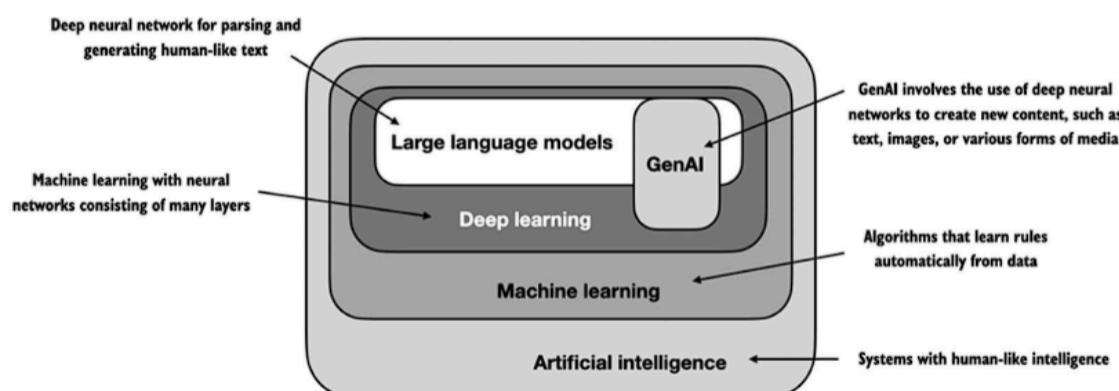


## 1 대규모 언어모델의 이해

### 1. LLM의 이해

#### 1.1 LLM이란 무엇인가?

- LLM, 대규모 언어 모델은 인간이 하는 것과 유사하게, 텍스트를 이해하고 생성하며 응답하기 위해 설계된 신경망이다.
  - ‘대규모’ : “모델의 크기(파라미터 수) + 훈련한 방대한 데이터셋”을 의미함
  - ‘파라미터’ : 시퀀스에서 다음 단어를 예측하기 위해 훈련 중 최적화되는 네트워크의 가중치
- LLM은 Transformer 아키텍처를 사용한다.
  - Attention을 통한 입력의 다양한 부분에 선택적으로 주의를 기울일 수 있게 해줌



- AI는 언어 이해, 패턴 인식, 의사 결정 등 인간과 유사한 지능이 요구되는 작업을 수행할 수 있는 기계를 만드는 넓은 분야를 포괄한다.
- 머신러닝은 AI를 구현하는 데 사용되는 알고리즘의 개발을 의미한다.
- 딥러닝은 다중 계층 신경망을 사용하는 데 중점을 둔 머신 러닝의 특화된 분야이다.
- 위의 계층적 그림이 보여주는 바와 같이, LLM은 인간과 유사하게 텍스트를 처리하고 생성하는 능력을 활용한 딥러닝 기법의 특정 어플리케이션을 나타낸다.

#### 1.2 LLM의 응용분야

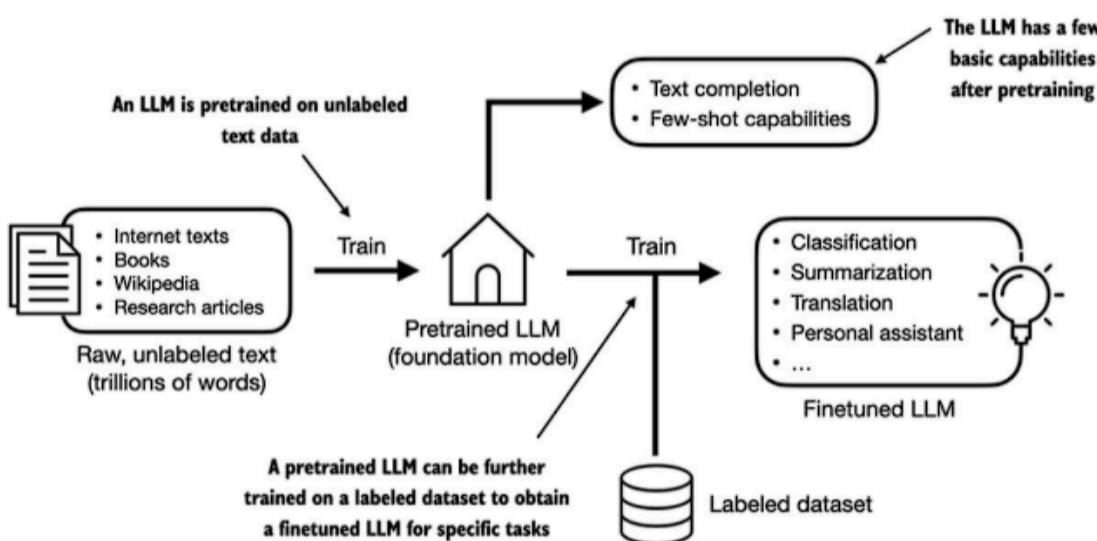
- LLM은 비정형 텍스트 데이터를 파싱하고 이해하는 뛰어난 능력 덕분에 다양한 분야에서 널리 응용되고 있다.

