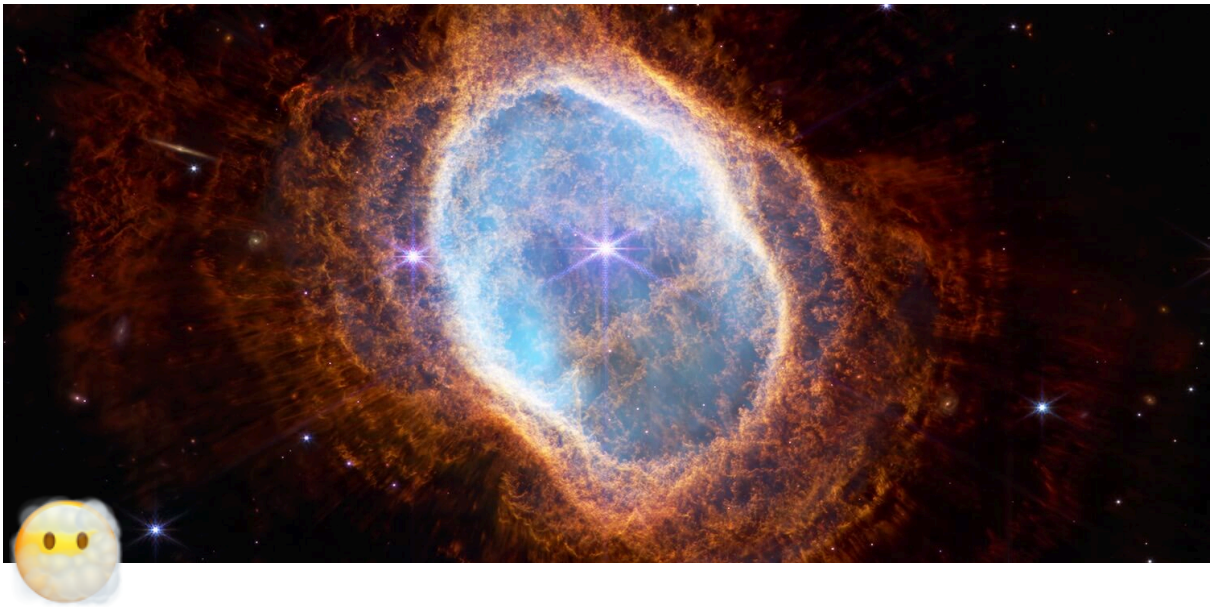

DeepSeek LLM: Scaling Open-Source Language Models with Longtermism

DeepSeek LLM 프로젝트 소개

사전 학습 데이터

DeepSeek LLM은 주로 중국어와 영어로 구성된 **2조(trillion) 개 토큰 규모의 데이터셋을 수집**하여 사전 학습에 활용하였으며, 해당 데이터셋은 지속적으로 확장되고 있습니다.

1. **아키텍처**

   LLaMA의 아키텍처를 기반으로 설계되었으며, 기존의 코사인 학습률 스케줄러(cosine learning rate scheduler) 대신 다단계 학습률 스케줄러(multi-step learning rate scheduler)를 사용함으로써 성능을 유지하면서도 지속적 학습(continual training)을 보다 용이하게 만들었습니다.

2. **미세 조정(Fine-tuning)**

   다양한 출처로부터 100만 개 이상의 인스턴스를 수집하여 SFT(Supervised Fine-Tuning)를 수행하였으며, 다양한 SFT 전략과 데이터 제거(data ablation) 기법에 대한 경험을 공유합니다.

3. **정렬(Alignment)**

직접 선호 최적화(Direct Preference Optimization, DPO) 기법(“Direct Preference Optimization: Your Language Model Is Secretly a Reward Model”)을 활용하여 모델의 대화 성능을 향상시켰습니다.

Pretraining

DATA

기존 연구(예: RedPajama, The Pile, RefinedWeb, LLaMA)에서 얻은 통찰을 통해, 데이터 처리 과정에서 **중복 제거, 필터링, 재혼합**이 핵심적으로 중요하다는 점을 확인하였습니다.

중복 제거 · 필터링 · 재혼합

중복 제거 및 재혼합 단계는 고유한 인스턴스를 샘플링함으로써 데이터 표현의 다양성을 보장합니다.

