Lecture 5

GAN, Diffusion model, LLM

Seoul National University

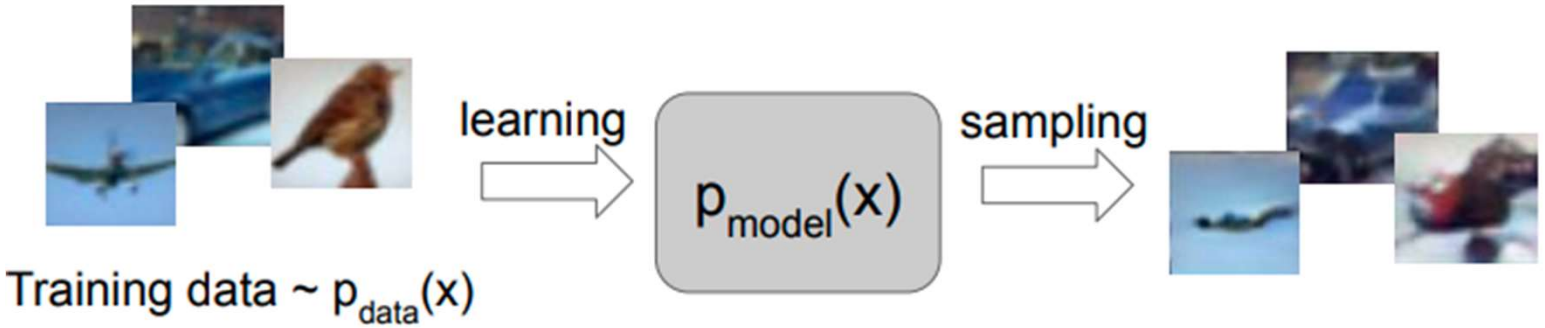
Human Interface Laboratory

Contents

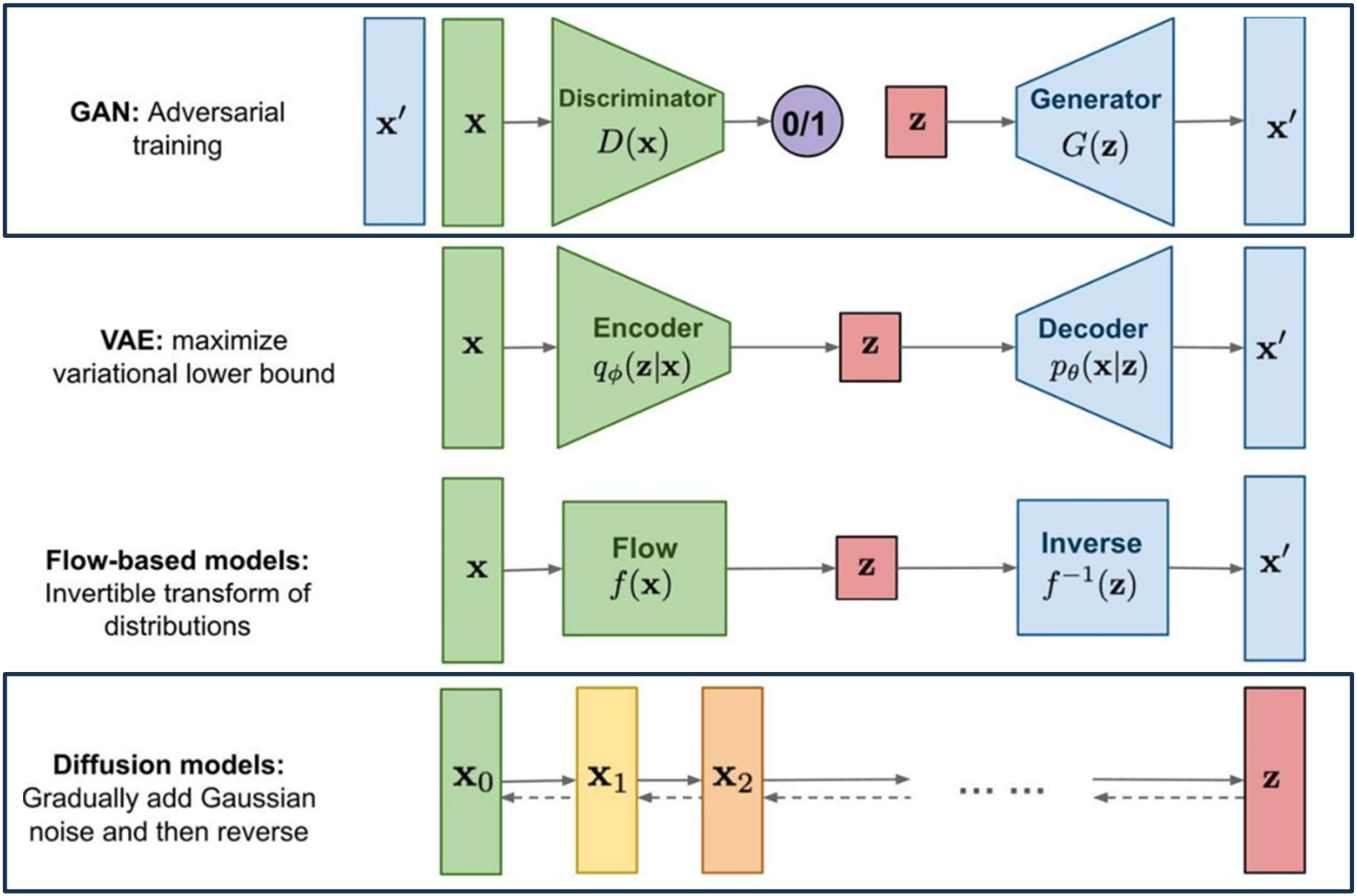
- GAN 기본 이론
- GAN 실습코드 실행
- Diffusion Model 기본 이론
- Diffusion Model 실습코드 실행
- LLM 기본 이론

Generative Models

- **왜 생성 모델인가?**
  - 기존 Discriminative Model:
    - 주어진 데이터 X에 대해 라벨 Y를 예측 $P(Y|X)$
    - 한계: $P(X)$, 즉 데이터 자체의 분포를 학습하지 않음 → 새로운 데이터 생성 불가능.
  - Generative Model
    - 데이터의 확률 분포: $P(X)$를 모델링.
    - $P(X)$에서 샘플링하여 새로운 데이터를 생성 가능

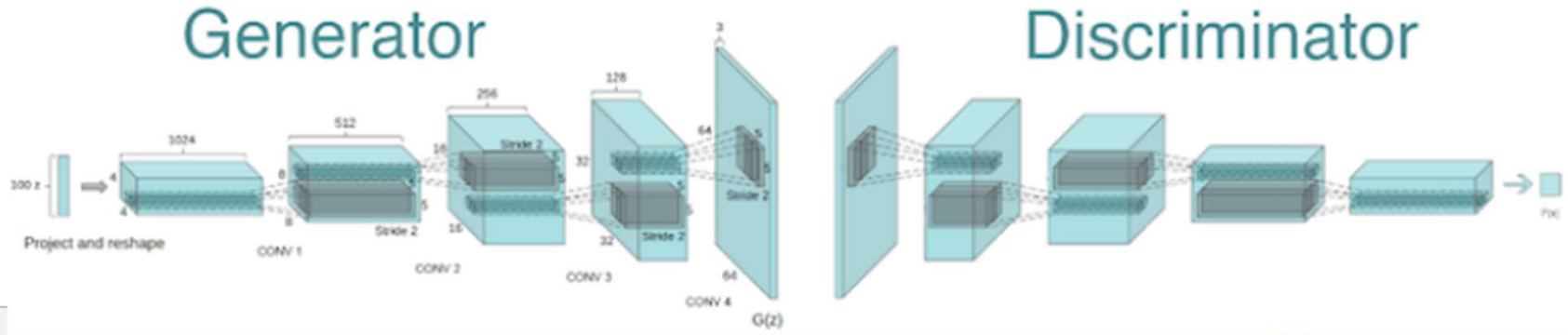


Generative Models



GAN 개요 및 구성 요소

- **Generative Adversarial Networks (GAN)**
  - 데이터 분포를 학습하여 새로운 데이터를 생성하는 모델
  - 구성 요소:
    - Generator (생성기)
      - 랜덤 노이즈 $z \sim p_z(z)$를 입력으로 받아 데이터를 생성
      - 데이터를 실제 데이터처럼 보이게 만들기 위해 학습.
    - Discriminator (판별기)
      - 생성된 데이터와 실제 데이터를 구별.
      - 이진 분류 문제로 작동 (P(실제)) vs P(가짜)).
  - Generator와 Discriminator는 서로 경쟁하며 성능을 개선.
  - Generator는 Discriminator를 속이기 위해 학습하고, Discriminator는 이를 구별하려고 학습.



