



생성 모델

알파팀

김민 김민우 김현우 임지훈 이수린 채희지

INDEX

1. 생성 모델의 정의 및 개념
2. 고전적 생성 모델
3. 이미지 생성 모델
4. 텍스트 생성 모델

1

생성 모델의 정의 및 개념

생성 모델(Generative Model)이란?

생성 모델 (Generative Model)

주어진 데이터를 학습하여, 학습된 데이터의 분포를 따르는
유사한 데이터를 생성하는 모델

▼ 생성 모델의 종류

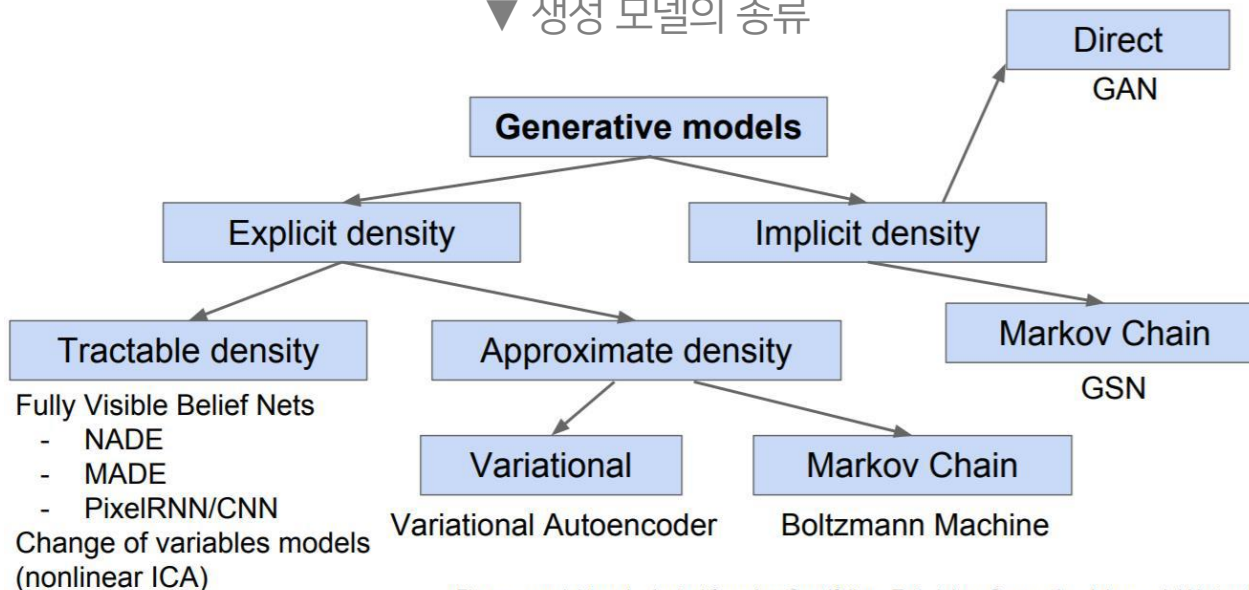


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

생성 모델(Generative Model)이란?

생성 모델 (Generative Model)

주어진 데이터를 학습하여, 학습된 데이터의 분포를 따르는
유사한 데이터를 생성하는 모델

▼ 생성 모델의 종류

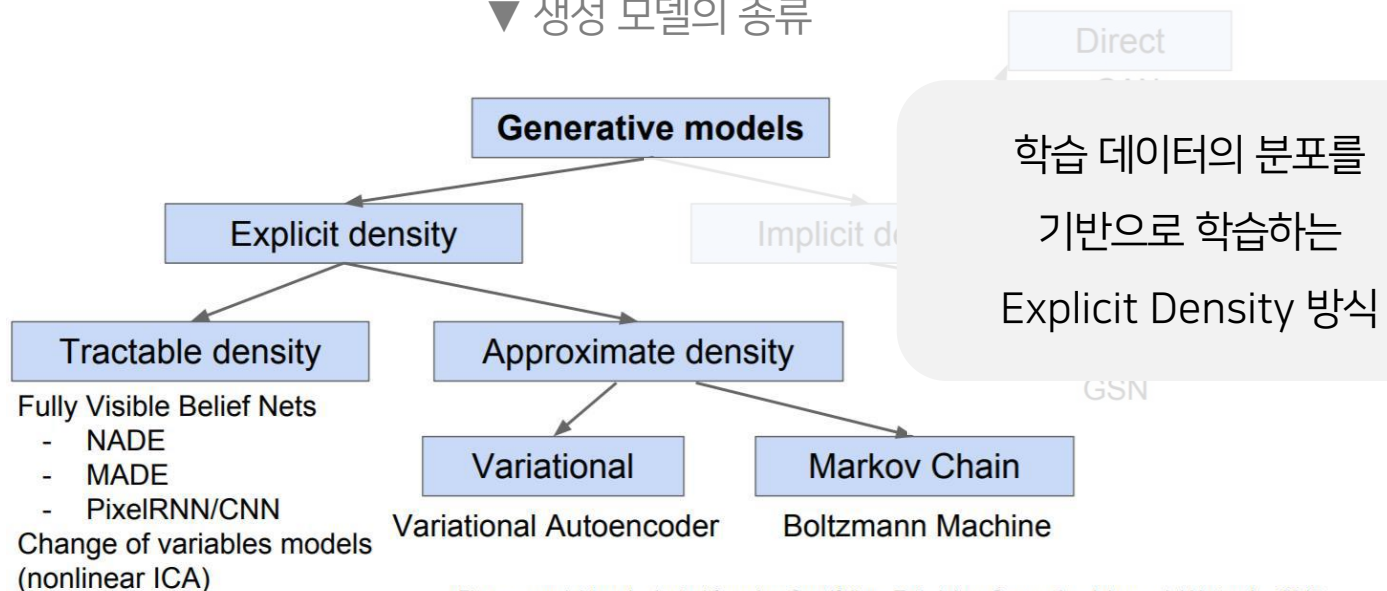


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

생성 모델(Generative Model)이란?

생성 모델 (Generative Model)

주어진 데이터를 학습하여, 학습된 데이터의 분포를 따르는 유사한 데이터를 생성하는 모델

▼ 생성 모델의 종류

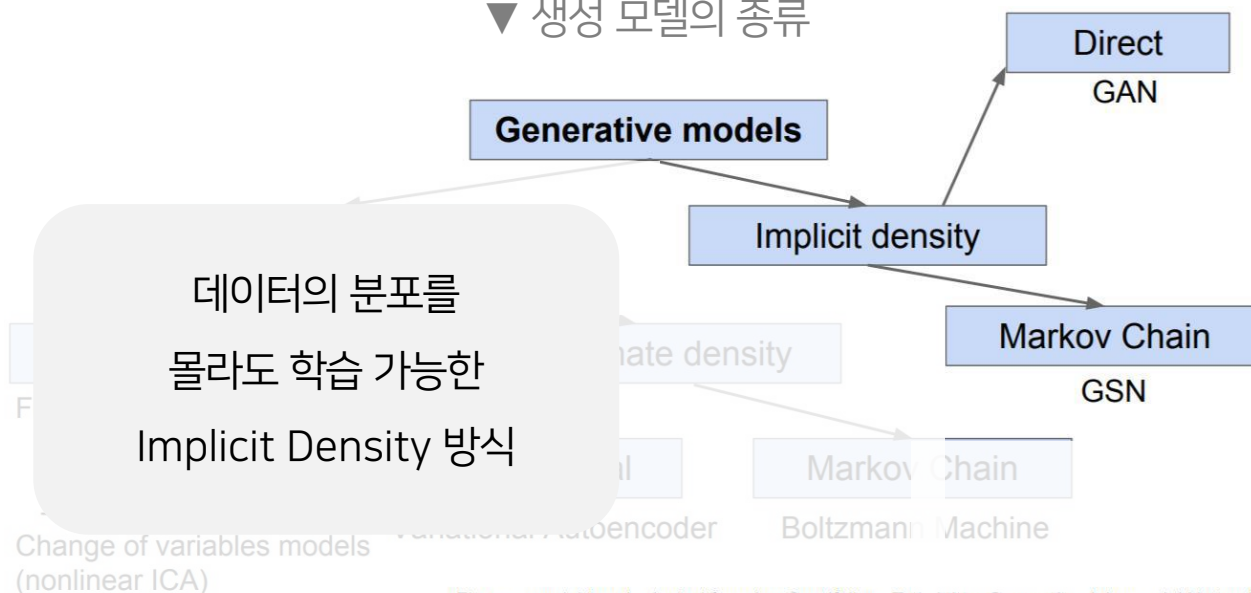
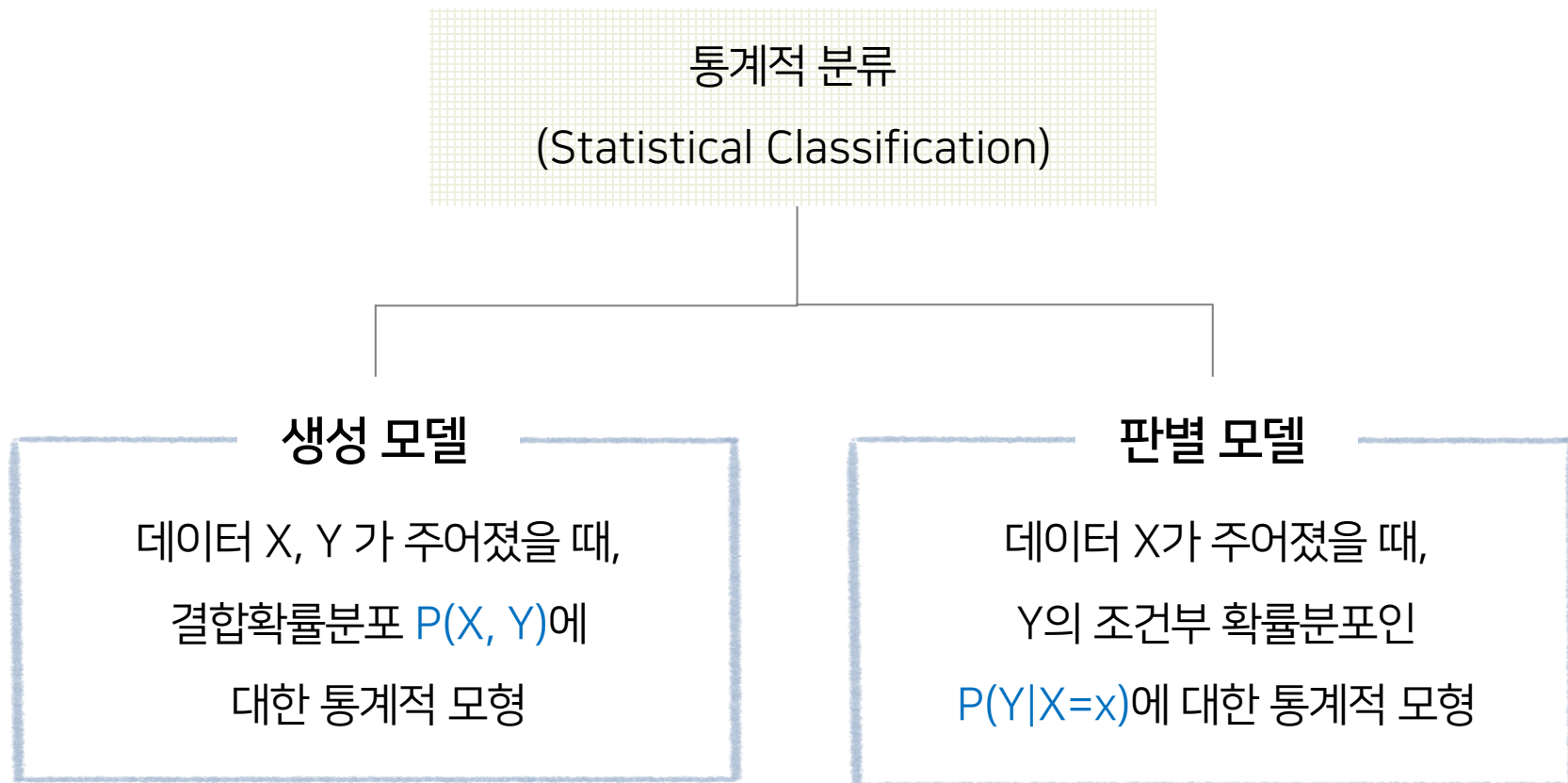


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

1

생성 모델의 정의 및 개념

생성 모델 vs. 판별 모델



1

생성 모델의 정의 및 개념

생성 모델 vs. 판별 모델

통계적 분류
(Statistical Classification)