필터링 단계는 데이터의 정보 밀도를 향상시켜, 보다 효율적이고 효과적인 모델 학습을 가능하게 합니다.

1. 중복 제거

중복 제거 범위를 확장하는 공격적인 중복 제거 전략을 채택하였습니다.

분석 결과, 전체 Common Crawl 코퍼스를 대상으로 중복 제거를 수행하는 것이 단일 덤프 내에서 중복 제거를 수행하는 것보다 더 많은 중복 인스턴스를 제거하는 것으로 나타났습니다.

즉, 전체 데이터를 하나의 코퍼스로 통합한 후 중복 제거를 수행합니다.

2. 필터링

언어적(linguistic) 분석과 의미론적(semantic) 분석을 통합하여 데이터 품질을 개별적 관점과 전반적 관점에서 평가합니다.

즉, 문장의 형태와 구조를 평가하는 언어적 분석과 의미 있는 정보를 포함하는지를 판단하는 의미론적 분석을 결합하여,

개별 문장이 자연스러운지 여부와 문서 전체가 일관된 의미를 가지는지를 함께 평가합니다.

3. 재혼합

재혼합 과정에서는 접근 방식을 조정하여, 상대적으로 과소 표현된 도메인의 비중을 확대하는 데 중점을 둡니다.

토크나이저

- **알고리즘**

  Hugging Face 팀(2019)이 개발한 tokenizers 라이브러리를 기반으로,

  Byte-level Byte-Pair Encoding(BBPE) 알고리즘을 구현하였습니다.

- **프리-토크나이제이션(Pre-tokenization)**

  GPT-2(Radford et al., 2019)와 유사하게, 줄바꿈, 구두점, 중국어·일본어·한국어(CJK) 기호 등 서로 다른 문자 범주에서 온 토큰이 병합되는 것을 방지하기 위해 사전 토큰화를 적용하였습니다.

- **숫자 처리**

  「LLaMA: Open and Efficient Foundation Language Models」

  https://arxiv.org/pdf/2302.13971

  에서 사용된 접근 방식을 따라, 숫자를 개별 숫자로 분할하는 방식을 채택하였습니다.

  예: 123 → 1, 2, 3

- **어휘 크기**

  초기 어휘는 약 100,000개의 일반 토큰으로 구성되었으며, 여기에 15개의 특수 토큰을 추가하여 총 100,015개의 어휘를 사용하였습니다.

  학습 중 계산 효율성을 보장하고 향후 추가될 수 있는 특수 토큰을 위한 여유 공간을 확보하기 위해, 최종 모델의 어휘 크기는 102,400으로 설정되었습니다.

아키텍처

- DeepSeek LLM은 LLaMA의 설계를 따릅니다.

  「LLaMA: Open and Efficient Foundation Language Model」

  https://arxiv.org/pdf/2302.13971

- RMSNorm(Zhang and Sennrich, 2019)을 사용하는 Pre-Norm 구조를 채택하였습니다.
- Feed-Forward Network(FFN)의 활성화 함수로 SwiGLU(Shazeer, 2020)를 사용하며, 중간 계층의 차원은 8 × d_model로 설정되었습니다.
- 위치 인코딩 방식으로 Rotary Embedding(Su et al., 2024)https://arxiv.org/abs/2104.09864을 통합하였습니다.

- 추론 비용 최적화를 위해 67B 모델에서는 전통적인 Multi-Head Attention(MHA) 대신 Grouped-Query Attention(GQA)(Ainslie et al., 2023)https://arxiv.org/abs/2305.13245를 사용합니다.

  대부분의 GQA 기반 연구와 달리, FFN 계층의 중간 너비를 확장하는 대신 네트워크 깊이를 확장하여 67B 모델의 파라미터를 구성함으로써 더 나은 성능을 목표로 하였습니다.

- DeepSeek LLM 7B는 30개 계층으로 구성되며, DeepSeek LLM 67B는 95개 계층을 가집니다.

  이러한 계층 설계는 다른 오픈 소스 모델과의 파라미터 일관성을 유지하면서도 모델 파이프라인 분할을 용이하게 하여 학습 및 추론 효율을 높입니다.

