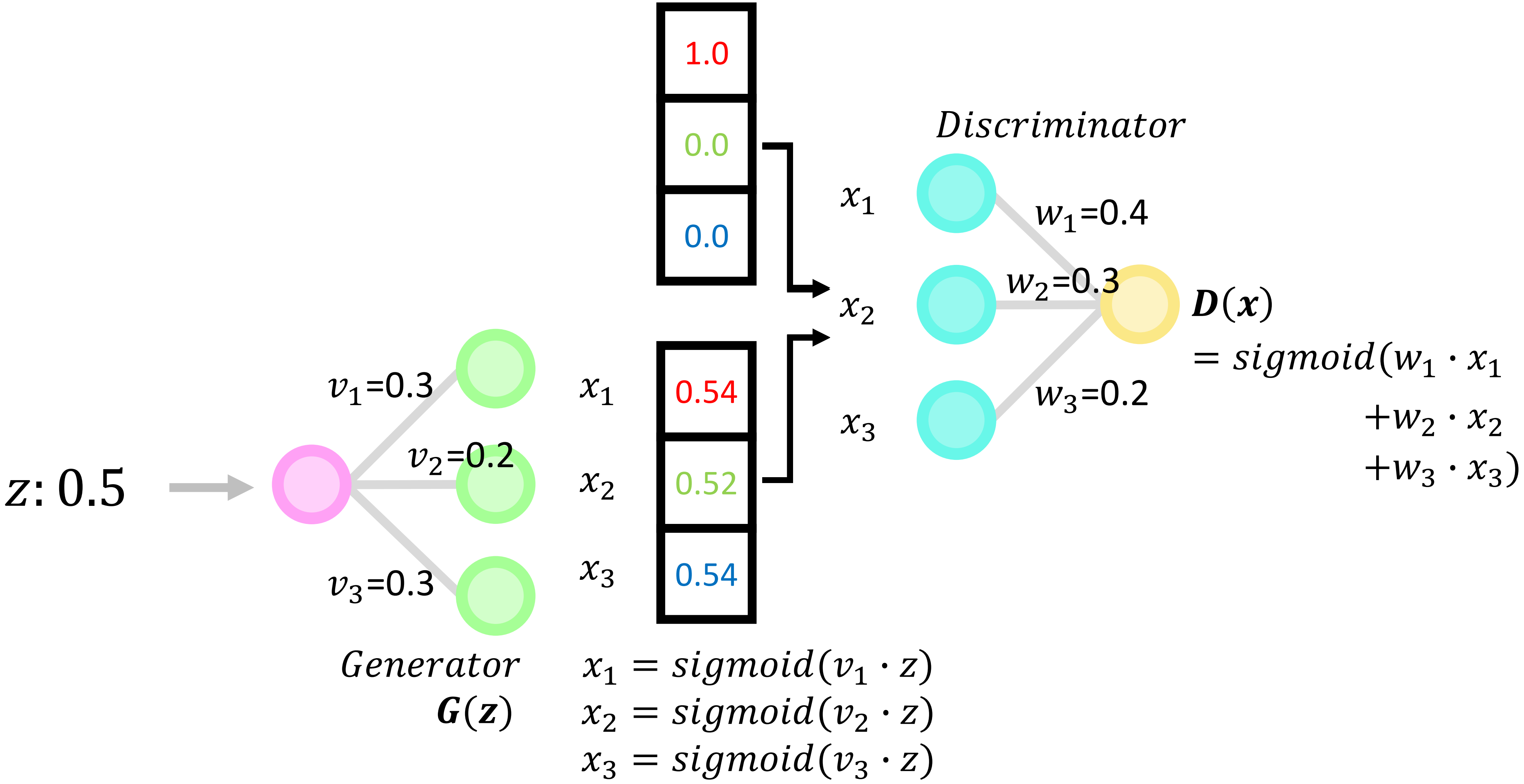
$x_1 = sigmoid(v_1 \cdot z) = sigmoid(0.3 \cdot 0.5) = 0.54$
$x_2 = sigmoid(v_2 \cdot z) = sigmoid(0.2 \cdot 0.5) = 0.52$
$x_3 = sigmoid(v_3 \cdot z) = sigmoid(0.3 \cdot 0.5) = 0.54$

그리고 generator의 경우와 마찬가지로, discriminator의 가중치도 랜덤으로 설정하도록 하겠습니다

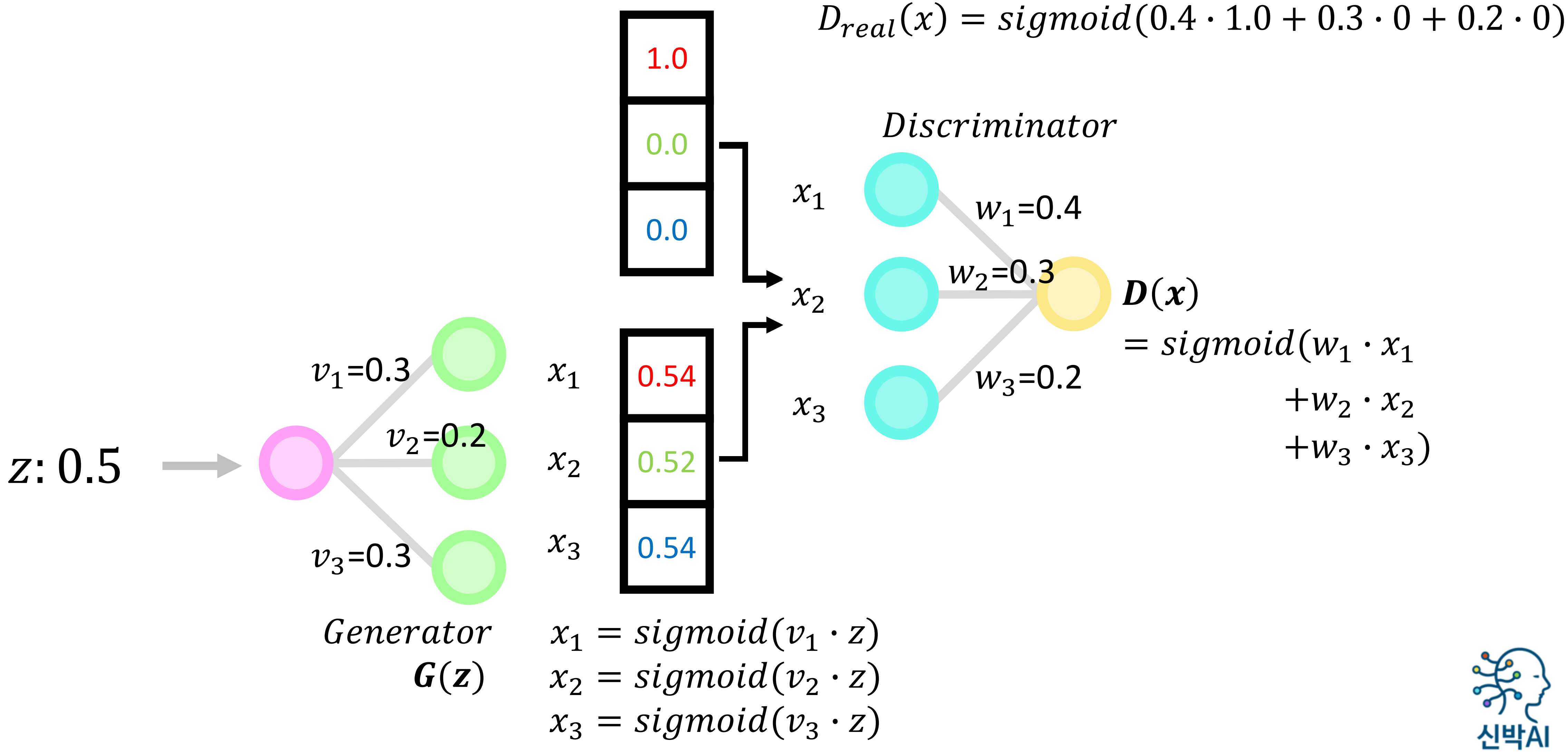


$$x_1 = sigmoid(v_1 \cdot z)$$
$$x_2 = sigmoid(v_2 \cdot z)$$
$$x_3 = sigmoid(v_3 \cdot z)$$

그럴경우, 만약 진짜 데이터인 (1.0, 0.0, 0.0)이 discriminator에 들어갈 경우 아웃풋은,

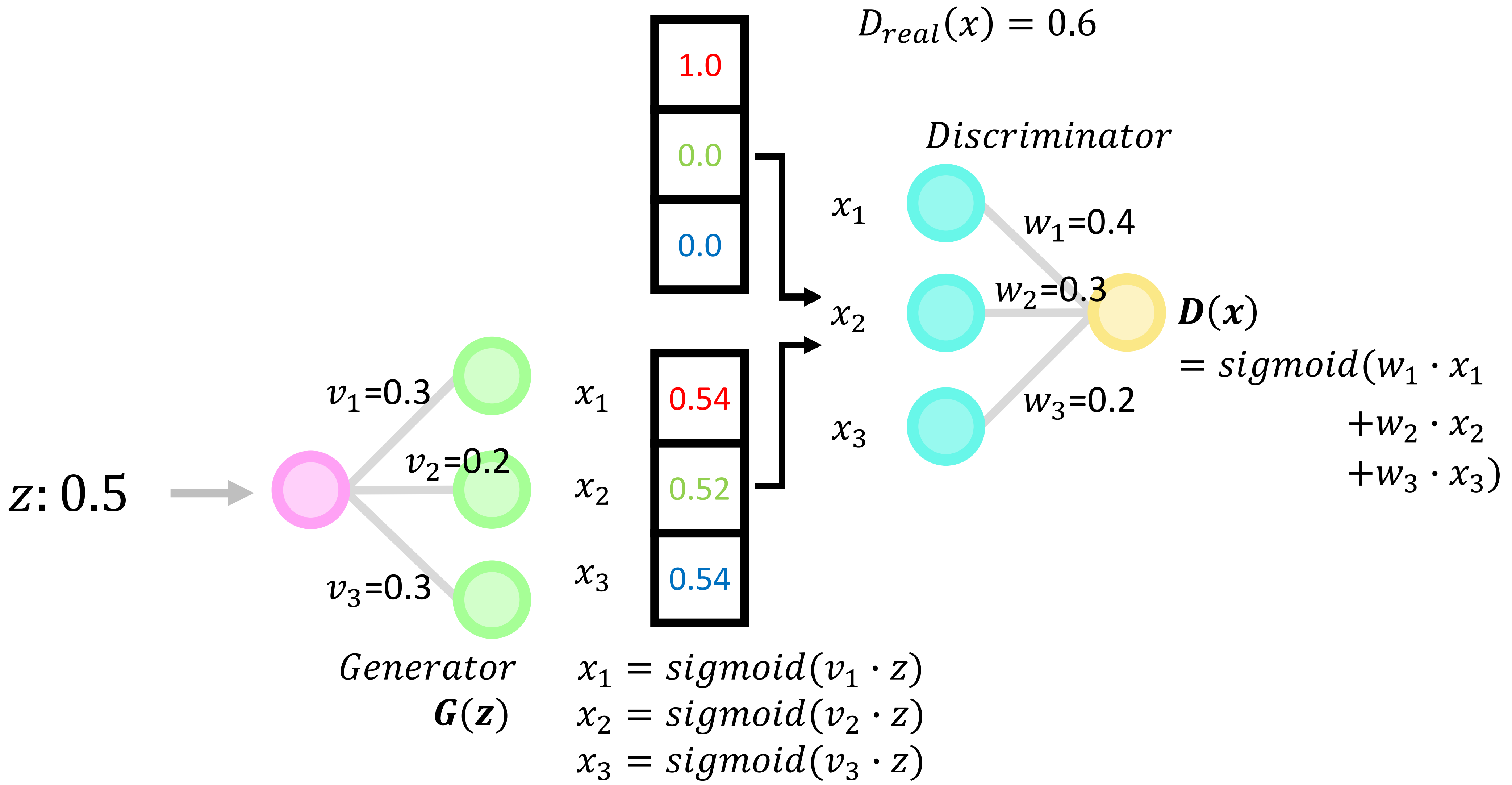


이렇게 계산할 수 있고,



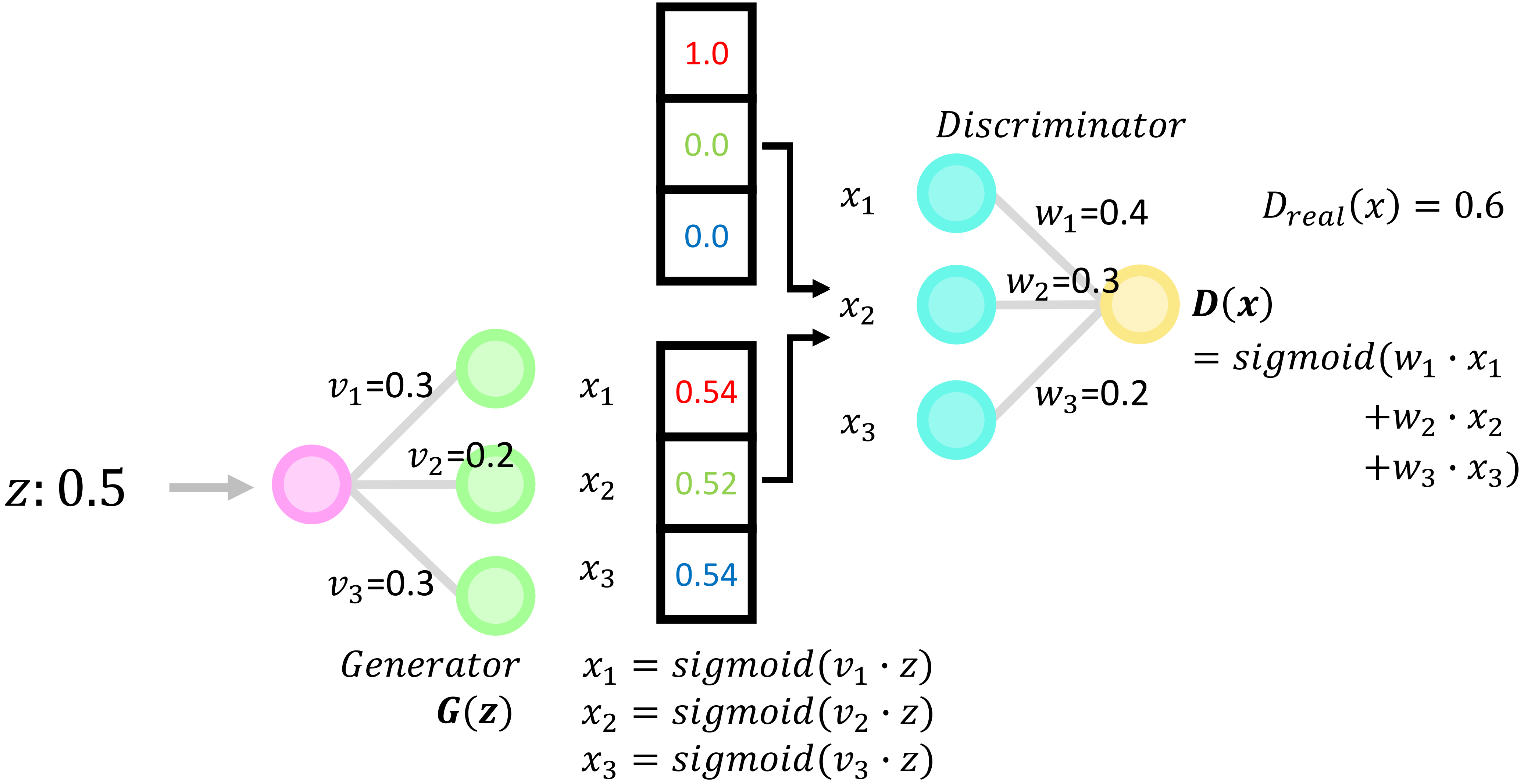
신박AI

0.6이 나옵니다.



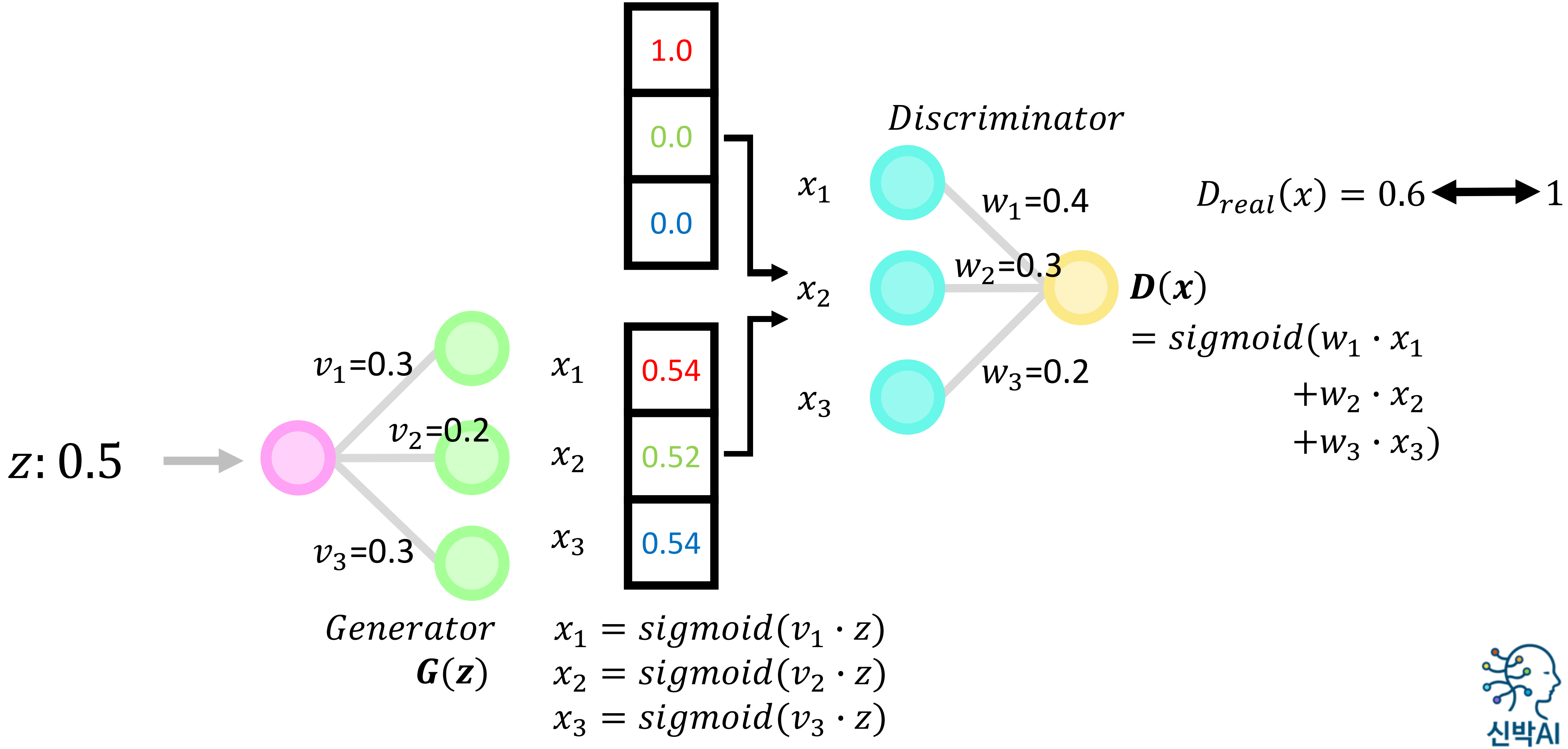
신박AI

0.6이 나옵니다.