생성 모델

데이터 X, Y 가 주어졌을 때,
결합확률분포 $P(X, Y)$ 에
대한 통계적 모형

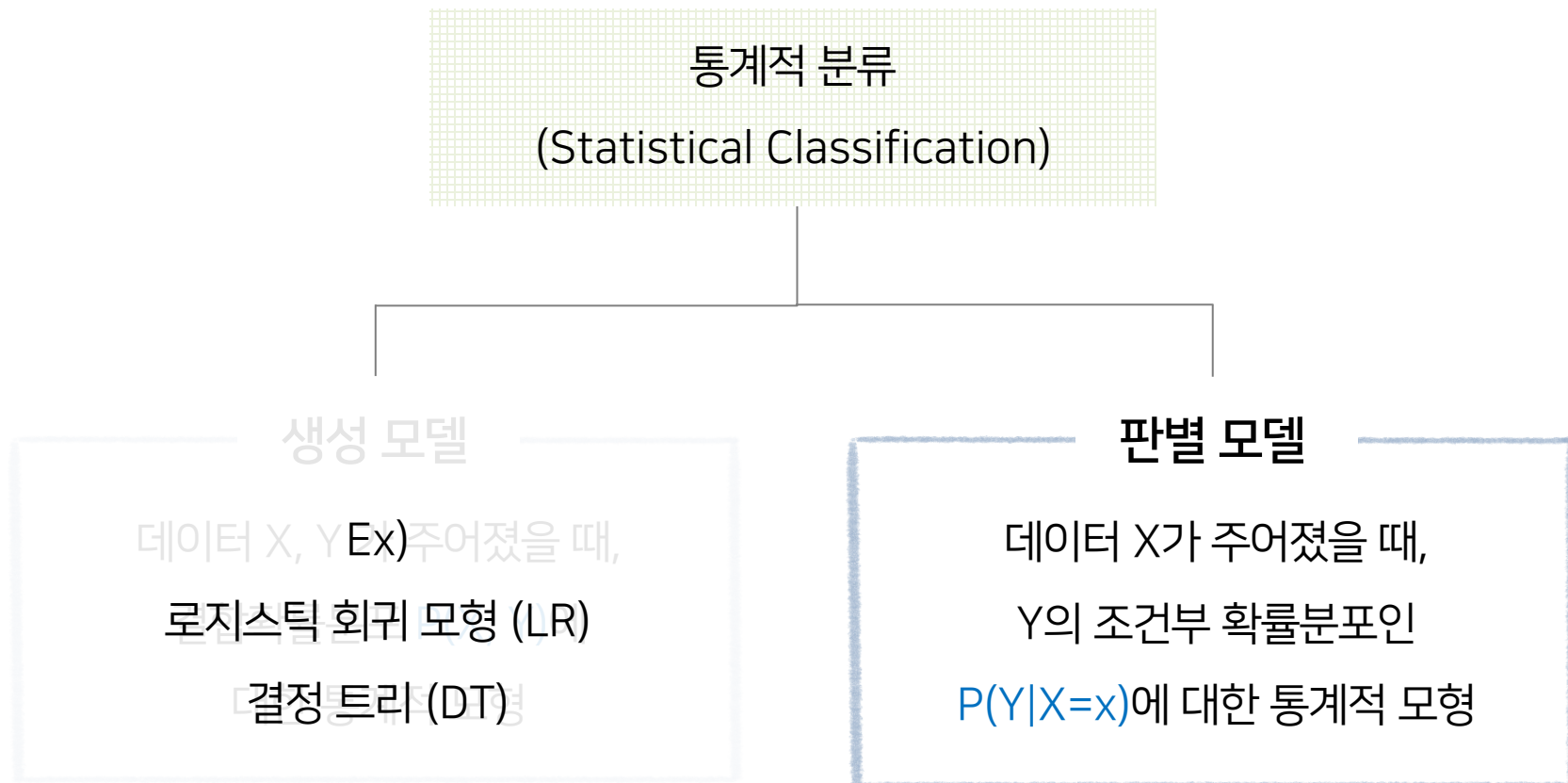
판별 모델

데이터 X 가 주어졌을 때,
나이브 베이즈 분류기 (NB)
 $P(Y|X)$ 선형 판별 분석 (LDA) 모형

1

생성 모델의 정의 및 개념

생성 모델 vs. 판별 모델



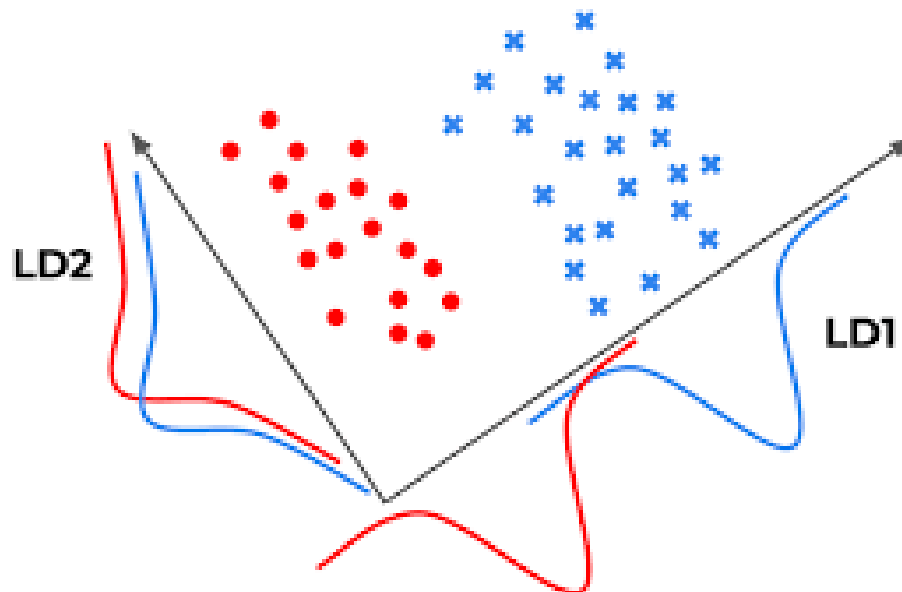
2

고전적 생성 모델

LDA (Linear Discriminant Analysis)

LDA

선형 판별 분석(LDA)은 클래스를 구분하는 변수 간의 선형 결합을 찾는 알고리즘으로, 분류 또는 차원 축소에 활용 가능



LDA (Linear Discriminant Analysis)

LDA

선형 판별 분석(LDA)은 클래스를 구분하는 변수 간의
선형 결합을 찾는 알고리즘으로, 분류 또는 차원 축소에 활용 가능



아래의 두 가지 성질을 만족하는 선형 결합을 찾는 것이 목표!



클래스 간 데이터의 거리가 최대가 되게 하는 벡터



클래스 내 데이터의 분산이 최소가 되게 하는 벡터

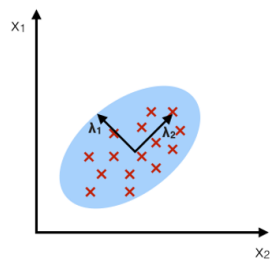
LDA (Linear Discriminant Analysis)

PCA(주성분 분석)와의 차이점

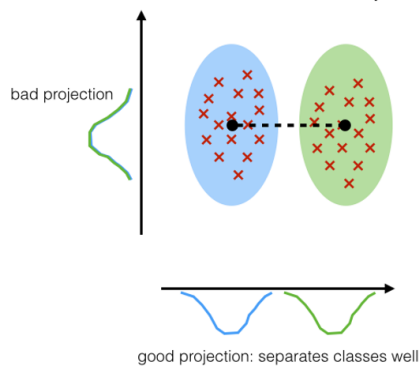
PCA(주성분 분석)	LDA(선형 판별 분석)
클래스 간 차이 고려 X	클래스 간 차이 고려 O
비지도학습 알고리즘	지도학습 알고리즘

PCA:

component axes that
maximize the variance

**LDA:**

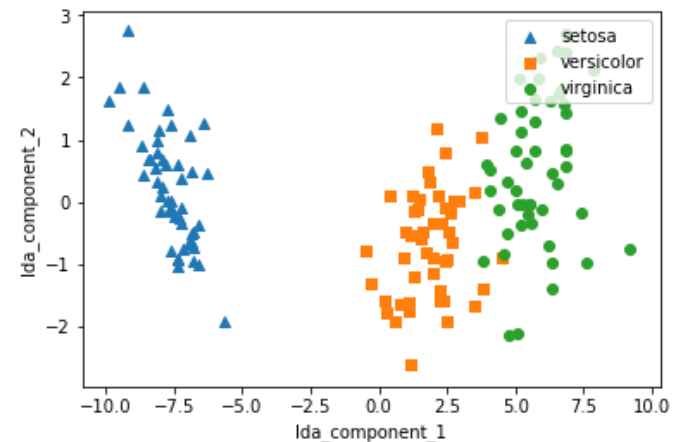
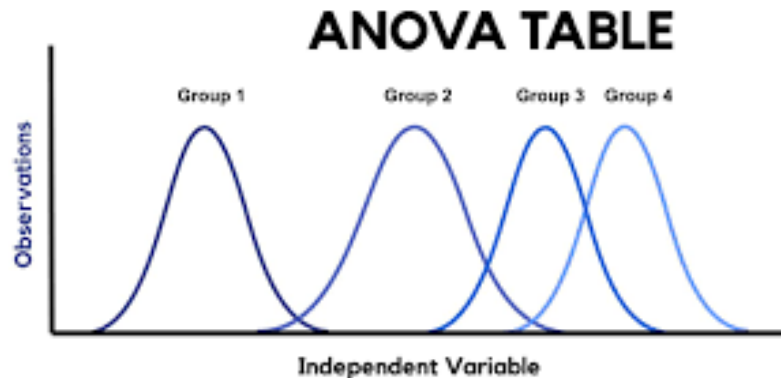
maximizing the component
axes for class-separation



LDA (Linear Discriminant Analysis)

ANOVA(분산 분석)와의 차이점

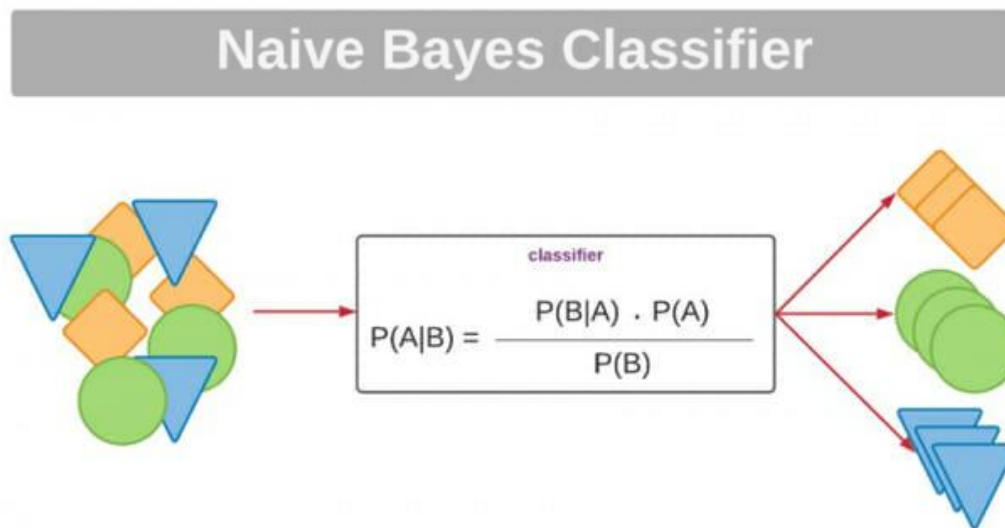
	ANOVA (분산 분석)	LDA (선형 판별 분석)
독립변수	범주형 변수	연속형 변수
종속변수	연속형 변수	범주형 변수



Naive-Bayes Classifier

Naive-Bayes Classifier

나이브 베이즈 분류기는 베이즈 정리에 기반한 선형 확률 분류 모형으로,
변수 간의 독립성을 가정하는 베이즈 정리에 기반함



Naive-Bayes Classifier

Naive-Bayes Classifier

나이브 베이즈 분류기는 베이즈 정리에 기반한 선형 확률 분류 모형으로,
변수 간의 독립성을 가정하는 베이즈 정리에 기반함

나이브 베이즈 분류기의 특징



비교적 간단한 알고리즘에도 불구하고 우수한 성능을 보임



소수의 데이터만 가지고도 모델 학습이 가능

Naive-Bayes Classifier

베이즈 정리

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$



사후확률

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

Naive-Bayes Classifier

베이즈 정리

$$P(A) = \text{prior}$$

사건 A가 일어날 확률
A는 각 클래스를 나타냄

$$P(B) = \text{evidence}$$

사건 B가 일어날 확률
B는 입력 데이터(X 벡터)를 나타냄

$$P(A|B) = \text{posterior}$$

사건 B가 일어났을 때,
사건 A가 일어날 확률

$$P(B|A) = \text{likelihood}$$

사건 A가 일어났을 때,
사건 B가 일어날 확률

3

이미지 생성 모델

GAN

GAN (Generative Adversarial Network)

서로 다른 두 개의 네트워크를 적대적으로(adversarial) 학습시키며
실제 데이터와 비슷한 데이터를 생성(generative)해내는 모델

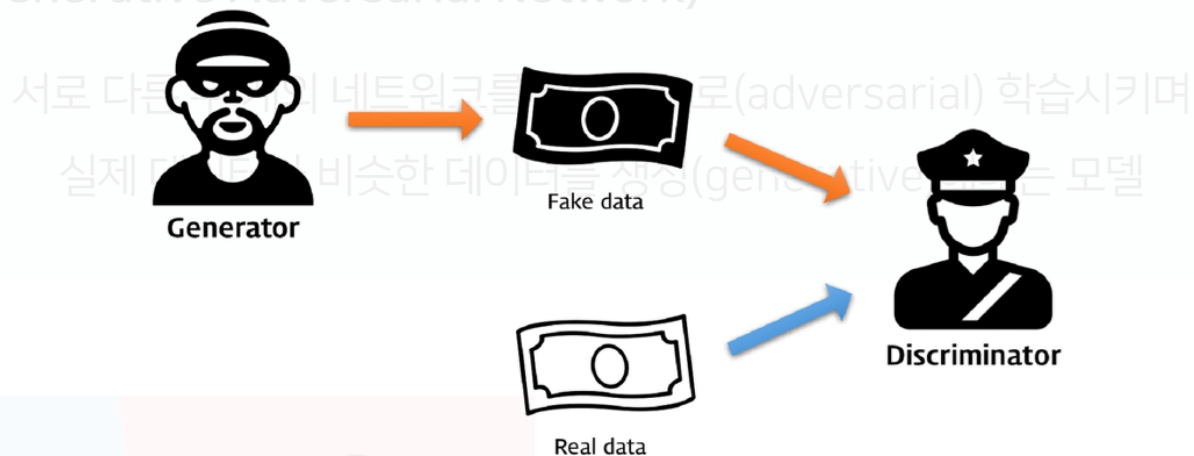


위조지폐범: 진짜 지폐와 비슷한 가짜 지폐 만들
경찰: 이 지폐가 진짜 지폐인지 위조 지폐인지 구분

위조지폐범과 경찰은 적대적인 관계

GAN

GAN (Generative Adversarial Network)



