데이터과학응용III Assignment3

데이터과학과 2021320322 윤민서

1. LLM

1. LLM이란 무엇인가

LLM은 대규모 언어 모델(Large Language Model)의 약자로 기본적인 상식부터 프로그래밍과 같은 창의성을 요구하는 전문 분야를 포함하는 추론 작업까지 뛰어나게 수행할 수 있는 언어 모델이다. LLM은 모델 훈련 과정에서 수십억 개 이상의 매개변수를 학습한다. 또한 이를 훈련하기 위해 대량의 데이터를 사용한다. LLM은 ChatGPT와 같은 서비스 덕분에 널리 퍼지게 되었다.

LLM은 입력된 텍스트를 다음 토큰 또는 단어를 예측하는 방식으로 작동한다. 수천억 개 수준의 LLM이 등장하기 전까지는 미세 조정(fine-tuning)만이 특정 작업을 수행할 수 있도록 하는 유일한 방법이었다. 하지만 GPT-3와 같은 더 큰 모델들이 등장한 후 프롬프트 엔지니어링을 통해 유사한 결과를 달성할 수 있다.

LLM이 좋은 성능을 낼 수 있던 것은 Auto-regressive한 특성의 Transformer 구조 덕분이다. Transformer를 깊게 쌓은 구조로 모델을 구축한 후 self-supervised learning을 통한 데이터 코퍼스에서 사전 훈련(pre-trained)된다. 이후 미세 조정 단계에서 인간의 선호도와 일치하는 결과를 생성하기 위해 최근에는 인간 피드백을 통한 강화학습(Reinforcement Learning with Human Feedback, RLHF)과 같은 기술을 사용한다.

그러나 이와 같이 훈련 방법론이 단순함에도 불구하고 고성능 컴퓨터의 GPU를 사용한 개발이 필수적으로 동반된다. 또한 ChatGPT의 경우에는 성능이 가장 좋다고 알려져 있지만 폐쇄적인 제품 정책으로 인해 성능 재현이 투명하게 이루어지지 않고 있기 때문에 AI 연구의 진전이 제한되는 부분도 있다.

2. 왜 LLM이 중요한가

LLM은 일반적인 언어 이해(NLU)와 언어 생성(NLG)을 할 수 있는 능력을 가지고 있으며 이것은 곧 자연어 처리(NLP) 분야의 작업(task)인 텍스트 요약, 질문 응답, 번역, 창의적 글쓰기 등 다양한 분야에서 활용될 수 있다는 증거이다. 또한 자연어 처리 분야의 모델의 평가를 위한 작업뿐만 아니라 다양한 분야에 응용될 수 있는 여지가 크다. 예를 들어 LLM은 의료, 법률, 교육, 금융 등 다양한 분야에서 전문 지식을 제공할 수 있다. 아직까지는 적극적으로 LLM을 전문 분야에 사용하고 있는 사례는 많지 않지만 현재의 LLM 성능으로는 고객 응대, 데이터 분석, 연구 보조 등의 역할은 충분히 수행한다고 생각할 수 있다.

LLM은 직관적인 채팅 인터페이스를 통해 입력 텍스트를 받는다. 이를 통해 인간과 효과적으로 상호작용 할 수 있다. 이것은 사용자 경험을 개선하고 보다 자연스러운 AI 서비스를 제공한다. 위에서 LLM이 ChatGPT와 같은 서비스 덕분에 널리 퍼지게 되었다고 언급하였는데, 이러한 것이 연구자가 아닌 일반인의 시선에서 LLM을 바라보았을 때 LLM에 접근하는 장벽을 크게 낮추는 역할을 하였다.