하이퍼파라미터

Params	n_{layers}	d_{model}	n_{heads}	$n_{\text{kv\_heads}}$	Context Length	Sequence Batch Size	Learning Rate	Tokens
7B	30	4096	32	32	4096	2304	4.2e-4	2.0T
67B	95	8192	64	8	4096	4608	3.2e-4	2.0T

- DeepSeek LLM은 표준 편차 0.006을 사용하여 초기화되었으며, AdamW 옵티마이저를 사용합니다.

$$\beta_1 = 0.9, \beta_2 = 0.95, weight\_decay = 0.1$$

- 코사인 스케줄러 대신 다단계 학습률 스케줄러를 사용하였으며, 해당 선택의 일관성이 실험적으로 검증되었습니다.

  2,000 스텝의 워밍업 이후 학습률은 최대값에 도달하고, 전체 토큰의 80%를 처리한 이후에는 최대값의 31.6%로 감소합니다.

  이후 토큰 90% 이후 구간에서는 학습률을 최대값의 10%로 추가 감소시킵니다.

  학습 과정에서 그래디언트 클리핑은 1.0으로 설정됩니다.

  지속적 훈련에서의 재사용 비율과 모델 성능 간 균형을 고려하여, 세 단계의 비율을 각각 80%, 10%, 10%으로 설정하였습니다.

  배치 크기와 학습률은 모델 크기에 따라 달라집니다.

- 다단계 학습률 스케줄러를 선택한 이유는, 모델 크기를 고정한 상태에서 훈련 규모를 조정할 경우 첫 번째 단계에서의 훈련 결과를 재사용할 수 있어 지속적 훈련에 유리하기 때문입니다.

인프라스트럭처

- 대규모 언어 모델의 훈련과 평가를 위해, 효율적이고 경량화된 훈련 프레임워크인 HAI-LLM(High-flyer, 2023)을 사용합니다.
- 데이터 병렬화, 텐서 병렬화,
  시퀀스 병렬화, 1F1B 파이프라인 병렬화가
  Megatron(Korthikanti et al., 2023; Narayanan et al., 2021; Shoeybi et al., 2019)과 유사한 방식으로 통합되어 있습니다.
- 하드웨어 활용도를 극대화하기 위해 Flash Attention(Dao, 2023; Dao et al., 2022)을 적용하였습니다.
- ZeRO-1(Rajbhandari et al., 2020)은 옵티마이저 상태를 데이터 병렬 랭크에 분할하는 데 사용되며, 계산과 통신을 오버랩하여 대기 오버헤드를 최소화합니다.

  여기에는 ZeRO-1의 마지막 마이크로 배치 역전파 및 reduce-scatter 연산, 그리고 시퀀스 병렬화에서의 GEMM 계산과 all-gather/reduce-scatter가 포함됩니다.
- LayerNorm, GEMM, Adam 업데이트 등 일부 연산은 훈련 속도 향상을 위해 융합(fused)되었습니다.
- 모델 훈련 안정성을 위해 bf16 정밀도로 훈련을 수행하되, 그래디언트는 fp32 정밀도로 누적합니다.
- GPU 메모리 사용량을 줄이기 위해 인플레이스(in-place) 교차 엔트로피를 적용합니다. 즉, HBM에서 미리 변환하는 대신 교차 엔트로피 CUDA 커널에서 bf16 로짓을 실시간으로 fp32로 변환하고, 해당 bf16 그래디언트를 계산한 후 로짓을 해당 그래디언트로 덮어씁니다.
- 모델 가중치와 옵티마이저 상태는 5분마다 비동기적으로 저장되며, 이로 인해 하드웨어 또는 네트워크 장애 발생 시에도 최대 5분 이내의 훈련 손실만 발생합니다. 이러한 임시 체크포인트는 저장 공간 낭비를 방지하기 위해 주기적으로 삭제됩니다.
- 컴퓨팅 클러스터 부하의 동적 변화에 대응하기 위해, 다른 3D 병렬 구성으로부터 훈련을 재개하는 기능을 지원합니다.
- 평가 단계에서는 생성 작업에 vLLM(Kwon et al., 2023)을 사용하고, 비생성 작업에는 연속 배치를 적용하여 수동 배치 크기 조정을 피하고 토큰 패딩을 최소화합니다.