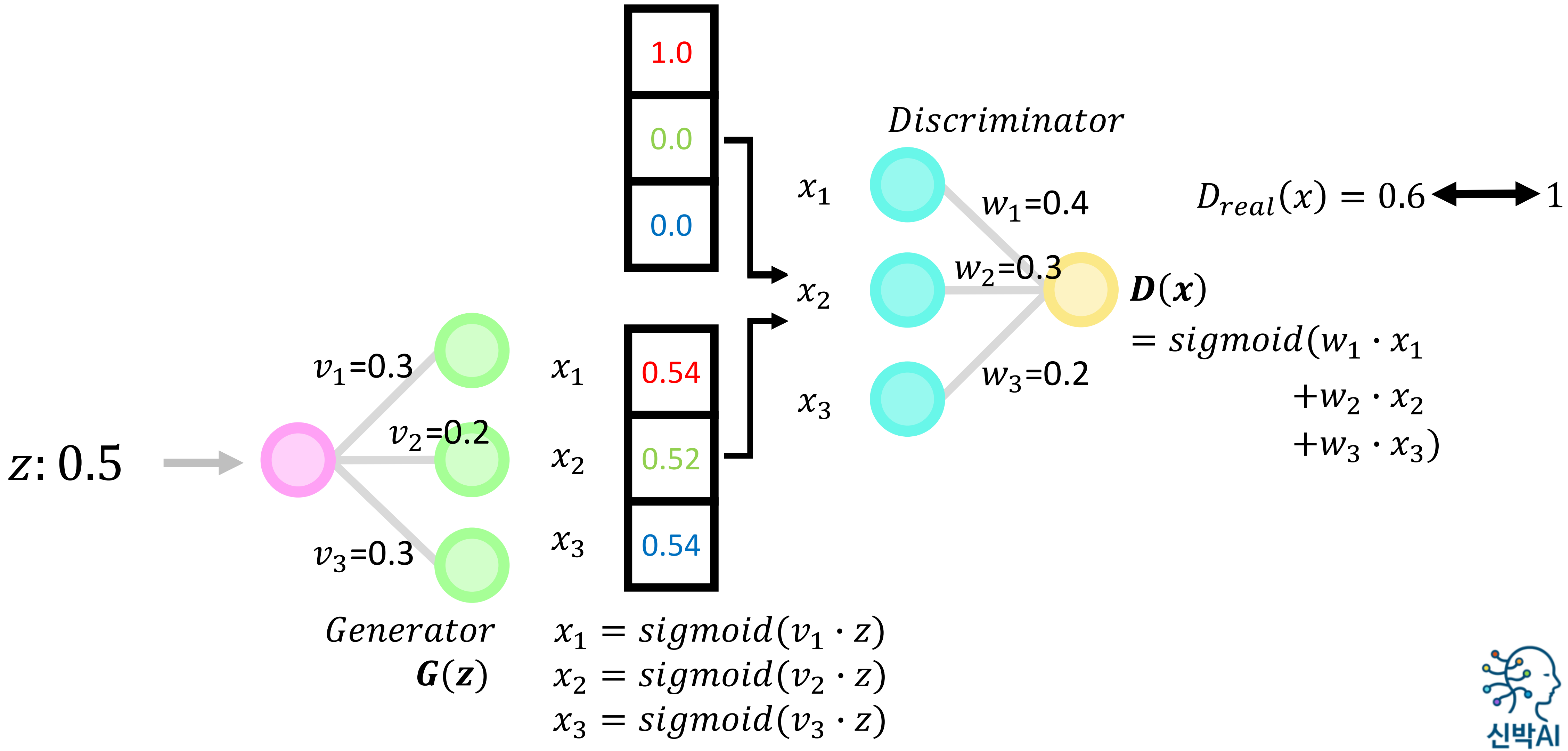
신박AI

원래 진짜 데이터에 대한 실제값은 1이어야 하기 때문에 이만큼의 손실 loss가 발생하게 됩니다



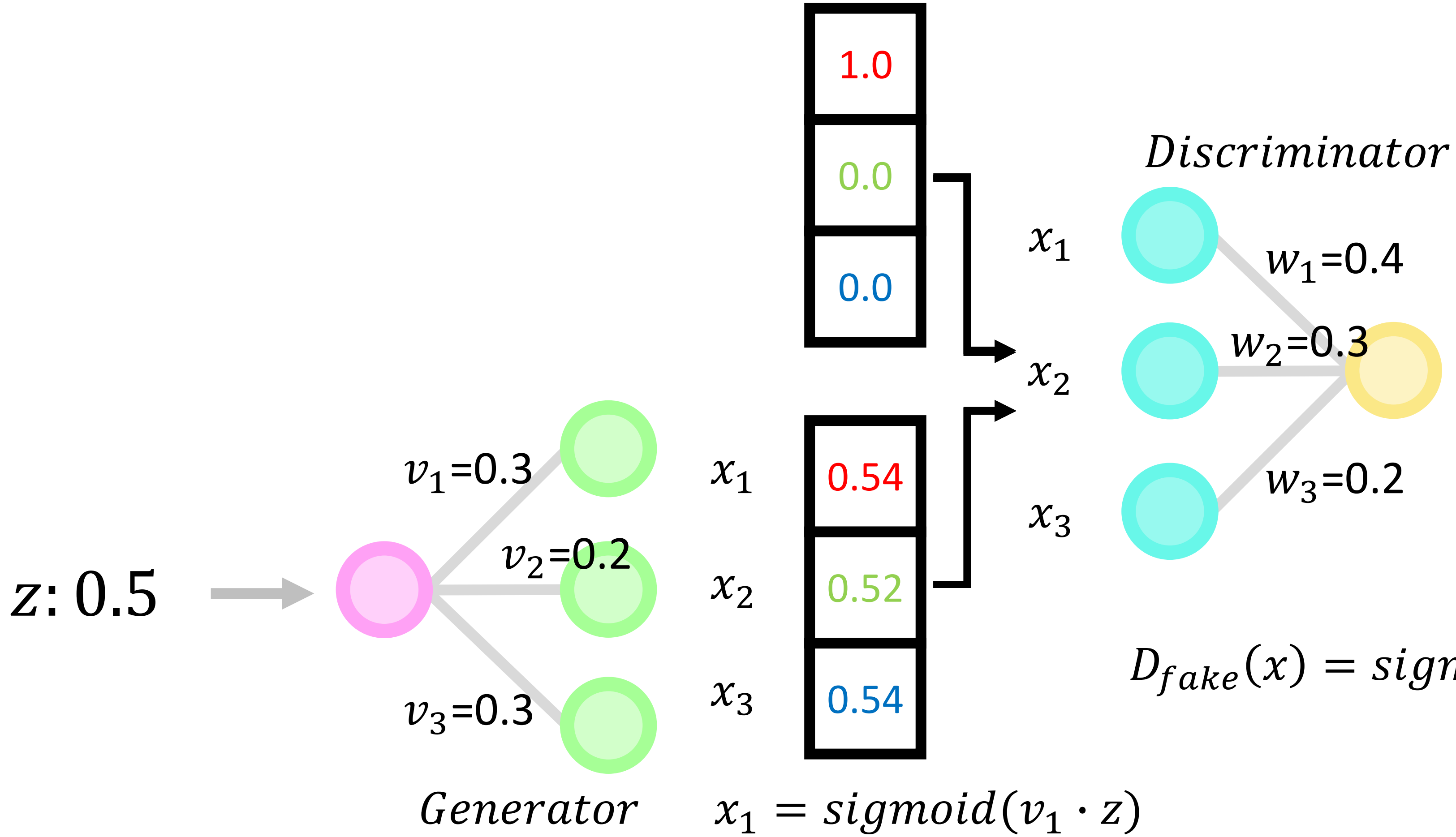
신박AI

마찬가지로 만약 가짜 데이터, 즉 생성 데이터인 (0.54,0.52,0.54)가 들어간다고 한다면,



신박AI

다음과 같이 대입하여 계산하면..



$D_{real}(x) = 0.6 \longleftrightarrow 1$

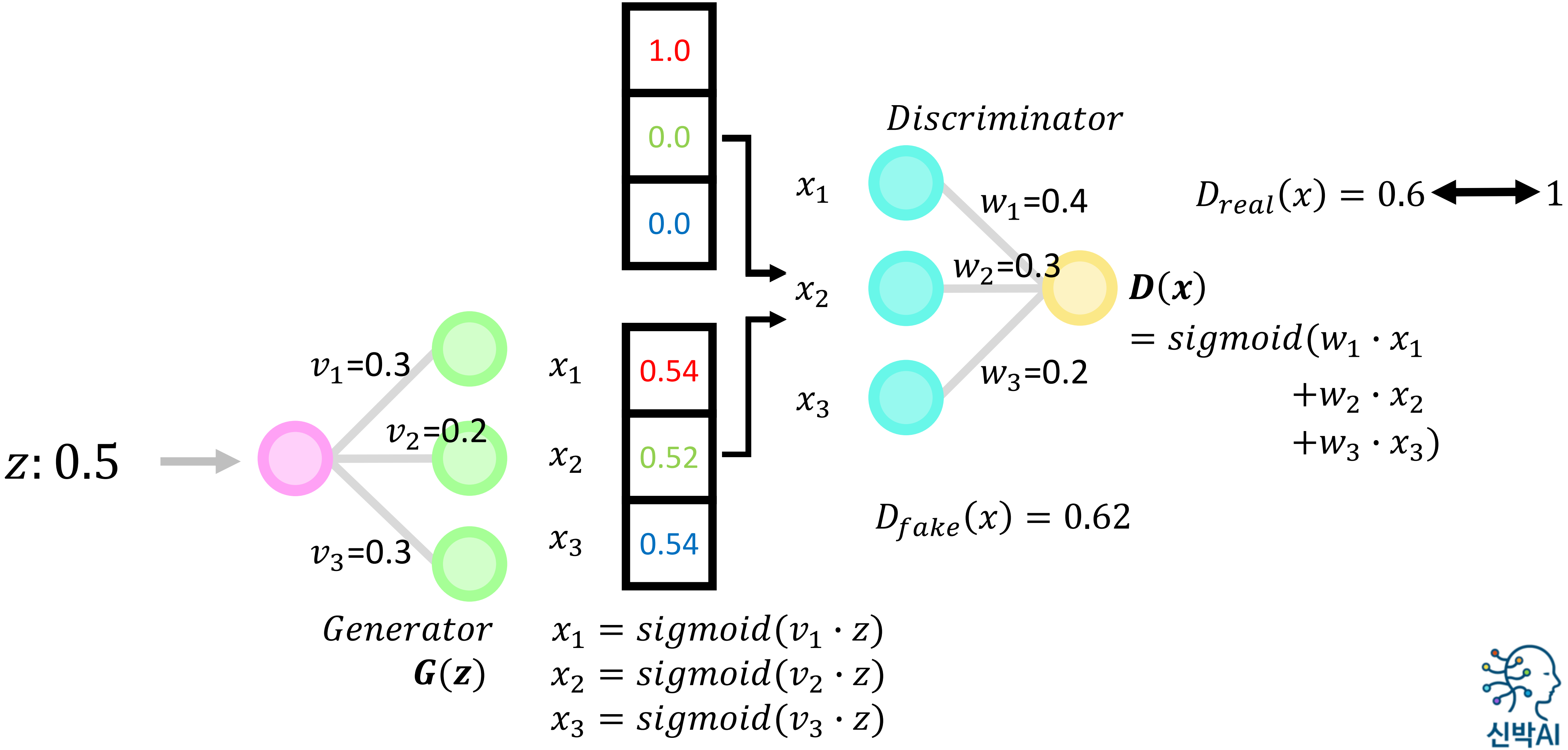
$$\boldsymbol{D(x)} = sigmoid(w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3)$$

$$D_{fake}(x) = sigmoid(0.4 \cdot 0.54 + 0.3 \cdot 0.52 + 0.2 \cdot 0.54)$$

$$x_1 = sigmoid(v_1 \cdot z)$$
$$x_2 = sigmoid(v_2 \cdot z)$$
$$x_3 = sigmoid(v_3 \cdot z)$$

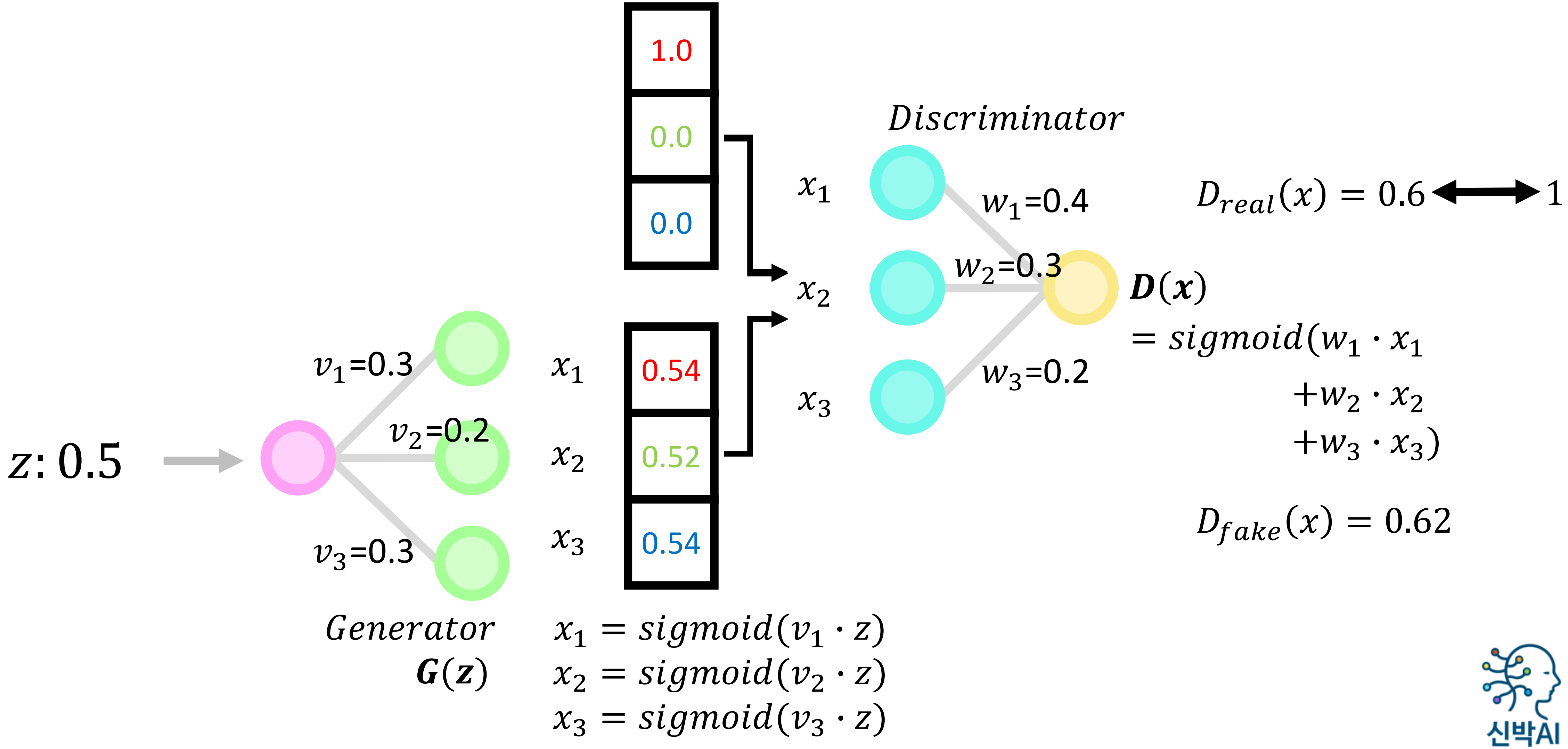
신박AI

이렇게 계산할 수 있습니다.



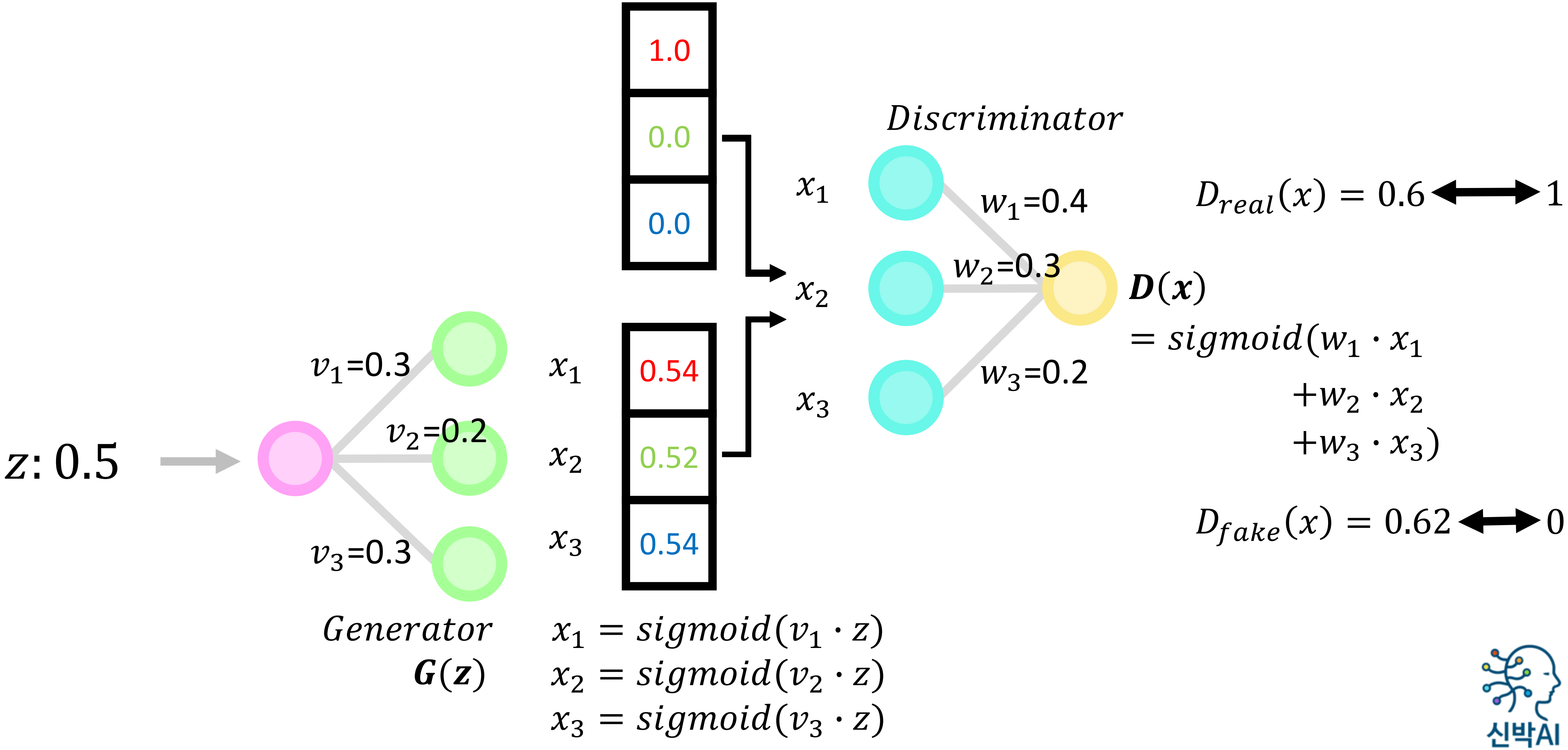
신박AI

이렇게 계산할 수 있습니다.