1. 자연어 생성 (Text Generation)
  - 사람이 쓴 것처럼 자연스러운 문장을 생성하는 기술
2. 기계 번역 (Machine Translation)
  - 한 언어의 문장을 다른 언어로 정확하게 번역
3. 요약 (Summarization)
  - 긴 텍스트를 핵심만 간추려 요약
4. 문서 분류 & 감정 분석 (Text Classification & Sentiment Analysis)
  - 문서나 문장의 카테고리 또는 감정을 자동으로 판단

- 위의 분야 이외에도 응용 분야는 거의 무궁무진하며, 텍스트를 파싱하고 생성하는 거의 모든 작업을 자동화하는 데 매우 유용하다.

### 1.3 LLM 구축 및 사용 단계

- 밑바닥부터 LLM을 코딩하는 것은 그 메커니즘과 한계를 이해하는 데 훌륭한 연습과 경험이 된다.
  - 또한, 이러한 코딩은 기존의 오픈 소스 LLM 아키텍처를 pretrain하거나 특정 도메인 데이터셋이나 작업에 맞게 fine-tuning하는 데 필요한 지식을 제공한다.
- 맞춤 제작된 LLM
  - 데이터 프라이버시 측면에서 여러가지 이점을 제공한다.
  - 노트북이나 스마트폰과 같은 **로컬에서 동작하도록 구현된 LLM은 개발자에게 완전한 자율성을 부여하여 모델 업데이트 및 수정을 자유롭게 할 수 있게 한다.**



- 위 그림은 LLM을 만드는 일반적인 과정을 나타낸다.
- LLM을 만드는 일반적인 과정은 **사전 훈련**과 **미세 조정**으로 구성된다.

#### 1. 사전 훈련

- LLM이 언어에 대한 광범위한 이해를 위해 대규모의 다양한 데이터셋으로 처음 훈련되는 초기 단계를 의미함
- 대규모의 “원시(raw) 텍스트”로 훈련시키는 것
- 초기 사전 훈련된 LLM을 base model 또는 foundation model이라 부른다
  - 사용자가 제공한 반쯤 작성된 문장을 완성하는 text completion 기능
  - 몇 가지 예제만으로도 새로운 작업을 수행하는 방법을 배울 수 있는 제한된 few-shot 학습 능력

#### 2. 미세 조정

- 사전 훈련된 모델을 특정 작업이나 도메인에 더 구체적인 데이터셋으로 세부적으로 훈련하는 것을 의미함
- 즉, LLM을 라벨이 지정된 데이터로 추가 훈련하는 것

##### 1. 지시 미세 조정 (Instruction tuning)

- 라벨이 지정된 데이터셋이 지시와 그에 대한 답변 쌍으로 구성됨
- ex) 텍스트를 번역하라는 질문하라는 질문과 함께 올바르게 번역된 텍스트가 포함됨

##### 2. 분류 작업을 위한 미세 조정 (finetuning for classification)

- 라벨이 지정된 데이터셋이 텍스트와 관련된 클래스 라벨로 구성됨
- ex) 스팸과 비스팸 라벨이 지정된 이메일

#### ▼ 전통적인 머신러닝 vs LLM의 사전 훈련 학습 방식

##### 1. 전통적인 머신 러닝

- 라벨이 있는 데이터를 필요로 함
- 즉, 입력(x)와 정답 출력(Y) 쌍이 있어야 학습 가능
  - ex) 이미지 → 개/고양이 라벨
- 지도학습 (Supervised Learning)

##### 2. LLM의 사전 훈련 단계

- LLM은 대부분 거대한 텍스트 코퍼스(책, 위키, 웹사이트 등)를 가지고 학습을 시작함
- 이때 정답 라벨이 따로 존재하지 않음,
  - 대신 모델이 스스로 학습 목표를 만들어내는 방식으로 학습함
- 자기 지도 학습 (Self-Supervised Learning)

