

5주차: GAN /Transformer

9차시:

◆ GAN

◆ Attention is All You Need

- ▷ LLM(Large Language Model)
- ▷ LLM(Large Language Model) 소개& 역사
- ▷ 파운데이션 모델(Foundation Model)
- ▷ LLM의 핵심 개념 이해
- ▷ LLM 분석과정 이해
- ▷ LLM 이용한 사례

GAN(**Generative Adversarial Network**) **적대적 생성** 신경망

- GAN은 실제로 가까운 이미지나 사람이 쓴 것과 같은 글 등 여러 가짜 데이터들을 **생성하는 모델**입니다.
- “적대적 생성 신경망”이라는 이름에서 알 수 있듯 GAN은 서로 다른 **두 개의 네트워크(Generator(G,생성모델/생성기)와 Discriminator(D,판별모델/판별기))**라는 서로 다른 2개의 네트워크를 적대적으로(adversarial) 학습시키며 실제 데이터와 비슷한 데이터를 생성(generative)해내는 모델 (위조범과 경찰)
- 이렇게 생성된 데이터에 정해진 label값이 없기 때문에 **비지도 학습 기반 생성모델**로 분류됩니다.

*META 인공지능 연구팀의 리더*

*얀 르쿤(Yann LeCun) 뉴욕대교수는 GAN(Generative Adversarial Network)을 가리켜 최근 10년간 머신러닝 분야에서 가장 혁신적인 아이디어라고 말했다.*

1. GAN의 역사

- **2014년: Ian Goodfellow와 동료들이 GAN을 처음 제안**한 논문 "Generative Adversarial Nets"가 발표되었습니다. 이 논문은 기존의 생성 모델과 달리 두 개의 모델을 서로 경쟁시키며 학습하는 혁신적인 아이디어를 제시하여 큰 관심을 받았습니다.
- 2015년 이후: GAN은 **이미지 생성, 비디오 생성, 음악 생성** 등 다양한 분야에서 적용되었으며, 이를 개선하는 여러 가지 변형 모델이 등장했습니다. 특히 **DCGAN(Deep Convolutional GAN), WGAN(Wasserstein GAN), StyleGAN** 등이 **대표적인 변형 모델**입니다.
- 현재: GAN은 **고화질 이미지 생성, 딥페이크 생성, 이미지 변환** 등 많은 응용 분야에서 성공적으로 사용되고 있습니다.

2. GAN의 작동 원리

- GAN은 **두 개의 신경망을 포함하는 모델**입니다.
- 각각의 네트워크는 서로 **적대적인 관계**에 있으며, 생성자(Generator)와 판별자(Discriminator)로 구성됩니다.

### (1) 생성자(Generator, G)

- 생성자의 목적은 실제 데이터와 유사한 가짜 데이터를 생성하는 것입니다.
- 이 신경망은 임의의 잠재 공간(Latent Space)에서 랜덤한 벡터를 받아들여, 이를 기반으로 가짜 데이터를 생성합니다.
- 생성자는 판별자가 가짜 데이터를 진짜로 오인하도록 학습합니다.
- 입력: 잠재 공간에서 샘플링한 랜덤 벡터  $z$
- 출력: 가짜 데이터  $G(z)$  (예: 가짜 이미지).

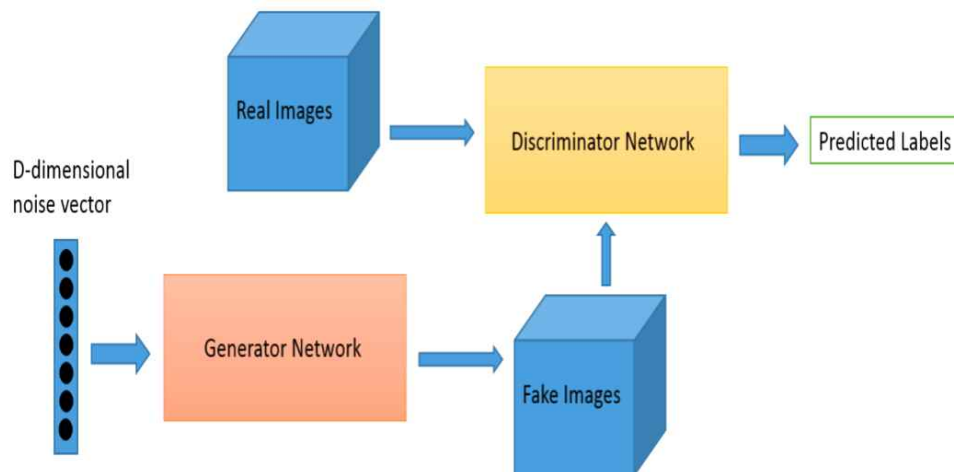
### (2) 판별자(Discriminator, D)