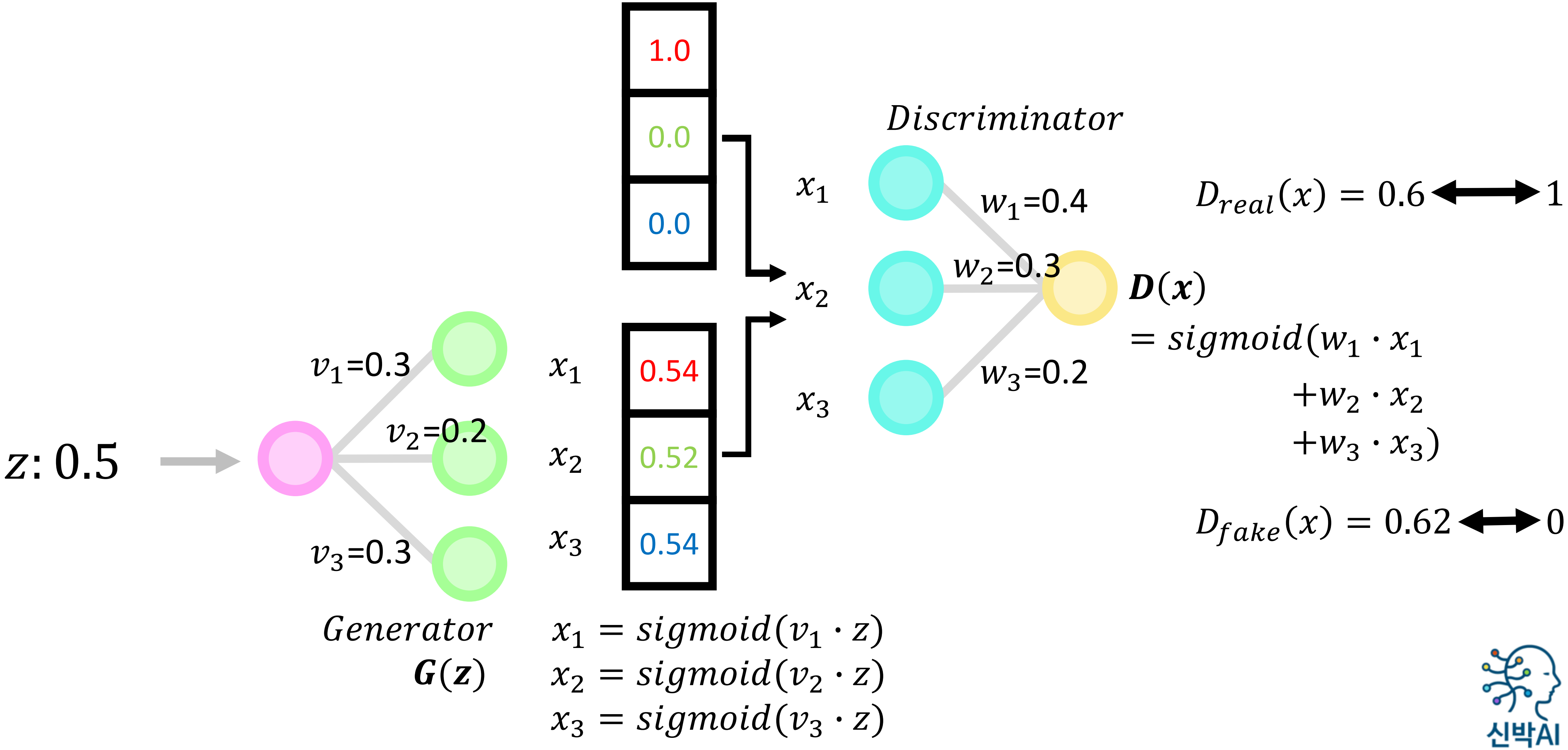
신박AI

가짜 데이터에 대한 실제값은 0이어야 하기 때문에 이만큼의 손실 loss가 발생하게 됩니다



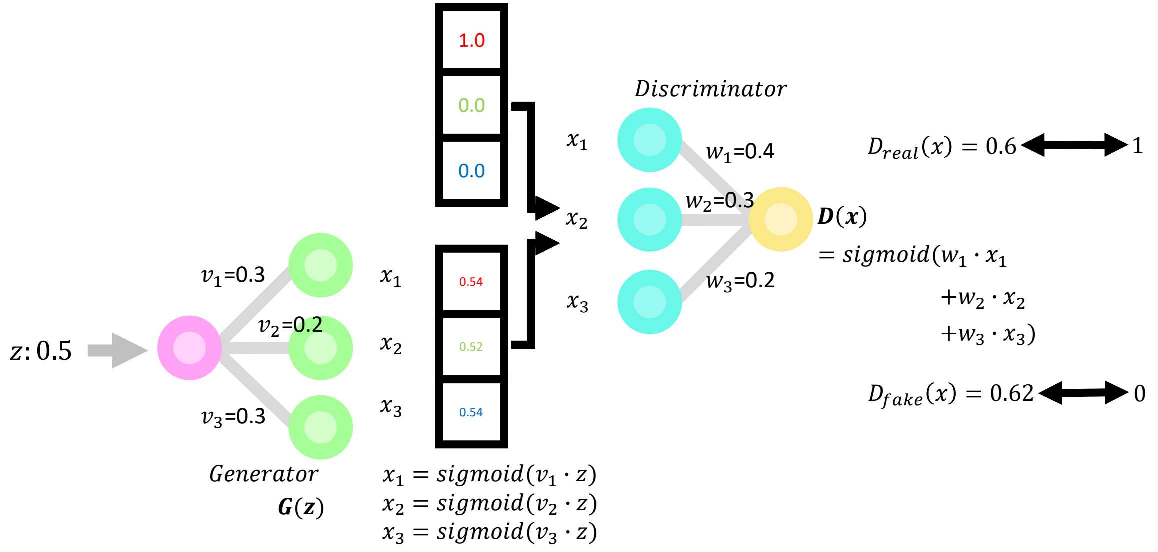
신박AI

이제부터는 손실함수부터 시작된 오류가 어떻게 역전파 되는지 알아볼 차례입니다

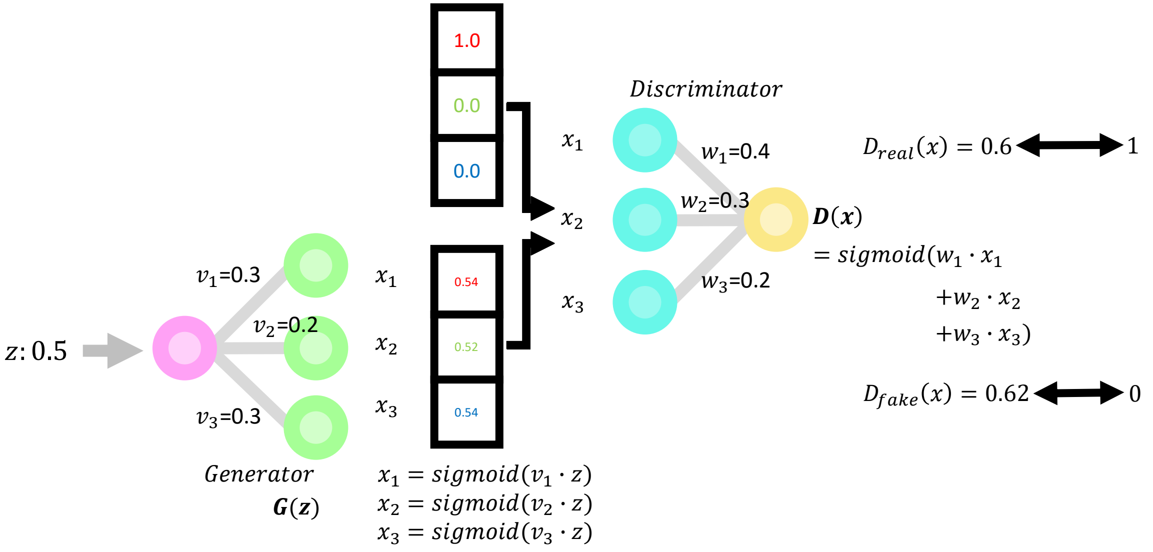


신박AI

이제부터는 손실함수부터 시작된 오류가 어떻게 역전파 되는지 알아볼 차례입니다

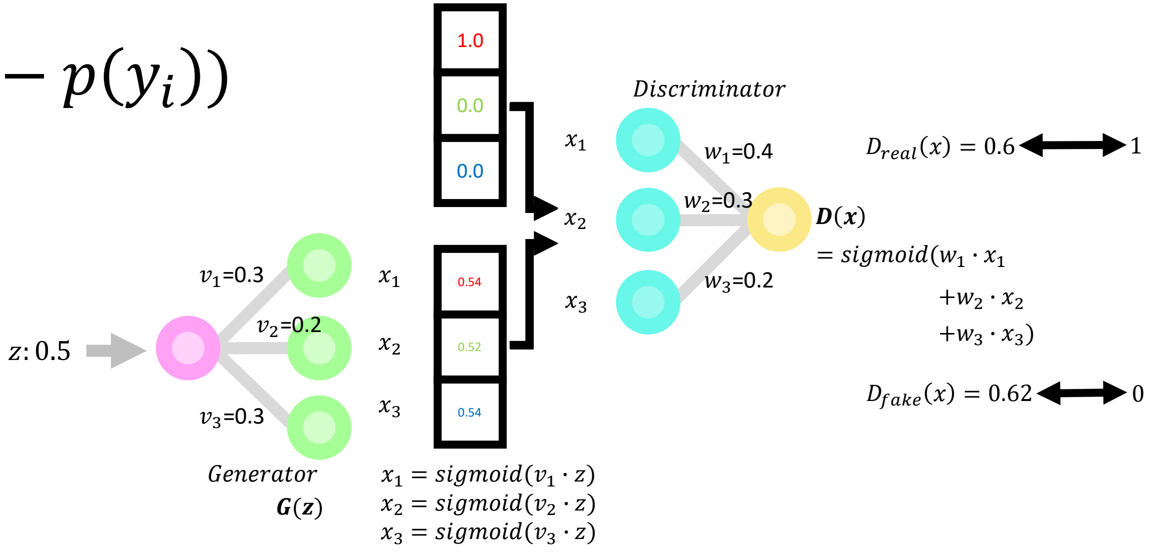


이제부터는 손실함수부터 시작된 오류가 어떻게 역전파 되는지 알아볼 차례입니다



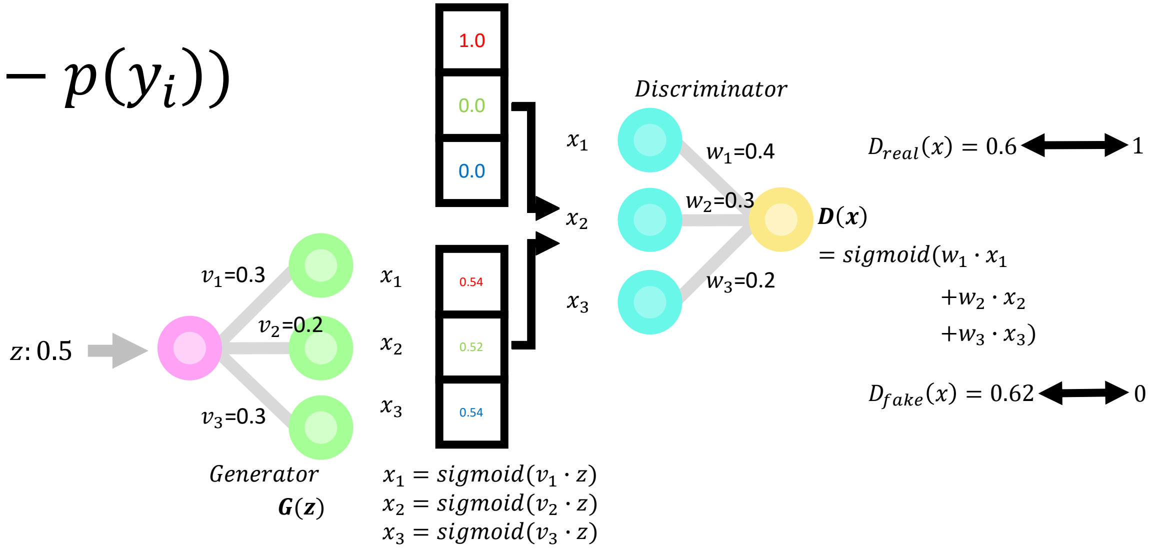
GAN모델의 손실함수는 이진 크로스 엔트로피 binary cross-entropy (BCE)함수를 기반으로 하고 있습니다

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$



사실상 GAN모델은 주어진 이미지가 진짜 데이터(1)냐 가짜 (0)냐 둘 중 하나만 판별해야 하기 때문에 이진binary 형태의 손실함수가 합당하고,

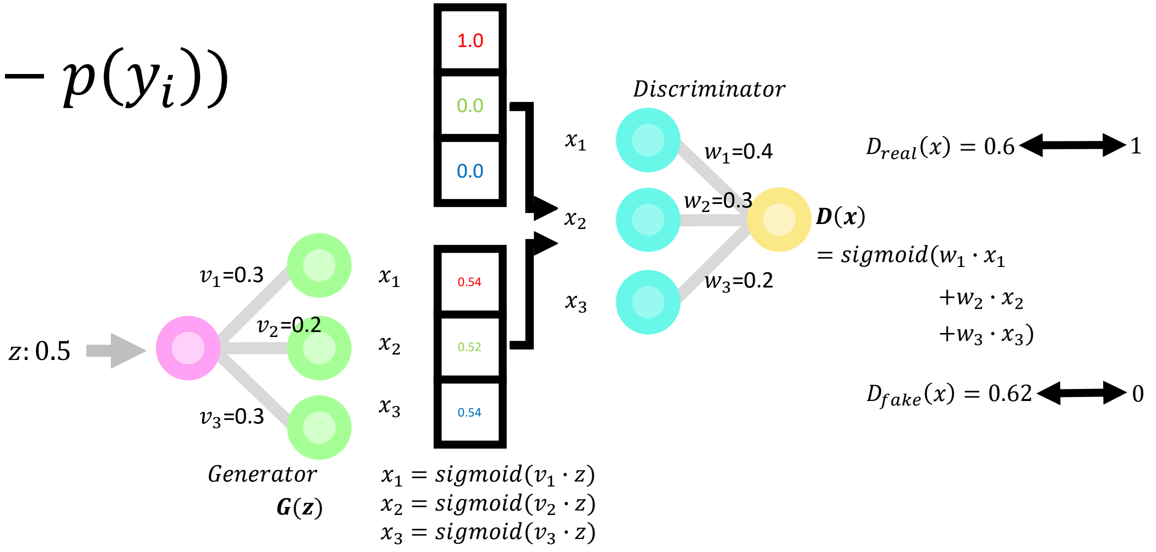
$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$



신박AI

또 지난 영상에서 본 바 처럼, 크로스 엔트로피 손실함수가 MSE에 비해 빨리 정답쪽으로 수렴하는 경향이 크기 때문에,

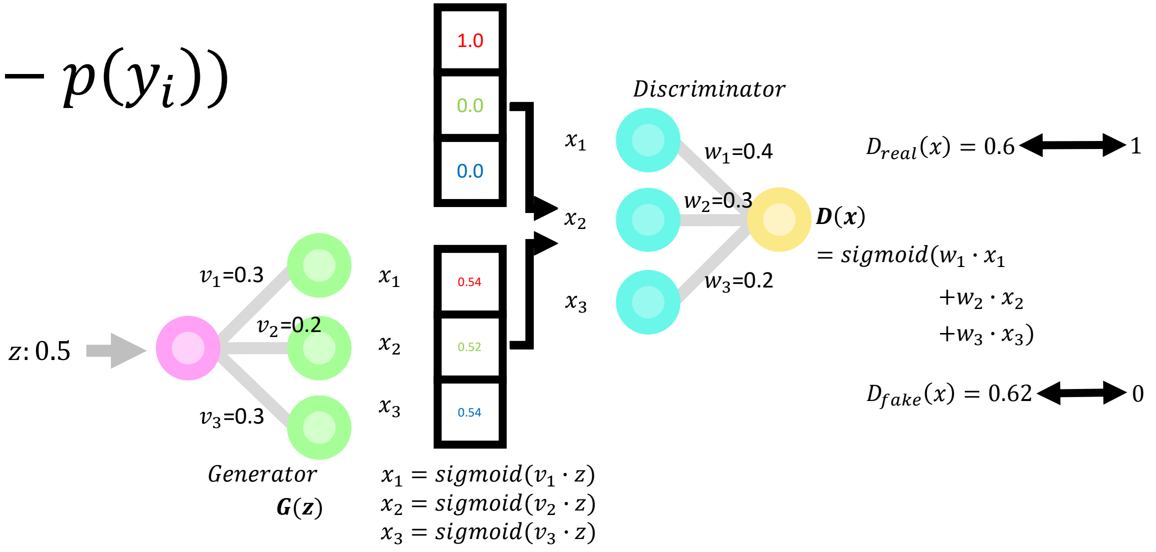
$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot \log(1-p(y_i))$$



신박AI

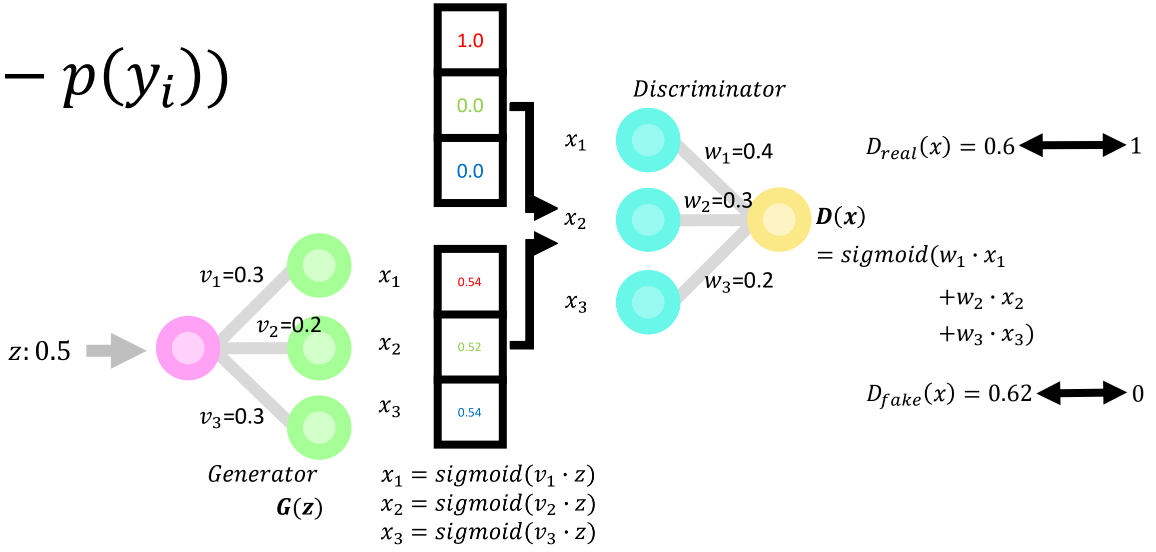
이진 크로스 엔트로피가 GAN모델의 손실함수로 적당해 보입니다.

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$



우선 discriminator의 손실함수부터 살펴보도록 하겠습니다.

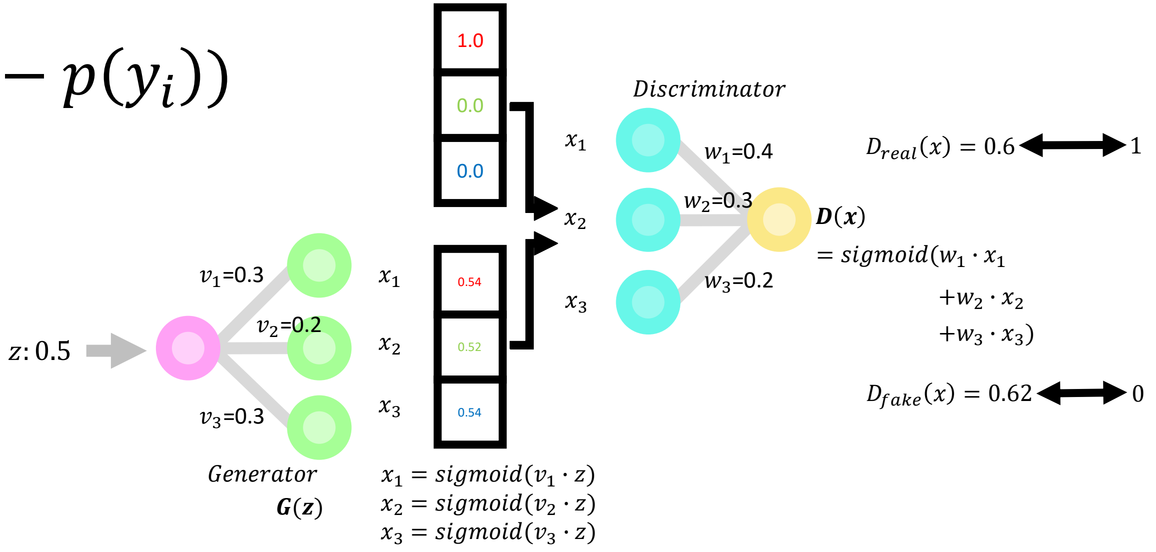
$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$



신박AI

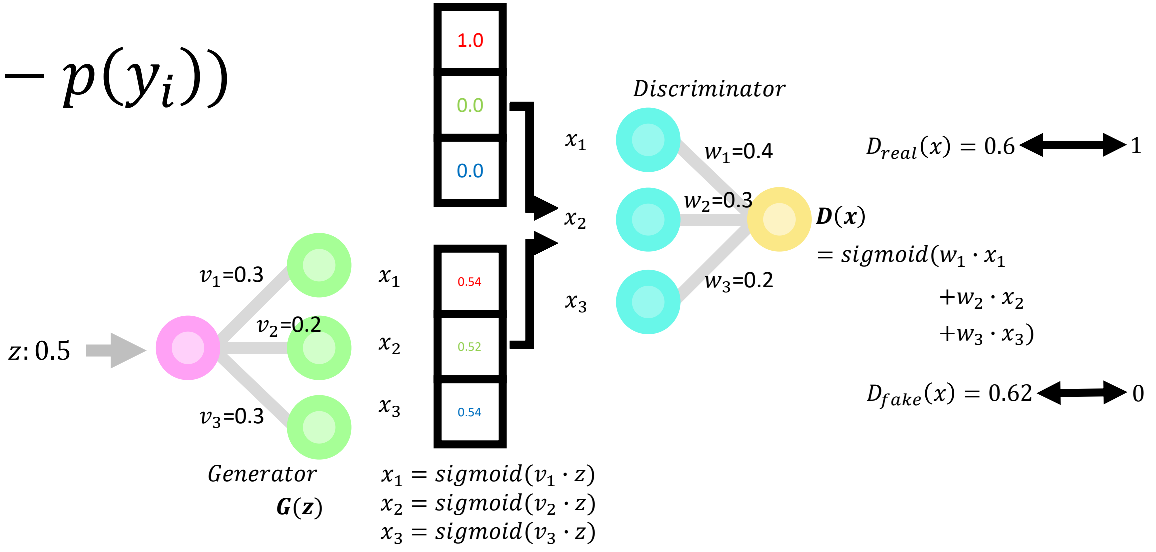
먼저 진짜 데이터가 들어올 경우를 가정해보면,

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$



진짜 데이터의 경우는 discriminator의 출력값이 1이어야 하기 때문에,

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

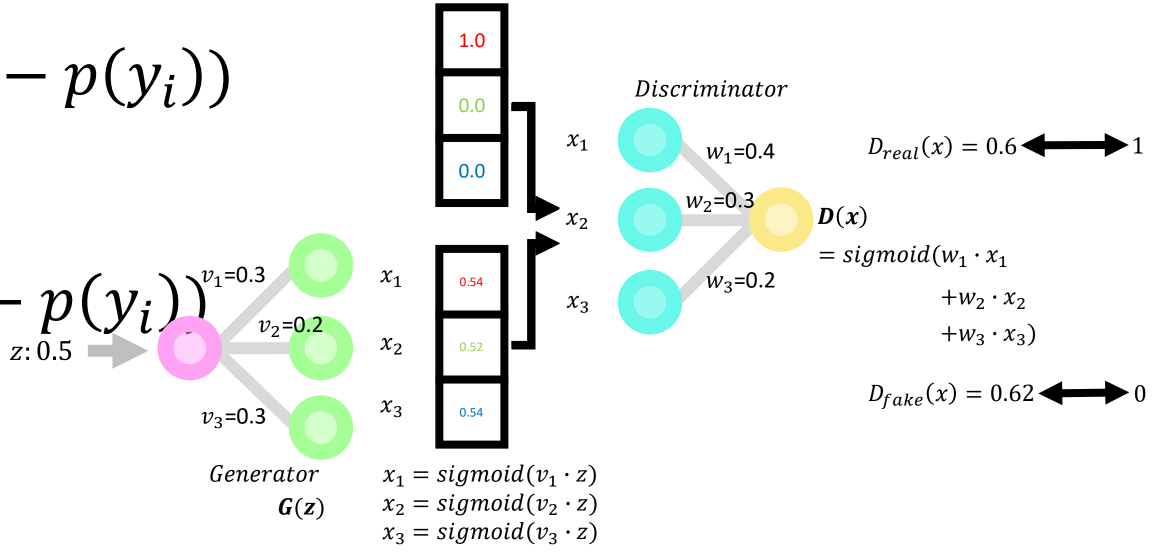


신박AI

손실함수는 이렇게 바꾸어 쓸수가 있습니다

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

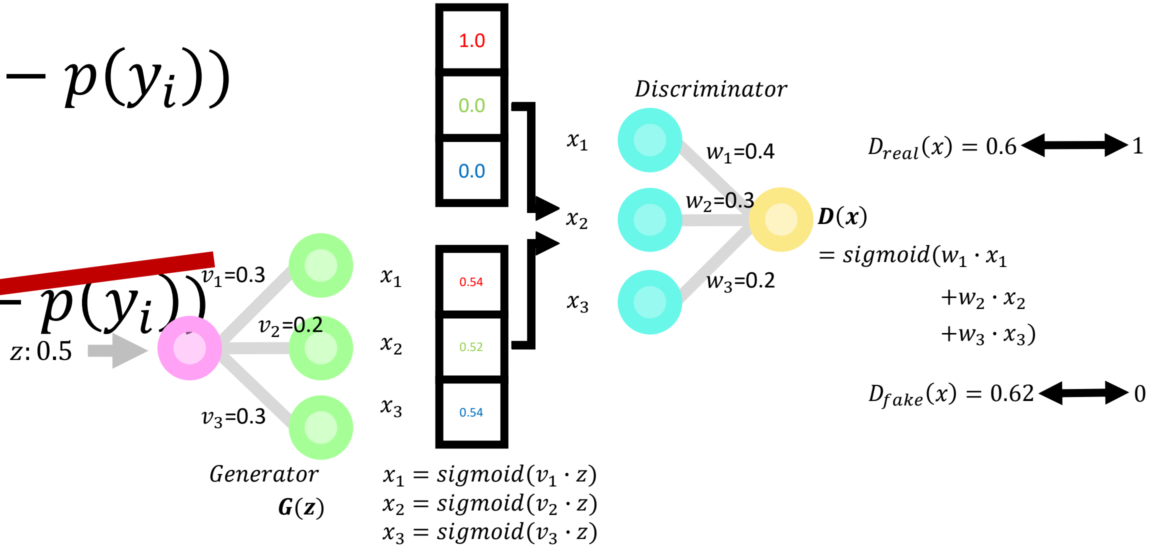
$$BCE = -\frac{1}{N}\sum_{i=1}^{N} 1 \cdot \log(p(y_i)) + (1 - 1) \cdot \log(1 - p(y_i))$$



이 부분은 0이 되어 사라지고,

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

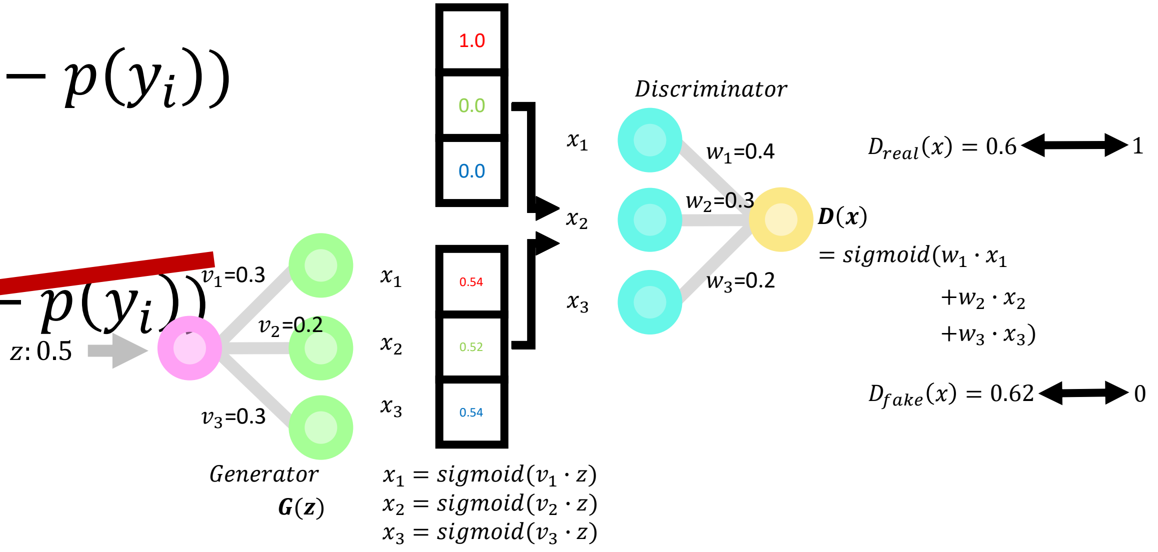
$$BCE = -\frac{1}{N}\sum_{i=1}^{N} 1 \cdot \log(p(y_i)) + \cancel{(1 - 1) \cdot \log(1 - p(y_i))}$$