위조지폐범(Generator)과 경찰(Discriminator)가

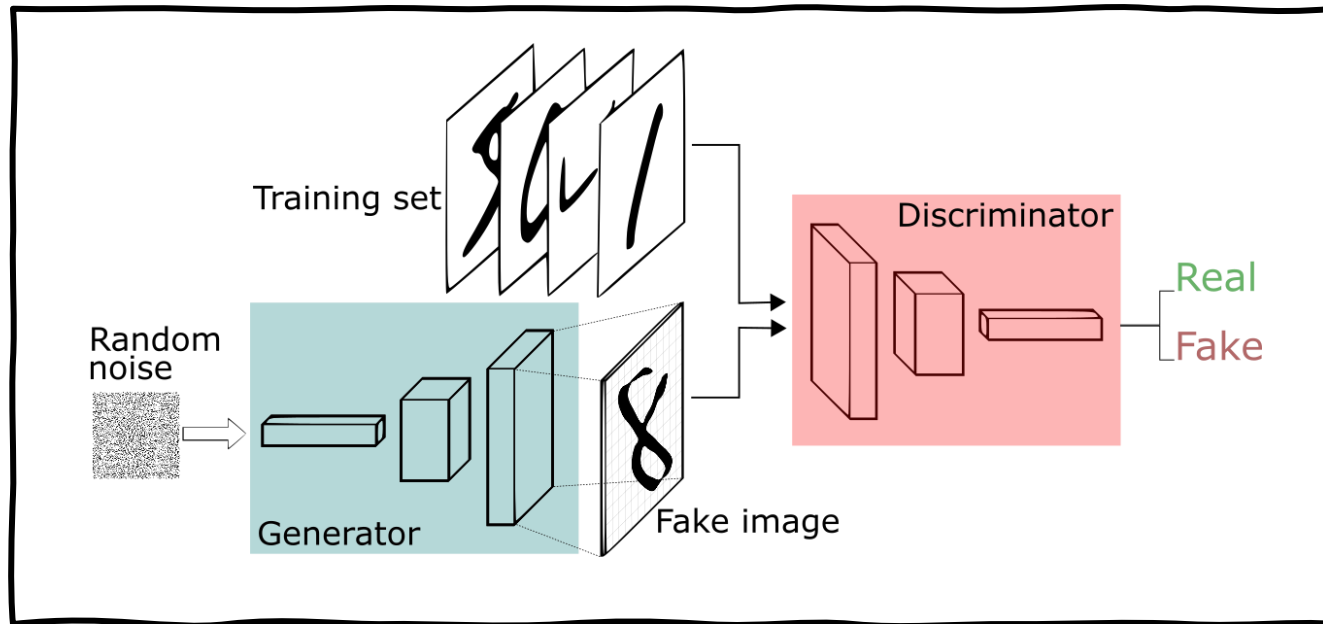
서로 위조지폐를 생성하고 구분하는 것을 반복하는 min-max game

경찰
(=분류모델)

위조지폐범
(=생성모델)

위조지폐범과 경찰은 적대적인 관계

GAN



Generator : 진짜 분포에 가까운 가짜 분포 생성

Discriminator : 표본이 진짜 분포에 속하는지 가짜 분포에 속하는지 판단

GAN

$$\min_G \max_D V(D, G) = E_{x \sim P_{data}}(x) [\log D(x)] + E_{z \sim P_z(z)} [\log (1 - D(G(z)))]$$

P_{data} : 실제데이터 / z : 노이즈

$\log D(x)$ 와 $\log (1 - D(G(z)))$ 가 최대



$$D(x)=1$$

실제 데이터를 진짜라고 분류하도록 학습

$$D(G(z))=0$$

G가 만들어낸 가짜 데이터를 가짜라고 분류하도록 학습

GAN

$$\min_G \max_D V(D, G) = E_{x \sim P_{data}}(\log D(x)) + E_{z \sim P_z(z)}[\log(1 - D(G(z)))]$$

P_{data} : 실제데이터 / z : 노이즈

$\log(1 - D(G(z)))$ 가 최소



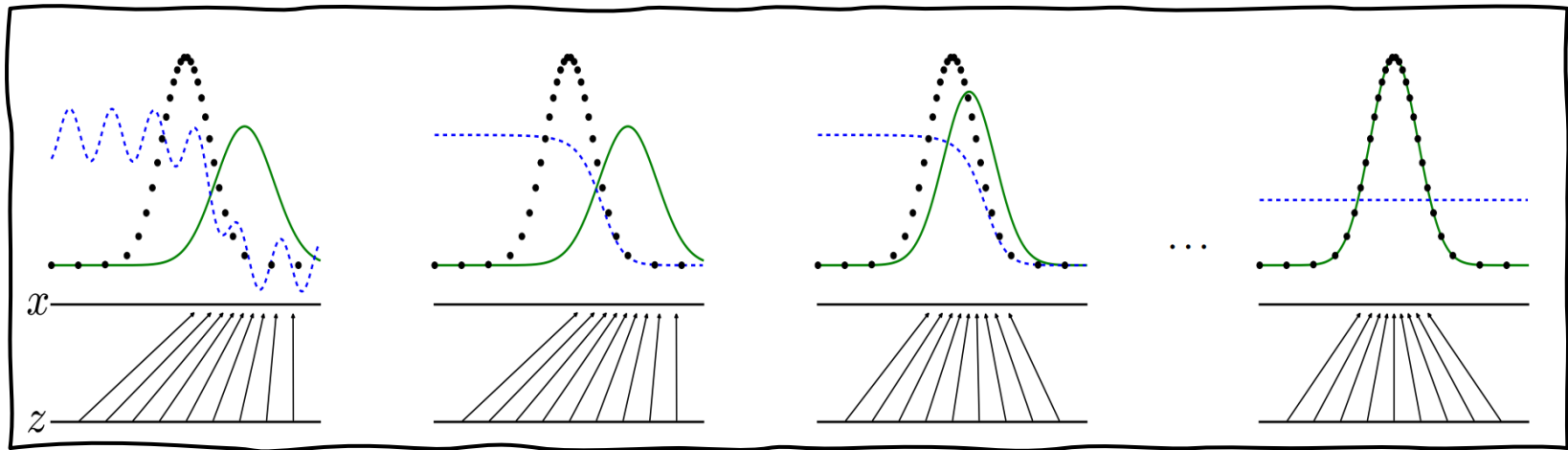
$$D(G(z))=1$$

D가 진짜로 분류할 만큼 완벽한 가짜 데이터를
생성하도록 G를 학습

3

이미지 생성 모델

GAN

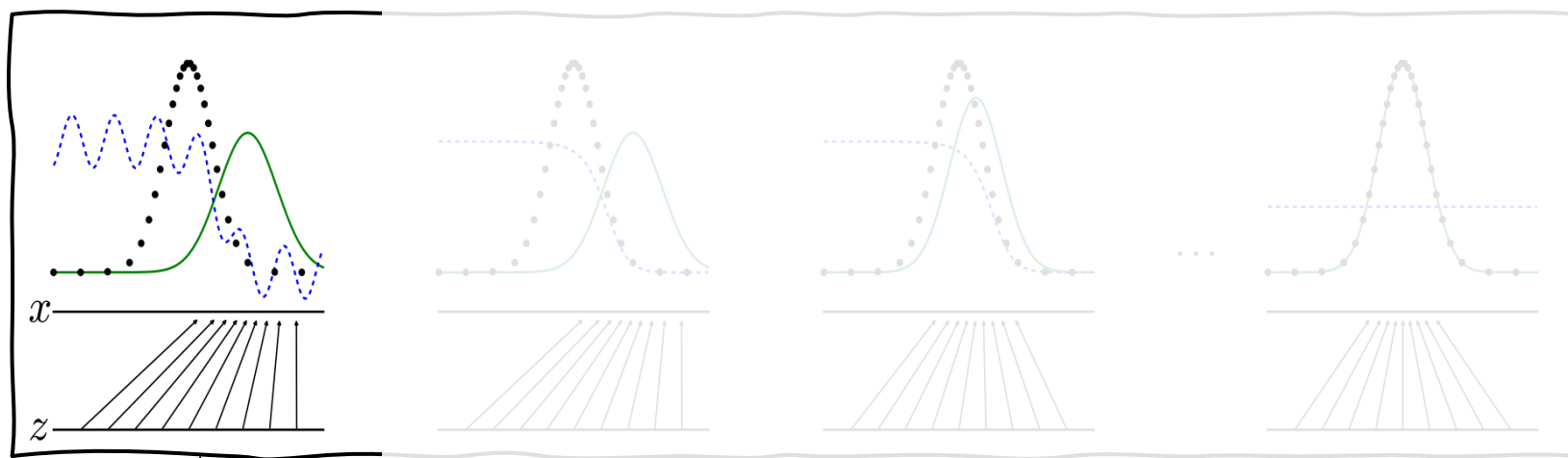


▲ 검은 점선: 실제 데이터 분포, 파란 점선: Discriminator Distribution, 녹색 실선: Generative Distribution

밑 화살표는 z노이즈가 생성모델을 거친 후 어디로 mapping이 되는지를 보여줌

3 이미지 생성 모델

GAN



학습 초반

실제이미지 \neq 생성이미지 분포

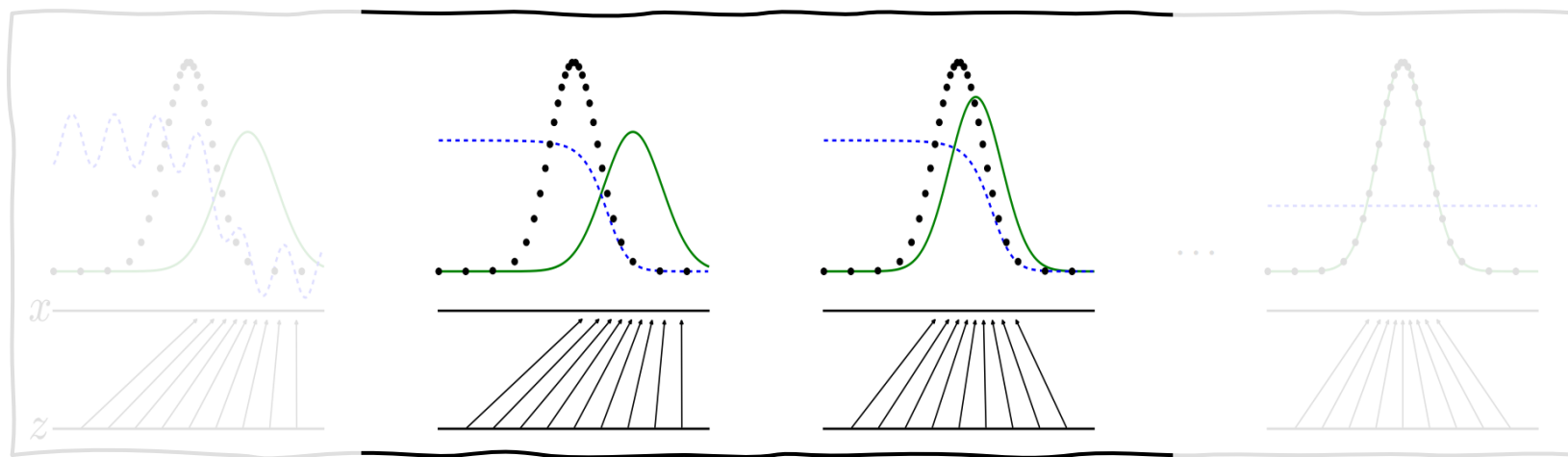
판별자의 불안정한 모습

점선: Discriminator Distribution, 녹색 실선: Generative Distribution

생성모형을 거친 후 어디로 mapping이 되는지를 보여줌

3 이미지 생성 모델

GAN



▲ 검은 점선: 실제 데이터 분포, P_{data}

밑 화살표는 z 노드

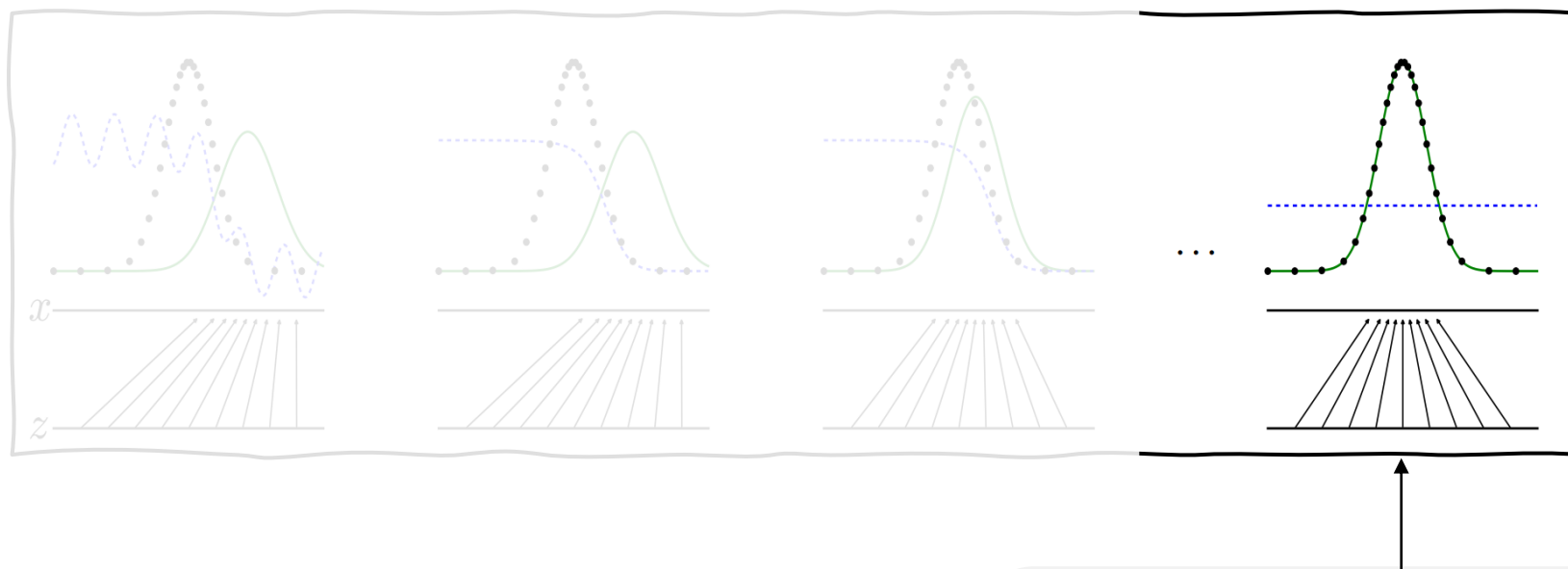
녹색 실선: Generative Distribution

는지를 보여줌

판별자는 학습을 하면서 좀 더 smooth하게
잘 구별하게 됨
생성자가 학습을 하여 녹색 실선 \approx 검은 실선

3 이미지 생성 모델

GAN



▲ 검은 점선: 실제 데이터 분포, 파란 점선: Discriminator Distribution

밑 화살표는 z 노이즈가 생성모델을 거친 후 어디로

학습을 반복하다 보면
실제 데이터 = 생성 데이터
판별자는 이를 구별하지 못하게
되기 때문에 1/2가 됨

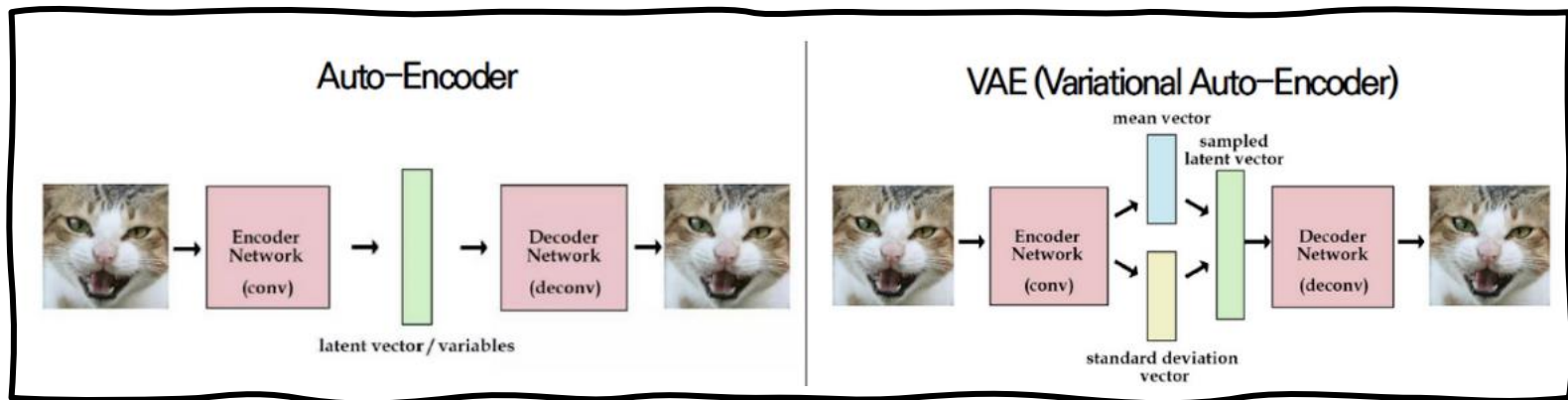
3

이미지 생성 모델

VAE

VAE (Variational Auto-Encoder)