- 판별자의 목적은 입력된 데이터가 진짜인지 가짜인지 구분하는 것입니다.
- 생성자가 만든 가짜 데이터를 받으면, 이 데이터를 진짜 데이터와 구분하려고 합니다.
- 판별자는 진짜 데이터에 대해서는 1을, 가짜 데이터에 대해서는 0을 출력하도록 학습합니다.
- 입력: 진짜 데이터 또는 생성자가 만든 가짜 데이터.
- 출력: 1(진짜) 또는 0(가짜)라는 이진 분류 결과.

생성모델(G)의 목적은 진짜 분포에 가까운 **가짜분포를 생성하는 것 !!!**

판별모델(D)의 목적은 표본이 **가짜분포에 속하는지 진짜분포에 속하는지**를 **결정하는 것 !!!**

Hamed Alqahtani. 2019. An Analysis Of Evaluation Metric Of GANs)



### (3) 적대적 학습 과정

GAN의 학습은 두 신경망이 경쟁하며 서로 발전하는 과정으로 설명할 수 있습니다.

#### ▪ **생성자(G)의 목적**

- ▷ 판별자가 가짜 데이터를 진짜 데이터로 구분하지 못하게 하는 것. 즉, 판별자를 속이는 것이 목표입니다.
- ▷ 생성자는 잠재 공간  $z$  에서 벡터를 샘플링하고, 이를 통해 가짜 데이터  $G(z)$ 를 생성합니다.

▷ 생성자가 학습할 때는 판별자가 생성한 가짜 데이터를 진짜라고 판단하도록 유도합니다.

#### ■ 판별자(D)의 목적

▷ 진짜 데이터를 가짜 데이터와 정확히 구분하는 것. 즉, 진짜 데이터와 가짜 데이터를 완벽히 구별하는 것이 목표입니다.

▷ 판별자는 생성자의 가짜 데이터를 0으로, 진짜 데이터를 1로 구분하도록 학습합니다.

학습 과정:

- 먼저 판별자가 고정된 상태에서 생성자를 학습합니다. 생성자가 만들어낸 가짜 데이터에 대해 판별자가 실수하도록 학습합니다.
- 그런 다음 생성자를 고정하고, 판별자를 학습시킵니다. 판별자가 진짜와 가짜를 잘 구분할 수 있도록 학습합니다.
- 이 과정을 반복하면서 두 네트워크는 서로의 성능을 개선하게 됩니다.

(4) 손실 함수

GAN의 학습 과정에서 두 네트워크는 서로 반대의 목표를 가지기 때문에, 이 게임을 수학적으로 최소/최대 게임으로 표현할 수 있습니다.

#### ▷ 판별자(D)의 손실

- 판별자는 진짜 데이터와 생성된 가짜 데이터를 구분하는 역할을 합니다.
- 진짜 데이터를 진짜로, 가짜 데이터를 가짜로 정확하게 구분하는 것이 목표이기 때문에 이진 분류 손실 함수(Binary Cross-Entropy Loss)를 사용하여 학습합니다.
- 진짜 데이터에 대해서는 1에 가까운 확률을 출력해야 하며, 가짜 데이터에 대해서는 0에 가까운 확률을 출력해야 합니다.

판별자 손실 함수는 다음과 같이 정의

$$D\_loss = -\mathbb{E}_{x \sim p_{data}} [\log D(x)] - \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))]$$

$D(x)$ : 진짜 데이터  $x$ 에 대해 판별자가 예측한 값 (진짜일 확률)

$D(G(z))$ : 생성자  $G$ 가 생성한 가짜 데이터  $G(z)$ 에 대해 판별자가 예측한 값 (진짜일 확률)

첫 번째 항은 진짜 데이터를 진짜로 예측하도록 학습하고,

두 번째 항은 가짜 데이터를 가짜로 예측하도록 학습합니다.