그리고 진짜 데이터의 경우는 generator와 아무 관련이 없기 때문에,

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot\log(1-p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} 1 \cdot \log(p(y_i)) + \cancel{(1-1)\cdot\log(1-p(y_i))}$$

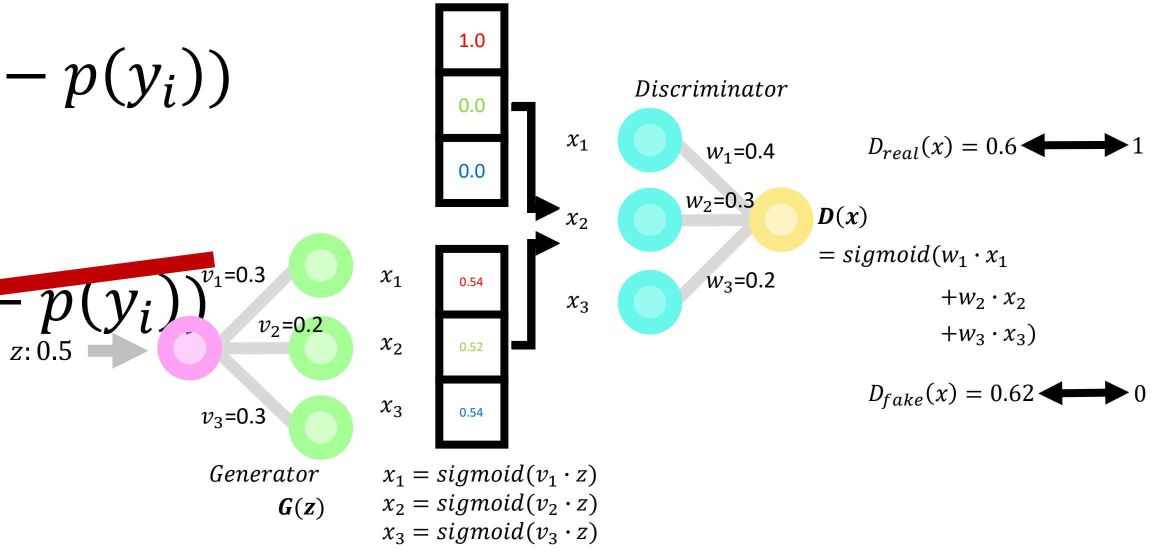


진짜 데이터를 받은 discriminator의 출력값 (확률)을 이렇게 다시 쓸 수 있습니다

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot \log(1-p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} 1 \cdot \log(p(y_i)) + \cancel{(1-1)\cdot \log(1-p(y_i))}$$

$$\log(D(x))$$

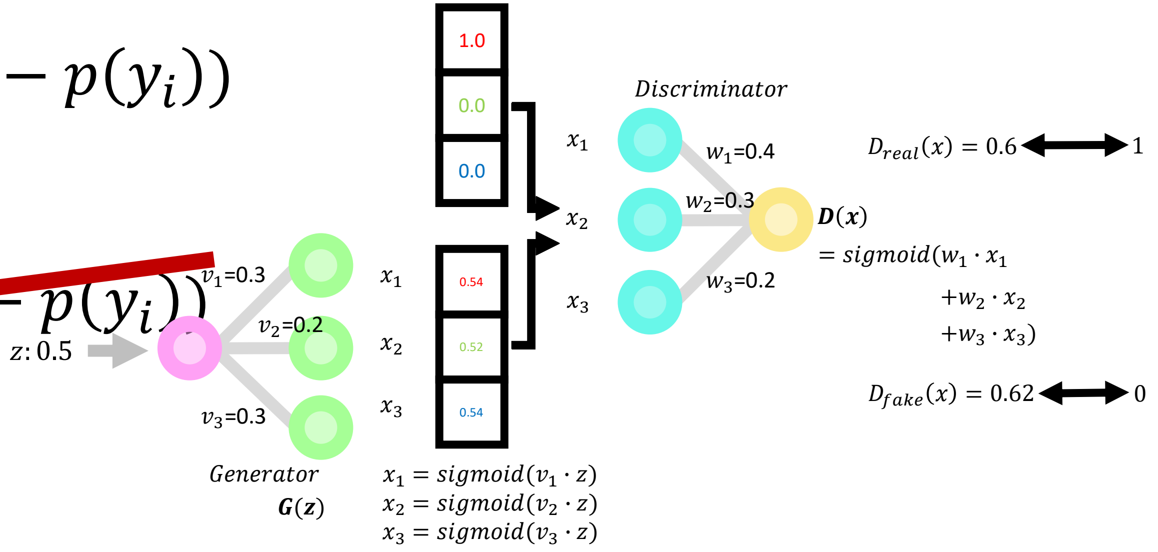


그래서 GAN모델을 학습시킬 때, 만약 입력이 진짜 데이터라면,

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot \log(1-p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} 1 \cdot \log(p(y_i)) + \cancel{(1-1)\cdot \log(1-p(y_i))}$$

$$\log(D(x))$$

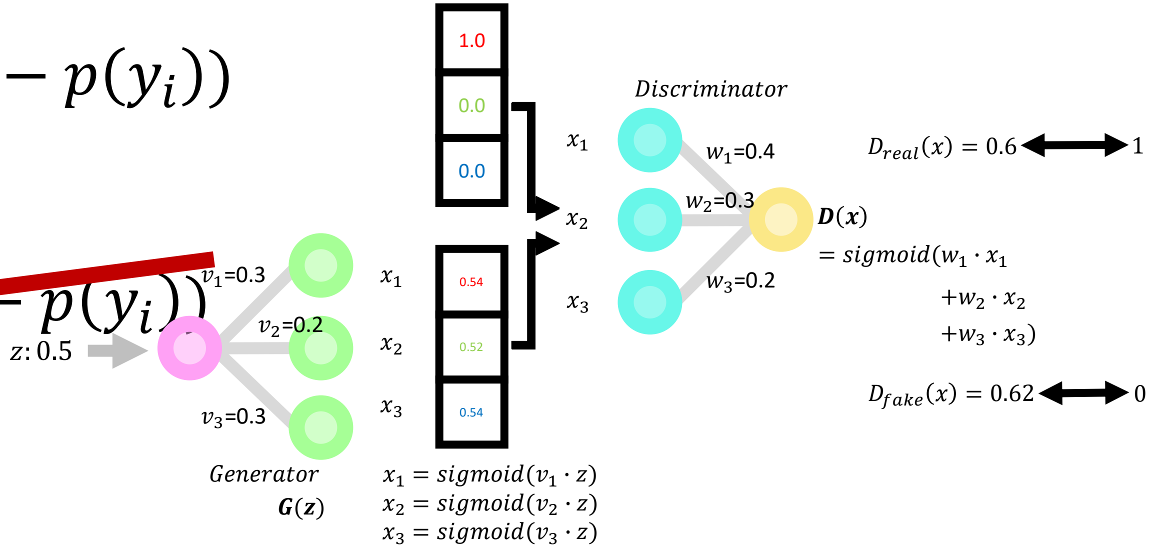


$\log(D(x))$를 최대화 하는 것이 앞의 −와 만나서 손실을 최소로 만들 수 있음을 알 수 있습니다.

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot\log(1-p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} 1 \cdot \log(p(y_i)) + \cancel{(1-1)\cdot\log(1-p(y_i))}$$

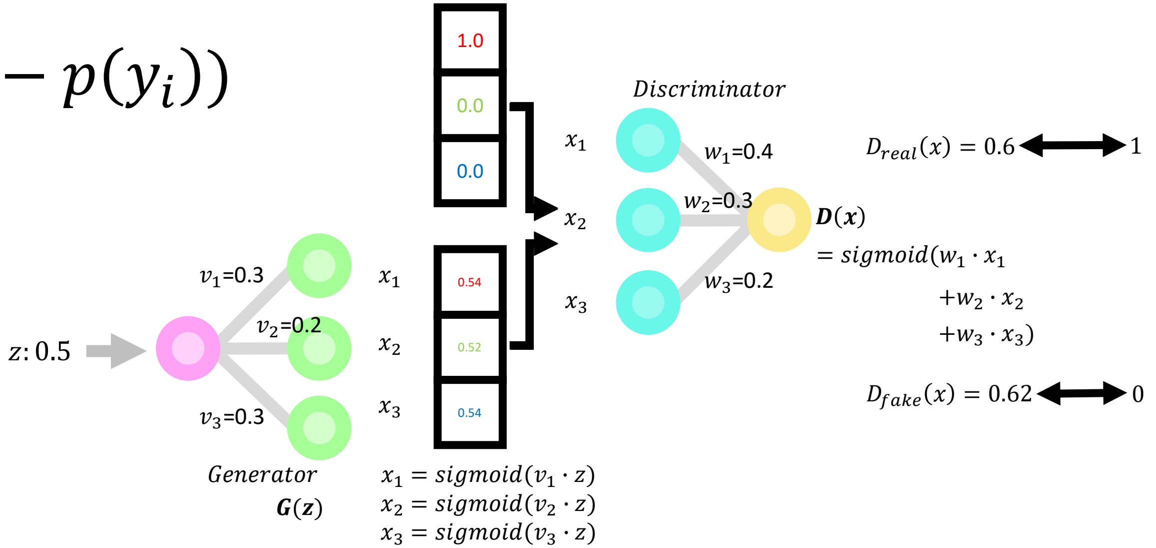
$\log(D(x))$



그런가 하면, 반대로 가짜 데이터의 경우는 discriminator의 출력값이 0이어야 하기 때문에,

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

$$\log(D(x))$$



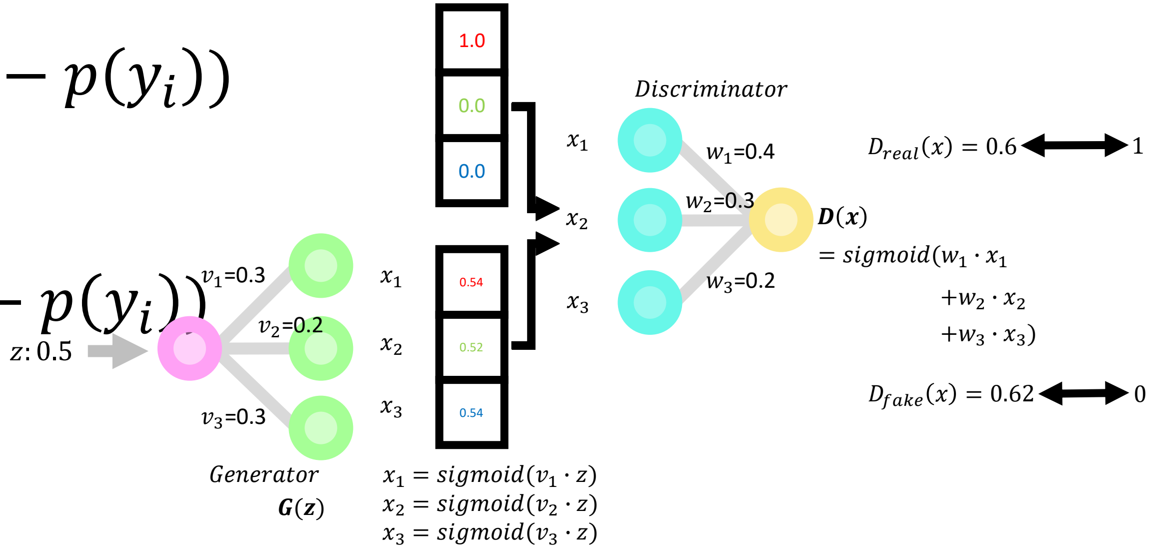
신박AI

손실함수는 이렇게 바꾸어 쓸수가 있습니다

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot\log(1-p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} 0 \cdot \log(p(y_i)) + (1-0)\cdot\log(1-p(y_i))$$

$$\log(D(x))$$



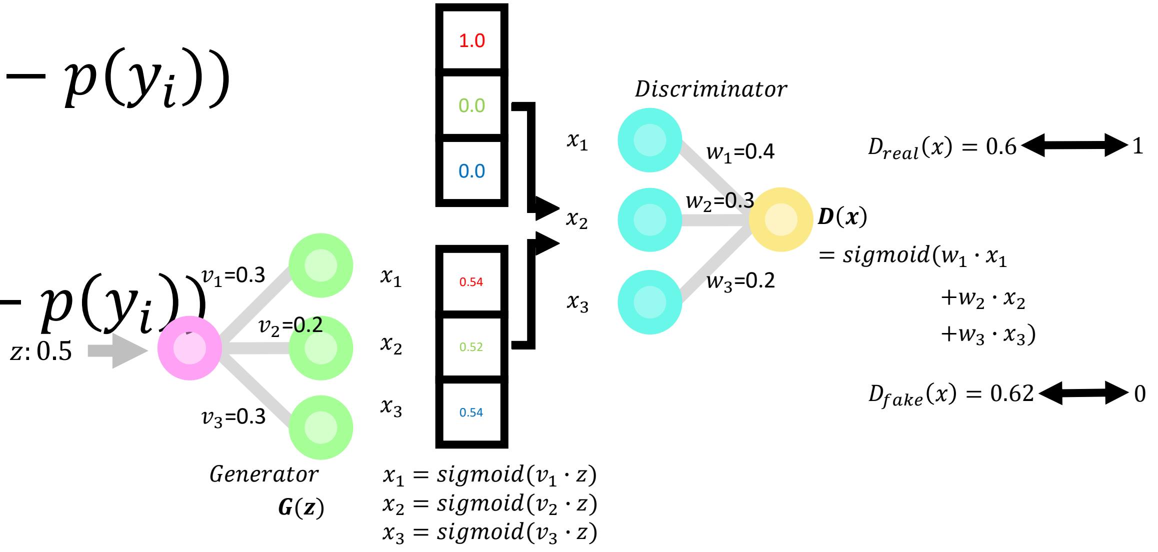
신박AI

마찬가지로 이 부분은 0이 되어 멀리 사라지고,

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot\log(1-p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} \cancel{0 \cdot \log(p(y_i))} + (1-0)\cdot\log(1-p(y_i))$$

$$\log(D(x))$$

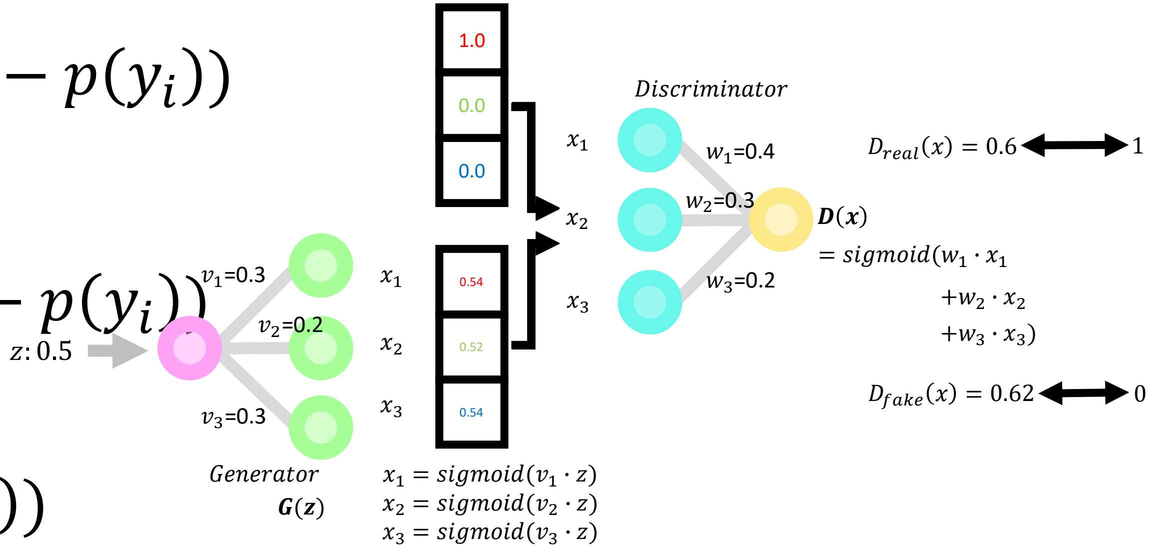


손실함수는 이렇게 바꾸어 쓸수가 있습니다

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot\log(1-p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} \cancel{0\cdot\log(p(y_i))} + (1-0)\cdot\log(1-p(y_i))$$

$\log(D(x))$ ↑ $\log(1-p(y_i))$

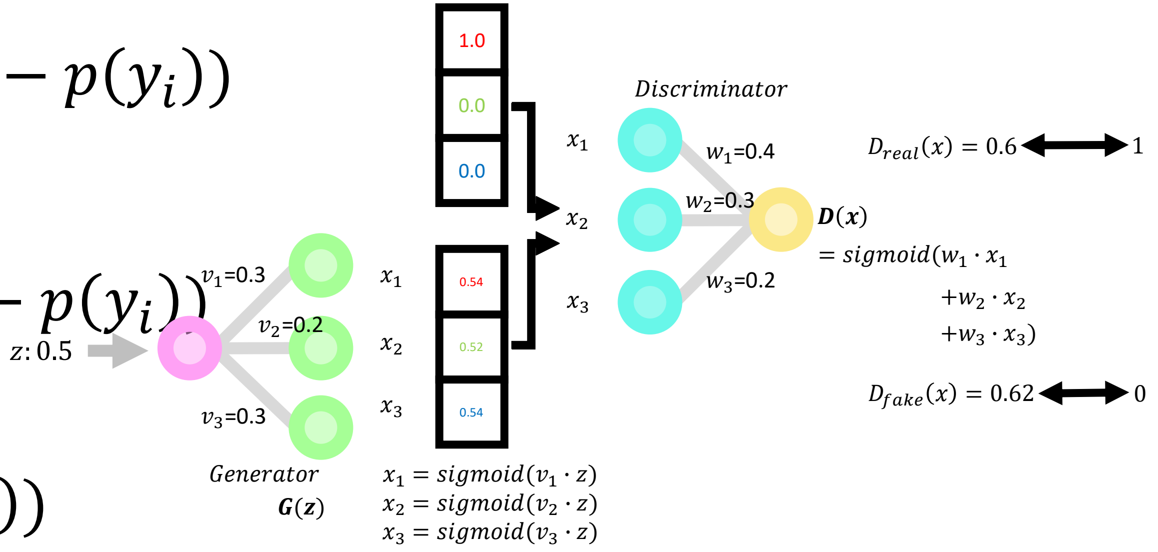


신박AI

가짜 데이터 혹은 생성 데이터는 z로 부터 시작하기 때문에,

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot\log(1-p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} \cancel{0 \cdot \log(p(y_i))} + (1-0)\cdot\log(1-p(y_i))$$



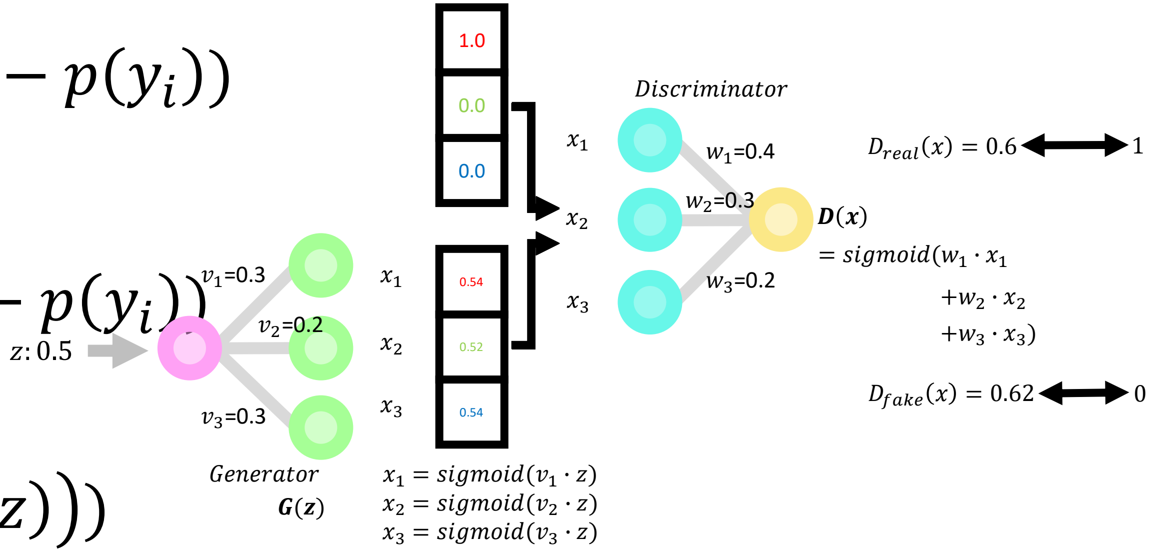
$\log(D(x))$ ↑ $\log(1-p(y_i))$

식을 다음과 같이 바꾸어 쓸 수 있습니다.