스케일링 법칙

스케일링 법칙이란?

스케일링 법칙은 컴퓨팅 예산(C), 모델 크기(N), 데이터 크기(D)가 증가함에 따라 모델 성능이 예측 가능하게 향상된다는 원리를 의미합니다.

이전 연구(Hestness et al., 2017; Kaplan et al., 2020; Hoffmann et al., 2022)에서는 컴퓨팅 예산 C가 모델 파라미터 수 N과 토큰 수 D의 곱으로 근사될 수 있으며,

C=6ND로 표현된다고 제시하였습니다.

$6N_1 = 72 \cdot n_{layer} \cdot d_{model}^2$

$6N_1 = 72 \cdot n_{layer} \cdot d_{model}^2 + 6 \cdot n_{vocab} \cdot d_{model}$

다만 이러한 근사는 어텐션 연산의 계산 오버헤드를 충분히 고려하지 않거나, 어휘 계산을 포함하는 방식으로 인해 정확성에 한계가 있었습니다.

LLM 개발에서의 중요성

스케일링 법칙은 제한된 컴퓨팅 예산을 어떻게 효율적으로 배분하여 모델을 스케일업할 것인지에 대한 핵심적인 기준을 제공합니다.

DeepSeek LLM의 스케일링 법칙 연구

- **하이퍼파라미터 스케일링 법칙**

  DeepSeek LLM 팀은 최적의 성능을 확보하기 위해 배치 크기 B와 학습률 η가 컴퓨팅 예산 C에 따라 어떻게 변화하는지를 분석하였습니다.

  실험 결과, 다음과 같은 파워 법칙 관계를 확인하였습니다.

  - 최적 학습률: $\eta_{\text{opt}} = 0.3118 \cdot C^{-0.1250}$
  - 최적 배치 크기:$B_{\text{opt}} = 0.2920 \cdot C^{0.3271}$

  이는 컴퓨팅 예산이 증가함에 따라 최적 배치 크기는 점진적으로 증가하고, 최적 학습률은 점진적으로 감소함을 의미합니다.

- **모델 및 데이터 스케일링 법칙**

  모델 스케일을 나타내기 위해 기존의 모델 파라미터(N) 대신 **비-임베딩 FLOPs/토큰 (non-embedding FLOPs/token, M)**을 도입했습니다.
  M은 어텐션 연산의 계산 오버헤드를 포함하지만, 어휘 계산은 제외하여 계산 비용을 더 정확하게 반영합니다.

  - 이 경우 컴퓨팅 예산(C)은 $C = M \cdot D$로 표현됩니다.

    $M = 72 \cdot n_{layer} \cdot d_{model}^2 + 12 \cdot n_{layer} \cdot d_{model} \cdot l_{seq}$

- Chinchilla 논문에서 제안된 IsoFLOP 프로파일 접근 방식(Hoffmann et al., 2022)을 사용하여 실험 비용과 피팅(fitting)의 어려움을 줄였습니다.
- 연구 결과, 최적 모델 스케일(M_{opt})과 데이터 스케일(D_{opt})은 다음과 같은 관계를 따릅니다:

$$M_{\text{opt}} = M_{\text{base}} \cdot C^a, \quad M_{\text{base}} = 0.1715,\ a = 0.5243$$

$$D_{\text{opt}} = D_{\text{base}} \cdot C^b, \quad D_{\text{base}} = 5.8316,\ b = 0.4757$$

이는 컴퓨팅 예산이 주어졌을 때 모델 크기와 데이터 크기를 어떻게 배분해야 가장 효율적인 성능을 얻을 수 있는지에 대한 가이드라인을 제공합니다.

- **데이터 품질의 영향**

  데이터 품질이 스케일링 법칙, 특히 최적 모델·데이터 할당 전략에 큰 영향을 미친다는 점을 확인하였습니다.

  데이터 품질이 높을수록 증가한 컴퓨팅 예산을 모델 스케일링에 더 많이 할당하는 것이 바람직하다는 결론을 제시합니다.

  즉, 고품질 데이터는 동일한 데이터 규모에서도 더 큰 모델을 학습시키는 데 유리합니다.

  이는 기존 연구들에서 서로 다른 최적 할당 전략이 도출된 이유를 설명할 수 있습니다.