GAN의 학습 과정

- Generator는 랜덤 벡터 $z \sim p_z(z)$ 를 입력으로 가짜 데이터를 생성 ($G(z)$).
- Discriminator는 실제 데이터 (x)와 가짜 데이터 $G(z)$를 구별.
- Generator는 Discriminator를 속이기 위해 학습
- Loss
  - Minimax Loss:

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

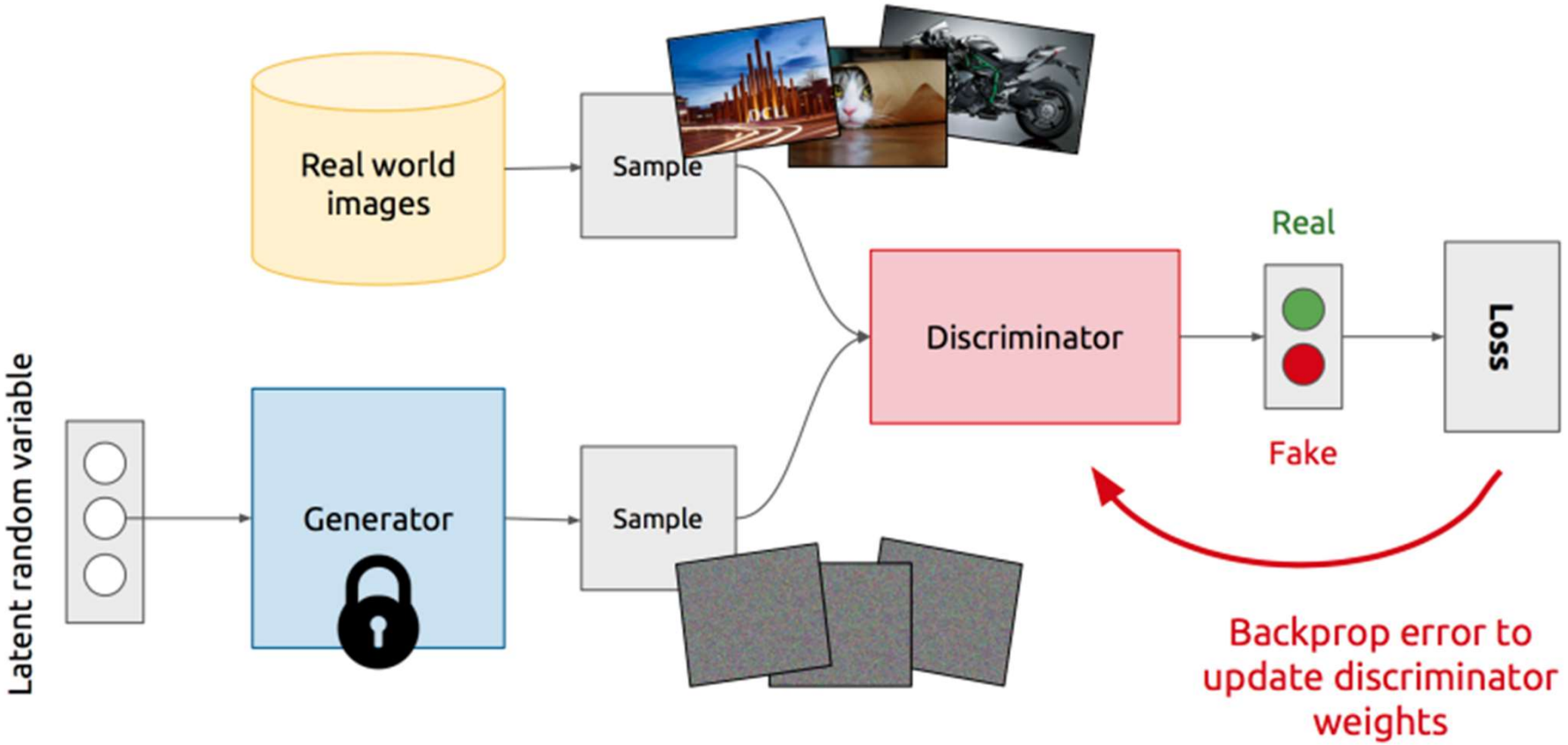
  - Discriminator loss:

$$L_D = -\left(\log D(x) + \log(1 - D(G(z)))\right)$$

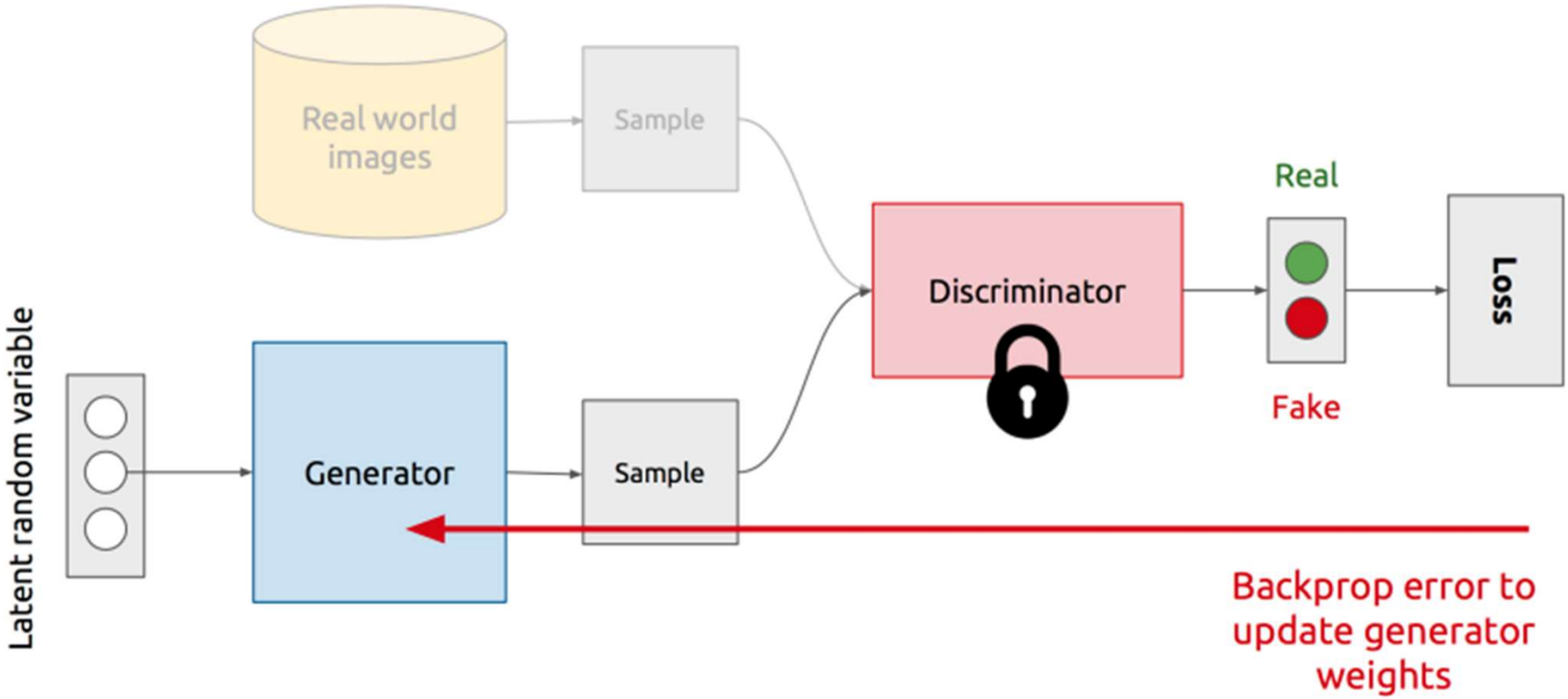
  - Generator loss:

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

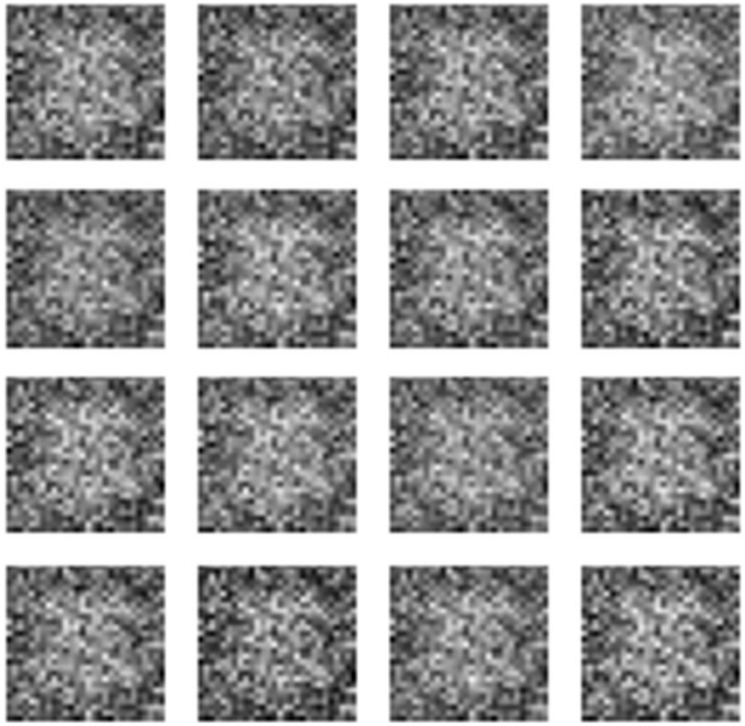
GAN의 학습 과정


Real world images
Sample
Latent random variable
Generator
Sample
Discriminator
Real
Fake
Loss
Backprop error to update discriminator weights

GAN의 학습 과정

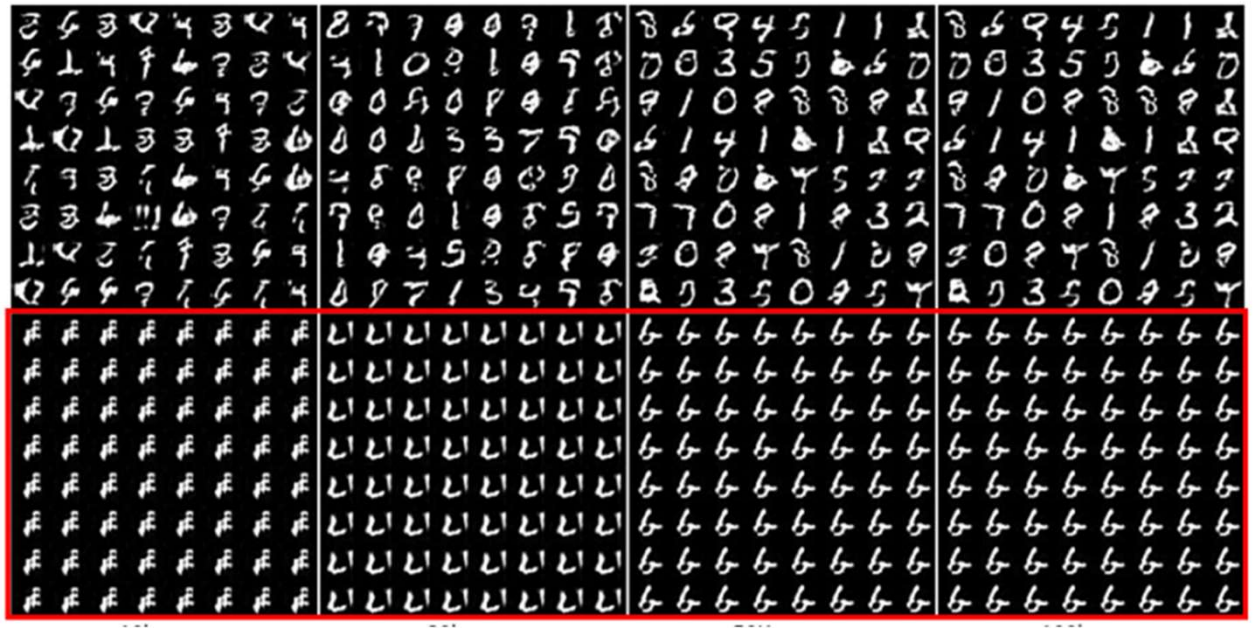

Real world images
Sample
Real
Discriminator
Loss
Fake
Latent random variable
Generator
Sample
Backprop error to update generator weights

GAN의 학습 과정



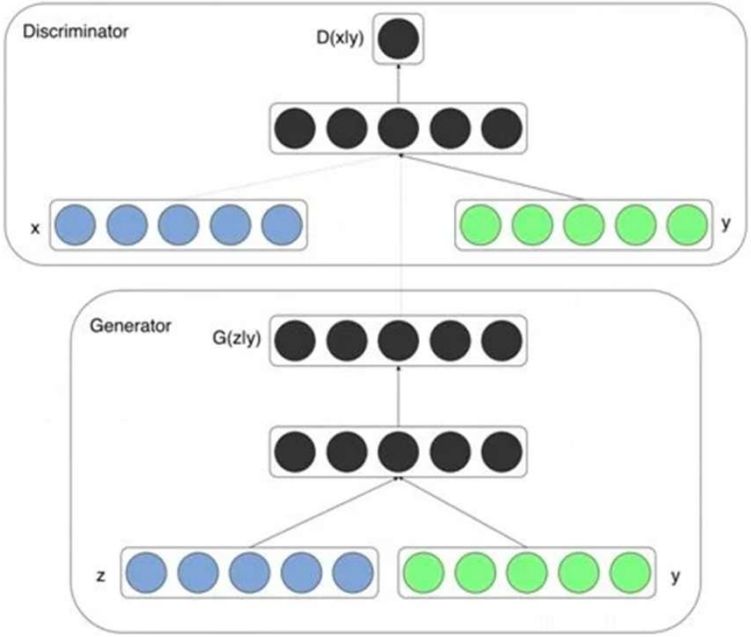
GAN의 한계

- **Mode Collapse**: Generator가 일부 모드만 학습.
  - 동일한 유형의 샘플만 생성.
- **Training Instability:** 학습이 불안정하고 수렴하지 않을 수 있음.
- **평가 어려움:** 생성 품질 평가 지표 부족.



GAN의 개선 방안

- **Wasserstein GAN (WGAN)**
  - 안정적인 학습을 위한 Wasserstein Distance 도입
- **Conditional GAN (cGAN)**
  - Class label y 추가를 통해 제어된 데이터 생성.
  - 예: 특정 카테고리의 이미지 생성.
- **StyleGAN**
  - 고품질 이미지 생성.
  - 스타일 변환 및 제어 가능.

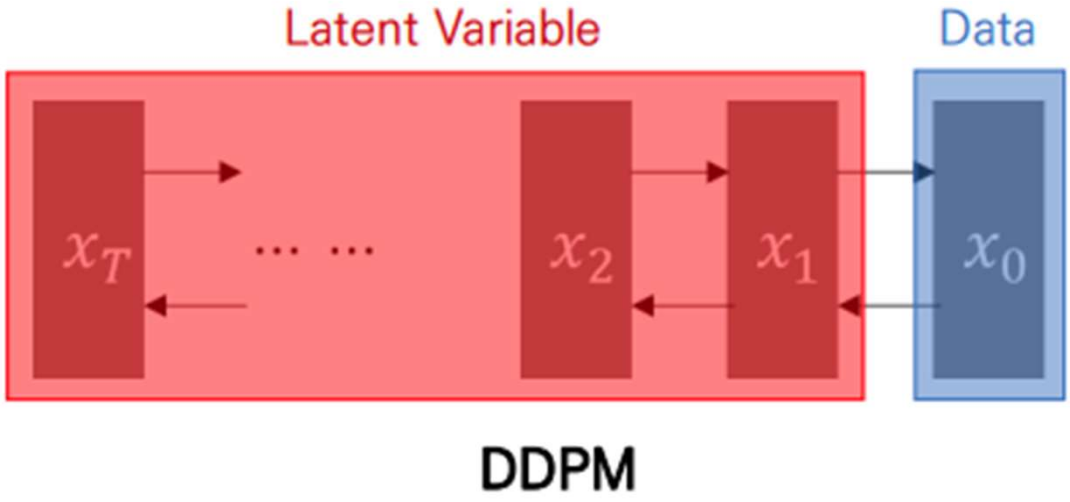


GAN 실습

- Lec5-1.ipynb

Diffusion Model란?