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot\log(1-p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} \cancel{0 \cdot \log(p(y_i))} + (1-0)\cdot\log(1-p(y_i))$$

$$\log(D(x)) \uparrow \quad \log(1-D(G(z)))$$



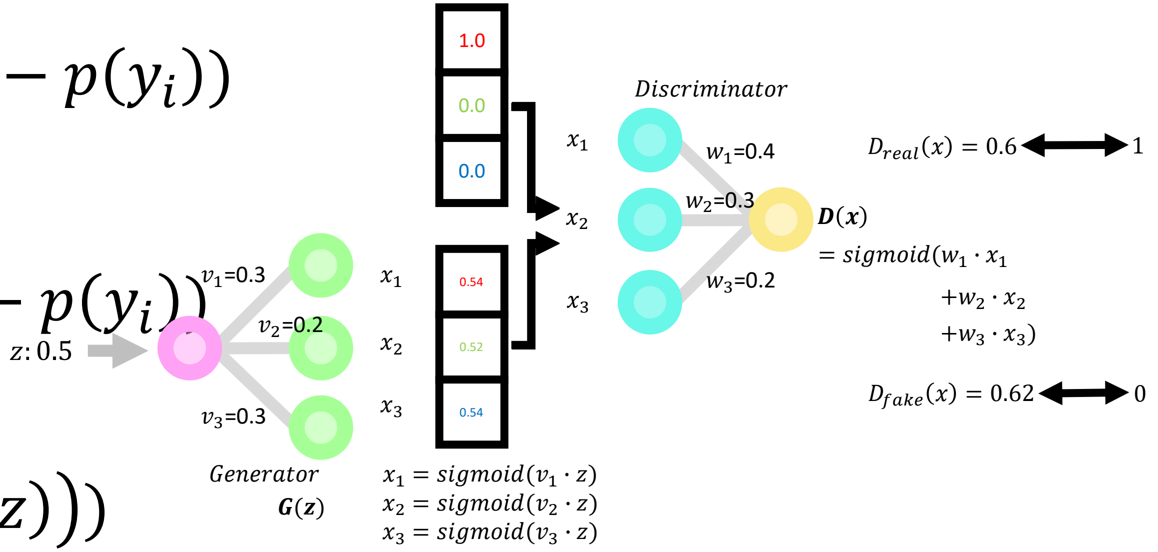
신박AI

그런데 $D(G(z))$는 가짜 (생성) 데이터를 넣은 discriminator의 출력값이기 때문에 0이 되어야 하고,

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i) \cdot \log(1-p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} \cancel{0 \cdot \log(p(y_i))} + (1-0) \cdot \log(1-p(y_i))$$

$$\log(D(x)) \qquad \log(1-D(G(z)))$$

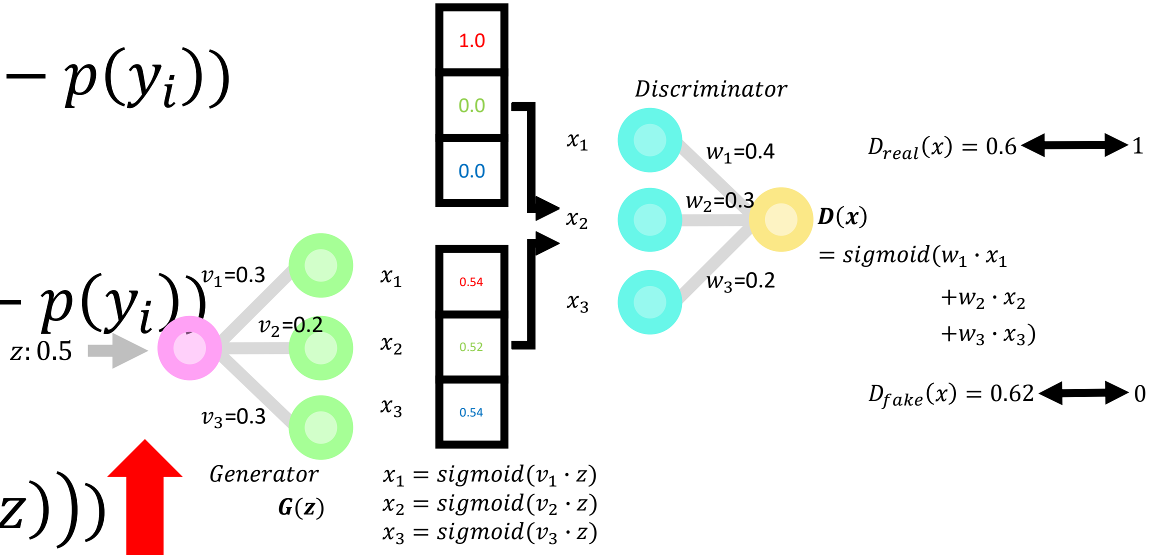


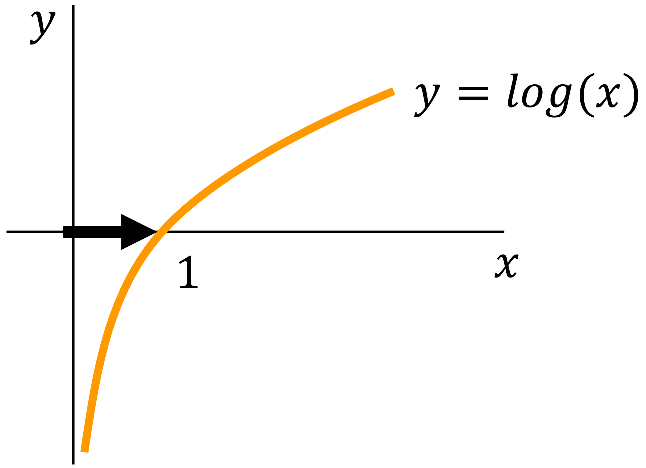
그러므로 $\log(1 - D(G(z)))$ 이 부분 또한 $D(G(z))$가 0일 때 최대값을 갖게 됨을 알 수 있습니다.

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot\log(1-p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} \cancel{0\cdot\log(p(y_i))} + (1-0)\cdot\log(1-p(y_i))$$

$\log(D(x))$ $\log(1-D(G(z)))$



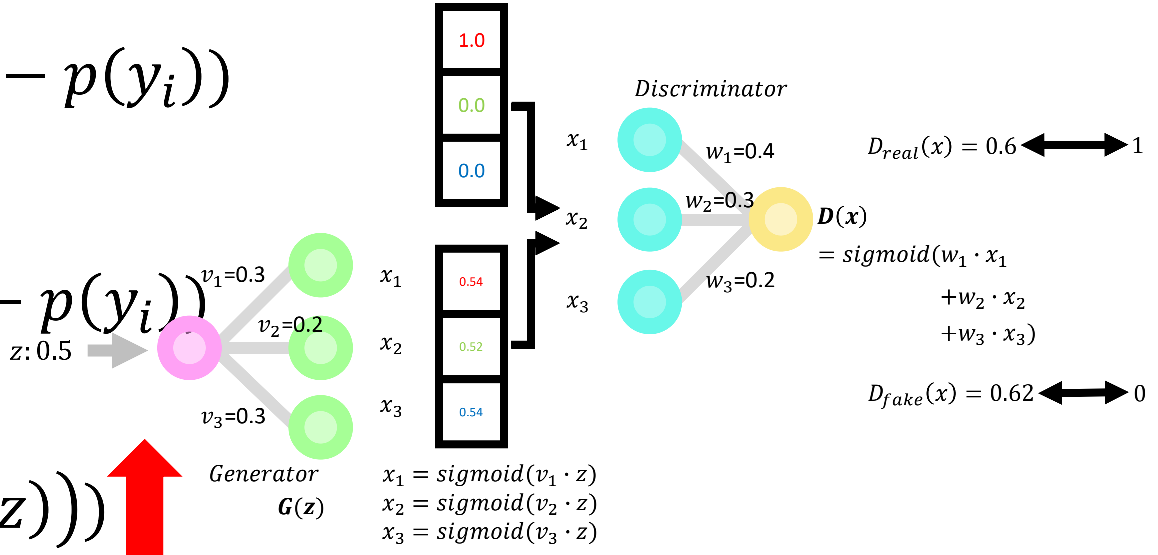


그래서 정리하자면, discriminator의 경우, real이 들어올 경우에는 $\log(D(x))$를 최대로 해야 하기 때문에,

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot\log(1-p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} \cancel{0\cdot\log(p(y_i))} + (1-0)\cdot\log(1-p(y_i))$$

$\log(D(x))$ $\log(1-D(G(z)))$

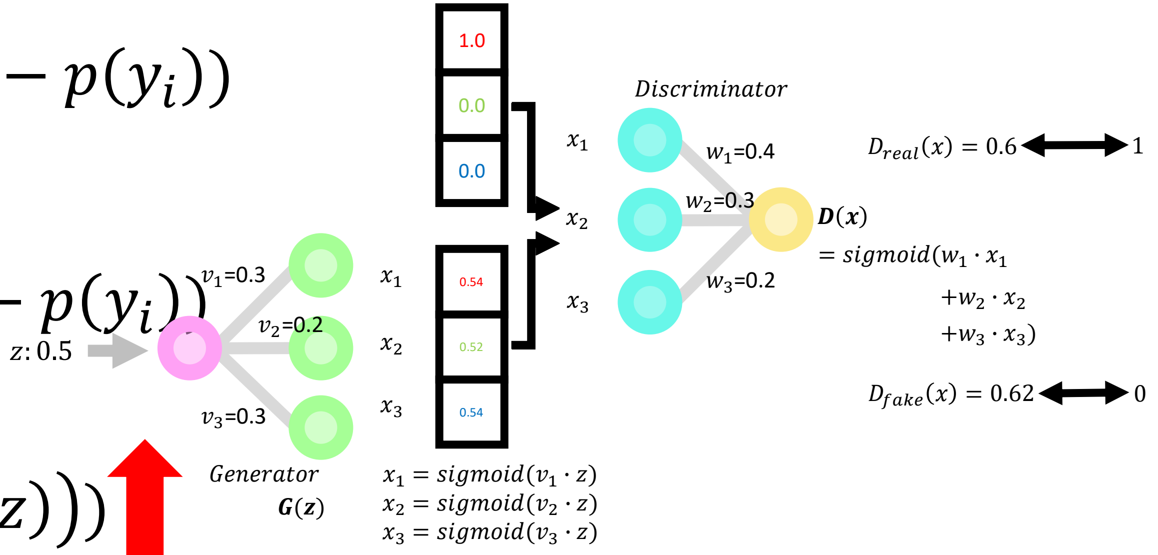


real 데이터에 대한 Discriminator의 손실함수는 다음과 같이 정의될 수 있고,

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot\log(1-p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} \cancel{0 \cdot \log(p(y_i))} + (1-0)\cdot\log(1-p(y_i))$$

$$\log(D(x)) \qquad \log(1-D(G(z)))$$



Discriminator loss function:
실제 (real)이미지가 들어올 경우

$$\text{Loss} = -\log(D(x))$$

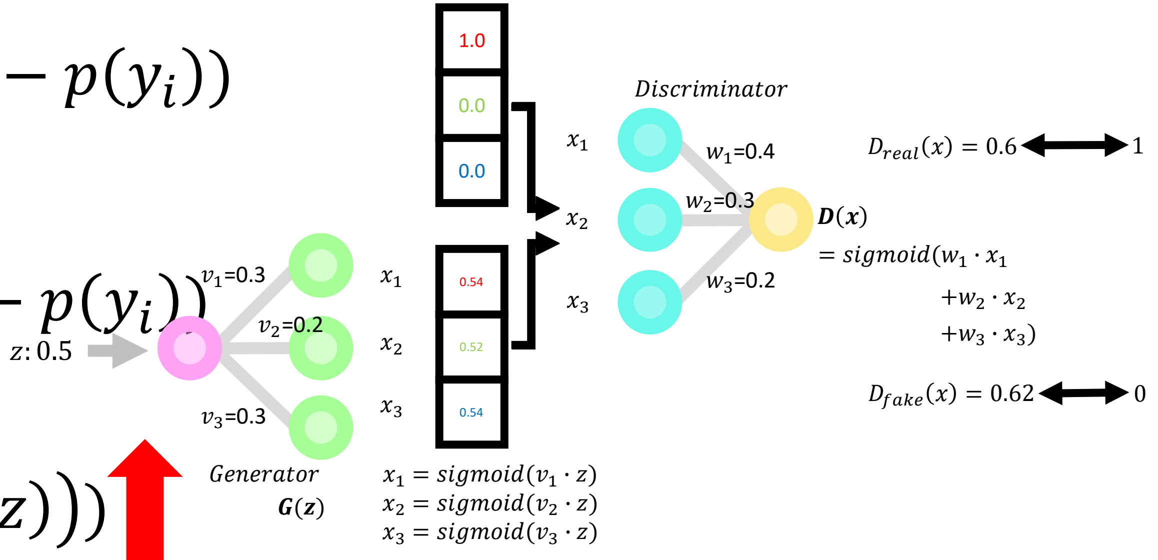
신박AI

Fake 데이터에 대한 Discriminator의 손실함수는 다음과 같이 정의될 수 있습니다.

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} \cancel{0 \cdot \log(p(y_i))} + (1 - 0) \cdot \log(1 - p(y_i))$$

$$\log(D(x)) \qquad \log(1 - D(G(z)))$$



Discriminator loss function:
실제 (real)이미지가 들어올 경우

$$\text{Loss} = -\log(D(x))$$

Discriminator loss function:
생성 (fake)이미지가 들어올 경우

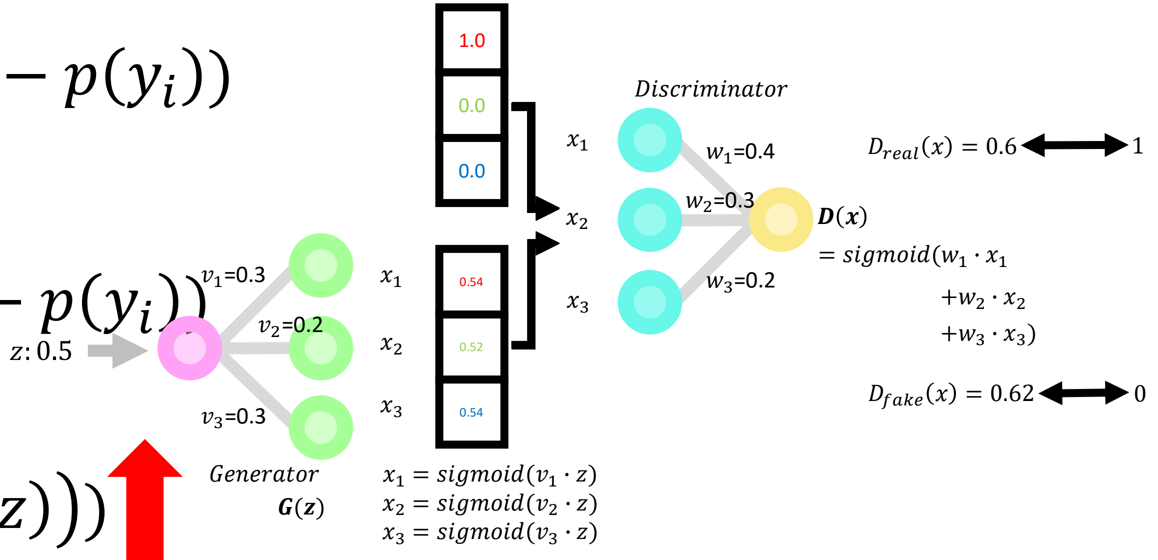
$$\text{Loss} = -\log(1 - D(x))$$

$x = G(z)$

Fake 데이터에 대한 Discriminator의 손실함수는 다음과 같이 정의될 수 있습니다.

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot\log(1-p(y_i))$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} \cancel{0 \cdot \log(p(y_i))} + (1-0)\cdot\log(1-p(y_i))$$



$$\log(D(x)) \quad \log(1-D(G(z)))$$

Discriminator loss function:
실제 (real)이미지가 들어올 경우

$$\text{Loss} = -\log(D(x))$$

Discriminator loss function:
생성 (fake)이미지가 들어올 경우

$$\text{Loss} = -\log(1-D(x))$$

$x = G(z)$

신박AI

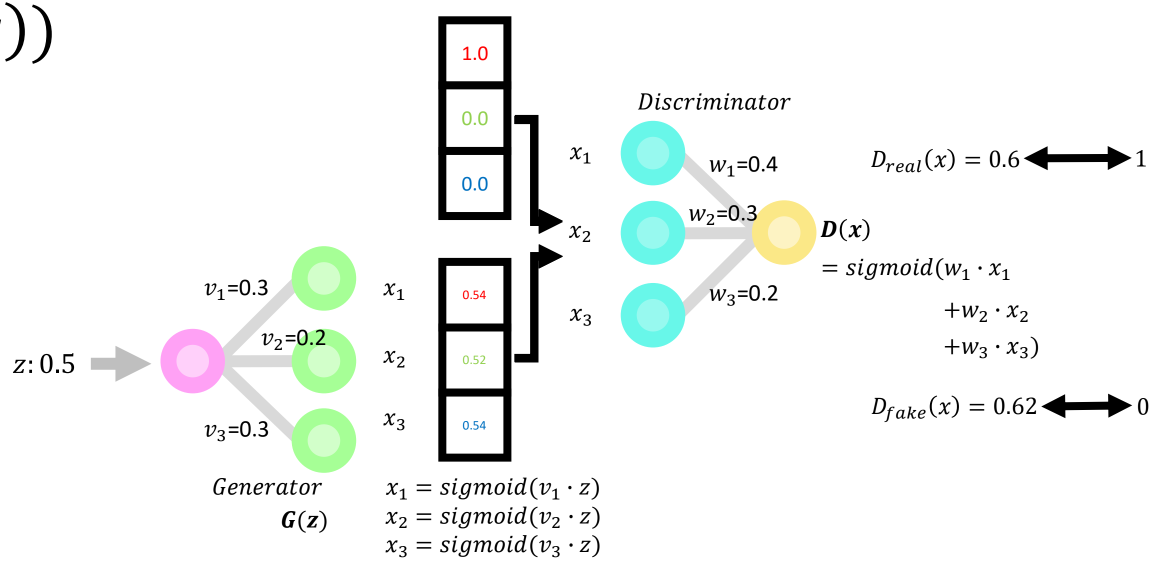
Fake 데이터에 대한 Discriminator의 손실함수는 다음과 같이 정의될 수 있습니다.

Discriminator loss function:
실제 (real)이미지가 들어올 경우

$$\mathrm{Loss} = -\log(D(x))$$

Discriminator loss function:
생성 (fake)이미지가 들어올 경우

$$\mathrm{Loss} = -\log(1 - D(x))$$



신박AI

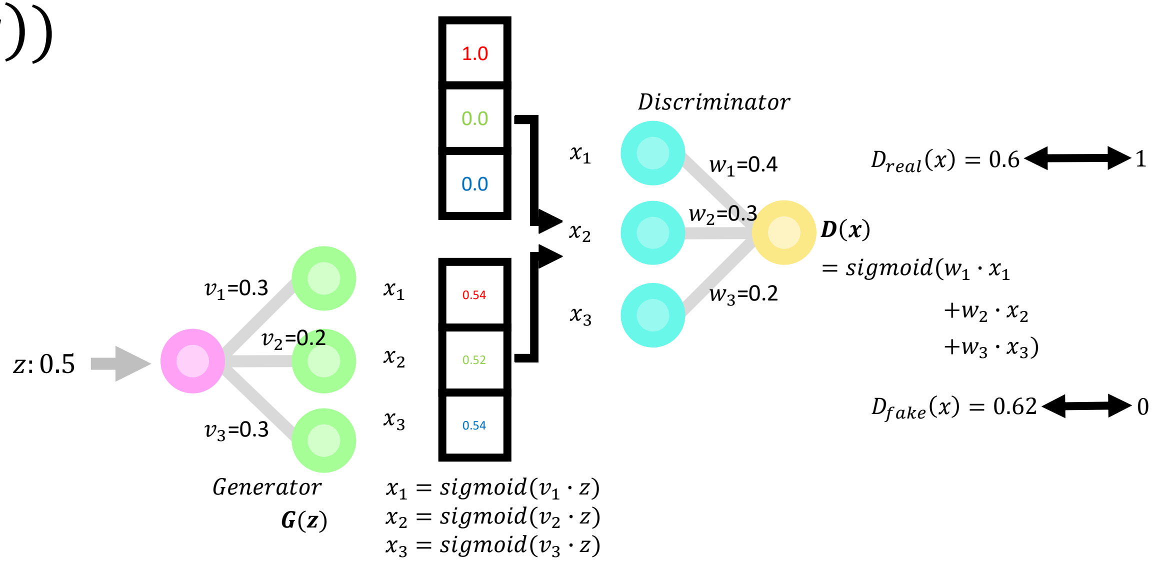
자 그러면 이제부터 discriminator부터 역전파를 이용한 기울기를 구해보도록 하겠습니다.

Real

$$\text{Loss} = -\log(D(x))$$

Fake

$$\text{Loss} = -\log(1 - D(x))$$



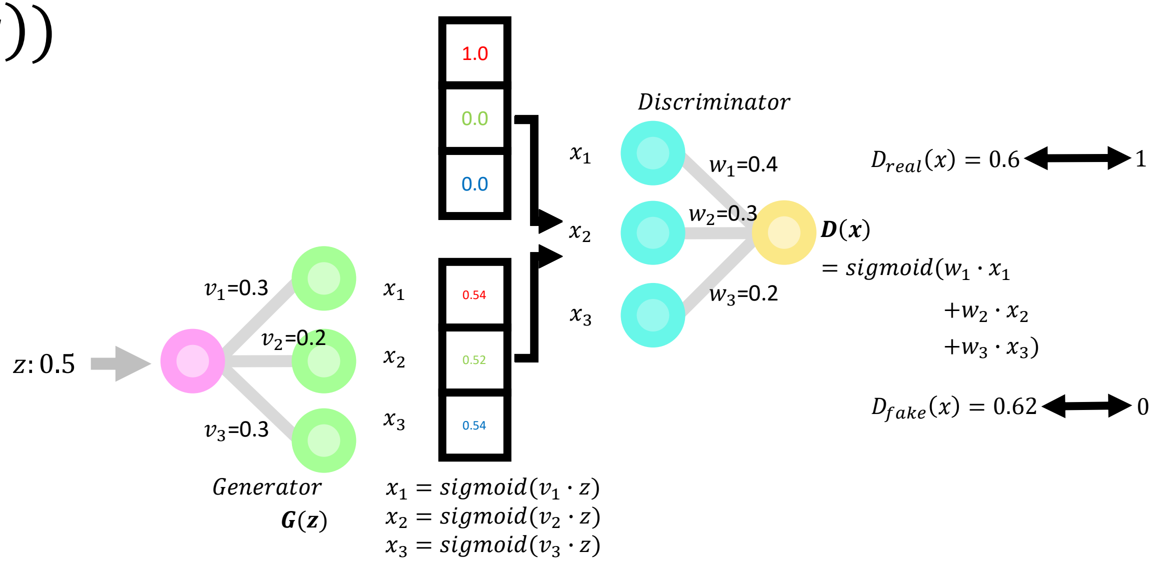
당연히 체인룰을 사용할 것입니다.