현재 LLM을 다른 modality와 결합한 연구도 활발하게 이루어지고 있다. 최신 연구 중에는 그래프 자료구조를 이용해 효과적으로 미세 조정하는 graph-of-thought와 같은 것이 있다. 일반적으로 컴퓨터 비전 분야에서 Diffusion 모델에 텍스트를 입력으로 주는 연구가 유명하다. 또한 비디오에서 주어진 텍스트에 관한 장면을 추출하는 연구도 존재하는데 입력으로 주어지는 텍스트의 이해도를 높이는 방법으로 LLM과 결합하는 방법이 있다. 또한 LLM은 대규모 데이터셋에서 패턴을 추출(supervised learning)할 수 있으며 이를 통해 연구 및 개발에 중요한 정보를 제공한다.

이러한 사례들을 AI 분야에 국한하지 않고 일반적인 시선으로 확장한다면 LLM은 방대한 양의 데이터를 분석하고 이를 기반으로 새로운 지식을 생성하며, 이를 사용자에게 효과적으로 전달할 수 있다는 것이 된다. 또한 LLM이 패턴을 추출할 수 있다는 점은 의사 결정에 도움을 줄 수 있다는 것으로 생각할 수 있다. 언급된 분야들뿐만 아니라 향후 더욱 다양한 분야에서 그 영향력이 확장될 것으로 기대되기 때문에 LLM 이러한 중요성은 계속해서 증가하고 있다.

3. Fine-tuning과 프롬프트 튜닝은 무엇이고 각각의 특징은 무엇인가

Fine-tuning

Fine-tuniung이란 이미 학습된 모델을 기반으로 구조를 새로운 목적에 맞게 변형하고 가중치를 미세하게 조정하여 학습시키는 방법을 말한다. Fing-tuning을 진행하는 방법은 크게 3가지가 있다.

1. 모델 전체의 가중치를 조정하는 방법

   사전 학습 모델의 구조만 사용하면서 자신의 데이터셋에 맞게 모델을 전부 새롭게 학습시키는 방법이라고 생각할 수 있다.

2. 모델의 가중치를 일부 고정하고 출력층에 가까운 은닉층 몇 개와 출력층을 새로 학습하는 방법

   특징을 추출하는 부분보다는 추출된 특징을 바탕으로 주어진 작업을 수행하는 부분과 가까운 가중치들을 조정하는 것이다. 낮은 레벨의 계층(layer)은 일반적이고 독립적인 특징을 추출하고, 높은 레벨의 계층은 보다 구체적이고 명확한 특징을 추출하기 때문에 이러한 특성을 고려하여 재학습시킬 계층의 정도를 정한다.

3. 출력층에 해당하는 완전연결계층(fully connected layer)만 새로 학습하는 방법

컴퓨팅 연산 능력이 부족하거나 데이터셋이 너무 작을 때 흔히 사용하는 방법이다. 또한 수행해야 하는 작업이 사전 학습된 모델의 데이터셋과 크게 유사한 경우에도 사용할 수 있다.

LLM을 fine-tuning한다는 것은 보통 두 번째나 세 번째 유형의 fine-tuning을 수행한다.

Fine-tuning의 특징으로는 다음과 같은 것들을 생각할 수 있다.

1. 모델이 특정한 작업에 더 적합하도록 미세 조정되어 성능 향상을 기대할 수 있다.
2. 주어진 컴퓨팅 자원과 시간에 따라 추가 훈련을 하는 정도를 직접 조절할 수 있다.
3. 각 작업에 대해 별도의 모델이 생성(transfer learning)될 수 있기 때문에 다양한 작업에 걸쳐 모델을 관리하는 것에 용이하다.

프롬프트 튜닝

프롬프트 튜닝은 사전 학습 모델의 가중치를 변경하거나 조정하지 않고, 입력 프롬프트를 조작하여 원하는 출력을 유도하는 기법이다. 이때 입력 프롬프트는 설명, 문맥, 입력 데이터, 출력 지표 등으로 구성할 수 있다. 예를 들어 모델이 사람의 사고 과정과 비슷하게 추론을 하도록 만들고 싶다면 “Let’s think step-by-step.”과 같은 프롬프트를 사용할 수 있다.