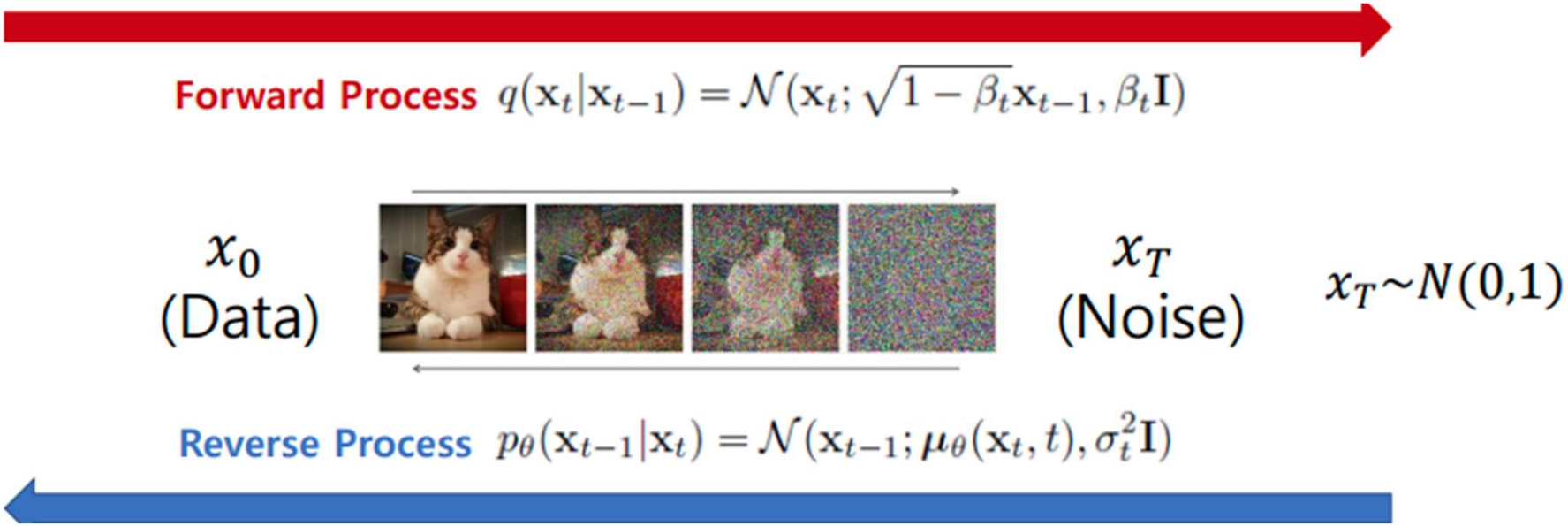
- Diffusion Model은 데이터 분포를 학습하여 노이즈를 제거하고 새로운 데이터를 생성하는 생성 모델.
- Forward Process: 데이터를 점진적으로 노이즈화하여 가우시안 분포로 변환.
- Reverse Process: 노이즈화된 데이터를 복원하여 원본 데이터로 재구성.
- DDPM (Denoising Diffusion Probabilistic Model)의 기본 아이디어:
  - 데이터를 점진적으로 노이즈화하고 이를 역으로 복원하는 과정.



DDPM

Forward and Reverse Process

- **Forward Process**:
  - 데이터를 점진적으로 노이즈화하여 가우시안 분포로 만듦.
  - 데이터(x_0) + 노이즈 -> 랜덤 노이즈 (x_T)
- **Reverse Process:**
  - 노이즈화된 데이터를 복원하여 원본 데이터로 재구성.
  - 랜덤 노이즈 (x_T) + 노이즈 제거-> 데이터(x_0)
  - 학습된 네트워크를 사용하여 원본 데이터를 재구성.



DDPM

- **Training**:
  - 각 시점에서 제거할 노이즈에 대해 학습
- **Sampling:**
  - 각 시점 마다 노이즈 제거하며 데이터 생성.

Algorithm 1 Training

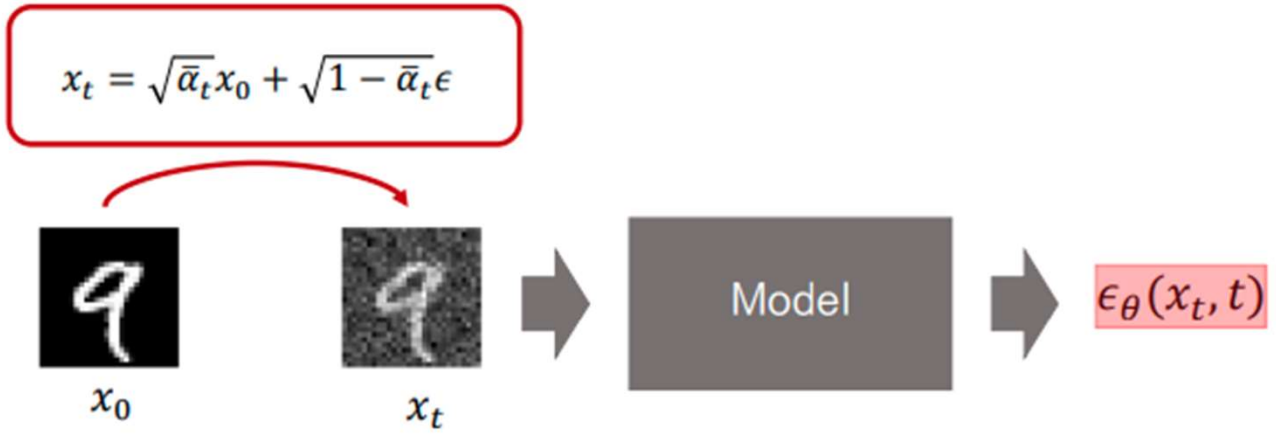
1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = 0$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

DDPM

- **Training**:
  - 각 시점에서 제거할 노이즈에 대해 학습
- Forward process는 모든 시점을 거치지 않음 → 효율적으로 학습 가능



DDPM

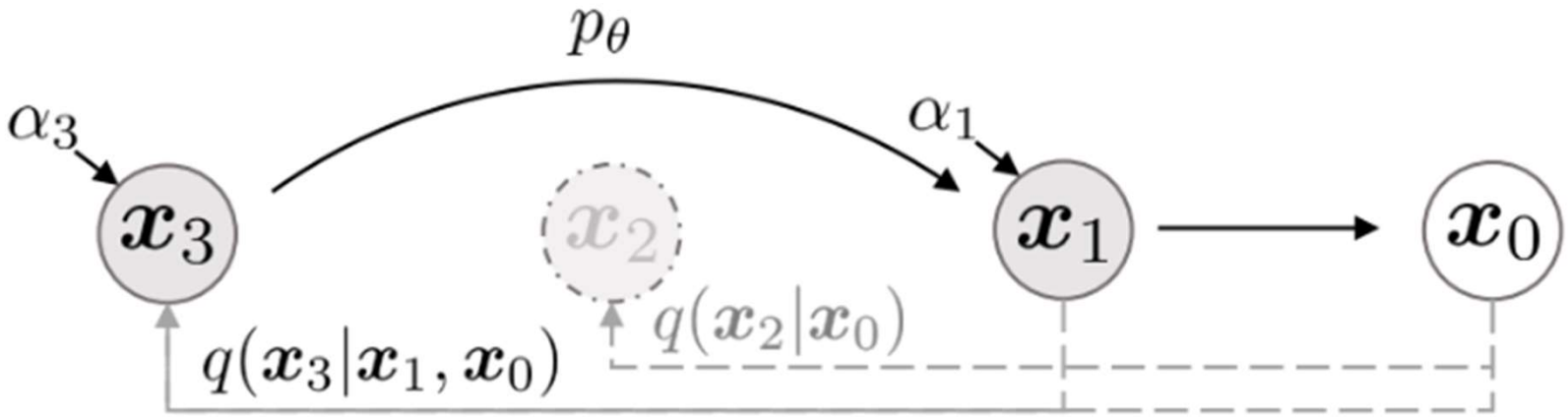
- **Sampling:**
  - 각 시점 마다 노이즈 제거하며 데이터 생성.
- Reverse process는 모든 timestep을 거침 → sampling speed 느림

DDPM의 한계

- 계산 비용:
  - Forward 및 Reverse Process로 인한 높은 연산 요구량.
- 샘플링 시간:
  - Reverse Diffusion의 느린 샘플링 속도.
- 복잡성:
  - 모델 설계 및 학습 과정의 복잡성.

DDPM의 개선 방안

- **샘플링 속도 개선**:
  - DDIM (Denoising Diffusion Implicit Models) 도입.
  - 가속화된 Reverse Process.
- **Hybrid Approach**:
  - GAN과 Diffusion Models의 결합.
- **Latent Diffusion**:
  - 데이터 표현을 압축하여 계산량 감소.