Real

$$\text{Loss} = -\log(D(x))$$

Fake

$$\text{Loss} = -\log(1 - D(x))$$



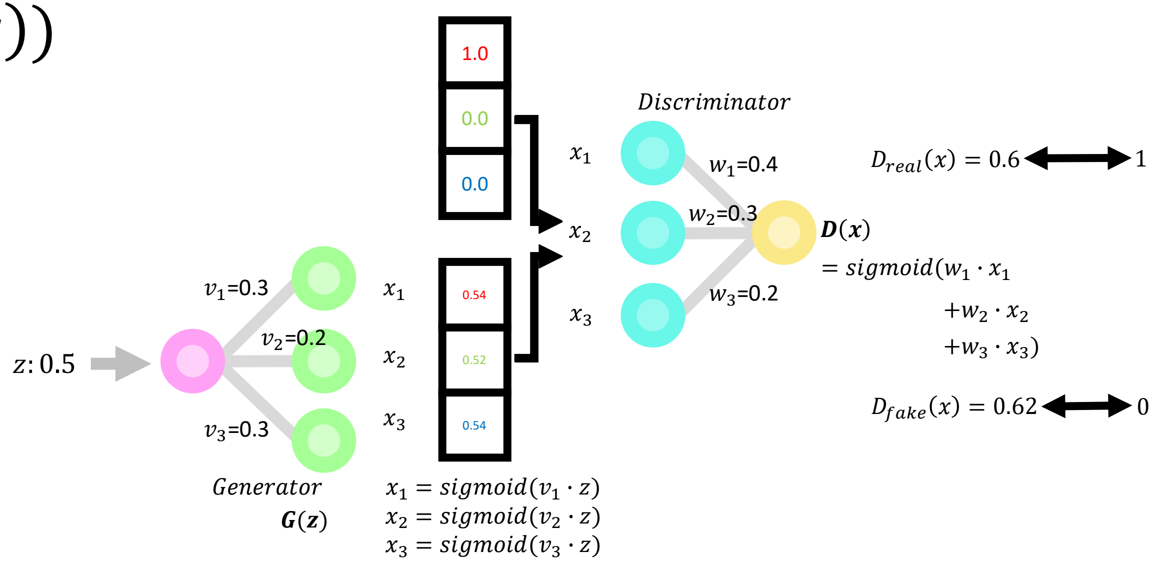
Discriminator의 학습이라는 것도 결국 가중치 변화량에 따른 손실의 변화량을 구하는 것이기 때문에,

Real

$$\text{Loss} = -\log(D(x))$$

Fake

$$\text{Loss} = -\log(1 - D(x))$$



신박AI

결국 discriminator의 학습은 다음의 값들을 구하면 되겠습니다

Real

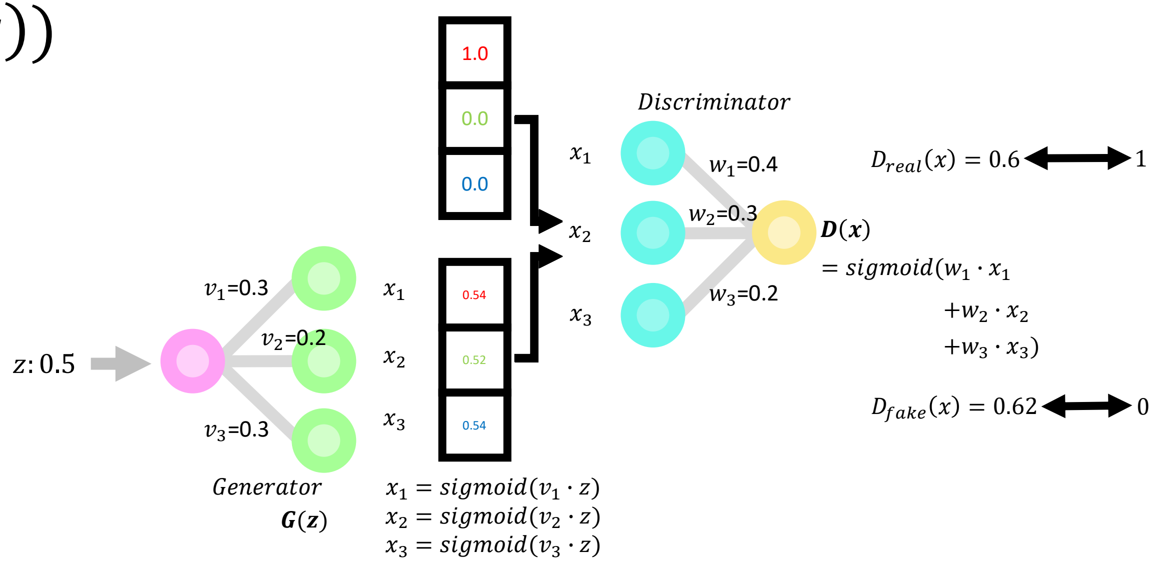
$$\text{Loss} = -\log(D(x))$$

$$\frac{\partial L}{\partial w_i} =$$

Fake

$$\text{Loss} = -\log(1 - D(x))$$

$$\frac{\partial L}{\partial w_i} =$$



신박AI

다음의 값들은 또한..

Real

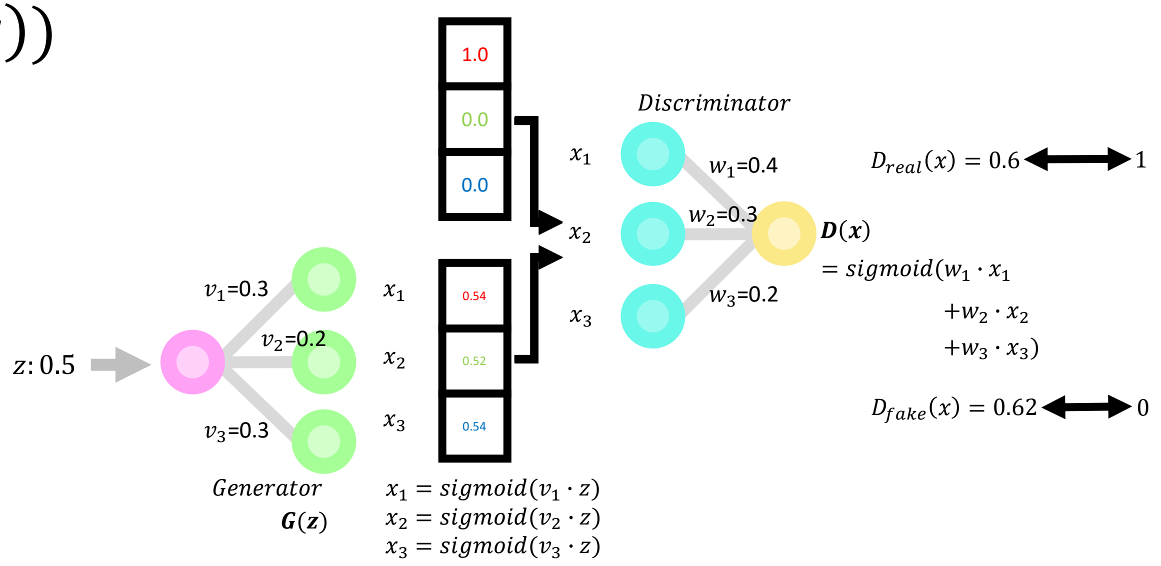
$$\text{Loss} = -\log(D(x))$$

$$\frac{\partial L}{\partial w_i} =$$

Fake

$$\text{Loss} = -\log(1 - D(x))$$

$$\frac{\partial L}{\partial w_i} =$$



Chain rule..

신박AI

다음의 값들은 또한..

Real

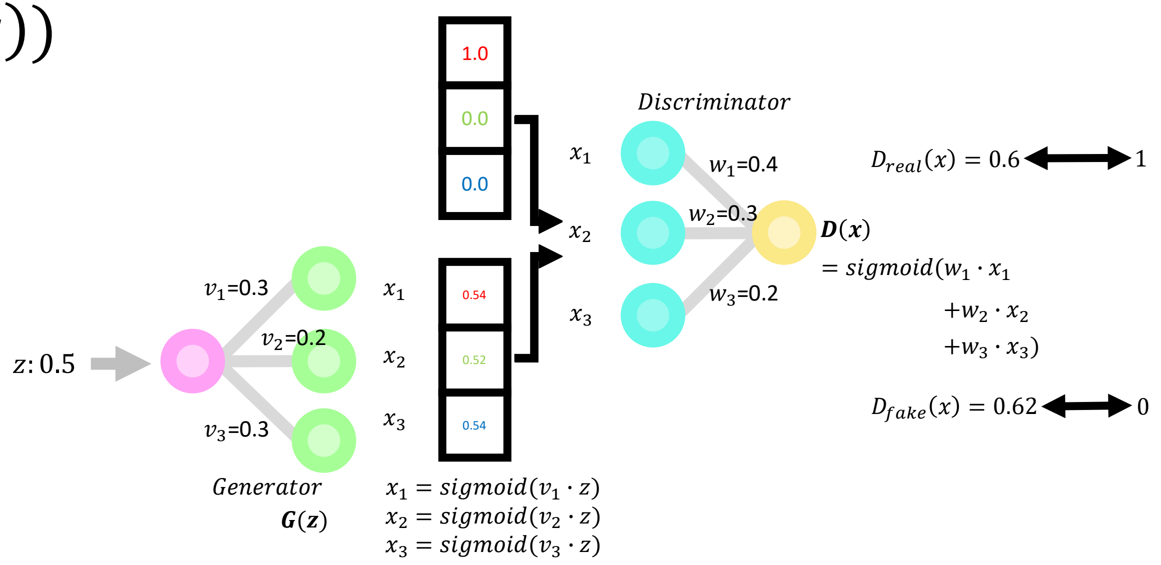
$$\text{Loss} = -\log(D(x))$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

Fake

$$\text{Loss} = -\log(1 - D(x))$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$



$\frac{\partial L}{\partial D}$ 는 log의 미분 공식에 의해서 −1/D(x)가 되고

Real

$$\text{Loss} = -\log(D(x))$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

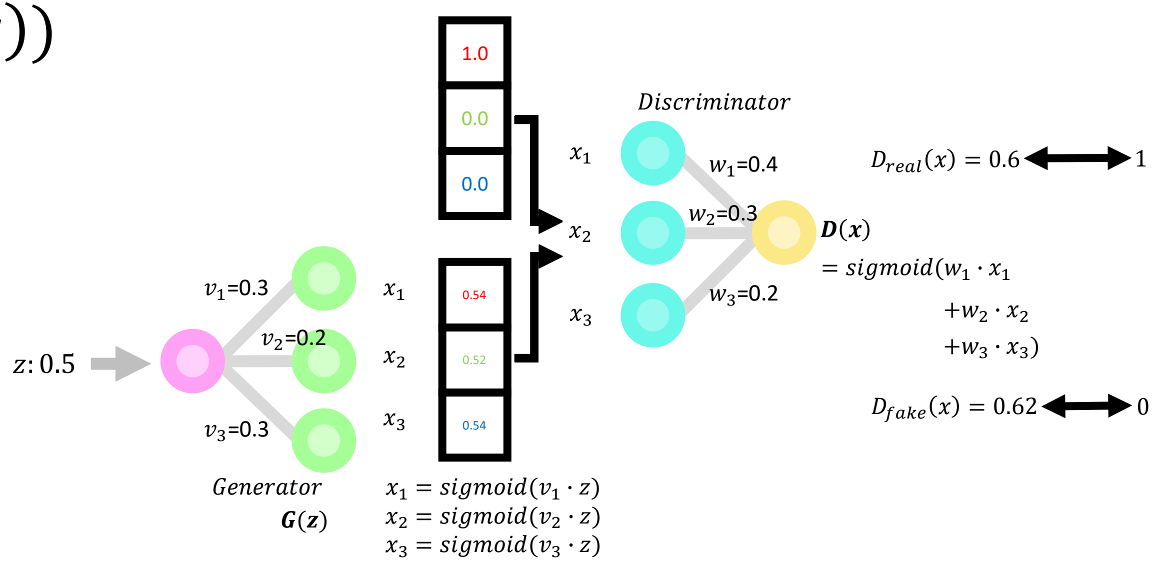
$$= \frac{-1}{D(x)} \cdot \frac{\partial D}{\partial w_i}$$

Fake

$$\text{Loss} = -\log(1 - D(x))$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{-1}{D(x)} \cdot \frac{\partial D}{\partial w_i}$$



신박AI

$\frac{\partial D}{\partial w_i}$ 는 각각 시그모이드 미분 공식에 의해서 다음과 같이 전개됩니다

Real

$$\text{Loss} = -\log(D(x))$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{-1}{D(x)} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{-1}{D(x)} \cdot D(x)(1 - D(x)) \cdot x_i$$

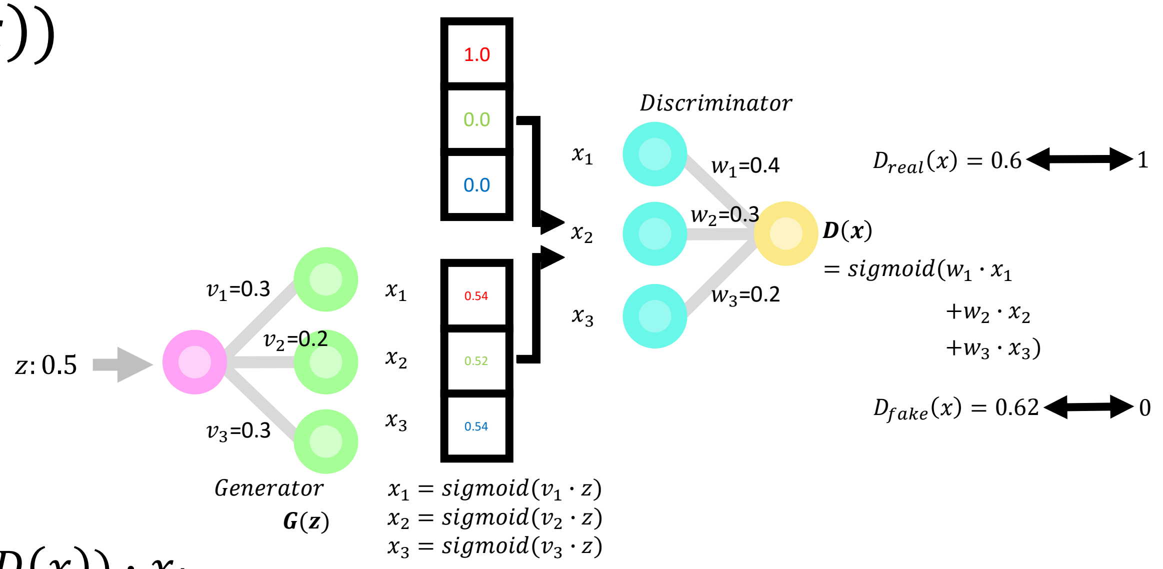
Fake

$$\text{Loss} = -\log(1 - D(x))$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{1}{1 - D(x)} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{1}{1 - D(x)} \cdot D(x)(1 - D(x)) \cdot x_i$$



그리고 사라질 것들은 사라져주면..다음과 같이 정리할 수 있습니다.

Real	*Fake*
$\text{Loss} = -\log(D(x))$	$\text{Loss} = -\log(1 - D(x))$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{-1}{D(x)} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{-1}{\cancel{D(x)}} \cdot \cancel{D(x)}(1 - D(x)) \cdot x_i$$

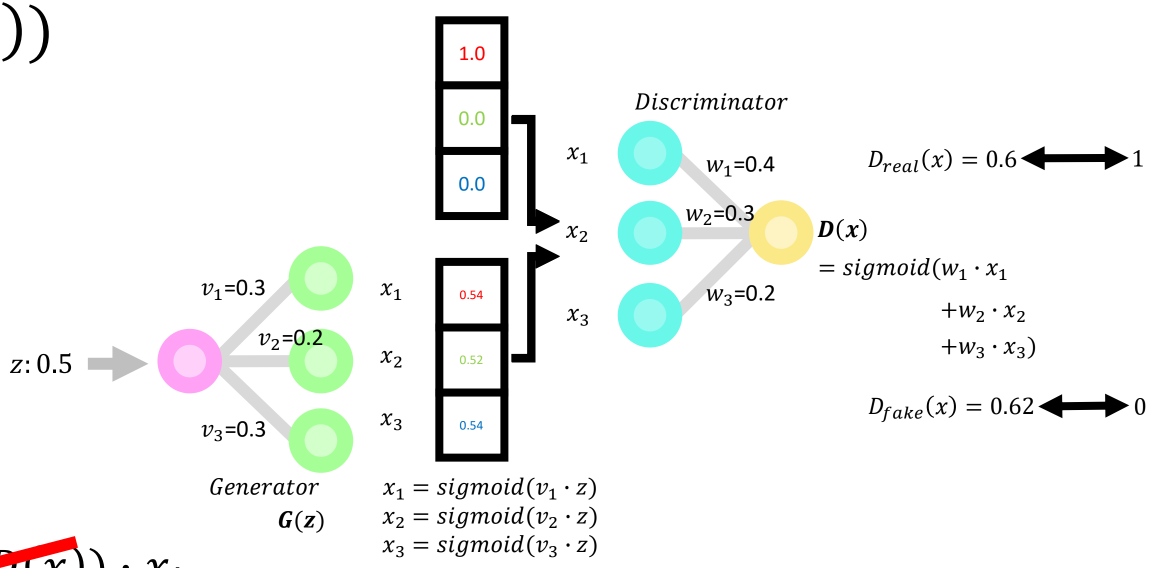
$$= -(1 - D(x)) \cdot x_i$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{1}{1 - D(x)} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{1}{\cancel{1 - D(x)}} \cdot D(x)\cancel{(1 - D(x))} \cdot x_i$$

$$= D(x) \cdot x_i$$



신박AI

그리고 각각의 값을 대입해주면 discriminator의 가중치 변화량을 다음과 같이 계산해 낼 수 있습니다

Real

$$\text{Loss} = -\log(D(x))$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{-1}{D(x)} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{-1}{\cancel{D(x)}} \cdot \cancel{D(x)}(1 - D(x)) \cdot x_i$$

$$= -(1 - D(x)) \cdot x_i$$

$$= -(1 - 0.6) \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Fake

$$\text{Loss} = -\log(1 - D(x))$$

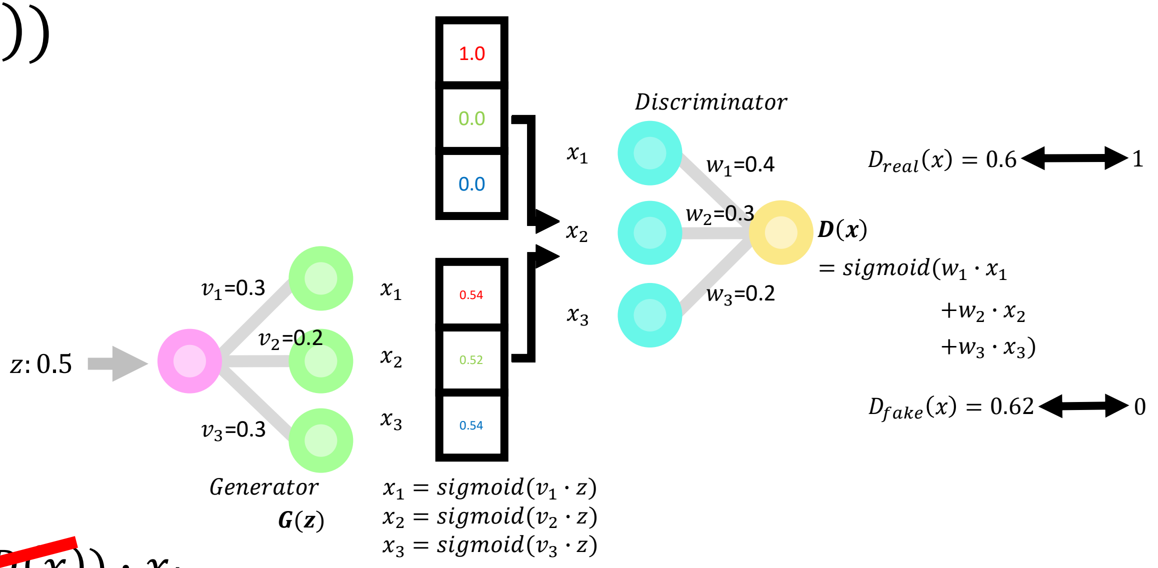
$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{1}{1 - D(x)} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{1}{\cancel{1 - D(x)}} \cdot D(x)\cancel{(1 - D(x))} \cdot x_i$$

$$= D(x) \cdot x_i$$

$$= 0.62 \cdot \begin{bmatrix} 0.54 \\ 0.52 \\ 0.54 \end{bmatrix}$$



신박AI

그리고 각각의 값을 대입해주면 discriminator의 가중치 변화량을 다음과 같이 계산해 낼 수 있습니다

Real

$$\text{Loss} = -\log(D(x))$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{-1}{D(x)} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{-1}{\cancel{D(x)}} \cdot \cancel{D(x)}(1 - D(x)) \cdot x_i$$

$$= -(1 - D(x)) \cdot x_i$$

$$= -(1 - 0.6) \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} -0.4 \\ 0 \\ 0 \end{bmatrix}$$

Fake

$$\text{Loss} = -\log(1 - D(x))$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

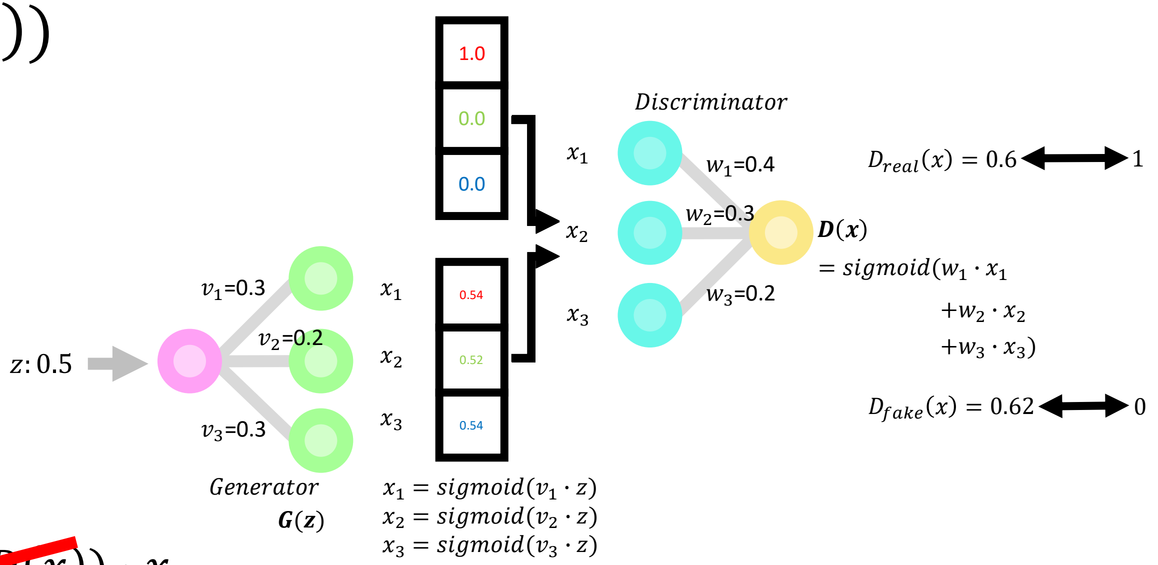
$$= \frac{1}{1 - D(x)} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{1}{\cancel{1 - D(x)}} \cdot D(x)\cancel{(1 - D(x))} \cdot x_i$$

$$= D(x) \cdot x_i$$

$$= 0.62 \cdot \begin{bmatrix} 0.54 \\ 0.52 \\ 0.54 \end{bmatrix}$$

$$= \begin{bmatrix} 0.332 \\ 0.324 \\ 0.332 \end{bmatrix}$$



그러면 학습률을 0.01로 했을 경우, discriminator의 가중치 는 다음과 같이 됩니다

Real

$$\text{Loss} = -\log(D(x))$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{-1}{D(x)} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{-1}{\cancel{D(x)}} \cdot \cancel{D(x)}(1 - D(x)) \cdot x_i$$

$$= -(1 - D(x)) \cdot x_i$$

$$= -(1 - 0.6) \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} -0.4 \\ 0 \\ 0 \end{bmatrix}$$

Fake

$$\text{Loss} = -\log(1 - D(x))$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

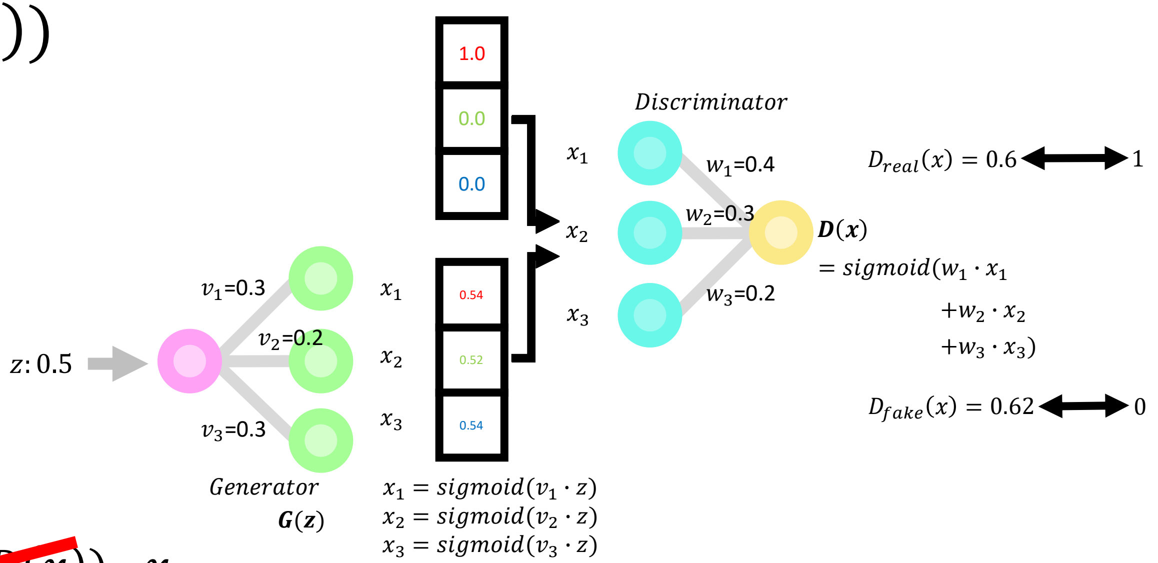
$$= \frac{1}{1 - D(x)} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{1}{\cancel{1 - D(x)}} \cdot D(x)\cancel{(1 - D(x))} \cdot x_i$$

$$= D(x) \cdot x_i$$

$$= 0.62 \cdot \begin{bmatrix} 0.54 \\ 0.52 \\ 0.54 \end{bmatrix}$$

$$= \begin{bmatrix} 0.332 \\ 0.324 \\ 0.332 \end{bmatrix}$$



$$w^* = \begin{bmatrix} 0.4 \\ 0.3 \\ 0.2 \end{bmatrix} - 0.01 * \begin{bmatrix} 0.332 \\ 0.324 \\ 0.332 \end{bmatrix}$$

$$w^* = \begin{bmatrix} 0.396 \\ 0.297 \\ 0.197 \end{bmatrix}$$

신박AI

GAN의 학습에서 유의할 사항은 이처럼, discriminator의 가중치를 업데이트 한 뒤, 그 가중치를 사용하여 generator를 학습합니다.