Auto-Encoder를 응용한 생성 모델



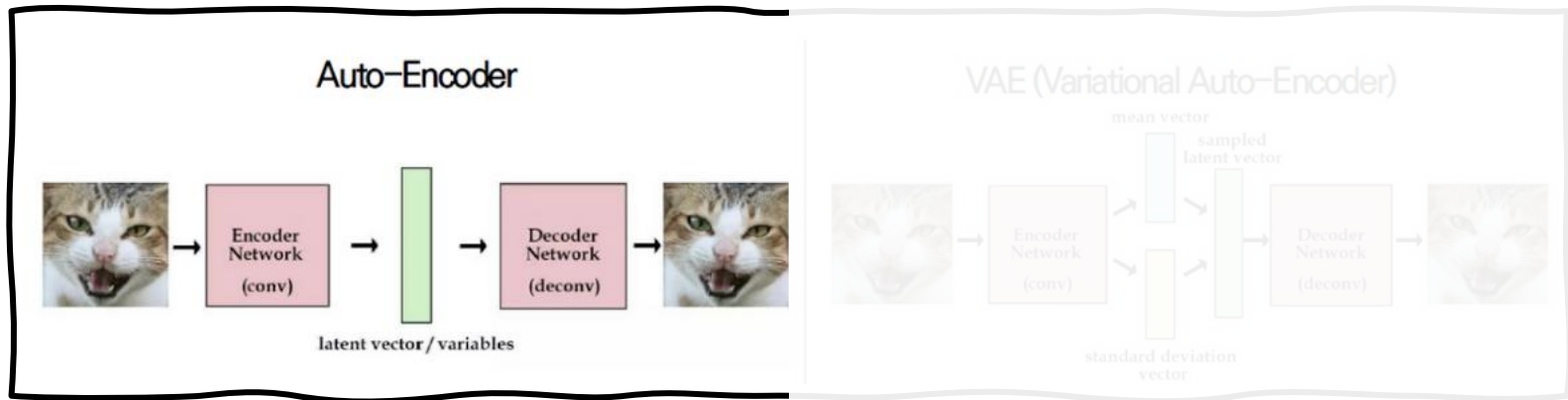
3

이미지 생성 모델

VAE

VAE (Variational Auto-Encoder)

Auto-Encoder를 응용한 생성 모델



input x 자신을 언제든지 reconstruct
할 수 있는 z를 만드는 것이 목적

3

이미지 생성 모델

VAE

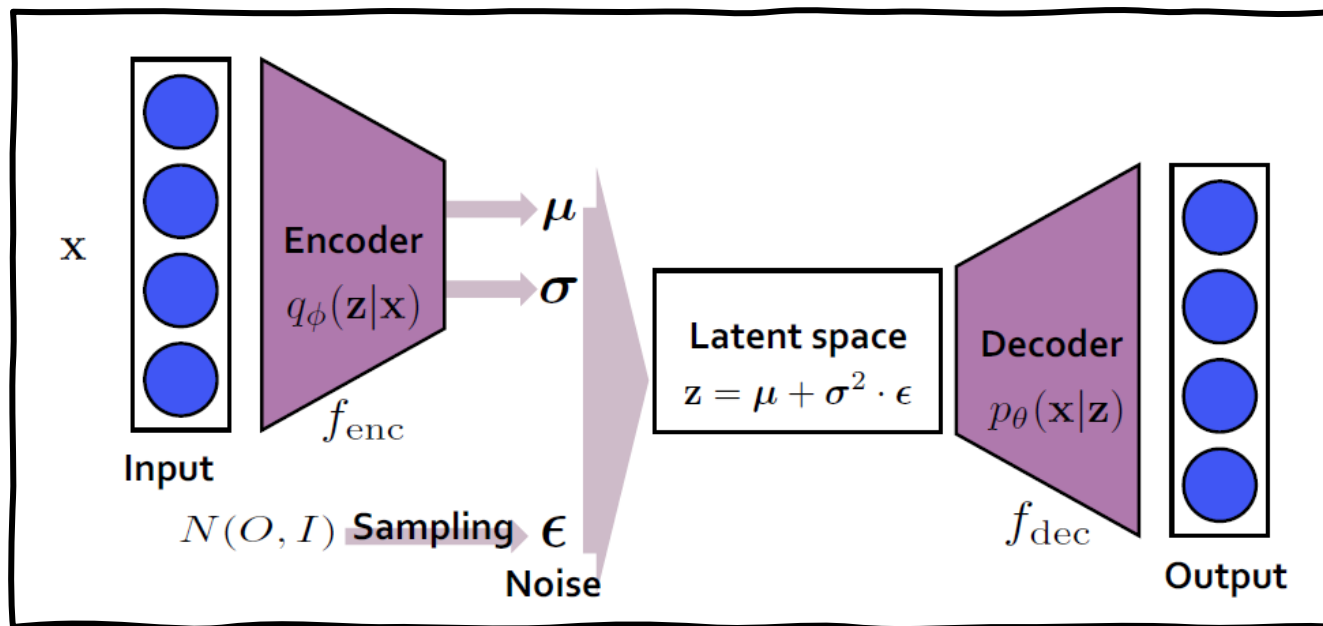
VAE (Variational Auto-Encoder)

Auto-Encoder를 응용한 생성 모델



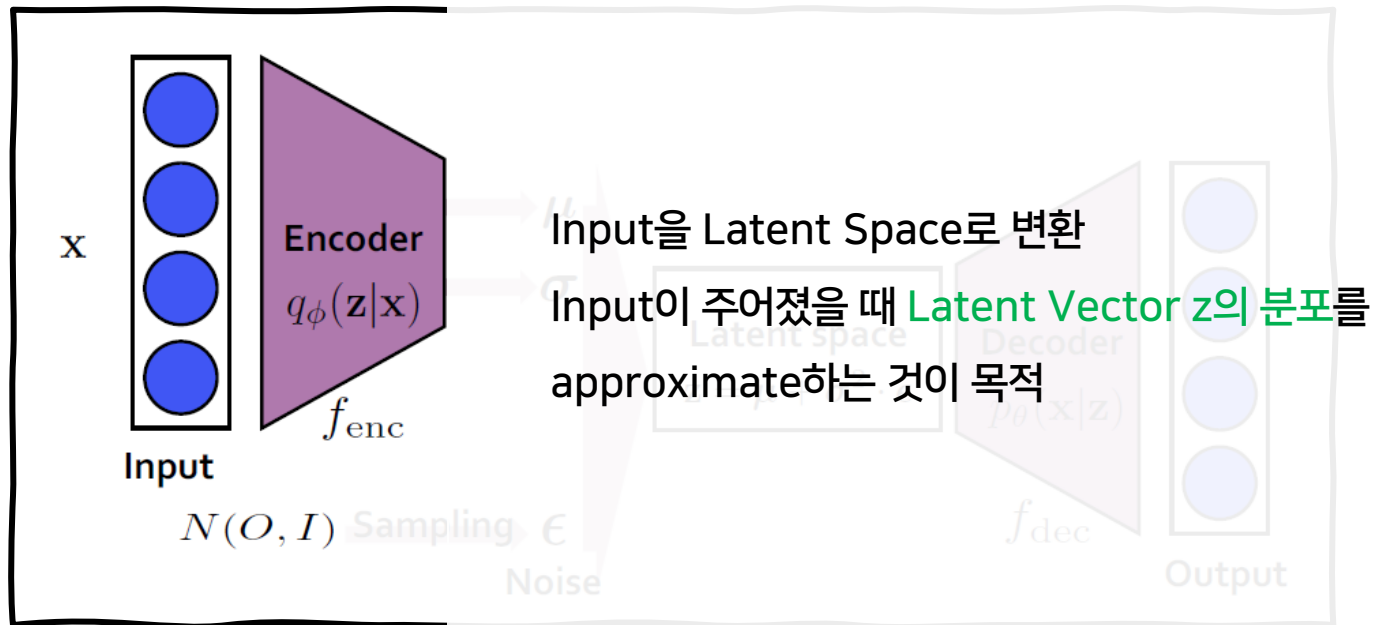
input x가 만들어지는 **확률분포**를 찾는 것이 목적

VAE



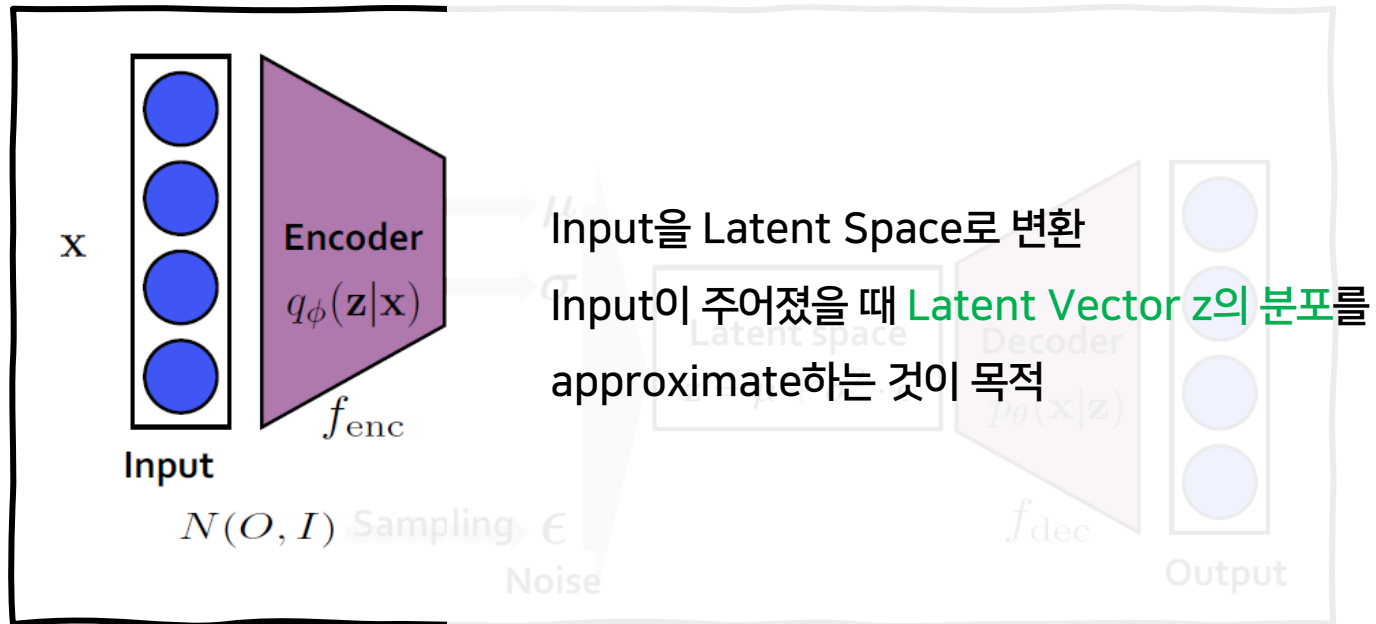
Encoder, Decoder, Latent Space로 구성
학습 단계에서 Encoder와 Decoder의 파라미터를 학습