#### 🔍 자기 지도 학습 자세히 알아보기

- 데이터에서 일부 정보를 숨기고, 그걸 스스로 예측하게 만듬
- 이 예측 작업이 곧 학습의 목적(라벨 역할)을 수행함
- 즉, 사람이 라벨링하지 않아도 데이터가 자체적으로 “문제 + 정답”을 만들어냄

Ex 01) Masked Language Modelig (BERT의 방식)

문장 : 나는 [MASK]을 좋아해요.  
정답 : 커피

- 이때 모델은 [MASK] 위치의 단어를 예측함
  - 입력 : “나는 [MASK]를 좋아해요” / 출력 : “커피”를 맞히도록 학습
- 사람이 직접 “커피”라고 라벨링하지 않아도 문장에서 정답을 유추할 수 있음!

Ex 02) 다음 단어 예측 (GPT의 방식)

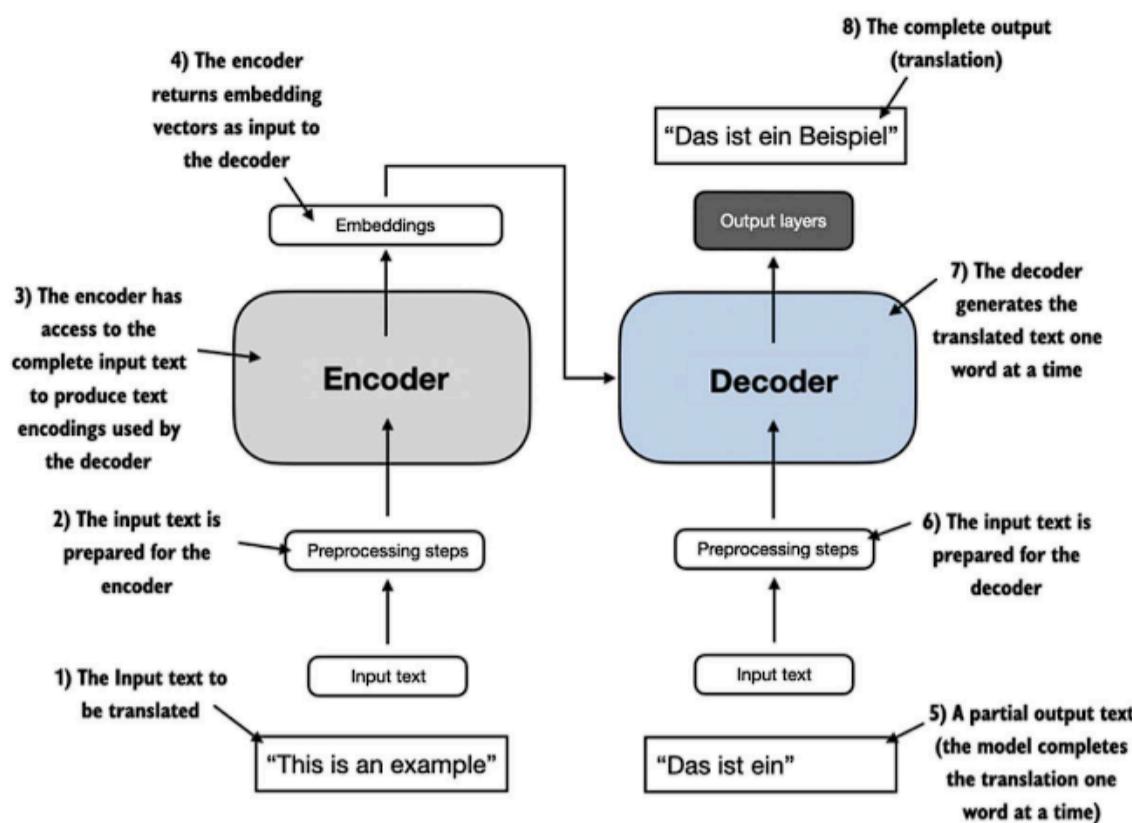
입력 : 나는 커피를  
출력(예측) : 좋아해요

- 문장의 앞부분을 주고, 다음에 올 단어를 예측하게 함
- 이 작업을 계속 반복하면서 언어 패턴을 학습

→ 이를 통해 라벨링 비용을 크게 줄일 수 있고, 범용적인 언어 능력을 학습 가능하게된다.