판별자는 진짜 데이터에서 높은 값을, 가짜 데이터에서는 낮은 값을 예측하려고 합니다.

판별자는 진짜와 가짜 데이터를 구별하며, 진짜는 진짜로, 가짜는 가짜로 인식하도록 학습됩니다.

#### ▷ 생성자(G)의 손실

- 생성자의 목표는 판별자를 속여서 가짜 데이터를 진짜로 인식하게 하는 것입니다.
- 그래서 생성자는 판별자가 가짜 데이터를 진짜로 잘못 예측하도록 학습합니다.
- 생성자는 판별자를 속일 수 있는 더 진짜 같은 데이터를 생성하는 방향으로 학습되며, Binary Cross-Entropy Loss를 사용합니다.

▷ 생성자 손실 함수는 다음과 같이 정의

$$G\_loss = -\mathbb{E}_{z \sim p_z} [\log D(G(z))]$$

$D(G(z))$ : 생성자가 생성한 데이터  $G(z)$ 에 대해 판별자가 예측한 값  
생성자는 판별자가 생성된 가짜 데이터를 진짜로 예측하게 만들어야 하므로,  
 $D(G(z))$ 가 1에 가까워지도록 학습됩니다.

생성자는 판별자가 생성한 가짜 데이터를 진짜로 분류하도록 유도하는 방향으로 학습됩니다.  
GAN은 이 최소/최대 게임을 통해, 생성자는 점점 더 현실적인 가짜 데이터를 생성하게 되고,  
판별자는 점점 더 가짜와 진짜를 잘 구분하게 됩니다.

*생성자는 판별자를 속여 가짜 데이터를 진짜로 인식하게 만드려고 학습됩니다.*

: Ian J. Goodfellow. 2014. Generative Adversarial Nets)

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

$V(D, G)$ 의 값은 확률값으로 도출

D의 관점에서 실제 데이터(x)를 입력하면  $D(x)$ 가 커지면서 log값이 커지면서 높은 확률값이 나오도록 하고, 가짜 데이터( $G(z)$ )를 입력하면 log값이 작아짐에 따라 낮은 확률값이 나오도록 학습됩니다. 다시 말해 D는 실제 데이터와 G가 만든 가짜 데이터를 잘 구분하도록 조금씩 업데이트되는 것입니다.

G에서는 Zero-Mean Gaussian 분포에서 노이즈  $z$ 를 멀티레이어 퍼셉트론에 통과시켜 샘플들을 생성하며 이 생성된 가짜 데이터  $G(z)$ 를 D에 input으로 넣었을 때 실제 데이터처럼 확률이 높게 나오도록 학습됩니다.

즉  $D(G(z))$ 값을 높도록, 그리고 전체 확률 값이 낮아지도록 하는 것이며 이는 다시 말해 G가 'D가 잘 구분하지 못하는' 데이터를 생성하도록 조금씩 업데이트되는 것입니다.

실제 학습을 진행할 때는 G와 D 두 네트워크를 동시에 학습시키지 않고 하나의 네트워크를 고정한 상태에서 다른 한 네트워크를 업데이트하는 방식으로 따로따로 업데이트합니다.

### 3. GAN의 응용 및 발전

#### (1) 응용 분야

- 이미지 생성: GAN은 고화질 이미지를 생성하는 데 널리 사용됩니다. 예를 들어, Face Generation에서는 가짜 인물 사진을 만들어낼 수 있습니다.
- 비디오 생성: GAN을 사용하여 현실적인 비디오 장면을 생성하거나, 딥페이크(Deepfake) 기술을 적용할 수 있습니다.
- 이미지 변환: CycleGAN이나 Pix2Pix 같은 GAN 모델은 하나의 이미지를 다른 이미지로 변환하는 데 사용됩니다. 예를 들어, 여름 풍경을 겨울 풍경으로 바꾸거나, 흑백 이미지를 컬러로 변환하는 것이 가능합니다.

#### (2) 주요 발전

DCGAN (Deep Convolutional GAN): CNN 구조를 기반으로 한 GAN 모델로, 더 깊고 복잡한 이미지 생성이 가능해졌습니다.