VAE



Encoder, Decoder, Latent Space로 구성
학습 단계에서 Encoder와 Decoder의 파라미터를 학습

VAE

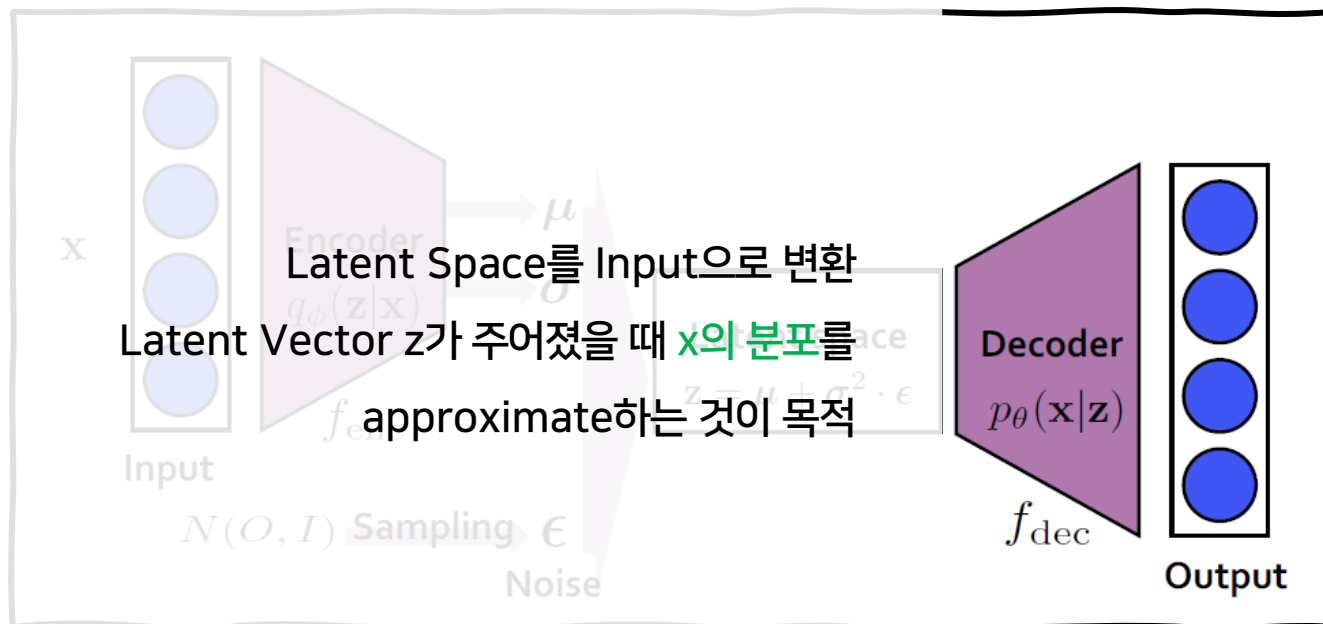


$q(z|x)$ 를 가장 잘 나타내는 분포가 정규분포라고 선택한다면,
 이 정규분포를 나타내는 평균과 표준편차를 찾아야 함

3

이미지 생성 모델

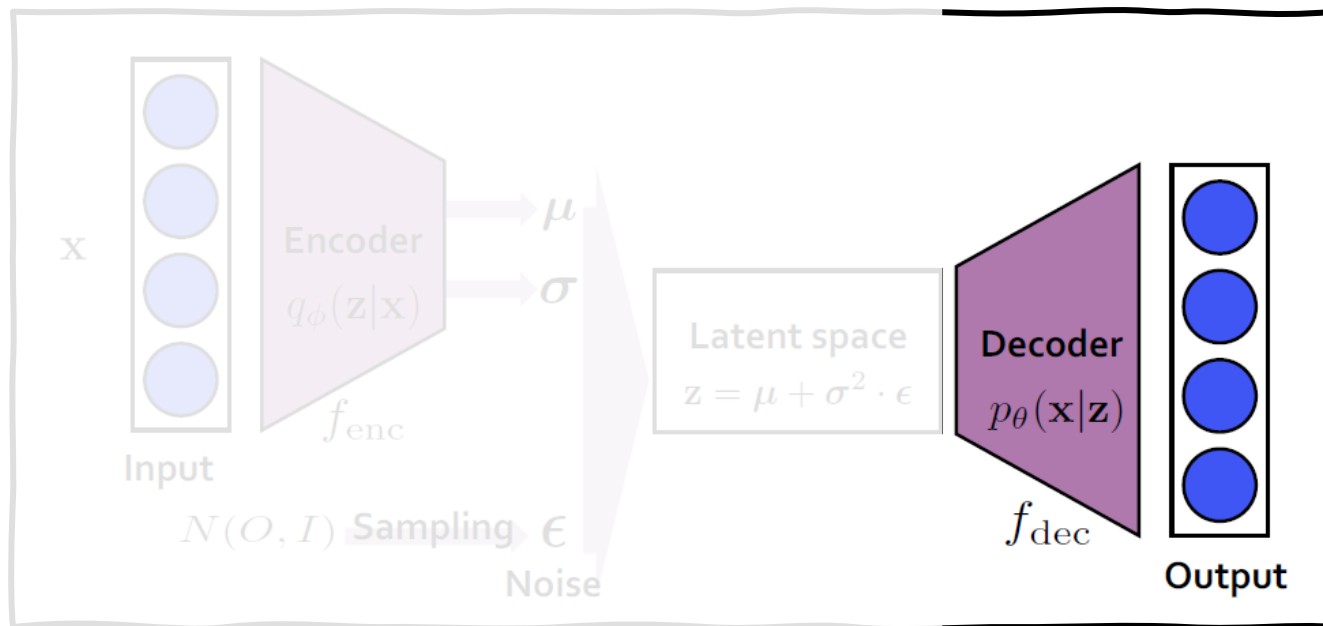
VAE



3

이미지 생성 모델

VAE

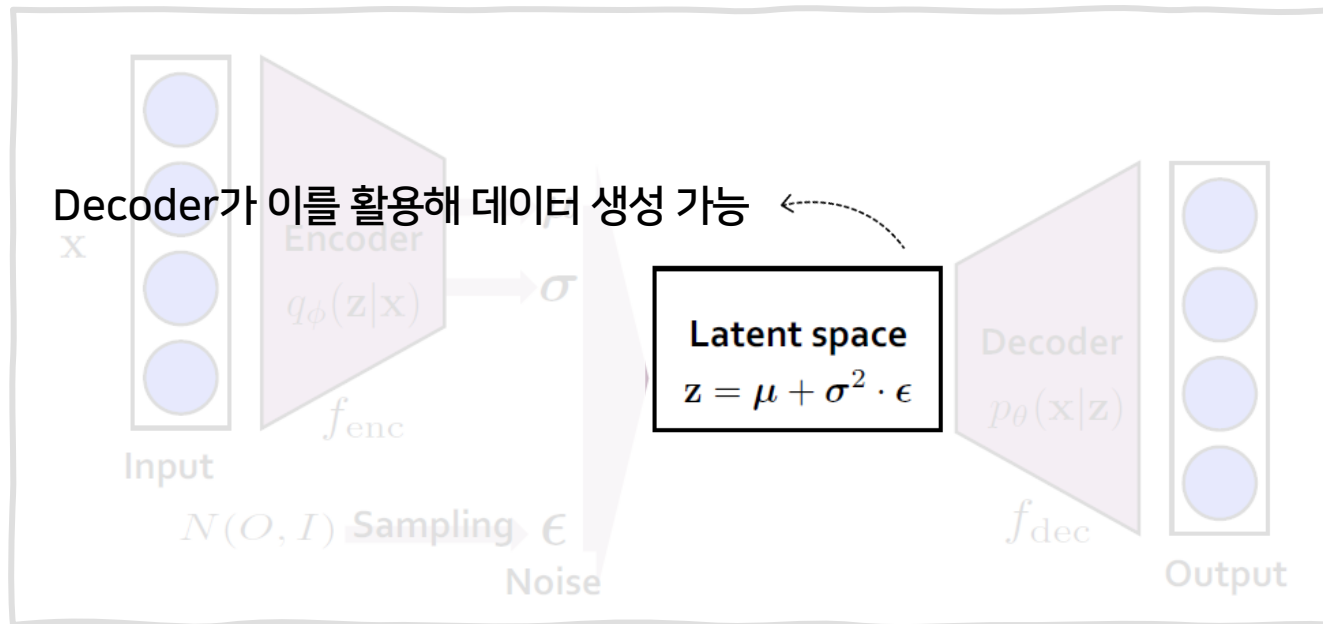


어떤 z 라는 vector가 주어짐에 따라 다시 데이터 x 를 generate하는 역할
Decoder가 Generative model의 역할을 하게 됨

3

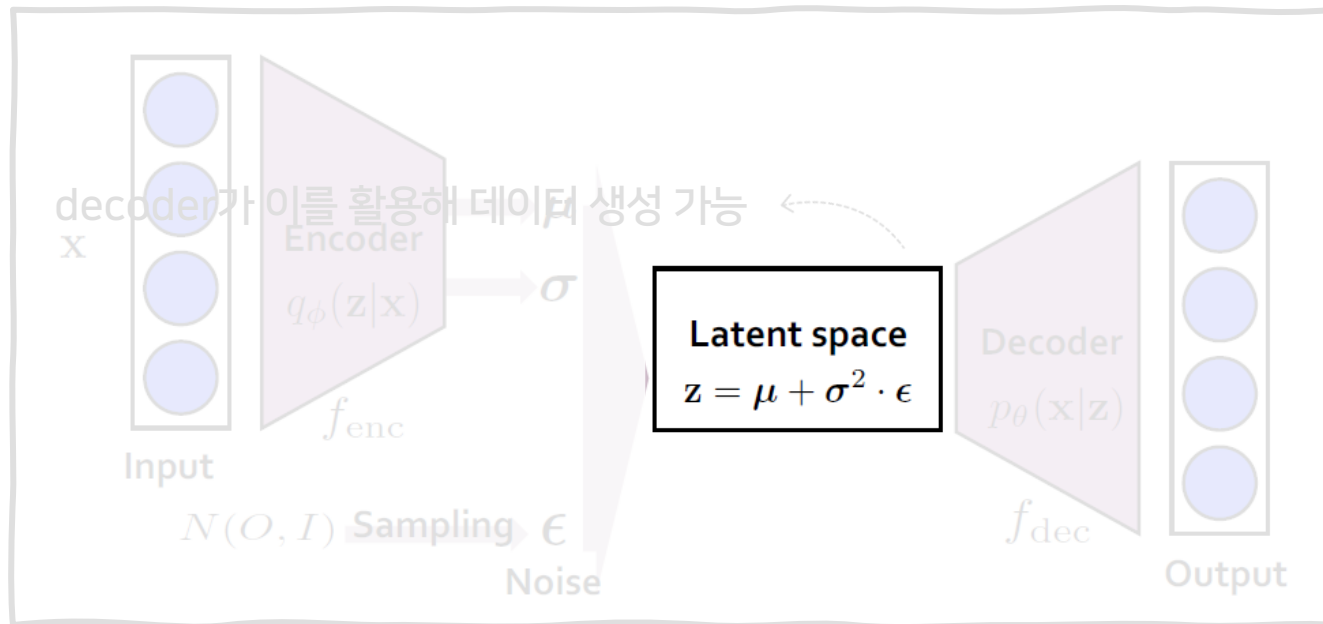
이미지 생성 모델

VAE



3 이미지 생성 모델

VAE

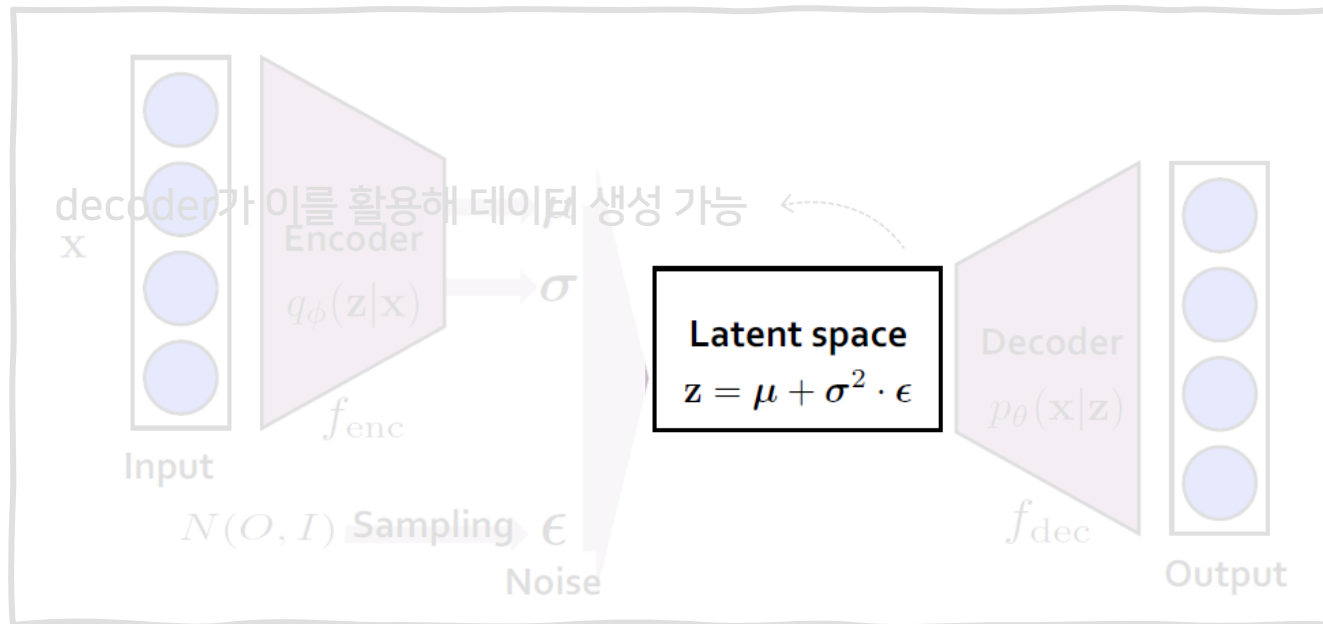


Auto-Encoder처럼 Input과 Output을 똑같이 만드는 것이 목적이라면
Latent Space는 항상 input과 같은 모양의 데이터를 만들 수 밖에..

3

이미지 생성 모델

VAE



이를 방지하기 위해

noise를 샘플링 하여 Latent Space 만듦!