Real

$$\text{Loss} = -\log(D(x))$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{-1}{D(x)} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{-1}{\cancel{D(x)}} \cdot \cancel{D(x)}(1 - D(x)) \cdot x_i$$

$$= -(1 - D(x)) \cdot x_i$$

$$= -(1 - 0.6) \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} -0.4 \\ 0 \\ 0 \end{bmatrix}$$

Fake

$$\text{Loss} = -\log(1 - D(x))$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial w_i}$$

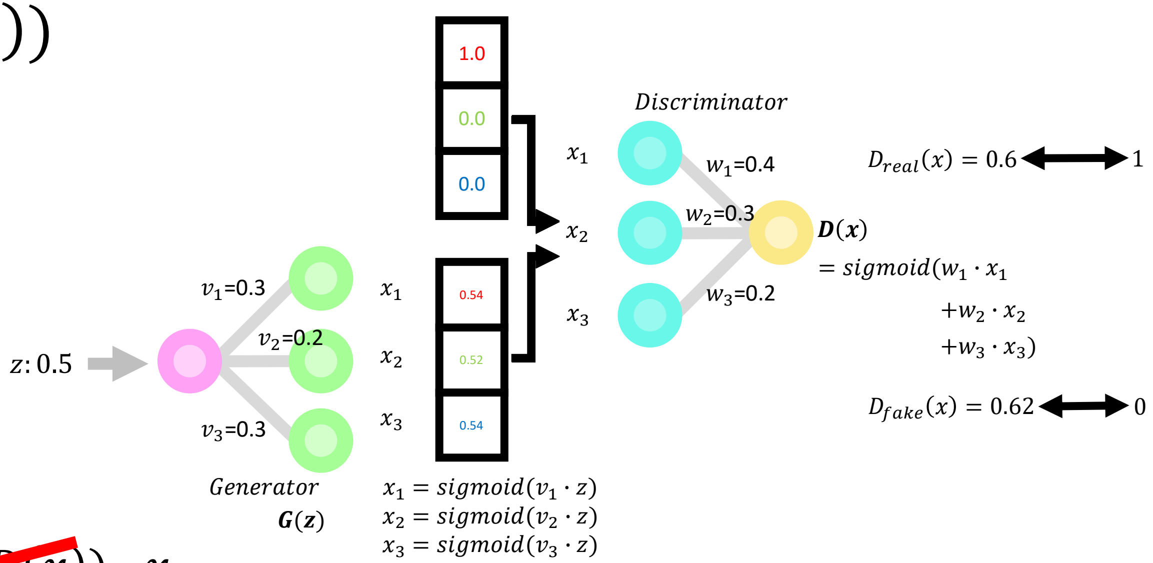
$$= \frac{1}{1 - D(x)} \cdot \frac{\partial D}{\partial w_i}$$

$$= \frac{1}{\cancel{1 - D(x)}} \cdot D(x)\cancel{(1 - D(x))} \cdot x_i$$

$$= D(x) \cdot x_i$$

$$= 0.62 \cdot \begin{bmatrix} 0.54 \\ 0.52 \\ 0.54 \end{bmatrix}$$

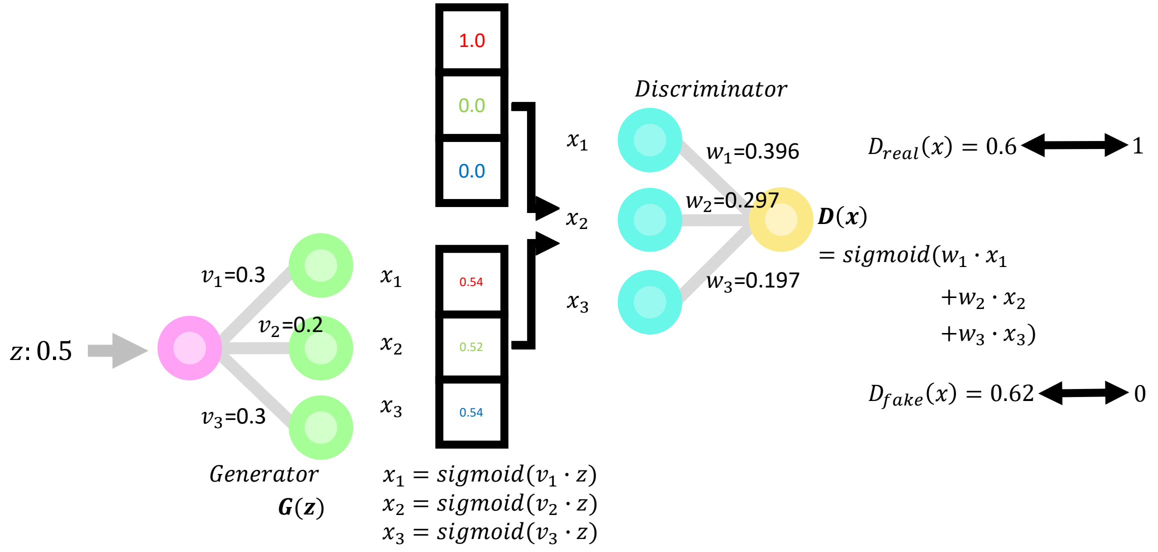
$$= \begin{bmatrix} 0.332 \\ 0.324 \\ 0.332 \end{bmatrix}$$



$$w^* = \begin{bmatrix} 0.4 \\ 0.3 \\ 0.2 \end{bmatrix} - 0.01 * \begin{bmatrix} 0.332 \\ 0.324 \\ 0.332 \end{bmatrix}$$

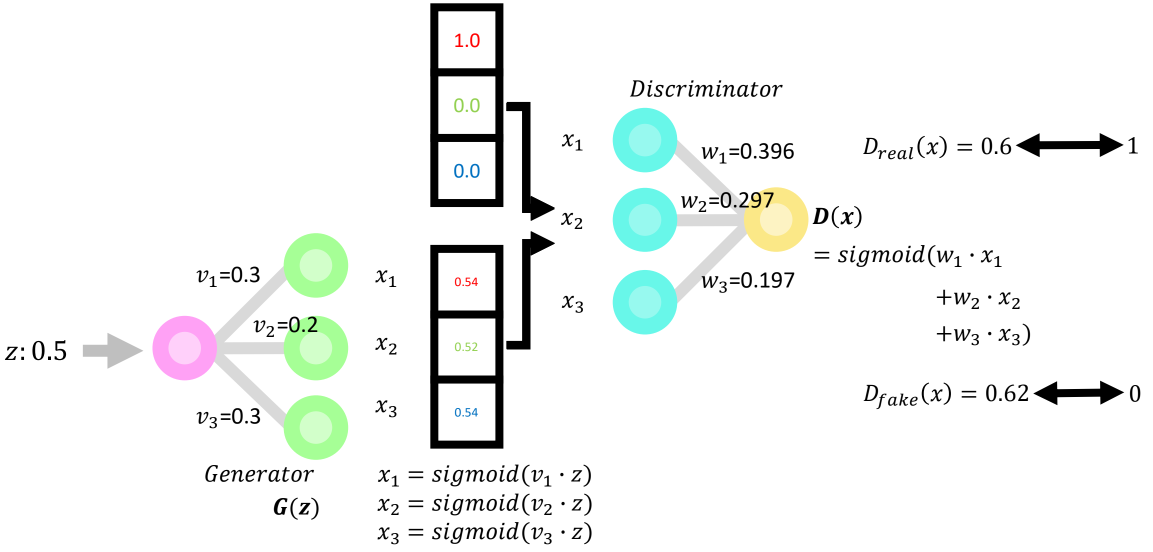
$$w^* = \begin{bmatrix} 0.396 \\ 0.297 \\ 0.197 \end{bmatrix}$$

이젠 generator의 손실함수를 알아보도록 하겠습니다.



좀전에 discriminator의 손실함수에서 real 데이터가 들어왔을 경우, $\log(D(x))$ 를 최대화 하는 것이 학습의 목표라고 말씀드렸습니다.

$$\text{Loss} = -\log(D(x))$$

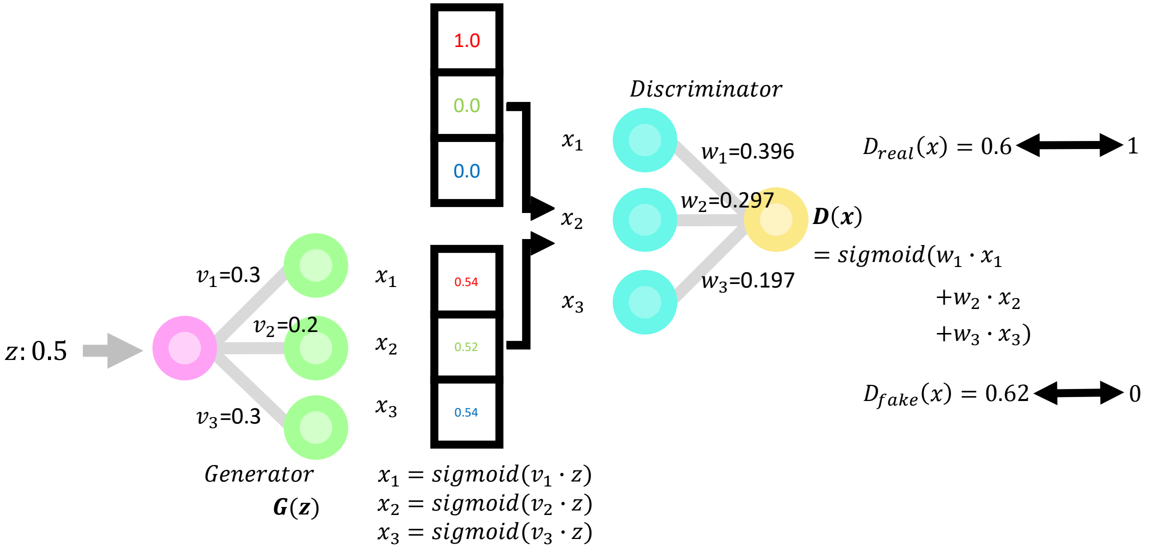


신박AI

Generator도 마찬가지로 fake 데이터를 이용하여 real 데이터 처럼 보이게 만드는 것이 바로 generator의 학습 목표입니다.

$$\text{Loss} = -\log(D(x))$$

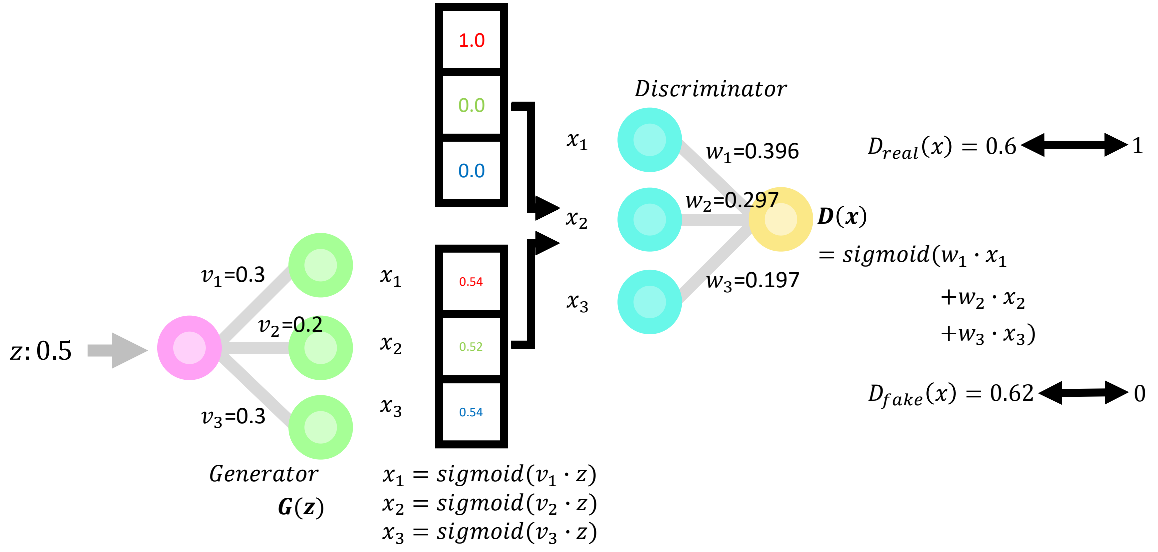
$$\text{Loss} = -\log(D(G(z)))$$



즉 generator의 학습은 다음 함수를 최대화 즉 discriminator가 fake데이터를 받았을 때 1을 출력하도록 만드는 것이 학습의 방향입니다.

$$\text{Loss} = -\log(D(x))$$

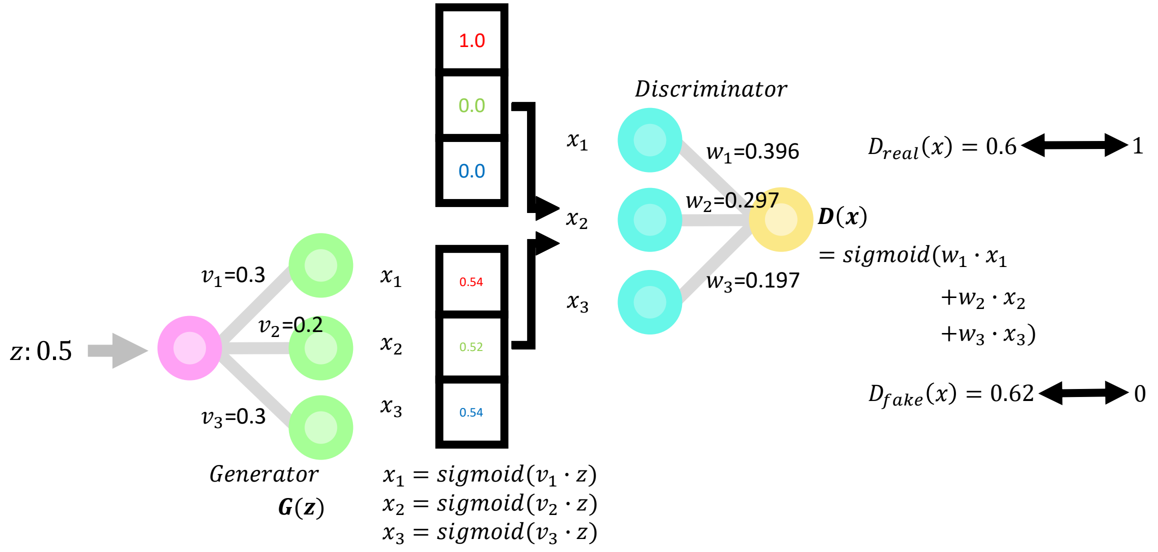
$$\text{Loss} = -\log(D(G(z))$$



신박AI

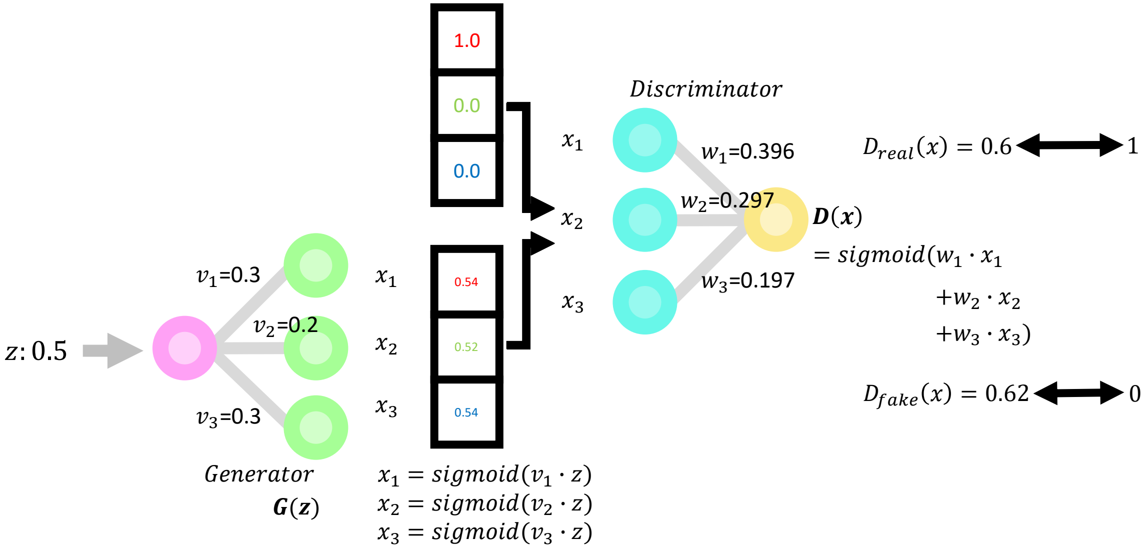
즉 generator의 학습은 다음 함수를 최대화 즉 discriminator가 fake데이터를 받았을 때 1을 출력하도록 만드는 것이 학습의 방향입니다.

$$\text{Loss} = -\log(D(G(z)))$$



Generator의 학습도 가중치 v의 변화량에 따른 손실의 변화량을 구하는 것이기 때문에,

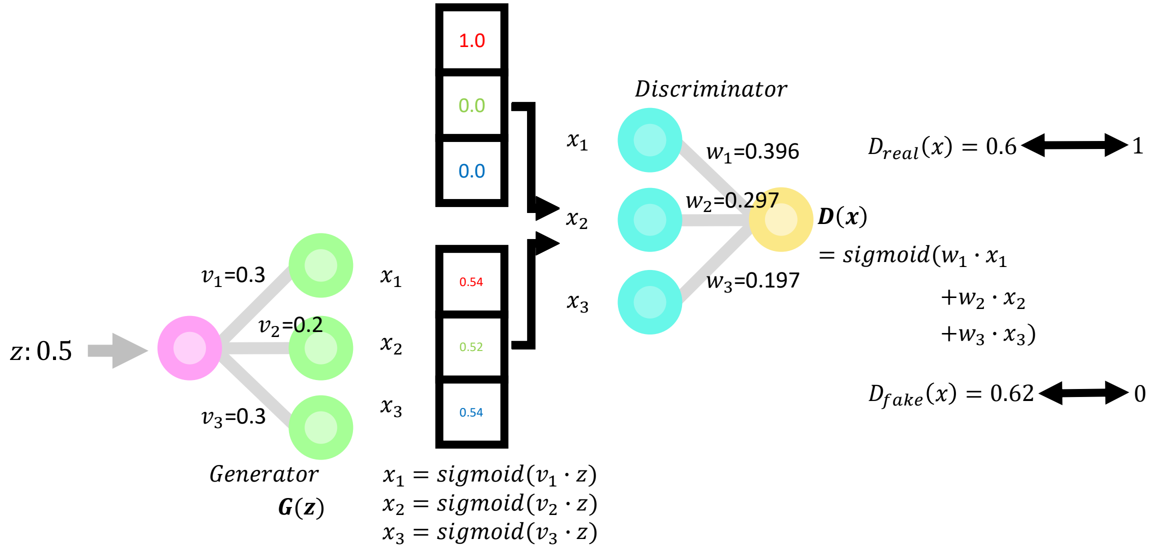
$$\text{Loss} = -\log(D(G(z)))$$



Generator의 학습은 결국 다음의 값을 구하면 되는 것입니다.

$$\text{Loss} = -\log(D(G(z)))$$

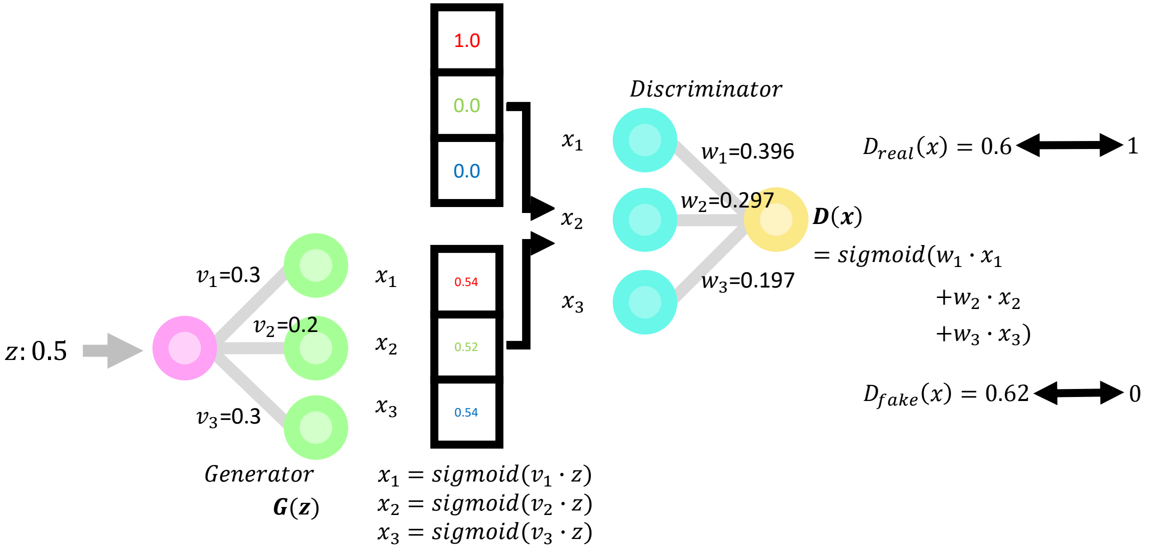
$$\frac{\partial L}{\partial v_i} =$$



체인룰에 의해서 다음과 같이 전개가 되고..