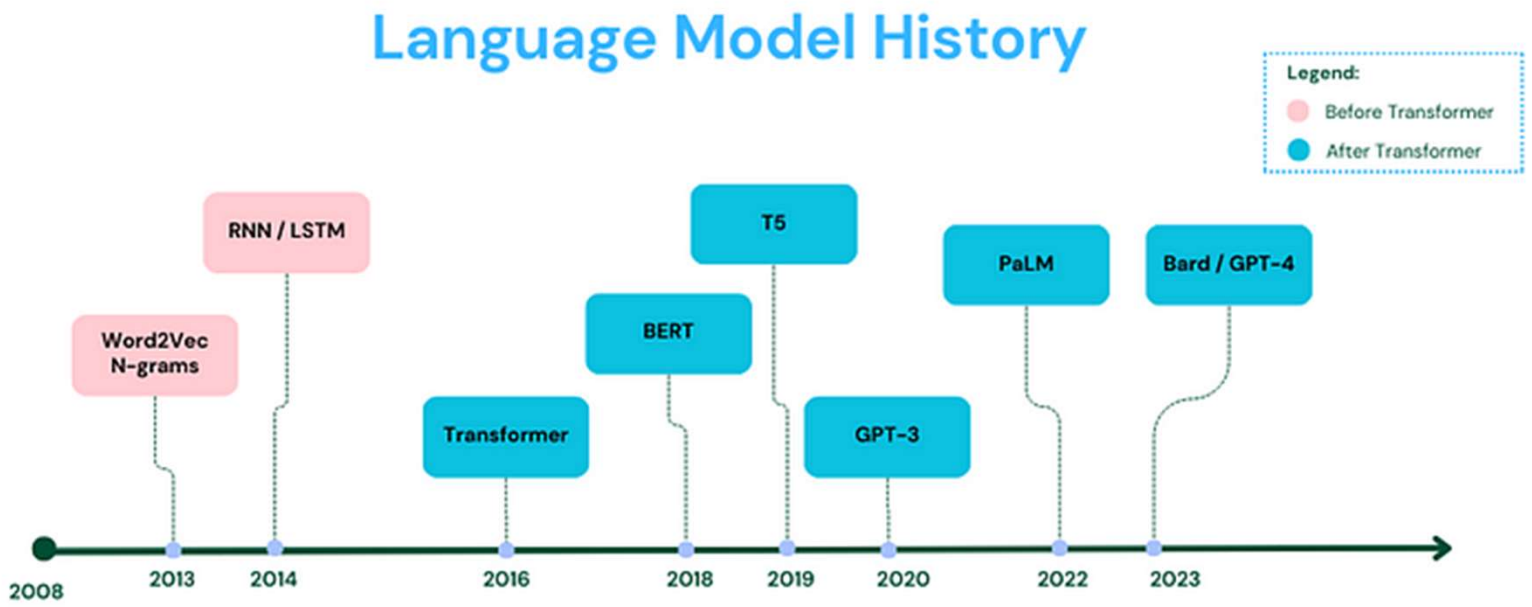


DDPM 실습

- Lec5-2.ipynb

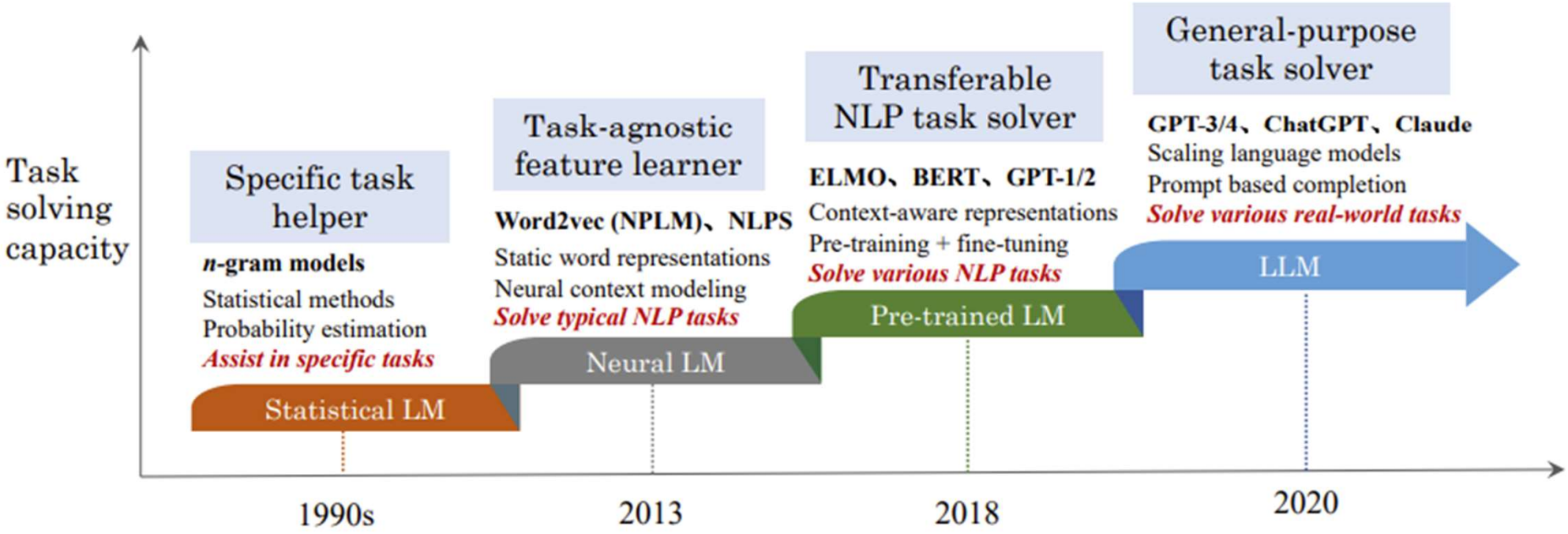
LLM

- LLM은 대규모 텍스트 데이터를 기반으로 학습하여 자연어 이해와 생성 능력을 갖춘 모델
- 발전 과정
  - n-그램 모델 → RNN, LSTM → Transformer → LLM



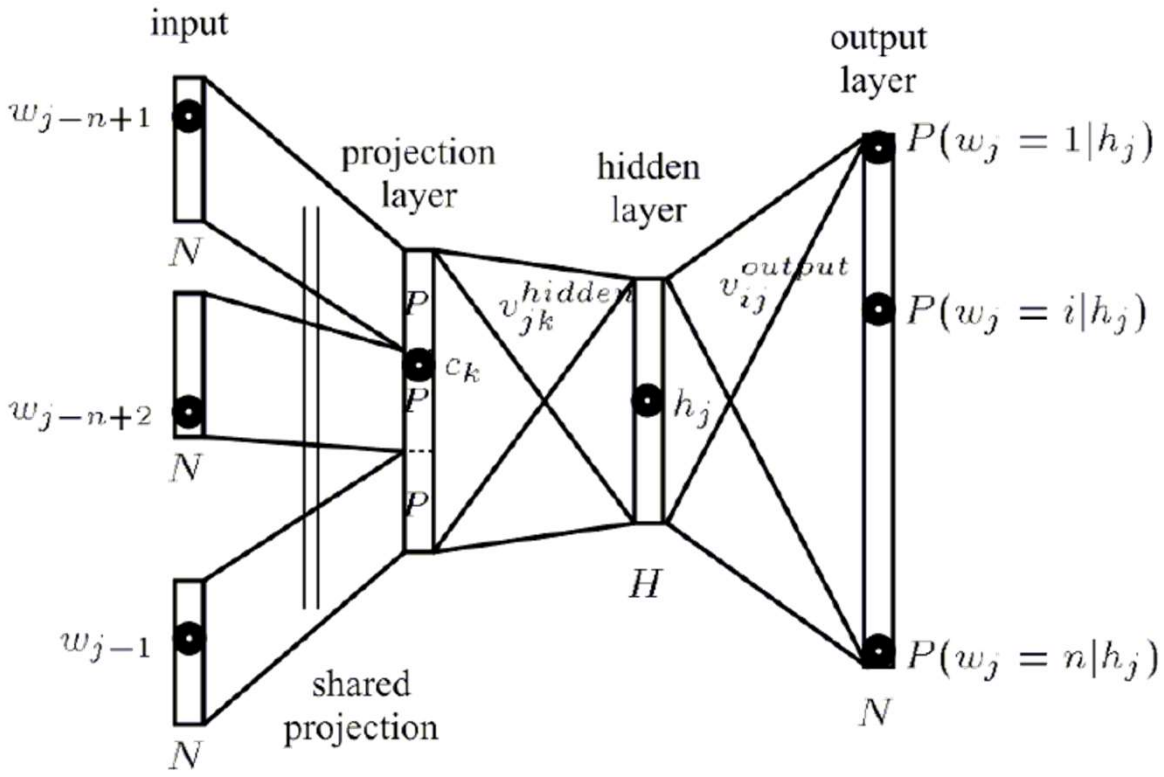
LLM

- LLM은 대규모 텍스트 데이터를 기반으로 학습하여 자연어 이해와 생성 능력을 갖춘 모델
- 주요 모델 예시: GPT, LLaMA, Gemini, Qwen등