WGAN (Wasserstein GAN): 기존 GAN의 불안정한 학습 문제를 해결하고자 등장한 변형. Wasserstein 거리 개념을 도입해 학습을 안정적으로 만들었습니다.

StyleGAN: 고화질 이미지 생성에 뛰어난 성능을 발휘하며, 특히 얼굴 이미지 생성에 특화된 모델입니다.

#### 4. GAN의 한계

- 모델 불안정성: GAN의 학습은 때때로 불안정할 수 있으며, 생성자와 판별자가 균형을 이루지 못할 때 학습이 실패할 수 있습니다.
- 모드 붕괴(Mode Collapse): 생성자가 일부 패턴만 학습하고 다른 패턴을 무시하는 현상이 발생할 수 있습니다. 이는 다양성 있는 데이터를 생성하지 못하는 문제입니다.
- 훈련의 복잡성: GAN은 매우 많은 훈련 데이터와 시간이 필요하며, 하이퍼파라미터 튜닝이 복잡한 경우가 많습니다.

#### 조건부 생성적 적대 신경망(Conditional Generative Adversarial Network, cGAN)

- GAN(Generative Adversarial Network)의 확장된 형태로, 생성된 데이터가 특정 조건을 따르도록 설계된 모델입니다.
- cGAN은 조건부 데이터를 입력받아 원하는 속성을 가진 이미지를 생성할 수 있도록 학습됩니다.

GAN과는 ‘샘플링을 어디서 해오는지’와 ‘데이터셋에 라벨이 있어야 하는지’, 2가지 측면에서 차이가 있습니다.

cGAN을 이용하면 인위적으로 원하는 클래스의 데이터를 생성할 수 있으며 생성기와 판별기를 훈련하는 데에 label을 사용합니다.

쉽게 말하면 Generator와 Discriminator에 특정 condition을 나타내는 정보 y를 추가해주는 것인데, 이 때 y는 형태가 정해진 것은 아니기 때문에 다양한 형태를 가질 수 있습니다.

예를 들어, 필기된 숫자를 인식하는 MNIST 데이터에서 원하는 숫자를 생성하고 싶다면 숫자의 class에 해당하는 label을 추가 정보 y로 입력해주는 것입니다.

#### cGAN 구조

기본적인 GAN 구조에 조건을 추가한 형태로, 생성자(Generator)와 판별자(Discriminator) 모델에 추가 정보를 함께 입력받아 학습합니다.

생성자: 무작위 노이즈 벡터와 조건  $y$ 를 입력받아, 조건에 맞는 가짜 데이터를 생성합니다. 예를 들어, 숫자 이미지를 생성하는 경우, 조건  $y$ 는 특정 숫자 클래스(0-9)가 될 수 있습니다.

판별자: 실제 데이터와 조건이 일치하는지 여부를 판단하며, 조건에 맞는 진짜/가짜를 분류하는 역할을 합니다.

#### cGAN의 손실 함수

cGAN의 손실 함수는 기존 GAN과 비슷하지만, 생성자와 판별자 모두 조건  $y$ 를 반영합니다.

$$D\_loss = -\mathbb{E}_{x \sim p_{data}} [\log D(x|y)] - \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z|y)|y))]$$

$$G\_loss = -\mathbb{E}_{z \sim p_z} [\log D(G(z|y)|y)]$$

$G(z|y)$ : 조건  $y$  를 고려하여 생성자가 생성한 가짜 데이터.

$D(x|y)$ : 판별자가 입력받은 데이터  $x$  와 조건  $y$  가 일치하는지를 평가하는 확률.