## 1.4 트랜스포머 아키텍처 소개

- 대부분의 현대 LLM은 트랜스포머 아키텍처에 기반한다.



- 위 그림은 초기 트랜스포머 아키텍처를 단순화한 것이다.
- 이는 언어 번역을 위한 딥러닝 모델이며, 트랜스포머는 두 부분으로 구성된다.
  - 인코더 : 입력 텍스트를 처리하고, 텍스트의 임베딩 표현을 생성함
  - 디코더 : 번역된 텍스트를 한 단어씩 생성하기 위해 인코더가 생성한 임베딩 표현을 사용함
    - 위의 그림에서 디코더는 초기 입력 텍스트 ("This is an example")와 부분적으로 번역된 문장("Das ist ein")을 사용하여 최종 단어("Beispiel")만을 생성하여 번역을 완료한다.
- 트랜스포머와 LLM의 핵심 구성 요소는 바로 **self-attention mechanism**이다.
  - 이는 모델이 시퀀스 내의 서로 다른 단어나 토큰의 중요성을 상대적으로 평가할 수 있게 함
  - 입력 데이터 내의 '장기적인 종속성 및 '문맥적 관계'를 포착할 수 있게 하여 일관되고 문맥적으로 적절한 출력을 생성할 수 있는 능력을 향상시킴**
- 트랜스포머 아키텍처의 후속 변형으로 BERT와 다양한 GPT 모델이 있다.

#### 1. BERT

- 초기 트랜스포머의 인코더 서브모듈을 기반으로 함
- 마스킹된 단어나 숨겨진 단어를 예측하는 'Masked Word'(prediction)에 특화됨
- '감정 예측', '문서 분류'와 같은 텍스트 분류 작업에서 강점을 발휘함

#### 2. GPT

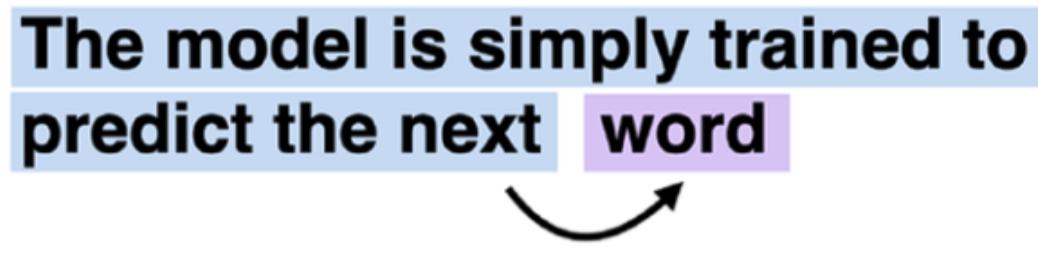
- 트랜스포머 아키텍처의 디코더 부분에 중점을 두고 텍스트 생성을 요구하는 작업을 위해 설계됨
- 기계 번역, 텍스트 요약, 소설 작성, 컴퓨터 코드 작성 등이 포함됨

## 1.5 대규모 데이터셋의 활용

- GPT 및 BERT와 같은 인기 있는 모델의 대규모 훈련 데이터셋은 다양한 주제와 자연 및 컴퓨터 언어를 포함한 수십억 단어의 텍스트를 포함한다.
- 중요한 점은 **훈련 데이터셋의 규모와 다양성**이 모델이 언어 구문, 의미 및 문맥을 포함한 다양한 작업에서도 우수한 성능을 발휘할 수 있게 한다는 것이다.
- LLM 사전 훈련에는 상당한 자원이 필요하며, 매우 비용이 많이 듈다.
  - GPT-3 사전 훈련 비용은 클라우딩 컴퓨팅 크레딧으로 약 460만 달러 정도로 추정된다.
- 다행히도, 사전 훈련된 많은 LLM은 오픈 소스 모델로 제공되어 훈련 데이터의 일부가 아닌 텍스트를 작성, 추출 및 편집하는데 일반적인 도구로 사용할 수 있다.
  - LLM은 상대적으로 작은 데이터셋으로 특정 작업에 맞게 미세 조정될 수 있어 필요한 계산 자원을 줄이고 특정 작업에서 성능을 향상 시킬 수 있다.