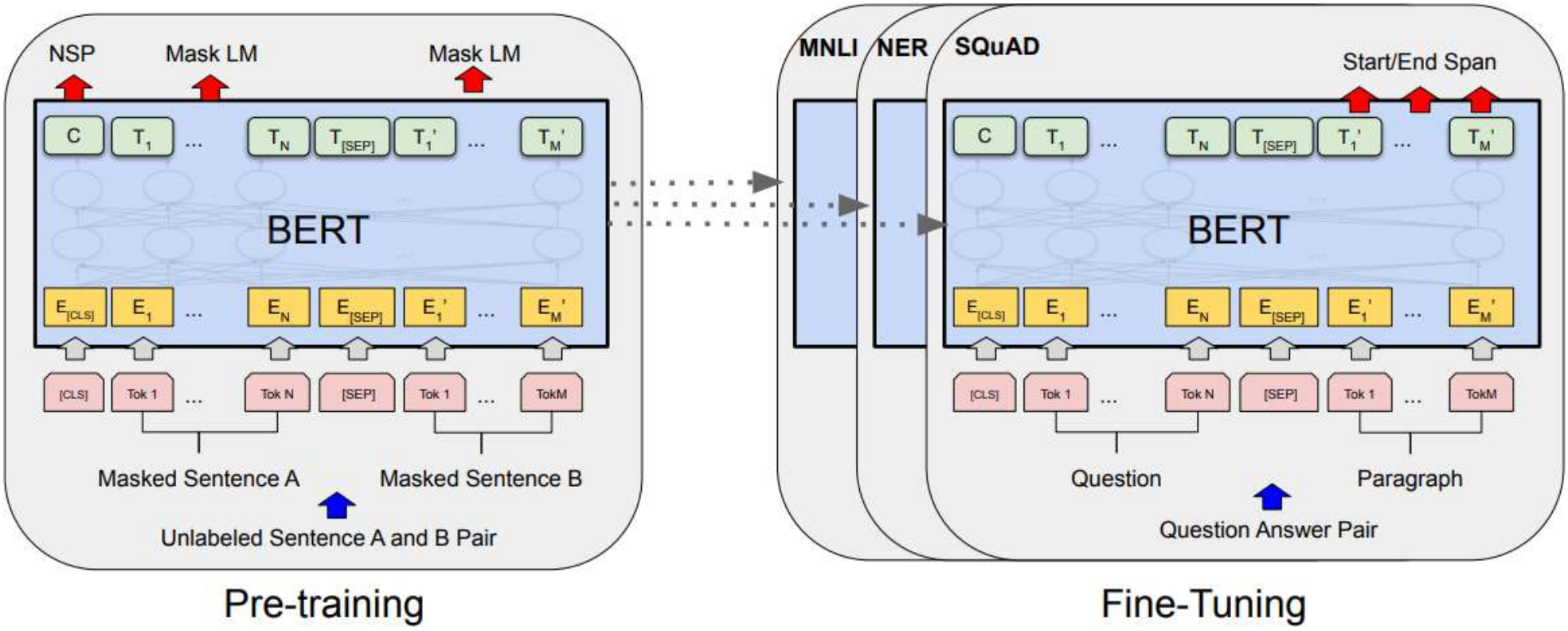
Word2Vec

- Define a model that assigns prediction between a center word w_t and $context$: $P(context|w\_t)$
- Loss function J=1−$P(w\_(-t)\ |w\_t)$
- Keep adjusting the vector representation of words to minimize the loss



BERT

- Pre-train via simple task to utilize in downstream tasks
- BERT has only encoder

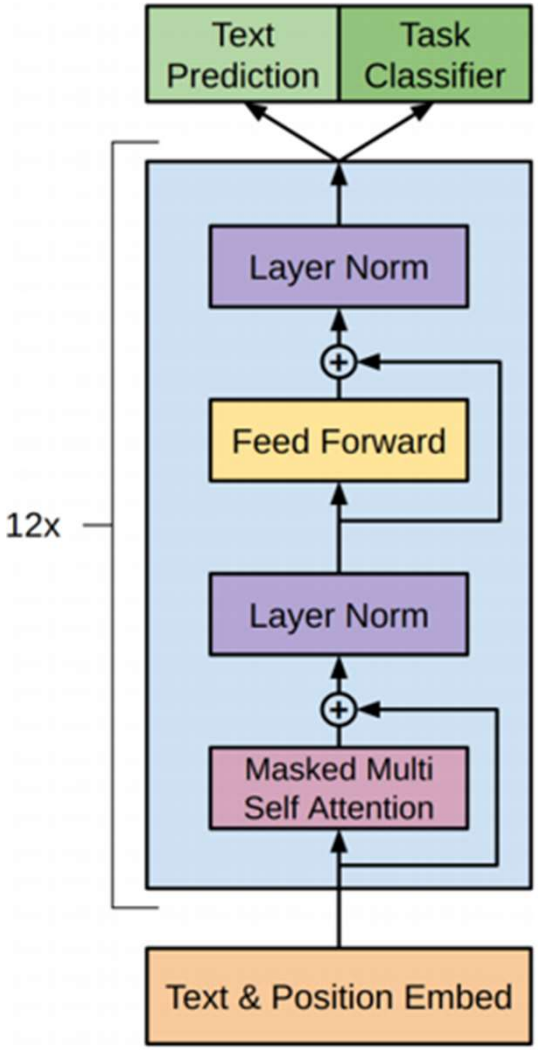


GPT

- 다음 단어를 맞추도록 학습

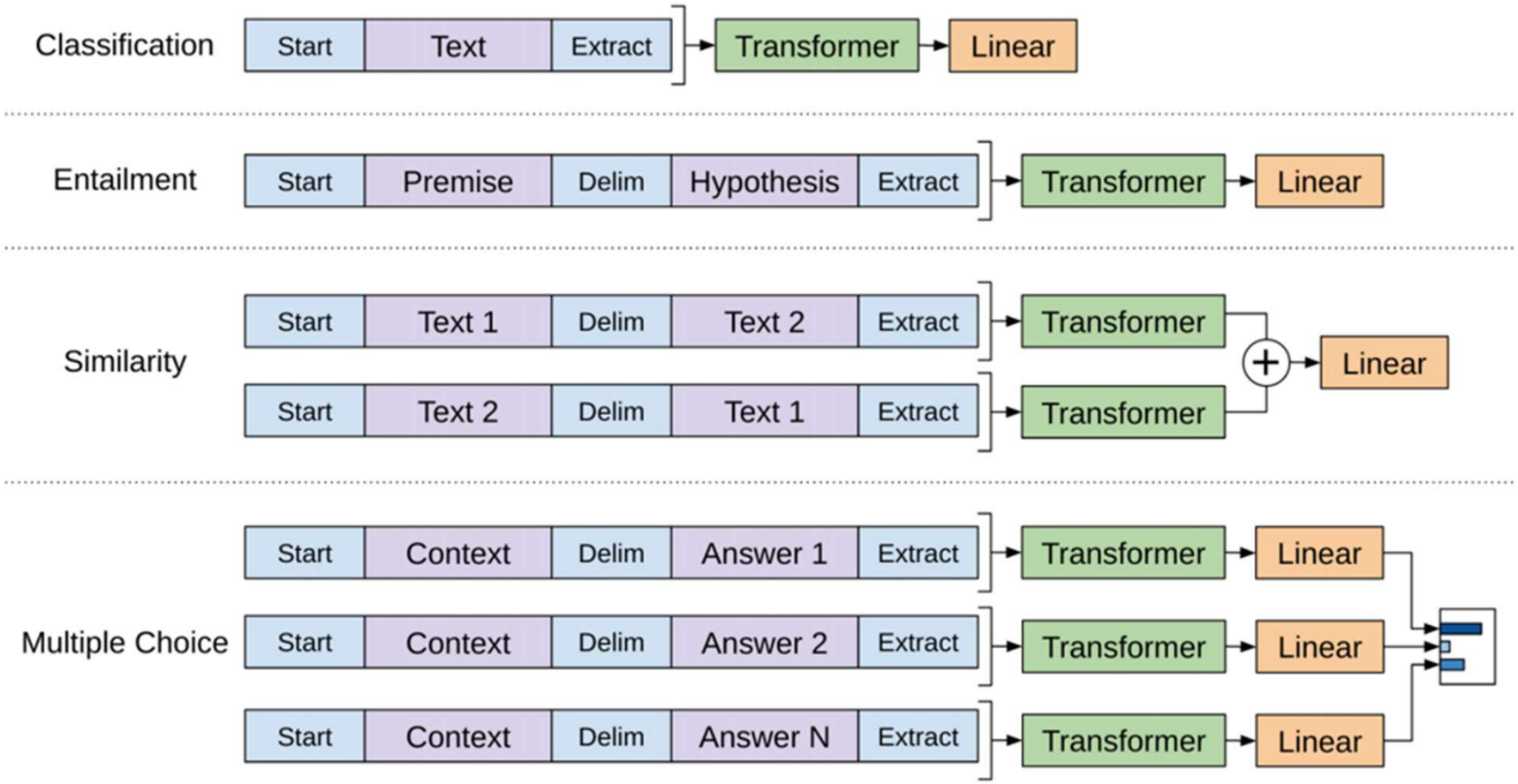
$$L_1(u) = \sum_i logP(u_i|u_{i-k}, \ldots, u_{i-1}; \theta)$$