### cGAN 응용 사례

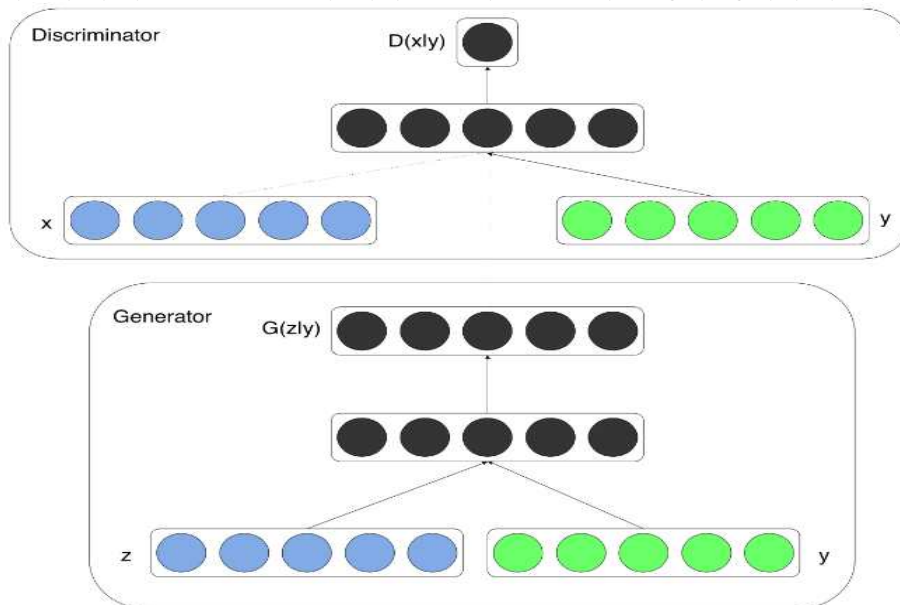
- 이미지-텍스트 생성: 특정 텍스트 설명에 따라 이미지를 생성하는 작업 (예: "파란 하늘과 녹색 산이 있는 풍경").
- 이미지 분할: 의료 영상에서 특정 부위를 분할하거나, 위성 사진에서 도시와 같은 특정 영역을 식별하는 데 사용.
- 이미지 변환: 흑백 이미지를 컬러 이미지로 변환하거나, 스케치 이미지를 실사 이미지로 변환하는 작업.

### cGAN의 장점

cGAN은 생성된 데이터에 대해 명확한 조건을 부여하여 특정한 특성을 반영할 수 있기 때문에, 제어 가능한 이미지 생성이 가능하여 다양한 실무 및 연구 분야에 응용될 수 있습니다.

### 평가 지표 (Evaluation Metrics)

GAN 창시자 Ian Goodfellow는 2014년 GAN을 소개한 이후 2016년에 학습(training)방식을 향상시키는 기술을 추가로 발표했는데 논문에서 저자들은 "GAN은 object function이 없으며 이는 서로 다른 모델들의 퍼포먼스를 비교하는 것을 어렵게 만드는 요인이다"라고 언급했습니다. 즉 주어진 GAN 모델에 대해 일반적으로 합의된 평가 방식이 없다는 것인데, 이는 학습



(training)이 실행되고 있는 동안 최종적인 GAN 모델을 선택할 때, GAN 모델의 활용성을 증명하기 위해 생성된 이미지를 선택할 때, GAN 모델 아키텍처 간 비교할 때 등의 경우에 이슈

가 되곤 합니다.

2018년 GAN의 평가 지표의 장단점을 기술한 논문에서 Ali Borji는 “GAN 모델의 강점과 한계점을 반영하는 가장 적절한 지표에 대해서는 아직도 의견의 일치가 이루어지지 않는다고 언급했습니다. 이와 같이 GAN 모델은 생성 대상의 도메인 문맥에 따라 생성된 이미지의 퀄리티에 기반하여 평가되고는 했습니다.

그 이후인 2019년 Macquarie 대학교의 Hamed Alqahtani가 발표한 논문에서는 GAN 모델의 평가지표에 대한 연구를 통해 10여 개의 GAN 모델의 평가지표를 제시하였습니다. Hamed에 따르면 GAN의 평가지표는 크게 정성적인 지표와 정량적인 지표 2가지로 분류되며 일반적으로 정성적인 지표는 사람이 이미지를 평가하므로 비용이 많이 든다고 합니다.

정성적인 지표에는 Nearest Neighbor, Rating and Preference Judgement, Rapid Scene Categorization이 있으며 정량적인 지표에는 FID(Frechet Inception Distance), IS(Inception Score), Mode Score, Maximum Mean Discrepancy 등이 있습니다. 우리는 아래에서 NVIDIA의 논문에 모델 비교시 핵심적으로 사용했던 IS와 자주 쓰이는 FID 2가지에 대해서 알아보겠습니다.