## 1.6 GPT 아키텍처 자세히 보기

- GPT는 Generative Pretrained Transformer의 약자이며, OpenAI의 Radford 등이 작성한 논문 'Improving Language Understanding by Generative Pre-Training'에서 처음 소개되었다.
  - GPT-3는 GPT의 확장된 버전으로, 더 많은 파라미터와 더 큰 데이터셋으로 훈련되었다.
  - ChatGPT에서 제공된 원래 모델은 OpenAI의 InstructGPT 논문에서 제시된 방법을 사용하여 큰 지시(Instruction) 데이터셋으로 GPT-3를 미세 조정하여 생성한 것이다.



- GPT 모델의 다음 단어 사전 훈련 작업에서, 시스템은 문장의 앞부분을 보고 다음에 올 단어를 예측하는 법을 학습한다.
  - 다음 단어 예측 작업은 **self-supervised learning**의 한 형태로, 자체 라벨링의 한 형태임

### ▼ [GPT 사전 훈련 과정이 왜 라벨링이 필요없는가?]

1. 입력 문장 - 사람이 제공

나는 오늘 아침에 커피를 마셨다.

2. 토큰화(Tokenization)

- 예를 들어, 이런 식으로 나누어졌다고 가정하면:

[나는, 오늘, 아침에, 커피를, 마셨다, .]

3. 토큰 시퀀스를 학습 데이터로 변환

- GPT는 다음처럼 (입력, 정답) 쌍을 만들어 학습한다.

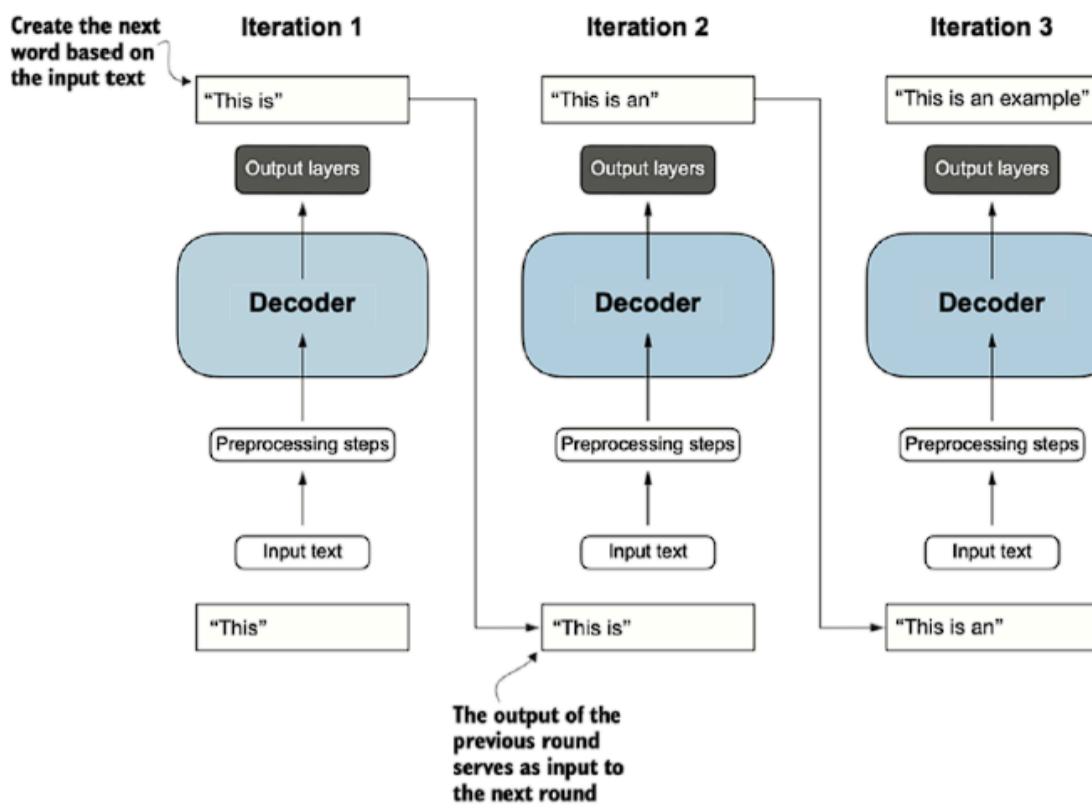
입력 시퀀스	정답(예측할 토큰)
[나는]	오늘
[나는, 오늘]	아침에
[나는, 오늘, 아침에]	커피를
[나는, 오늘, 아침에, 커피를]	마셨다
[나는, 오늘, 아침에, 커피를, 마셨다]	.