- GPT has only decoder



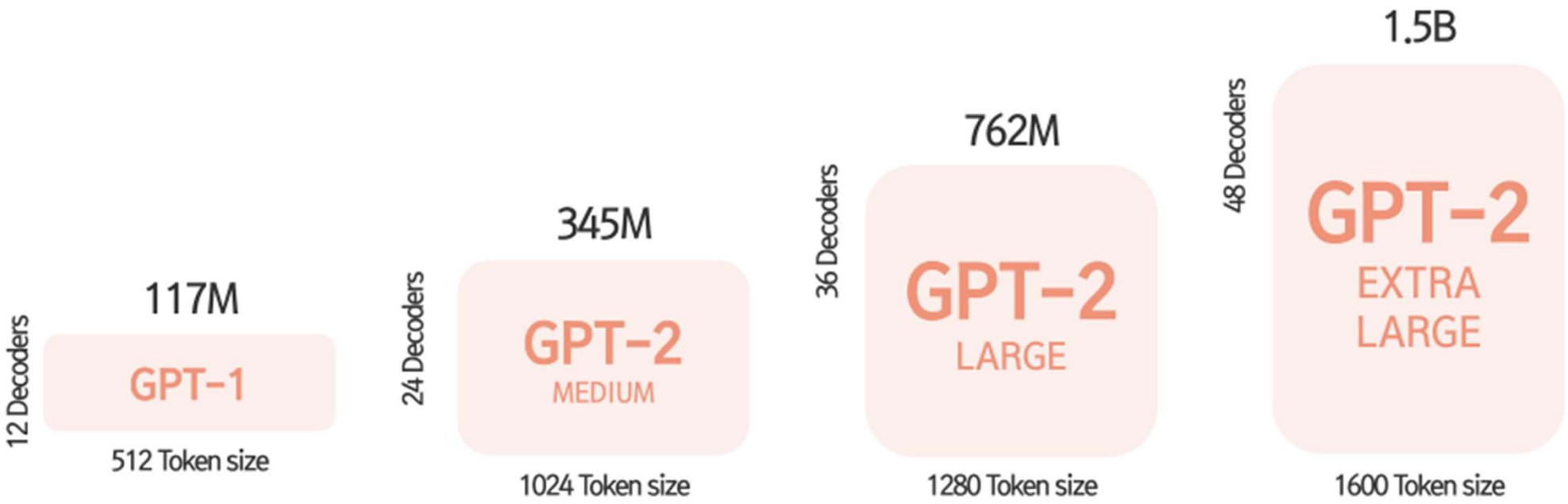
GPT

- Downstream task에 대해 fine-tuning



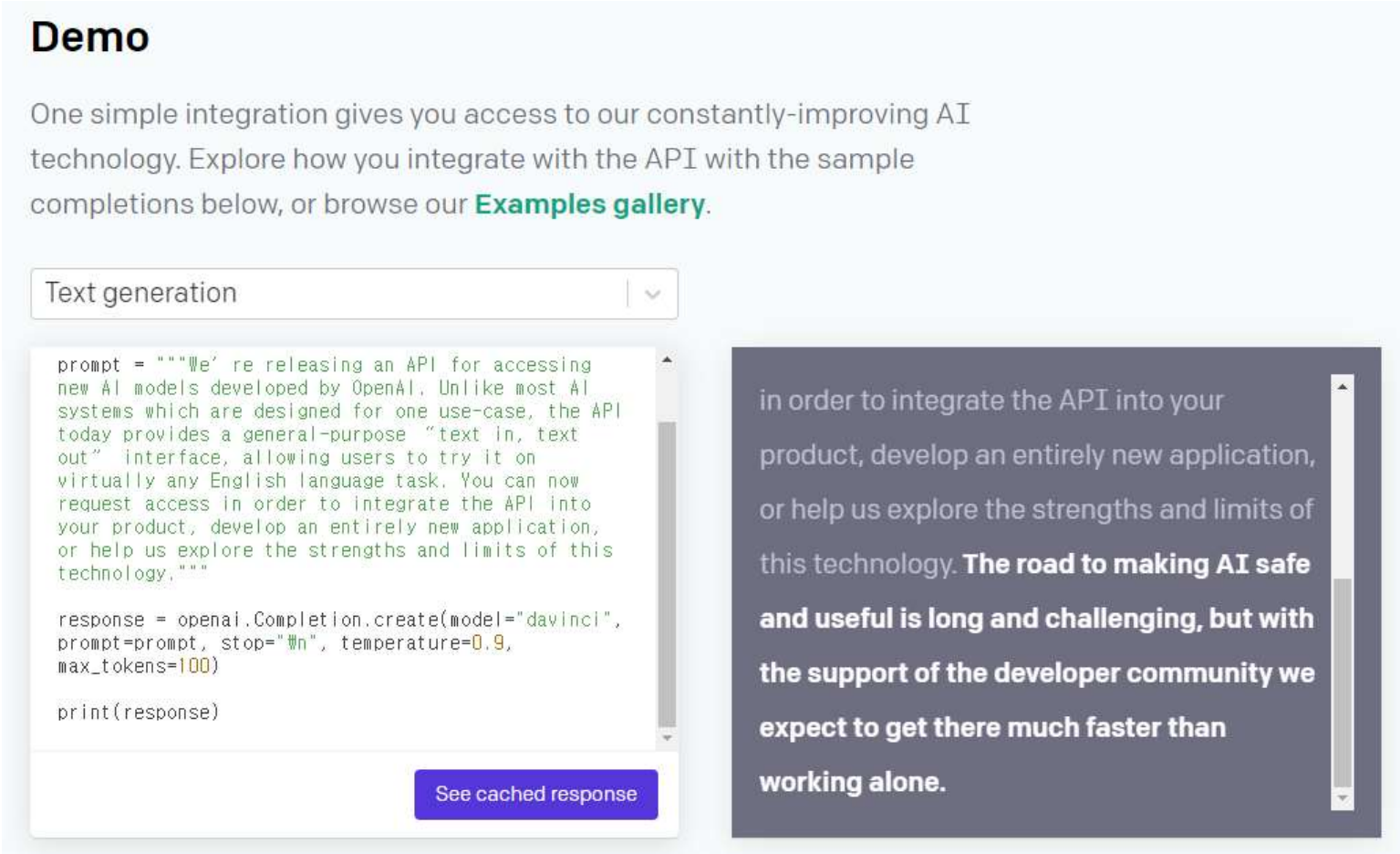
GPT2

- 더 큰 데이터셋 & 더 많은 parameter를 통해 학습
- Downstream task를 학습시킬 때 GPT1처럼 ordered sequence로 변환하지 않고 그대로 학습