VAE

$$\begin{aligned}
 \log p_{\theta}(x) &= \int q_{\phi}(z|x) \log p_{\theta}(x) dz \\
 &= \int q_{\phi}(z|x) \log \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(z|x)} dz && \text{Bayes rule} \\
 &= \int q_{\phi}(z|x) \log \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(z|x)} \frac{q_{\phi}(z|x)}{q_{\phi}(z|x)} dz && \text{KL divergence} \\
 &= \int q_{\phi}(z|x) \log p_{\theta}(x|z) dz - KL(q_{\phi}(z|x) || p(z)) + KL(q_{\phi}(z|x) || p_{\theta}(z|x))
 \end{aligned}$$

Maximize

VAE

$$\begin{aligned}
 \log p_{\theta}(x) &= \int q_{\phi}(z|x) \log p_{\theta}(x) dz \\
 &= \int q_{\phi}(z|x) \log \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(z|x)} dz && \text{Bayes rule} \\
 &= \int q_{\phi}(z|x) \log \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(z|x)} \frac{q_{\phi}(z|x)}{q_{\phi}(z|x)} dz && \text{KL divergence} \\
 &= \int q_{\phi}(z|x) \log p_{\theta}(x|z) dz - KL(q_{\phi}(z|x) || p(z)) + KL(q_{\phi}(z|x) || p_{\theta}(z|x))
 \end{aligned}$$

Encoder로부터 z 를 샘플링하고,
이 z 로부터 Decoder를 활용해 $p(x|z)$ 를 계산

VAE

$$\begin{aligned}
 \log p_{\theta}(x) &= \int q_{\phi}(z|x) \log p_{\theta}(x) dz \\
 &= \int q_{\phi}(z|x) \log \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(z|x)} dz && \text{Bayes rule} \\
 &= \int q_{\phi}(z|x) \log \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(z|x)} \frac{q_{\phi}(z|x)}{q_{\phi}(z|x)} dz && \text{KL divergence} \\
 &= \int q_{\phi}(z|x) \log p_{\theta}(x|z) dz - \text{KL}(q_{\phi}(z|x) || p(z)) + \text{KL}(q_{\phi}(z|x) || p_{\theta}(z|x))
 \end{aligned}$$

정규분포 사이의 KL divergence 계산

VAE

$$\begin{aligned}
 \log p_{\theta}(x) &= \int q_{\phi}(z|x) \log p_{\theta}(x) dz \\
 &= \int q_{\phi}(z|x) \log \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(z|x)} dz && \text{Bayes rule} \\
 &= \int q_{\phi}(z|x) \log \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(z|x)} \frac{q_{\phi}(z|x)}{q_{\phi}(z|x)} dz && \text{KL divergence} \\
 &= \int q_{\phi}(z|x) \log p_{\theta}(x|z) dz - KL(q_{\phi}(z|x) || p(z)) + KL(q_{\phi}(z|x) || p_{\theta}(z|x))
 \end{aligned}$$

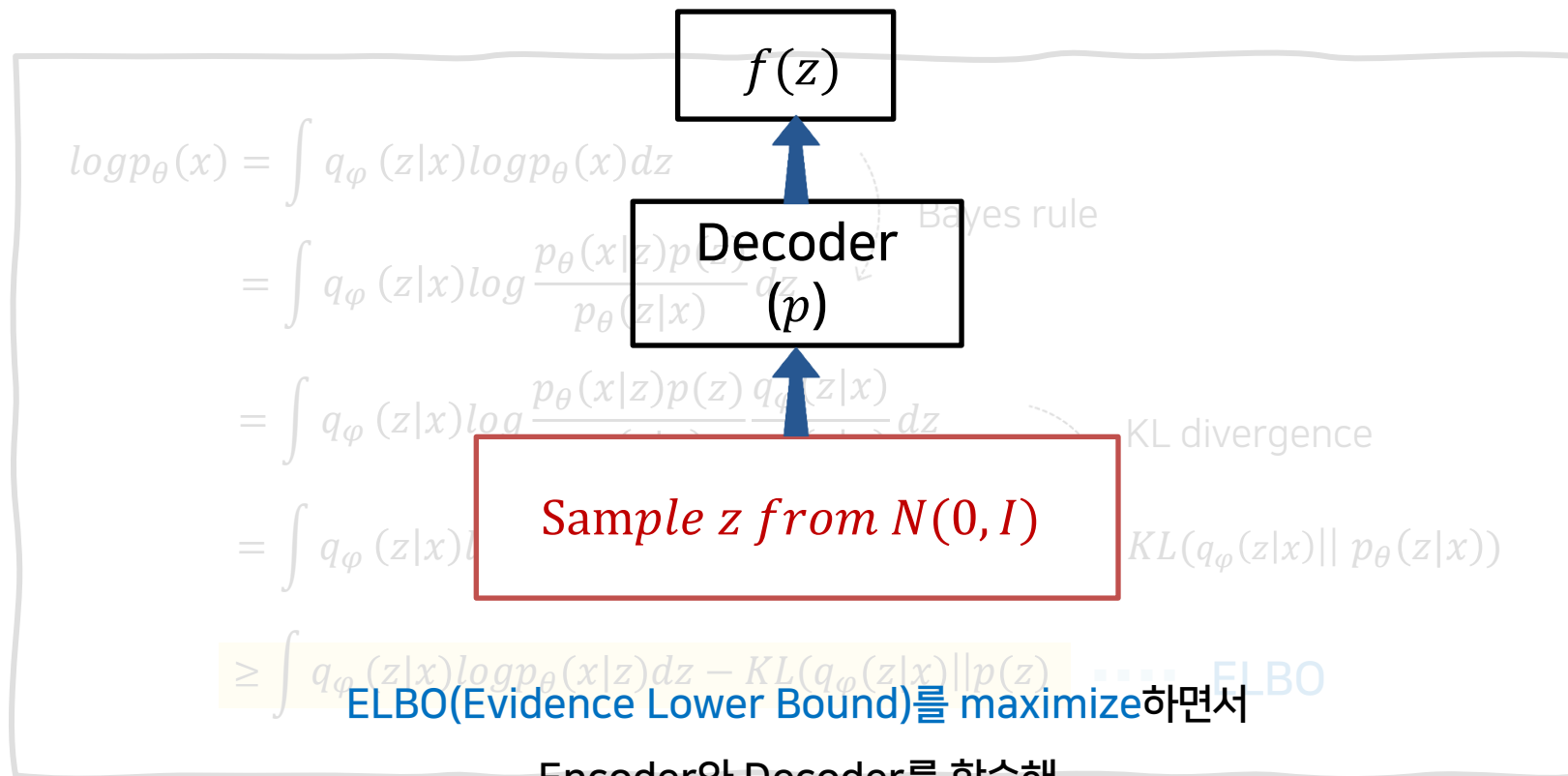
$p(z|x)$ 계산 불가능

KL divergence 양수라는 것 이용

VAE

$$\begin{aligned}
 \log p_{\theta}(x) &= \int q_{\phi}(z|x) \log p_{\theta}(x) dz \\
 &= \int q_{\phi}(z|x) \log \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(z|x)} dz && \text{Bayes rule} \\
 &= \int q_{\phi}(z|x) \log \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(z|x)} \frac{q_{\phi}(z|x)}{q_{\phi}(z|x)} dz && \text{KL divergence} \\
 &= \int q_{\phi}(z|x) \log p_{\theta}(x|z) dz - KL(q_{\phi}(z|x) || p(z)) + KL(q_{\phi}(z|x) || p_{\theta}(z|x)) \\
 &\geq \int q_{\phi}(z|x) \log p_{\theta}(x|z) dz - KL(q_{\phi}(z|x) || p(z)) \quad \dots \text{ELBO}
 \end{aligned}$$

VAE



Encoder와 Decoder를 학습해
 학습된 Decoder를 활용해 데이터 생성

4

텍스트 생성 모델

텍스트 생성 모델



ENCODER

A light blue oval containing the word 'ENCODER' in black capital letters.

Input Sentence의 맥락 파악



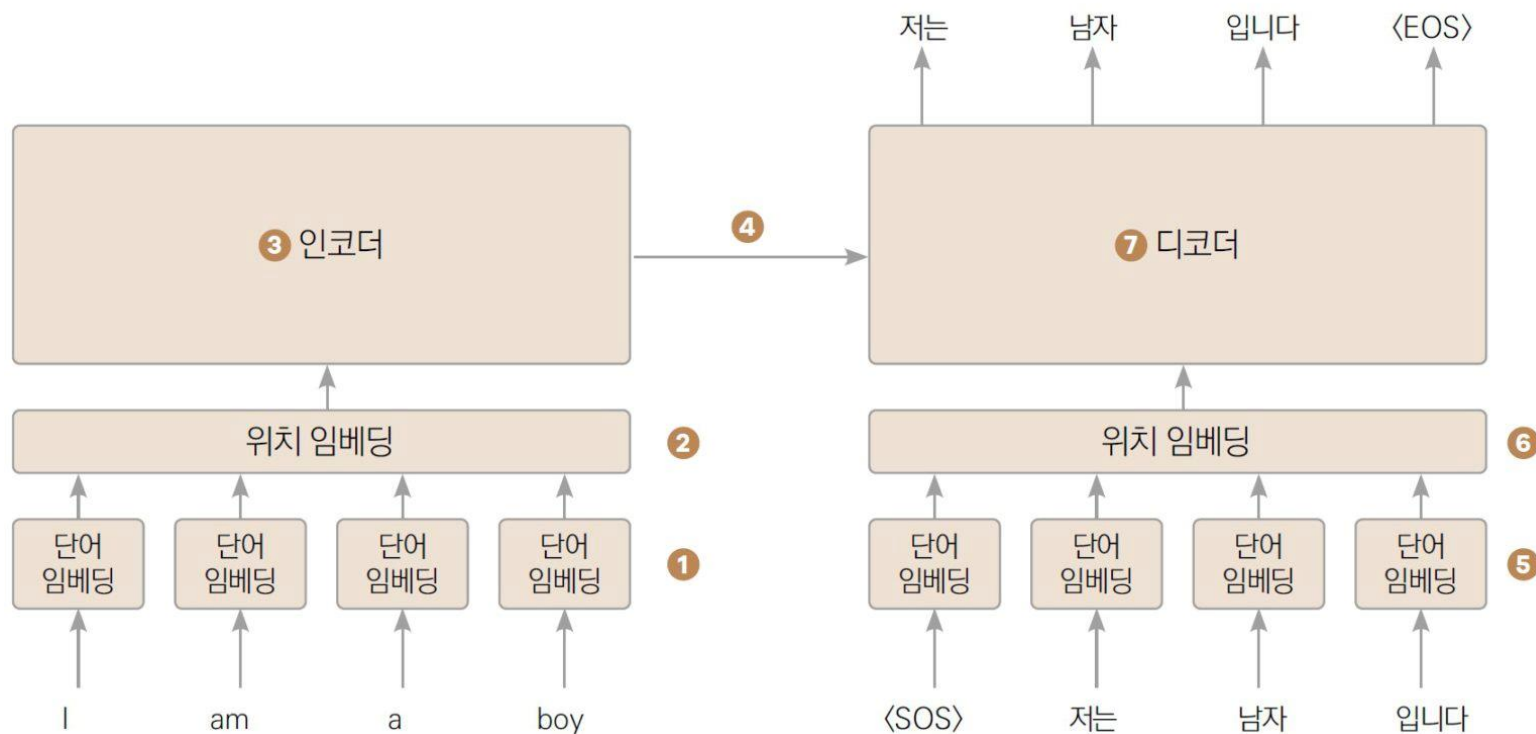
DECODER

A light blue oval containing the word 'DECODER' in black capital letters.

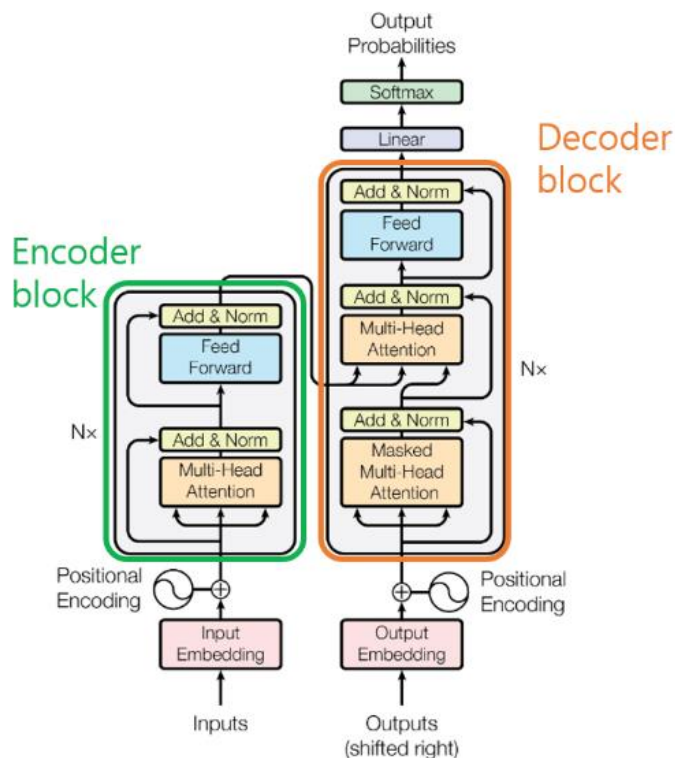
이전 문장이 주어졌을 때 다음 단어를 잘 예측

텍스트 생성 모델

▼ 트랜스포머의 구조



Transformer | 구조



기존의 Seq2Seq 모델

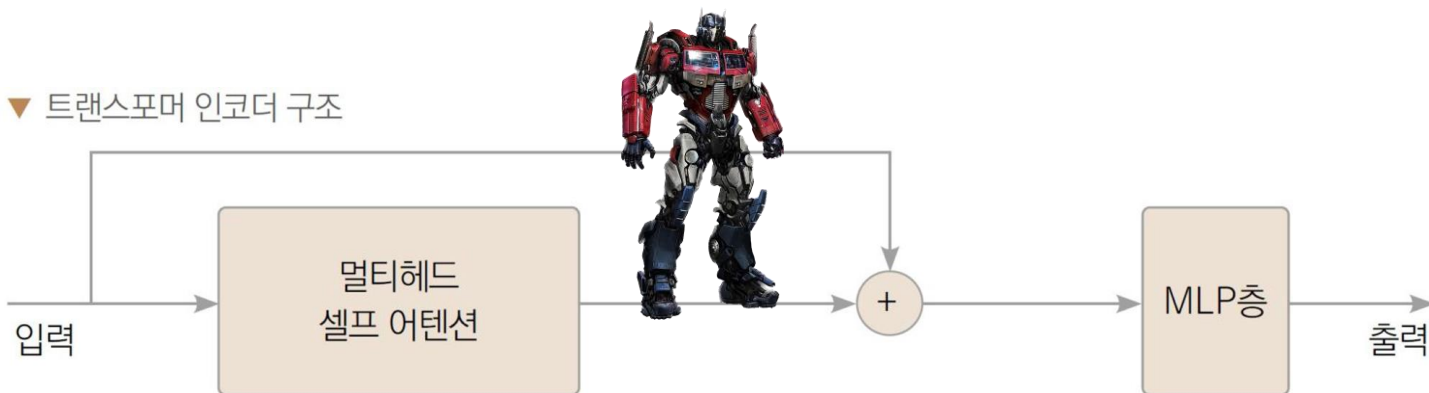
인코더가 입력 시퀀스를 하나의 벡터로 압축하는 과정에서 정보가 일부 손실된다는 한계점 존재