#### Inception Score (IS)¶

Inception Score는 Salimans et al.에 의해 제안되었는데 GAN의 평가에 널리 쓰이는 지표입니다. 이 지표는 클래스 label과 관련하여 특징적인 속성들을 잡아내기 위해 pre-trained 신경망을 사용합니다. 아래는 IS 값을 도출하는 수식인데 샘플의 조건부 분포  $p(y|x)$ 와 모든 샘플에서 얻은 주변분포  $p(y)$  사이의 평균적인 KL 발산 정도(Average KL Divergence)를 측정하는 것이며 이 값이 높을수록 좋은 성능을 낸다고 해석할 수 있습니다. 하지만 IS에는 실제 샘플 대신 생성된 이미지를 사용해 계산하고 클래스 당 하나의 이미지만 생성하면 다양성이 부족하더라도  $p(y)$ 가 균등 분포에 가깝게 나오기 때문에 성능을 왜곡할 수 있다는 단점이 있습니다.

$$\exp(E_x[KL(p(y|x) || p(y))]) = \exp(H_y) - E_x[H(y|x)]$$

Hamed Alqahtani. 2019. An Analysis Of Evaluation Metric Of GANs)

#### Frechet Inception Distance (FID)¶

Frechet Inception Distance는 생성되는 이미지의 퀄리티 일관성을 유지하기 위해 이용되는 지표입니다. 실제 데이터의 분포를 활용하지 않는 단점을 보완하여 실제 데이터와 생성된 데이터에서 얻은 feature의 평균과 공분산을 비교하는 방식이며 FID가 낮을수록 이미지의 퀄리티가 더 좋아지는데 이는 실제 이미지와 생성된 이미지의 유사도가 높아지는 것을 말합니다. 즉 쉽게 말해 FID는 생성된 샘플들의 통계와 실제 샘플들의 통계를 비교하는 것입니다.

문제: 손글씨 숫자 생성 (MNIST 데이터셋)

GAN을 사용하여 MNIST 데이터셋과 유사한 새로운 손글씨 숫자 이미지를 생성할 것입니다. MNIST 데이터셋은 0에서 9까지의 손글씨 숫자 이미지로 구성되어 있습니다. GAN 모델의 목표는 무작위 노이즈에서 생성된 이미지를 통해 실제 데이터셋의 이미지와 비슷한 새로운 숫자 이미지를 만드는 것입니다.

## 2. 아키텍처 개요

전형적인 GAN은 두 가지 주요 구성 요소로 이루어집니다:

생성자(Generator, G): 랜덤한 노이즈를 입력으로 받아 가짜 숫자 이미지를 생성합니다.

판별자(Discriminator, D): MNIST 데이터셋에서 가져온 실제 이미지나 생성자가 만든 가짜 이미지를 받아, 이를 진짜인지 가짜인지 분류하려고 합니다.

이 두 네트워크는 \*\*적대적 학습(Adversarial Training)\*\*을 진행합니다:

생성자는 진짜처럼 보이는 이미지를 생성하여 판별자를 속이려고 합니다.

판별자는 데이터셋에서 가져온 진짜 이미지와 생성자가 만든 가짜 이미지를 구분하려고 합니다.

학습 과정:

생성자는 시간이 지남에 따라 점점 더 현실적인 이미지를 생성하는 법을 배우게 됩니다.

판별자는 점점 더 진짜와 가짜를 구분하는 능력이 향상되며, 결국 생성자는 판별자를 속일 수 있을 정도로 현실적인 이미지를 만들게 됩니다.

## 3. GAN 작동 과정 (단계별)

### 1단계: 입력 데이터

실제 이미지: MNIST 손글씨 숫자 데이터셋.

랜덤 노이즈: 생성자에 입력할 무작위 벡터로 생성됩니다.

### 2단계: 생성자의 역할

생성자는 랜덤한 노이즈를 시작점으로 하여 가짜 이미지를 생성합니다. 처음에는 단순한 노이즈를 생성하지만, 학습이 진행되면서 점점 실제 손글씨 숫자와 유사한 이미지를 생성하게 됩니다.

생성자의 입력: 랜덤 노이즈 벡터  $z$  여기서  $z \sim N(0,1)$ (평균 0, 표준편차 1의 정규분포).

생성자의 출력: 28x28 픽셀 크기의 숫자 이미지 (가짜 이미지).