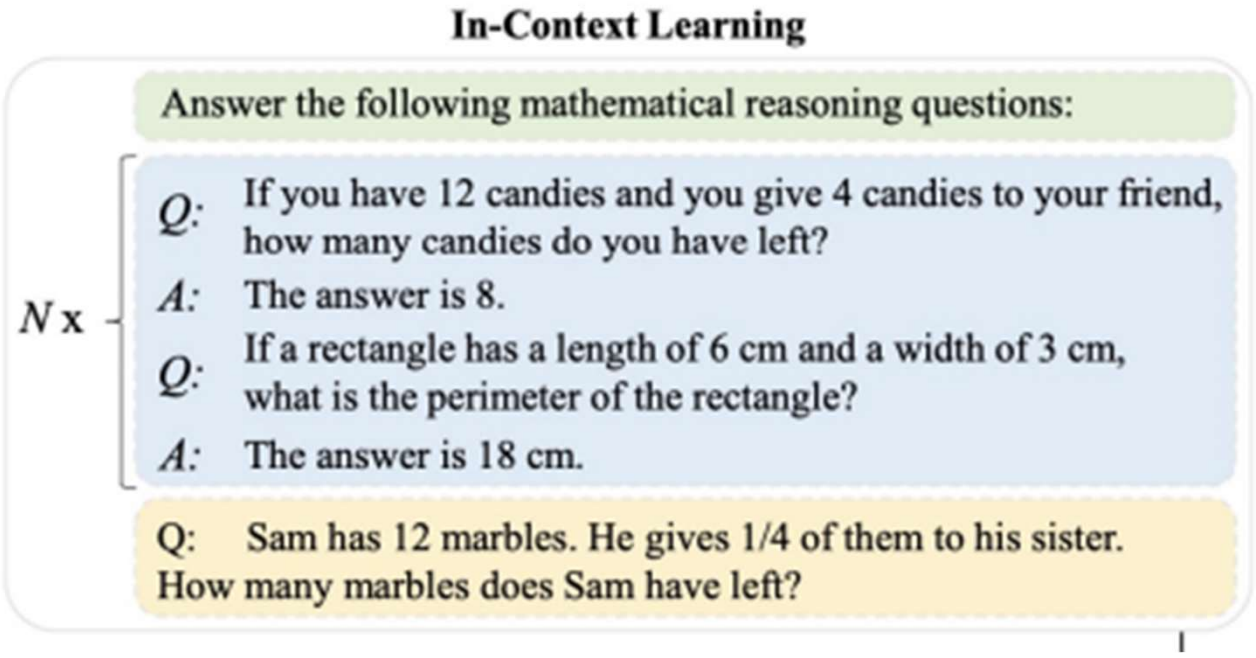
GPT3

- 175B parameter
- Sparse transformer 사용



GPT3

- Finetuning 진행하지 않음
- In-context learning으로도 충분한 성능 확보



GPT3

- Zero-shot
- One-shot
- Few-shot

Prompt // Zero Shot

```
Classify this review:
I loved this movie!
Sentiment:
```

Prompt // One Shot

```
Classify this review:
I loved this movie!
Sentiment: Positive

Classify this review:
I don't like this
chair.
Sentiment:
```

Prompt // Few Shot

```
Classify this review:
I loved this movie!
Sentiment: Positive

Classify this review:
I don't like this
chair.
Sentiment: Negative

Classify this review:
Who would use this
product?
Sentiment:
```

ChatGPT

- Aligning LLM to Follow Instructions

Prompt:
What is the purpose of the list C in the code below?

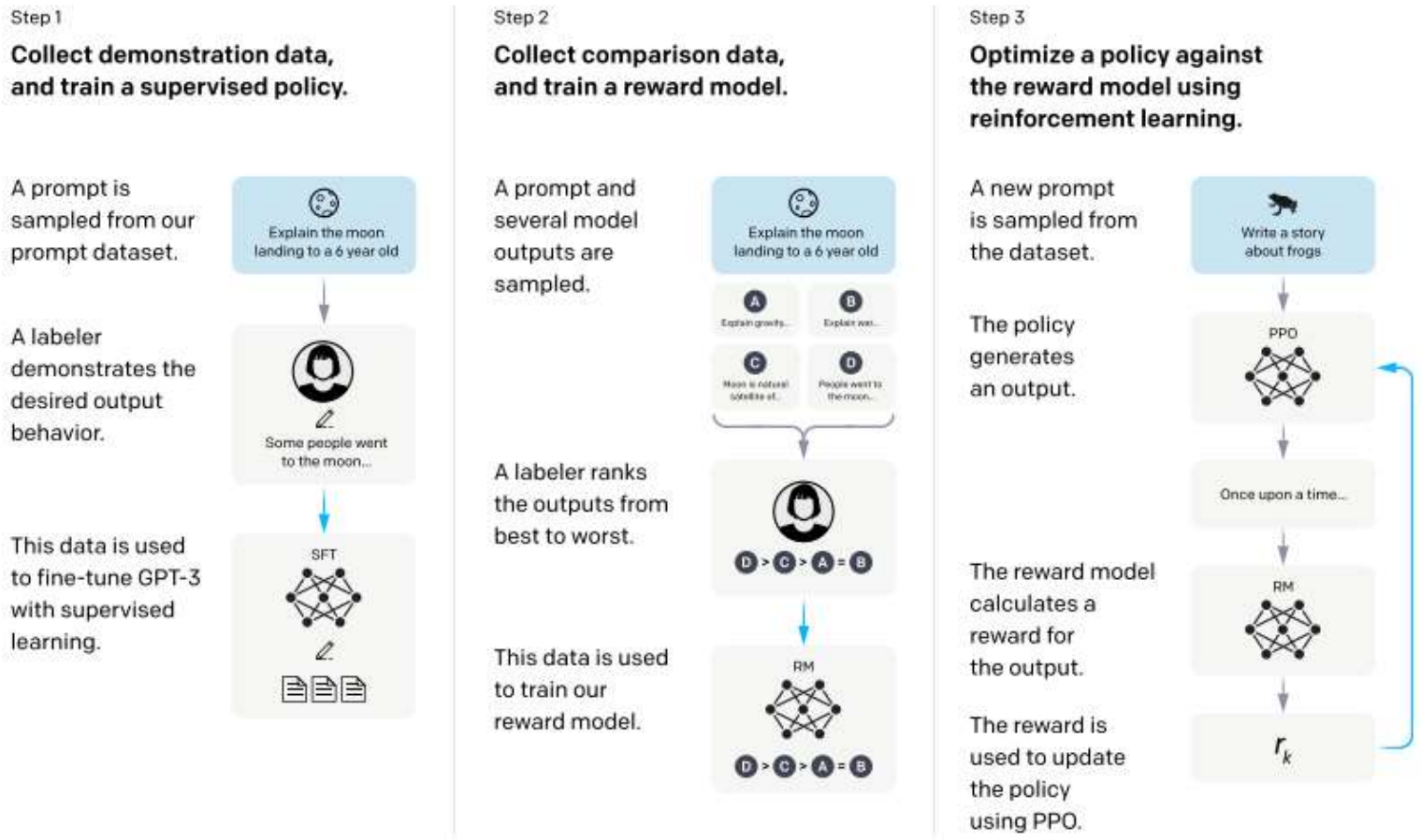
```
def binomial_coefficient(n, r):
    C = [0 for i in range(r + 1)];
    C[0] = 1;
    for i in range(1, n + 1):
        j = min(i, r);
        while j > 0:
            C[j] += C[j - 1];
            j -= 1;
    return C[r]
```

GPT-3 175B completion:
A. to store the value of C[0]
B. to store the value of C[1]
C. to store the value of C[i]
D. to store the value of C[i - 1]

InstructGPT 175B completion:
The list C in this code is used to store the values of the binomial coefficient as the function iterates through the values of n and r. It is used to calculate the value of the binomial coefficient for a given value of n and r, and stores the result in the final return value of the function.

ChatGPT

- Reinforcement Learning from Human Feedback(RLHF)



더 알아볼만한 것들

- Small Language Model
  - 대규모 언어 모델에 비해 경량화된 모델로, 특정 태스크나 환경에 최적화되어 빠르고 효율적으로 작동함.
- Prompt Engineering
  - 언어 모델이 원하는 출력을 생성하도록 입력 프롬프트를 설계하거나 조정하는 기법.
- Parameter Efficient Fine Tuning
  - 전체 모델이 아닌 일부 파라미터(예: 어댑터 레이어)만 학습시켜 효율적으로 성능을 향상시키는 기술.
- Agent
  - 특정 목표를 수행하기 위해 언어 모델을 활용해 데이터를 처리하고 의사 결정을 내리는 자동화된 시스템.