4. 예측 → 손실 계산 → 파라미터 업데이트

- 모델은 예측한 토큰 분포를 출력한다(e.g. softmax 결과)
- 예측값과 실제 정답 토큰을 비교해 cross-entropy loss 계산

- 일반적인 GPT 아키텍처는 기본 트랜스포머 아키텍처와 비교했을 때 상대적으로 간단하다.

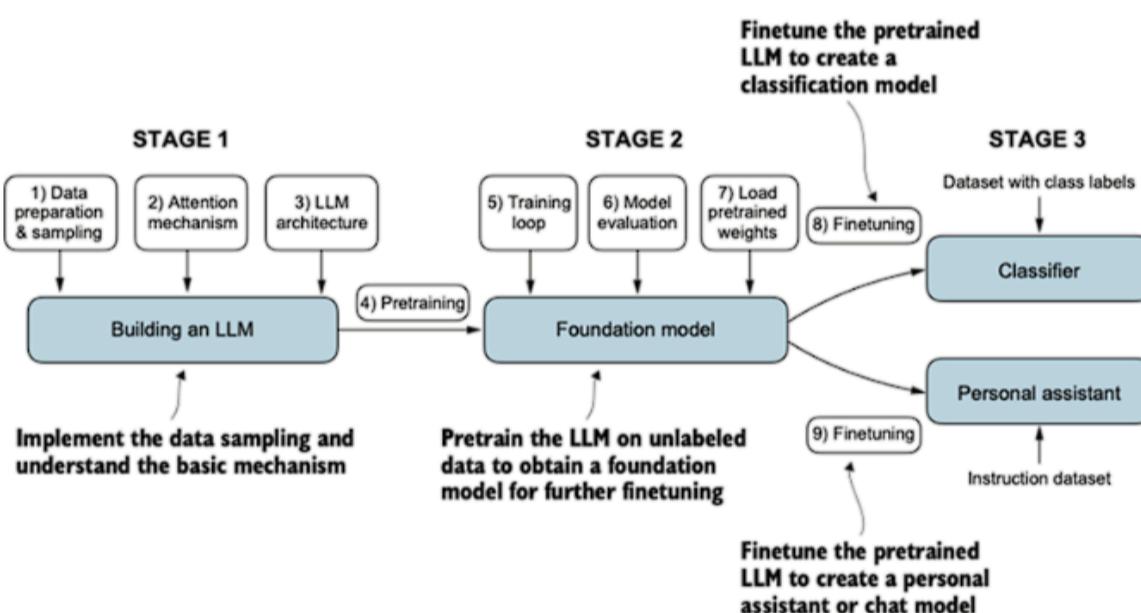
- 본질적으로, 인코더 없이 **디코더 부분만을 사용함**
- GPT와 같은 디코더 스타일 모델은 텍스트를 한 번에 한 단어씩 예측하여 생성함
  - 자동 회귀 모델(auto-regressive model)로 간주됨
- GPT에서는 각 새로운 단어가 이전 시퀀스를 기반으로 선택되어 결과 텍스트의 일관성을 향상시킴



- 위처럼 GPT 아키텍처는 초기 트랜스포머의 디코더 부분만을 사용한다.
- 이 아키텍처는 단방향, 왼쪽 → 오른쪽으로 처리하도록 설계되어 텍스트 생성 및 다음 단어 예측 작업에 잘 부합하며, 한 번에 한 단어씩 반복적으로 텍스트를 생성한다.

## 1.7 대형 언어 모델 구축

- 다음 챕터에서는 밑바닥부터 LLM을 코딩할 것이다.
- GPT의 기본 아이디어를 청사진으로 삼아, 밑의 그림과 같이 3단계를 통해 이를 다룰 것이다.



- 기본적인 데이터 전처리 단계를 배우고, 모든 LLM의 핵심인 Attention mechanism을 코딩
- 새로운 텍스트를 생성할 수 있는 GPT 유사 LLM을 코딩하고, 사전 훈련 + LLM 평가의 기본 사항을 다룸
  - 작은 데이터셋을 사용하여 교육 목적으로 훈련을 구현하는 데 중점을 둘 것임
- 사전 훈련된 LLM을 가져와 질문에 답하거나, 텍스트를 분류하는 것과 같은 지시를 따르도록 fine-tuning할 것