$G(z)$ =Fake Image

### 3단계: 판별자의 역할

판별자는 MNIST 데이터셋에서 가져온 실제 이미지와 생성자가 만든 가짜 이미지를 받아 이를 진짜인지 가짜인지 분류합니다.

판별자의 입력: 실제 이미지

$x$  와 가짜 이미지  $G(z)$  판별자의 출력: 이미지가 진짜(1)인지 가짜(0)인지 확률 점수

$$D(x) = 1 \quad (\text{real images})$$

$$D(G(z)) = 0 \quad (\text{fake images})$$

### 4단계: 적대적 학습



판별자 학습: 판별자는 실제 이미지를 진짜로, 가짜 이미지를 가짜로 정확하게 분류하는 법을 학습합니다.

실제 이미지에 대한 손실:

$$L_{\text{real}} = \log(D(x))$$

가짜 이미지에 대한 손실:

$$L_{\text{fake}} = \log(1 - D(G(z)))$$

판별자의 총 손실:

$$L_D = L_{\text{real}} + L_{\text{fake}}$$

생성자 학습: 생성자는 판별자를 속여 가짜 이미지를 진짜로 분류하게 만들기 위해 학습됩니다.

생성자의 손실:

$$L_G = \log(D(G(z)))$$

생성자의 목표는 이 손실을 최대화하여 판별자가 가짜 이미지를 진짜로 분류하도록 유도하는 것입니다.

#### 4. GAN 학습 과정

- 단계 1: 생성자를 고정하고, 판별자를 업데이트하여 진짜와 가짜 이미지를 더 효과적으로 구분할 수 있도록 합니다.
- 단계 2: 판별자를 고정하고, 생성자를 업데이트하여 더 좋은 가짜 이미지를 생성하여 판별자를 속이도록 합니다.
- 단계 3: 이 교차 학습 과정을 여러 번 반복합니다(에포크).

시간이 지남에 따라 생성자는 점점 더 현실적인 이미지를 만들게 되고, 판별자는 더 나은 구분 능력을 갖추게 됩니다. 결국 생성자가 만든 이미지는 육안으로 보았을 때 실제 MNIST 숫자와 구별할 수 없을 정도로 현실적입니다.

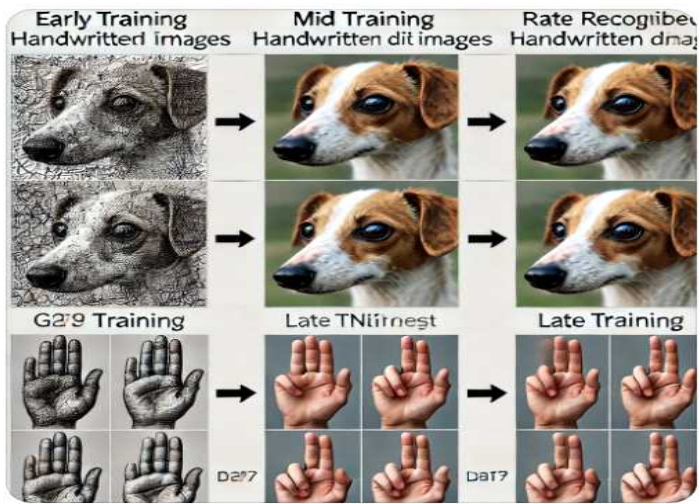
#### 5. 결과: 생성된 이미지

학습이 진행될수록 GAN은 점점 더 현실적인 이미지를 생성하게 됩니다. 초기에는 생성된 이미지가 단순한 노이즈일 뿐이지만, 시간이 지남에 따라 생성자는 더욱 선명하고 실제와 유사한 숫자 이미지를 만들어냅니다.

- 초기 학습: 생성된 이미지는 노이즈나 흐릿한 모양으로 숫자와 유사하지 않습니다.
- 중간 학습: 일부 숫자가 나타나지만, 왜곡되거나 불분명한 경우가 많습니다.
- 후기 학습: 생성자는 선명하고 실제 손글씨와 구별하기 어려운 숫자 이미지를 생성할 수 있습니다.

#### 6. 학습 단계별 생성 이미지 예시

GAN 모델이 시간이 지나면서 출력 이미지를 어떻게 발전시키는지 시각적 예시



(파이썬 실습)

GAN