프롬프트 튜닝과 아주 밀접한 관계를 가지는 프롬프트 엔지니어링은 현재 LLM의 동작은 대부분 인간과 컴퓨터가 상호작용을 할 때 사용되는 자연어에 의해 이루어진다는 것에서 연구되기 시작했다. 앞서 Auto-regressive한 특성의 Transformer 구조에 대한 장점을 언급했는데, 사실 이러한 구조는 단어의 순서가 조금 변화하는 것에도 응답의 품질이 달라질 수 있는 것의 원인이 된다.

프롬프트 엔지니어링을 더욱 자동화시킨 조정 방법이 프롬프트 튜닝이다. 프롬프트 튜닝을 이용하면 LLM의 내부 표현 방식을 fine-tuning하는 효과를 가져올 수 있다. 그렇기 때문에 LLM을 광범위하게 재학습하거나 수정하지 않고도 특정 작업이나 프롬프트에 대한 내용을 더욱 잘 수행하게 할 수 있다.

프롬프트 튜닝의 특징으로는 다음과 같은 것들을 생각할 수 있다.

1. 입력 방식의 변화만을 통해 결과를 조정하는 방식이기 때문에 모델 자체의 가중치는 변경되지 않는다.
2. 추가적인 훈련이 필요 없기 때문에 비용 면에서 효율적으로 빠르게 적용할 수 있다.
3. 효과적인 프롬프트를 설계하기 위해서는 그 분야의 지식이 어느 정도 필요할 수 있고 이에 대한 창의적인 접근이 필요하다.

2. LLaMA

1. LLaMA란 무엇이고 어떻게 설계되었는가

LLaMA 이전의 LLM은 파라미터 개수를 늘리면 무조건 좋을 것이라는 가정에 기반한다. 그러나 컴퓨팅 자원이 제한된 상황에서는 더 작은 모델을 더 많은 데이터로 학습시키는 것이 좋다는 것이 알려졌다. 하지만 선행 연구들은 추론 단계에서 소모되는 자원을 무시하였다. 그래서 더 많은 토큰을 훈련하여 목표 추론 자원 내에서 가장 좋은 성능을 내도록 하기 위해 설계된 모델이 LLaMA이다.

데이터셋의 대부분은 English CommonCrawl이다. 2017년부터 2020년까지의 데이터를 수집하였고 CCNet Pipeline을 통해 데이터를 정제한다. 이때 CCNet은 전처리, 중복 제거, 언어 식별, 언어 모델 필터링, 결과 재생성 순서대로 진행되는 방법론으로 품질이 비교적 낮은 웹 크롤링 데이터에서 고품질의 단일 언어 데이터를 추출하는 방법이다. 이외에 C4, Github, Wikipedia 등의 사전 학습 데이터가 존재한다.

Tokenizer는 BPE 알고리즘을 통해 토큰화하고 SentencePiece로 구현되어 있다. BPE는 OOV(Out of Vocabulary) 문제를 해결할 수 있는 방법이기 때문에 웹 크롤링 데이터의 비율이 높은 데이터셋에 적합하다고 볼 수 있다. 이때 구글이 만든 BPE Package인 SentencePiece를 사용한다.

2017년에 Transformer 구조가 공개된 이래로 많은 개선이 이루어졌는데 LLaMA는 그 중 몇 개를 이용하는 방식으로 설계되었다. 이외 구조적인 특징은 다음 절에서 설명할 것이다.

2. LLaMA의 핵심 기술(특징)은 무엇인가

Pre-normalization (GPT-3)

LLaMA는 훈련 시의 안정성 개선을 위해 Transformer의 각 sub-layer에서 출력값이 아닌 입력값을 정규화(normalize)한다. 이때 RMSNorm Normalizing 함수를 사용한다. RNN을 사용하여 가변 길이의 시퀀스(sequence)를 처리하는 데 있어서는 BatchNorm보다 LayerNorm이 우위에 있으나 이는 컴퓨팅 자원을 너무 많이 소모한다.