Transformer 모델

이러한 한계점을 극복하여 **Attention**만으로 기존 Seq2Seq 모델의 Encoder-Decoder 구조 구현

Transformer | Encoder 구조

▼ 트랜스포머 인코더 구조

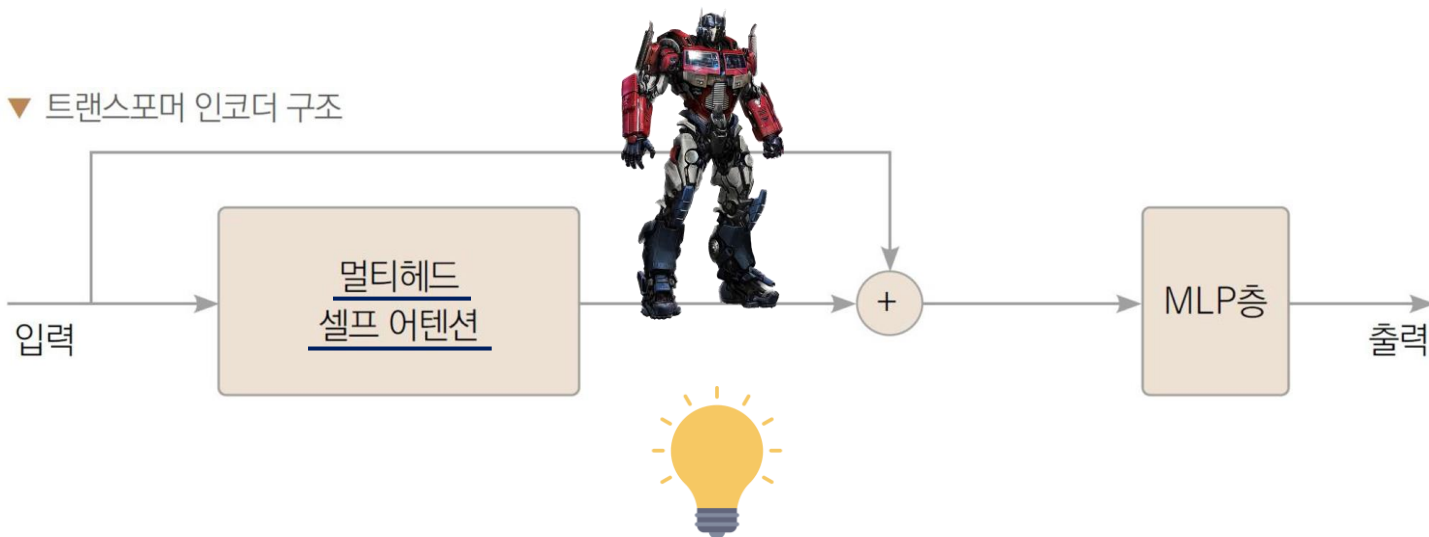


- ▶ RNN 계열 알고리즘과 달리 인코더에서 모든 단어를 동시에 고려
- ▶ Transformer의 Encoder는 입력 시 순서의 정보를 고려하지 않음

↳ 입력이 순차적으로 들어오지 않기 때문!

Transformer | Encoder 구조

▼ 트랜스포머 인코더 구조



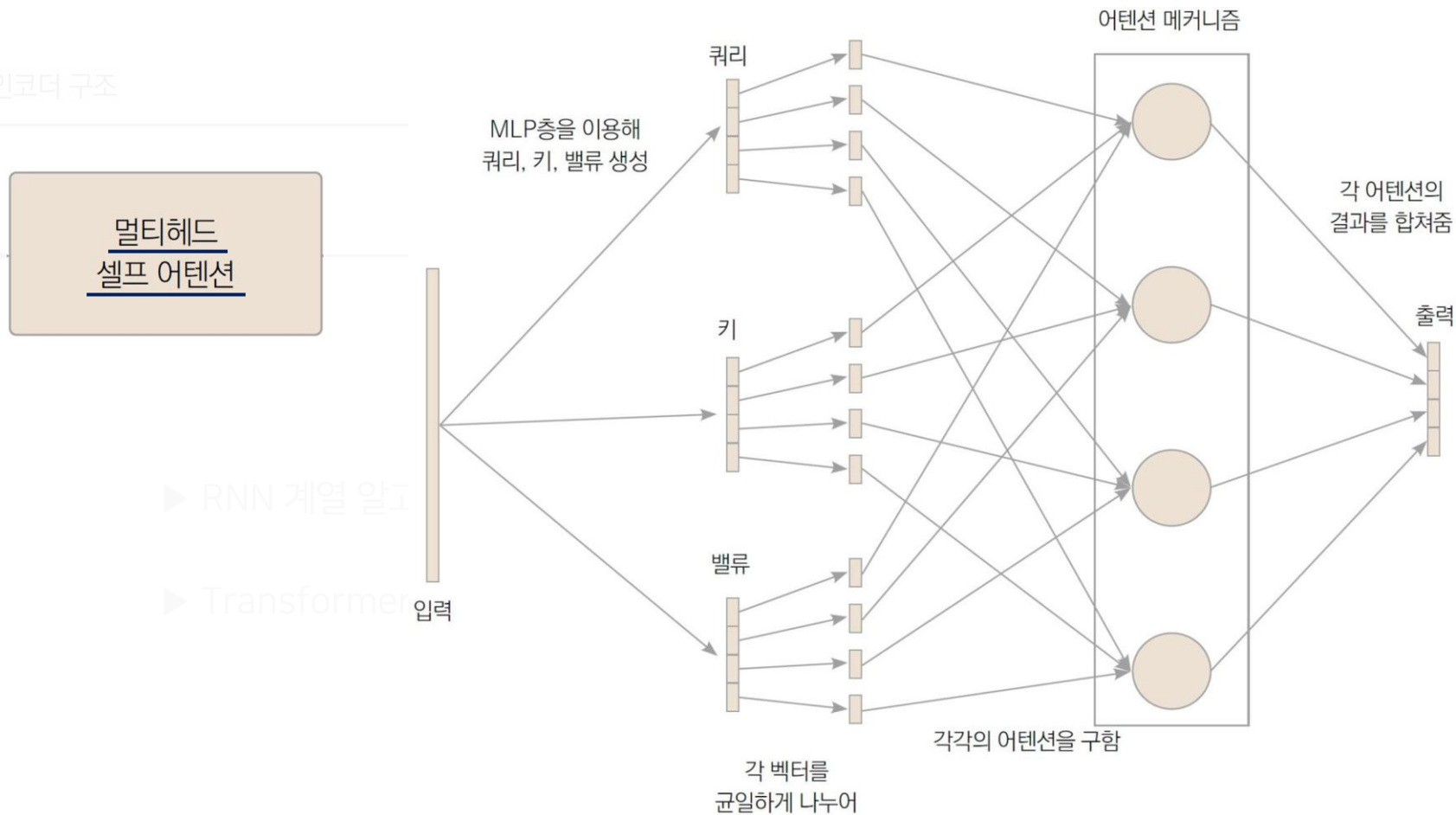
- ▶ 기존의 일반적인 셀프 어텐션은 한 번에 하나의 특징만을 추출했다면,
멀티헤드 셀프 어텐션은 쿼리, 키, 밸류를 활용해
 ▶ **한 번에 여러 개의 특징을 추출하도록** 벡터를 나누어 계산하는 알고리즘

* 셀프 어텐션 : 쿼리, 키, 밸류 3개 요소 사이의 문맥적 관계성을 추출하는 과정

Transformer | Encoder 구조

▼ 멀티 헤드 셀프 어텐션의 계산

트랜스포머 인코더 구조

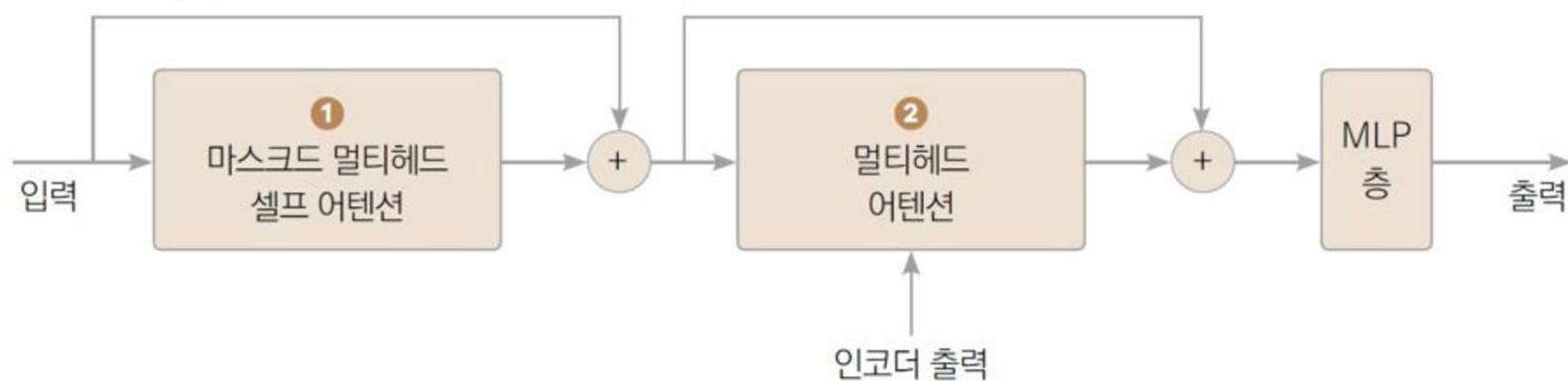


▶ RNN 계열 알고

▶ Transformer

Transformer | Decoder 구조

▼ 트랜스포머 디코더 구조

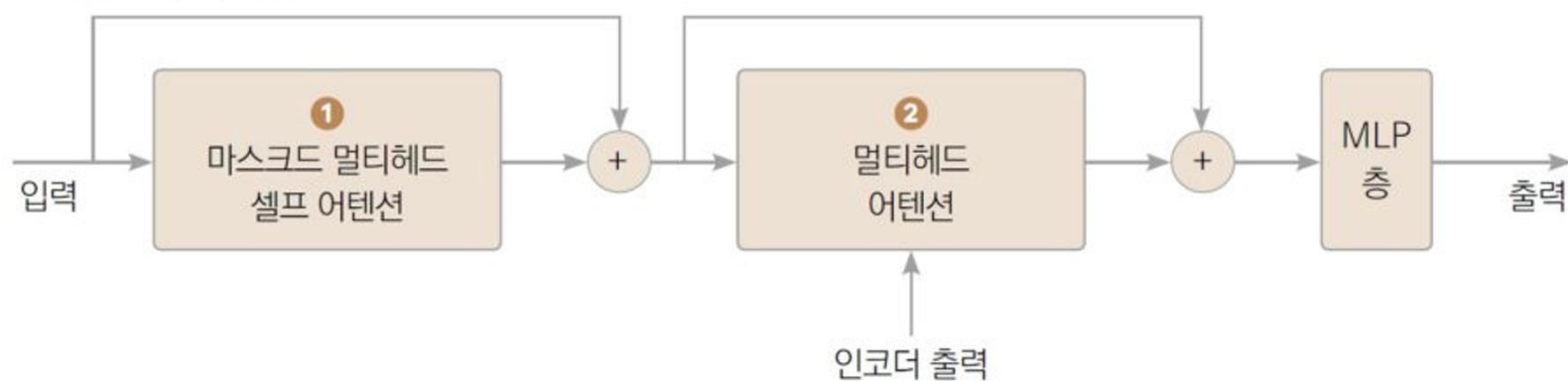


- ▶ 디코더는 아주 큰 음의 값을 미래 시점에 가하여 **소프트맥스**를 계산해 미래의 단어가 고려되지 않도록 하는 **마스크드 멀티헤드 셀프 어텐션**을 거침

* **소프트맥스** : 로지스틱 함수의 다차원 일반화로, 인공지능망에서 확률분포를 얻기 위한 마지막 활성화함수로 많이 사용

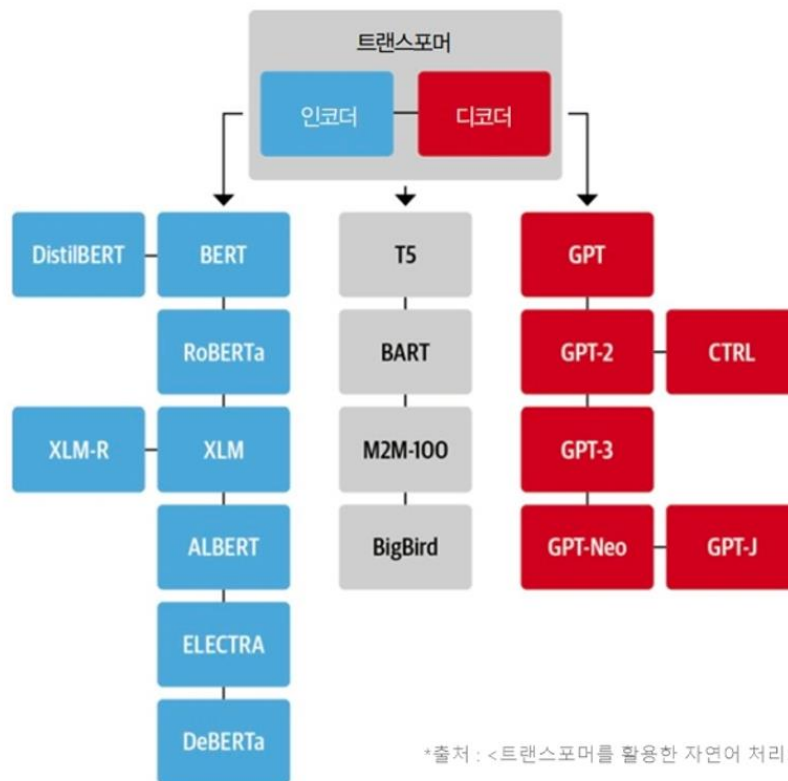
Transformer | Decoder 구조

▼ 트랜스포머 디코더 구조



- ▶ 디코더는 아주 큰 음의 값을 미래 시점에 가하여 **소프트맥스**를 계산해 미래의 단어가 고려되지 않도록 하는 **마스크드 멀티헤드 셀프 어텐션**을 거침
- ▶ 이후, 인코더의 출력과 디코더의 어텐션을 함께 계산한 결과를 MLP 층의 입력으로 넣어 디코더의 출력 계산