$$\text{Loss} = -\log(D(G(z)))$$

$$\frac{\partial L}{\partial v_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial x_i} \cdot \frac{\partial x_i}{\partial v_i}$$



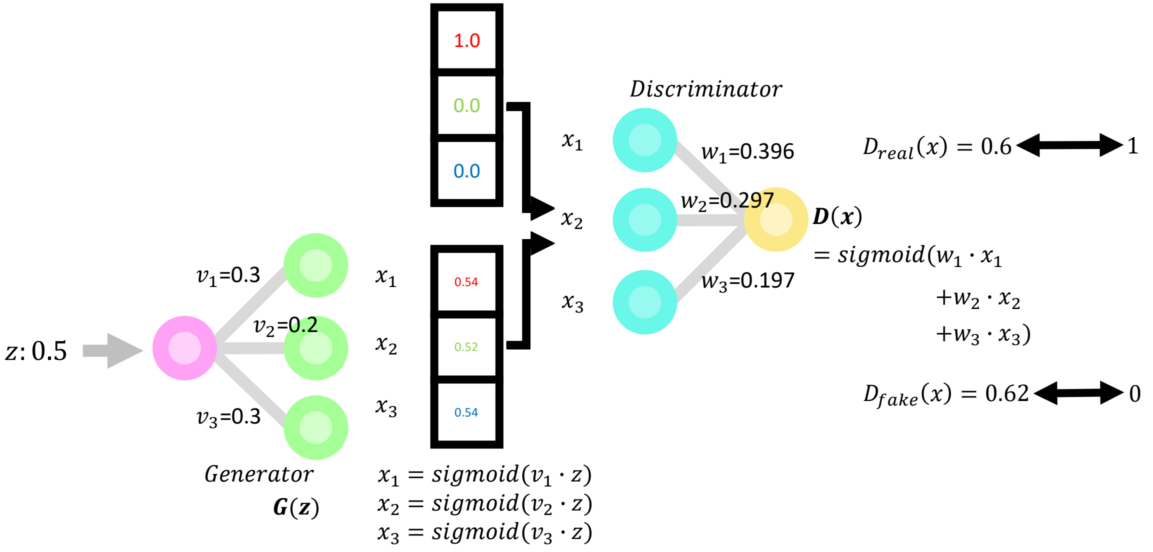
신박AI

$\frac{\partial L}{\partial D}$ 은 로그의 미분에 의해 다음과 같이 바꾸어 쓸 수가 있고,

$$\text{Loss} = -\log(D(G(z)))$$

$$\frac{\partial L}{\partial v_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial x_i} \cdot \frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \frac{-1}{D(G(z))} \cdot \frac{\partial D}{\partial x_i} \cdot \frac{\partial x_i}{\partial v_i}$$



신박AI

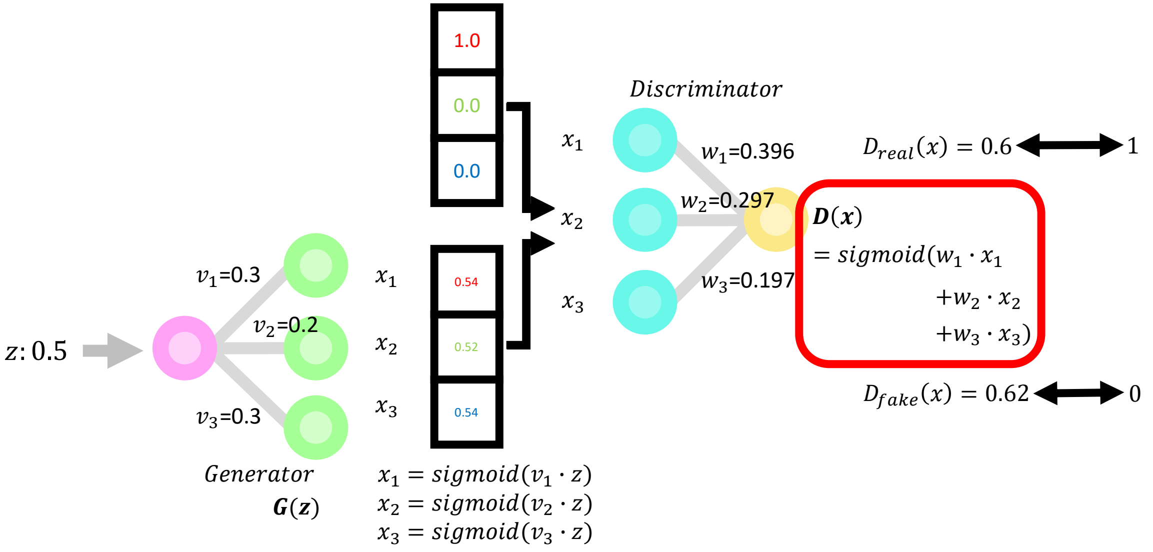
$\frac{\partial D}{\partial x_i}$ 는 D(x)의 정의에 의해서 다음과 같이 전개됩니다

$$\text{Loss} = -\log(D(G(z))$$

$$\frac{\partial L}{\partial v_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial x_i} \cdot \frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \frac{-1}{D(G(z))} \cdot \frac{\partial D}{\partial x_i} \cdot \frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \frac{-1}{D(G(z))} \cdot D(G(z)) \cdot (1 - D(G(z))) \cdot w_i \cdot \frac{\partial x_i}{\partial v_i}$$



신박AI

또한 $\frac{\partial x_i}{\partial v_i}$ 도 G(z)의 정의에 의해서 다음과 같이 전개 됩니다

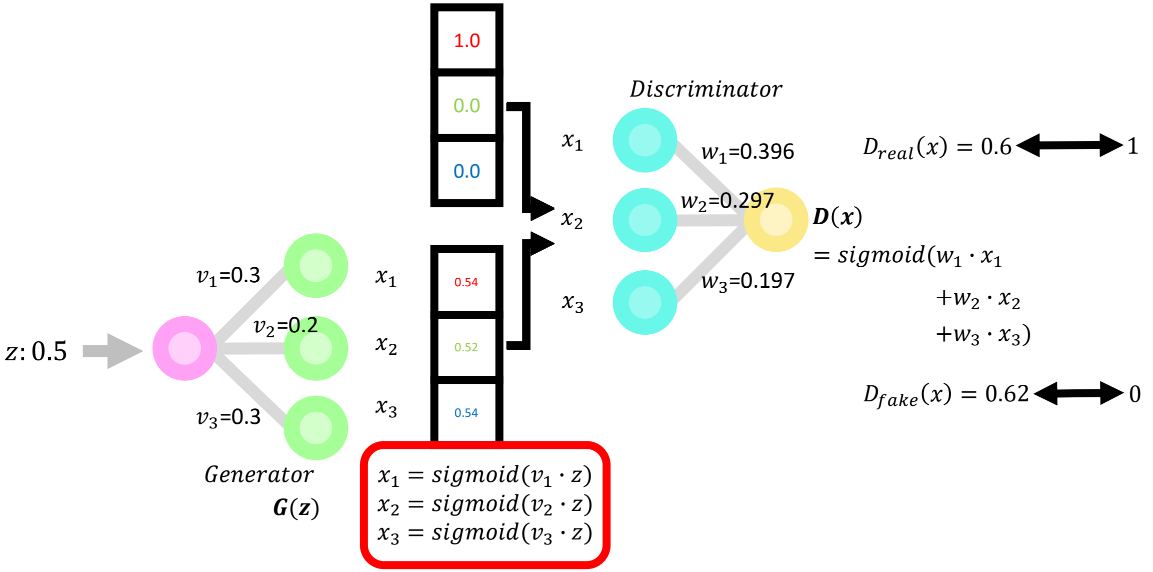
$$\text{Loss} = -\log(D(G(z))$$

$$\frac{\partial L}{\partial v_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial x_i} \cdot \frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \frac{-1}{D(G(z))} \cdot \frac{\partial D}{\partial x_i} \cdot \frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \frac{-1}{D(G(z))} \cdot D(G(z)) \cdot (1 - D(G(z))) \cdot w_i \cdot \frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \frac{-1}{D(G(z))} \cdot D(G(z)) \cdot (1 - D(G(z))) \cdot w_i \cdot G(z)(1 - G(z) \cdot z$$



사라질 것은 사라지고, 또 각각의 값들을 대입하면,

$$\text{Loss} = -\log(D(G(z))$$

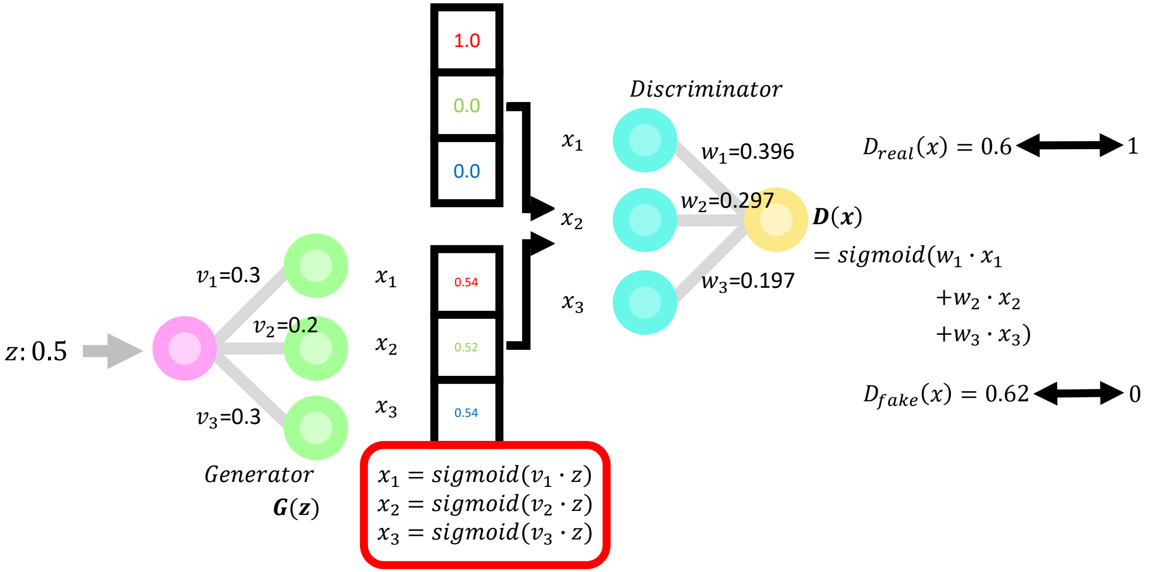
$$\frac{\partial L}{\partial v_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial x_i} \cdot \frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \frac{-1}{D(G(z))} \cdot \frac{\partial D}{\partial x_i} \cdot \frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \frac{-1}{D(G(z))} \cdot D(G(z)) \cdot (1 - D(G(z))) \cdot w_i \cdot \frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \frac{-1}{\cancel{D(G(z))}} \cdot \cancel{D(G(z))} \cdot (1 - D(G(z))) \cdot w_i \cdot G(z)(1 - G(z) \cdot z$$

$$\frac{\partial L}{\partial v_i} = (1 - 0.62) \cdot \begin{bmatrix} 0.396 \\ 0.297 \\ 0.197 \end{bmatrix} \cdot \begin{bmatrix} 0.54 \\ 0.52 \\ 0.54 \end{bmatrix} \cdot (1 - \begin{bmatrix} 0.54 \\ 0.52 \\ 0.54 \end{bmatrix}) \cdot 0.5$$



다음과 같은 generator의 가중치 변화량을 구할 수 있습니다.

$$\mathrm{Loss} = -\log(D(G(z)))$$

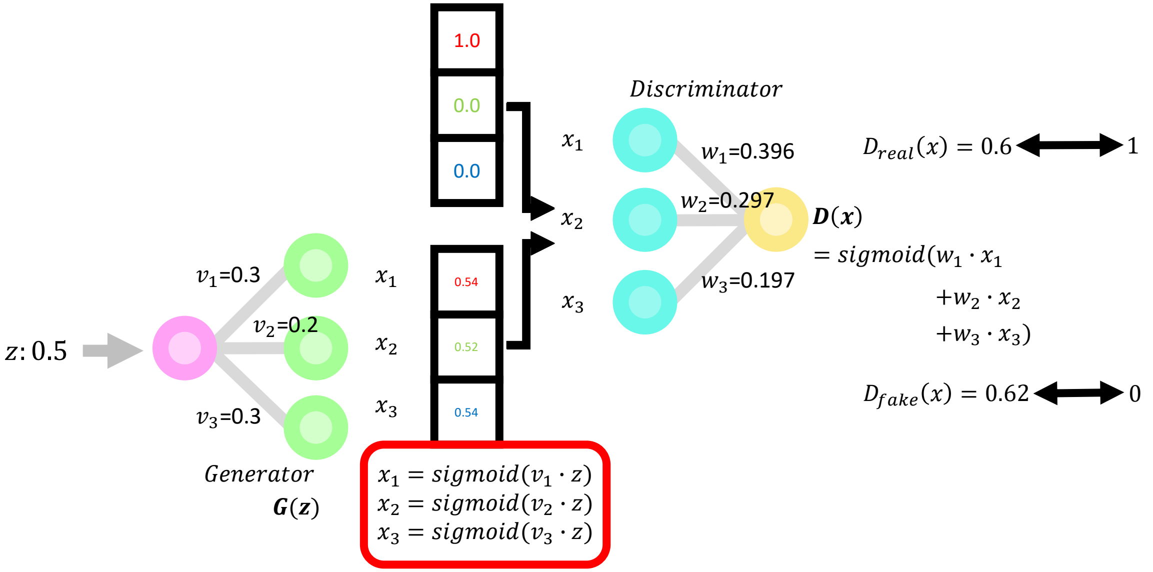
$$\frac{\partial L}{\partial v_i} = \frac{\partial L}{\partial D}\cdot\frac{\partial D}{\partial x_i}\cdot\frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \frac{-1}{D(G(z))}\cdot\frac{\partial D}{\partial x_i}\cdot\frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \frac{-1}{D(G(z))}\cdot D(G(z))\cdot(1-D(G(z)))\cdot w_i\cdot\frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \cancel{\frac{-1}{D(G(z))}\cdot D(G(z))}\cdot(1-D(G(z)))\cdot w_i\cdot G(z)(1-G(z)\cdot z$$

$$\frac{\partial L}{\partial v_i} = (1-0.62)\cdot\begin{bmatrix}0.396\\0.297\\0.197\end{bmatrix}\cdot\begin{bmatrix}0.54\\0.52\\0.54\end{bmatrix}\cdot\left(1-\begin{bmatrix}0.54\\0.52\\0.54\end{bmatrix}\right)\cdot 0.5 = \begin{bmatrix}-0.019\\-0.014\\-0.009\end{bmatrix}$$



이렇게 discriminator와 generator가 서로 경쟁하듯 가중치를 계속 업데이트 해나가는 것이 GAN모델 학습의 대략적인 과정이 되겠습니다.

$$\text{Loss} = -\log(D(G(z))$$

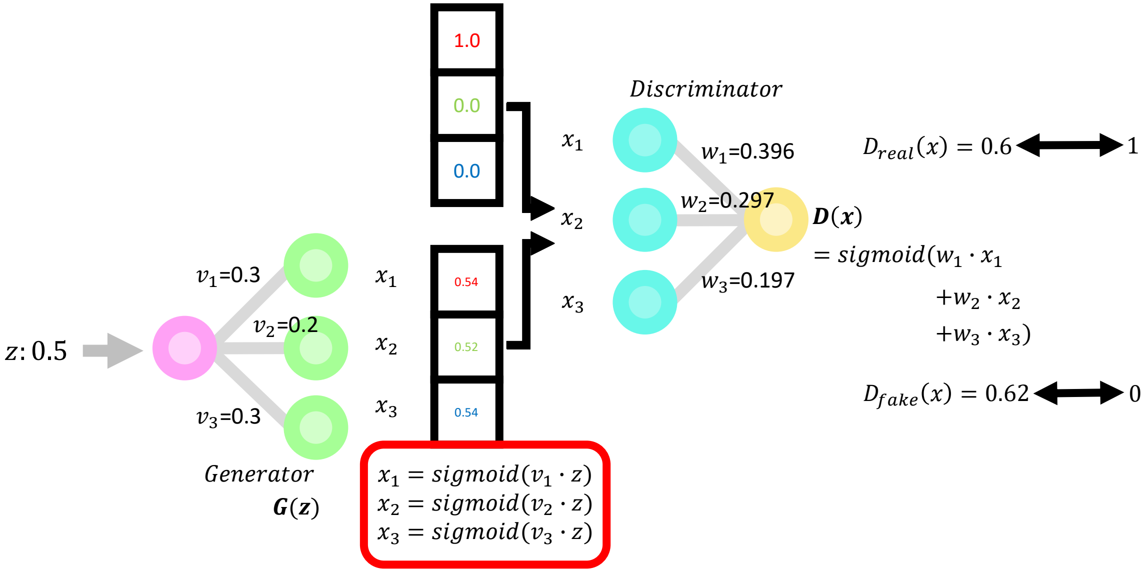
$$\frac{\partial L}{\partial v_i} = \frac{\partial L}{\partial D} \cdot \frac{\partial D}{\partial x_i} \cdot \frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \frac{-1}{D(G(z))} \cdot \frac{\partial D}{\partial x_i} \cdot \frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \frac{-1}{D(G(z))} \cdot D(G(z)) \cdot (1 - D(G(z))) \cdot w_i \cdot \frac{\partial x_i}{\partial v_i}$$

$$\frac{\partial L}{\partial v_i} = \frac{-1}{\cancel{D(G(z))}} \cdot \cancel{D(G(z))} \cdot (1 - D(G(z))) \cdot w_i \cdot G(z)(1 - G(z) \cdot z$$

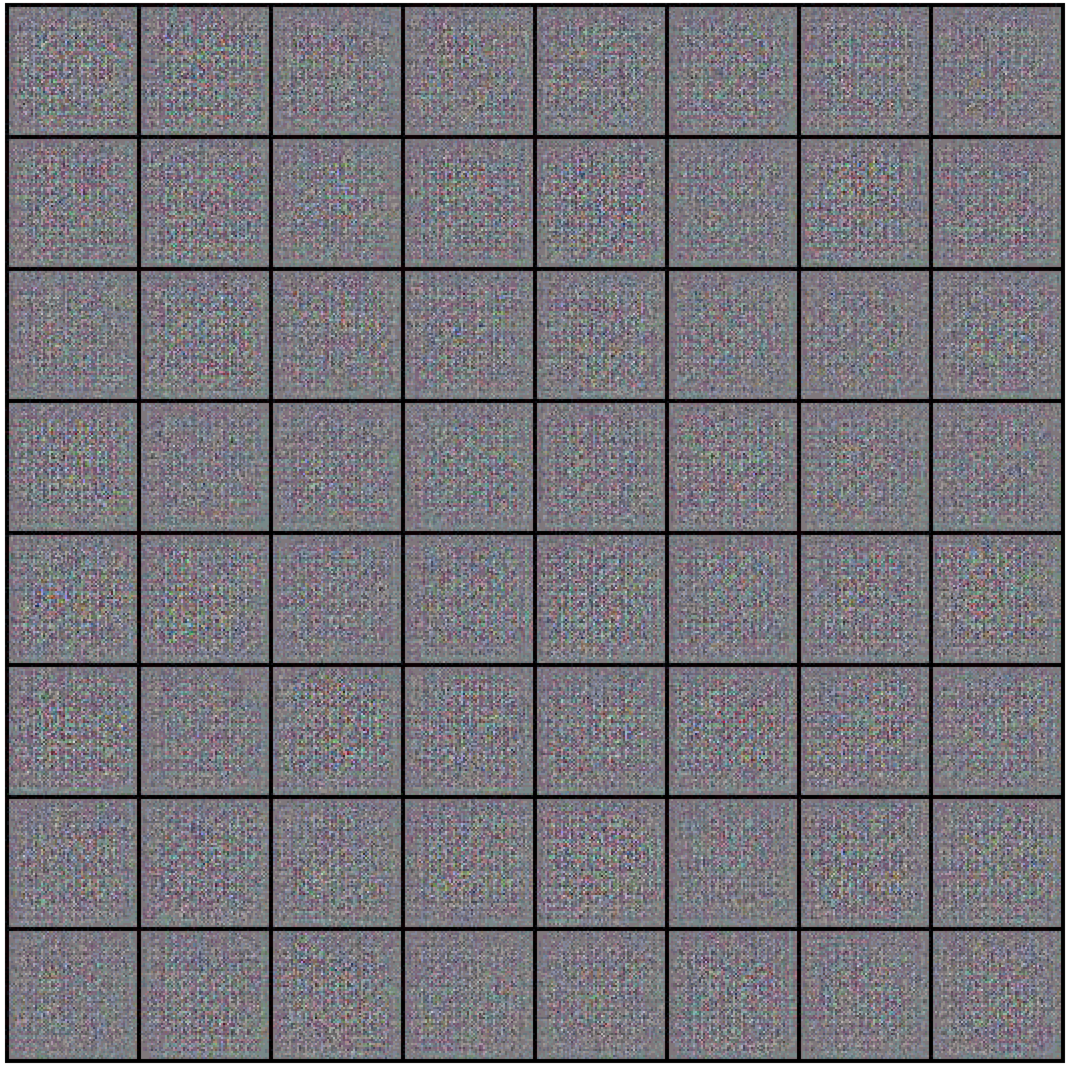
$$\frac{\partial L}{\partial v_i} = (1 - 0.62) \cdot \begin{bmatrix} 0.396 \\ 0.297 \\ 0.197 \end{bmatrix} \cdot \begin{bmatrix} 0.54 \\ 0.52 \\ 0.54 \end{bmatrix} \cdot (1 - \begin{bmatrix} 0.54 \\ 0.52 \\ 0.54 \end{bmatrix}) \cdot 0.5 = \begin{bmatrix} -0.019 \\ -0.014 \\ -0.009 \end{bmatrix}$$



신박AI

여기까지가 오늘 제가 준비한
GAN모델에 대한 대략적인 소개 및
예제입니다.

다음 시간에는 오늘 배운 내용을 바탕으로 다음 예제와 같이 fake 얼굴이미지를 생성해 내는 GAN모델을 작성해 보도록 하겠습니다.



신박AI

그럼 다음 시간에 또 만나요!

감사합니다!

좋은 하루 되세요!!

이 채널은 여러분의 관심과 사랑이 필요합니다

신박AI

‘좋아요’와 ‘구독’버튼은 강의 준비에 큰 힘이 됩니다!

신박AI

그리고 영상 자료를 사용하실때는
출처 ‘신박AI’를 밝혀주세요

신박AI