LayerNorm이 성공한 원인으로 추측되는 것 중 하나는 다시 센터링 연산을 해도 연산되는 값이 변하지 않는다는 것과 다시 스케일링 연산을 해도 연산되는 값이 변하지 않는다는 점이다. 이때 센터링 연산의 불변성이 필요 없다고 판단한 연구진들이 RMSNorm을 설계하게 되었다. 즉 RMS(Root Mean Square)만으로 정규화를 진행한다.

$$\bar{a}_i = \frac{a_i}{\mathrm{RMS}(\mathbf{a})} g_i \text{ where } \mathrm{RMS}(\mathbf{a}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} a_i^2}$$

SwiGLU activation function (PaLM)

대형 언어 모델 중 하나인 PaLM 이전에 사용되던 Swish 활성화 함수와 GLU 활성화 함수를 함께 사용하는 방식이다.

$$\text{Swish}_\beta(x) = x \cdot \sigma(\beta x)$$
$$\text{GLU}(x, W, V, b, c) = \sigma(xW + b) \otimes (xV + c)$$
$$\text{SwiGLU}(x, W, V, b, c) = \text{Swish}_\beta(xW + b) \otimes (xV + c)$$

SwiGLU는 ReLU보다 더욱 부드러운 곡선 모양이기 때문에 더 나은 최적화 효율과 더 빠른 수렴을 유도한다. 또한 단조함수가 아니기 때문에 입력값과 출력값 사이의 비선형 관계를 잘 파악할 수 있다. 그리고 게이팅 메커니즘(gating mechanism)을 사용하기 때문에 가중치를 선별적으로 활성화할 수 있다. 이렇게 Swish와 GLU의 강점을 모두 합친 SwiGLU 활성화 함수에 대하여 더 좋은 성능을 기대할 수 있다.

Rotary Embeddings (GPTNeo)

절대적인 위치 임베딩 대신 회전 위치 임베딩(Rotary Positional Embedding, RoPE)를 사용한다. 위치 임베딩은 Transformer 모델로 하여금 단어들의 위치 정보를 알 수 있게 한다. 그러나 Transformer 모델의 특성 상 이러한 위치 정보를 정확하게 학습하는 것은 어렵다. 이때 RoPE는 시퀀스 내 단어 위치 정보를 각 단어의 상대적 회전 각도로 나타낸다.

Casual multi-head attention

xformers 라이브러리에서 사용 가능한 casual multi-head attention을 통해 어텐션의 가중치를 저장하지 않고 mask된 key와 query의 점수를 계산하지 않을 수 있다. 따라서 메모리 사용량과 실행 시간을 효과적으로 감소할 수 있다.

Manual back-propagation

PyTorch의 autograd에 의존하는 대신 transformer 계층의 역전파 함수를 수동으로 구현함으로써 역전파 중 재계산되는 활성화 함수의 양을 줄인다. 예를 들어 linear layer의 출력과 같이 계산 비용이 높은 활성화 함수를 저장하는 것이다.

3. LLaMA2는 LLaMA와 어떤 차이를 갖고있는가

1. LLaMA2는 LLaMA에 비하여 40% 더 많은 데이터로 학습되었다. 또한 두 배의 문맥 길이를 가진다.
2. LLaMA2는 인간 피드백을 통한 강화학습(RLHF)이라는 기술을 사용하여 모델을 fine-tuning하였다. 이 과정은 사람이 선호하는 정도에 따라 출력을 생성하고, 모델이 생성한 출력 중 사람이 선호하는 패턴을 생성한다.
3. 학술 전용 오픈소스인 LLaMA와 달리 LLaMA2는 상용 오픈소스 모델이다.
4. LLaMA2 34B와 LLaMA2 70B는 추론 과정의 확장성을 개선하기 위하여 Grouped-Query Attention(GQA) 메커니즘을 사용하였다.