Transformer | 파생 모델

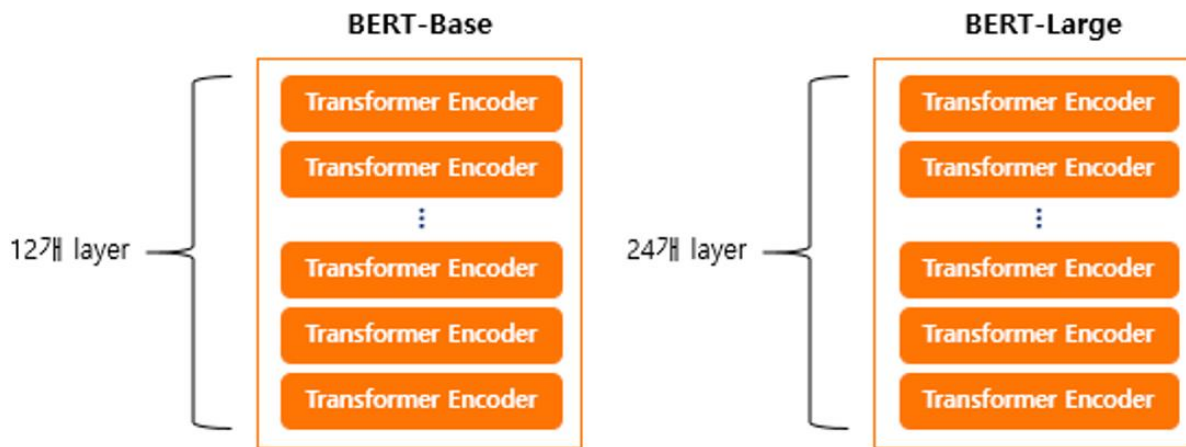


트랜스포머의 인코더-디코더 구조가 세분화되며 여러 파생 모델들이 생겨났음
그 중 대표적인 인코더 유형의 모델이 **BERT**

BERT

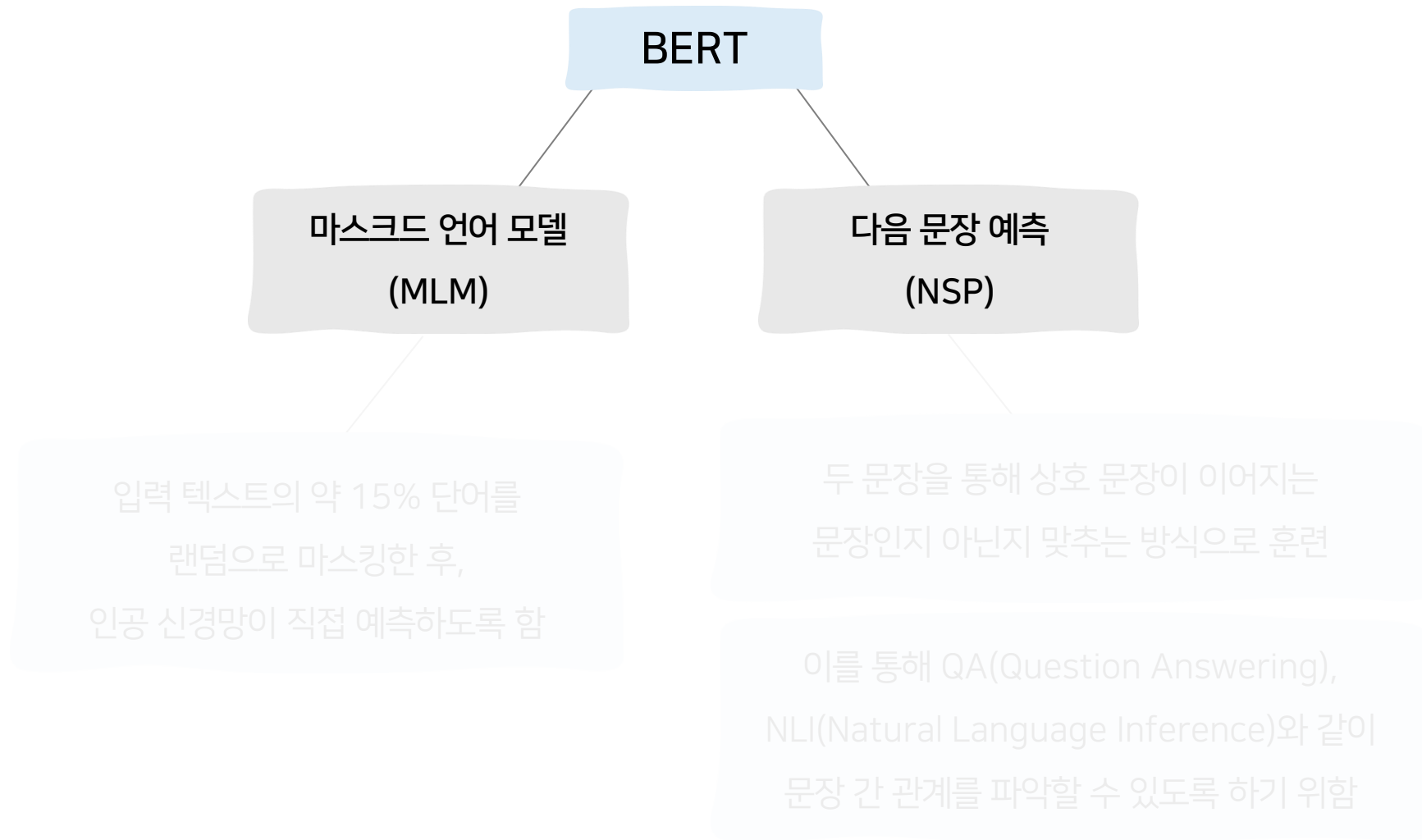
BERT

Transformer의 인코더를 쌓아올린 구조

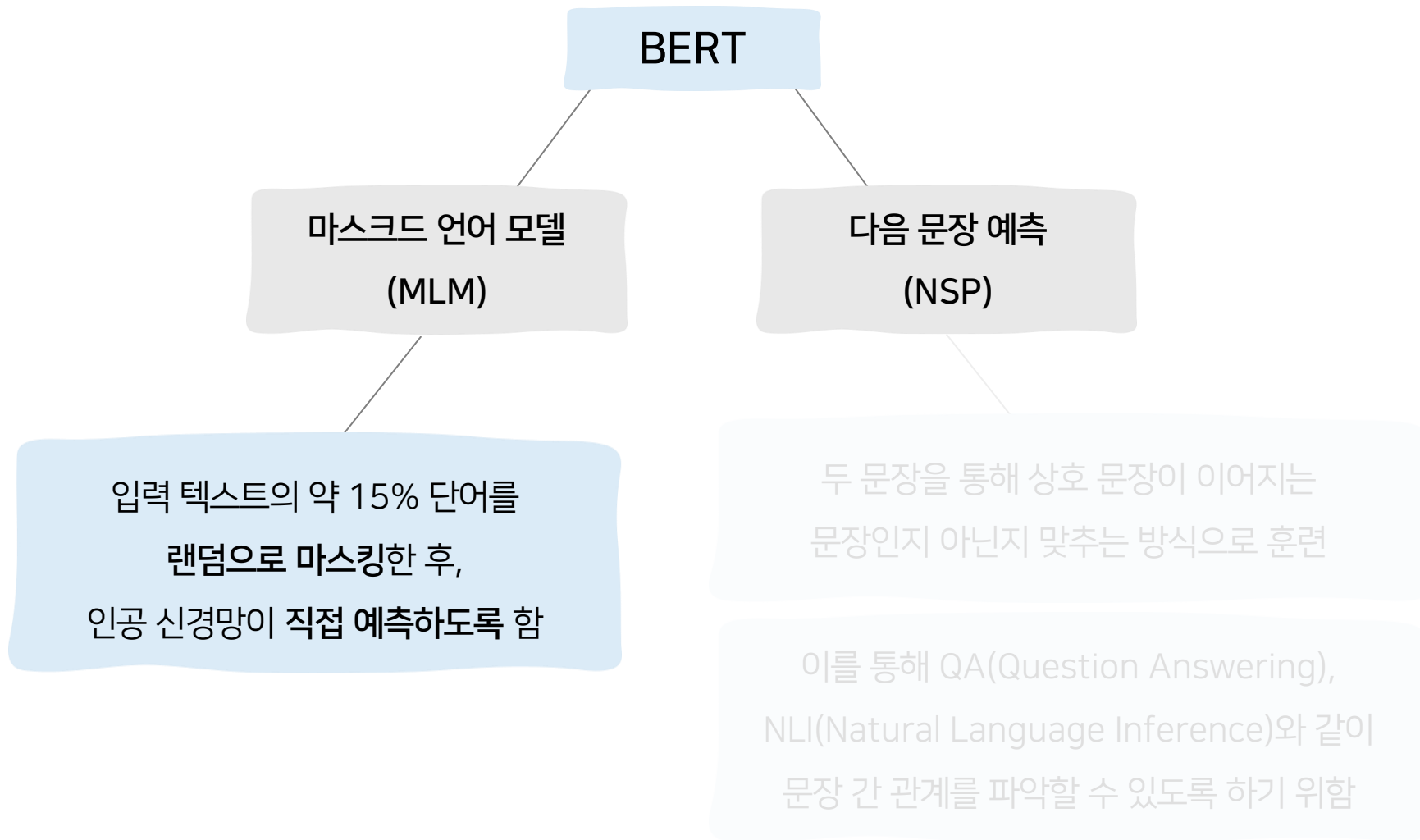


크게 마스크드 언어 모델(MLM), 다음 문장 예측(NSP)이라는 두 가지 사전 훈련 방법을 가짐

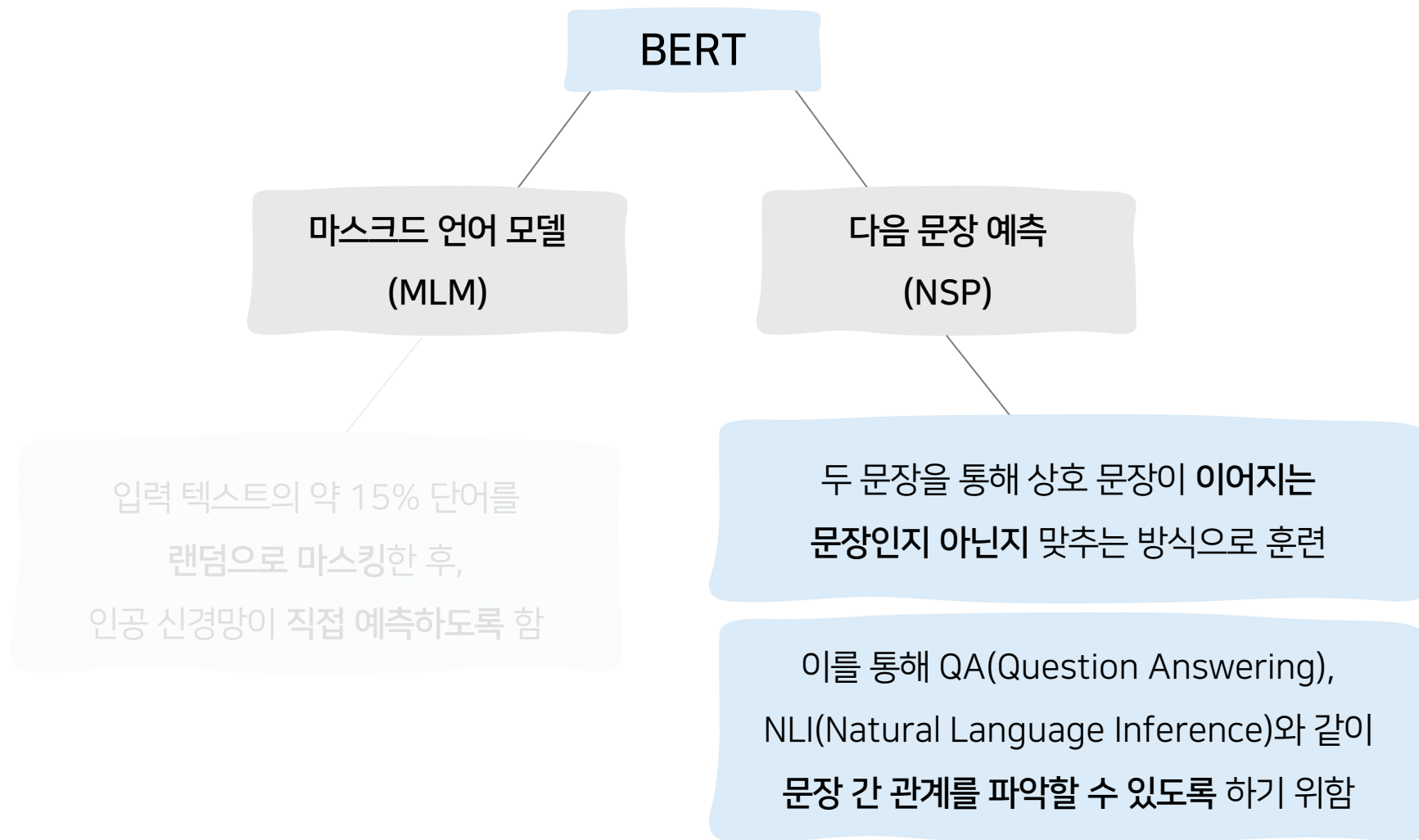
BERT



BERT



BERT



BERT

BERT

마스크드 언어 모델
(MLM)다음 문장 예측
(NSP)

이와 같은 사전 학습법을 통해, 입력 텍스트의 약 15% 단어를 랜덤으로 마스킹한 후, **BERT**는 Bidirectional 한 구조를 가짐! 이를 통해 상호 문장이 이어지는 문장을 찾아내는 방식으로 훈련
인공 신경망이 직접 예측하도록 함

이를 통해 QA(Question Answering), NLI(Natural Language Inference)와 같이 문장 간 관계를 파악할 수 있도록 하기 위함

GPT

GPT (Generative Pre-trained Transformer)

하나의 토큰이 입력되면 그 다음에 올 적절한 토큰을 생성해내는 언어 모델

Bidirectional한 BERT와 달리 **Unidirectional**한 구조로,
사전학습으로 **NSP**만을 활용



NSP?

- ▶ 이전 단어들을만 기반으로 다음 단어를 예측하도록 학습
- ▶ 따라서 문장 뒷부분에 중요한 정보가 있는 경우 GPT가 이를 고려하지 못할 수 있음

GPT

GPT (Generative Pre-trained Transformer)

하나의 토큰이 입력되면 그 다음에 올 적절한 토큰을 생성해내는 언어 모델

Bidirectional한 BERT와 달리 **Unidirectional**한 구조로,
사전학습으로 **NSP**만을 활용



대표적인 디코더 유형의
모델이 **GPT!**

- ▶ 이전 단어들만을 기반으로 다음 단어를 예측하도록 학습
- ▶ 따라서 문장 뒷부분에 중요한 정보가 있는 경우 GPT가 이를 고려하지 못할 수 있음

감사합니다

