LLaMA: Open and Efficient Foundation Language Models

Hugo Touvron[*], Thibaut Lavril[*], Gautier Izacard[*], Xavier Martinet
Marie-Anne Lachaux, Timothee Lacroix, Baptiste Rozière, Naman Goyal
Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin
Edouard Grave[*], Guillaume Lample[*]

Meta AI

- Problem / objective
  - Propose Large Language Model, called **LLaMA**
- Contribution / Key idea
  - Propose Large Language Model, called **LLaMA**

Chinchilla 논문에서 말하기를,

- 기존의 LLM 들은 under-trained 되었다!
  왜냐하면, 학습 데이터 양은 그대로 두고, 모델 크기만을 증가시키며 실험해왔음.

- ***LLM 을 최적으로 학습하려면, 주어진 예산 안에서, '모델 크기' 및 '학습 데이터 양' 이 동등하게 scaling 되어야 한다.***
  '적은 양의 데이터로 학습된 큰 모델' 보다, ***'많은 양의 데이터로 학습된 작은 모델'*** *의 성능이 더 좋다.*

-> 이를 LLaMA 에서도 동일하게 적용.

Hoffmann, Jordan, et al. "Training compute-optimal large language models." *arXiv preprint arXiv:2203.15556* (2022).

Pre-training data

- 공개적으로 사용 가능하고, open-source 인 데이터셋만 학습에 사용.
- SentencePiece 프레임워크를 사용해서 BPE 기반 토크나이저 사용.
- 토큰화 결과, 학습 데이터셋에 총 1.4T 개수의 토큰 포함.
- 학습중 대부분 토큰들은 한번만 비춰짐, 그러나 Wikipedia 와 Books 의 경우 대략 2 에포크 학습.

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Table 1: **Pre-training data.** Data mixtures used for pre-training, for each subset we list the sampling proportion, number of epochs performed on the subset when training on 1.4T tokens, and disk size. The pre-training runs on 1T tokens have the same sampling proportion.

Touvron, Hugo, et al. "Llama: Open and efficient foundation language models." *arXiv preprint arXiv:2302.13971* (2023).

ArXiv 2023

Architecture

- 기본 트랜스포머 구조에 아래 3가지 사항 추가.
  1. Pre-normalization
     - GPT3 에서 제안.
  2. SwiGLU activation function
     - PaLM 에서 제안.
  3. Rotary Embeddings
     - GPTNeo 에서 제안.

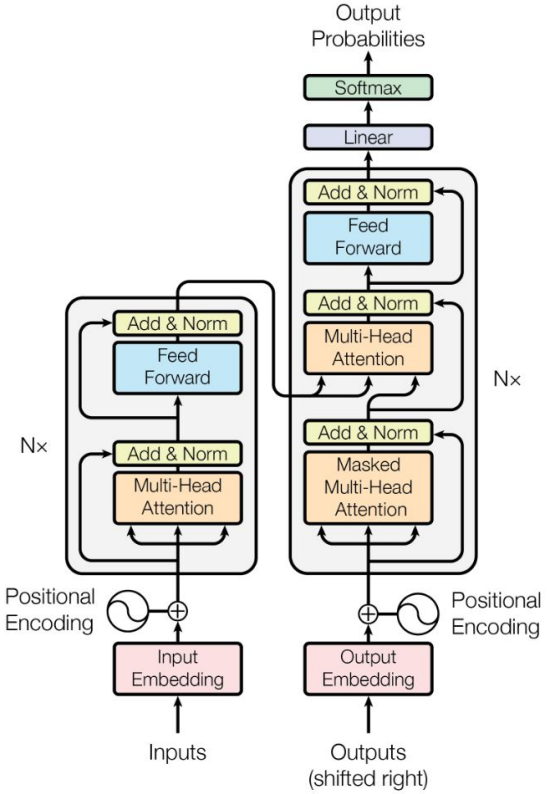

Figure 1: The Transformer - model architecture.

Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

전유진

Architecture

- **Pre-normalization [GPT3]**

  - 각 sublayer 의 인풋을 정규화 (원래는 아웃풋을 정규화했음.)

  - RMS normalization 사용

  - 효과 : 학습 안정성 향상.

$$\mathbf{y} = \gamma \cdot \frac{\mathbf{x}}{\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2 + \epsilon}}$$

Architecture

- **SwiGLU activation function [PaLM]**

  - SwiGLU = Swish + GLU activation 사용. (원래는 ReLU 사용했었음.)
  - Swish = sigmoid를 게이트로 사용하는 자기-게이트형 활성화 함수
  - GLU = 입력을 둘로 분리해서 하나는 정보, 하나는 게이트로 사용

$$Swish(x) = x\sigma(\beta x)$$

게이트 역할.

σ: Sigmoid Function $\sigma(x) = \frac{1}{1+e^{-x}}$

$$GLU(x, W, V, b, c) = \sigma(xW + b) \otimes (xV + c)$$

$$SwiGLU(x, W, V, b, c, \beta) = Swish_\beta(xW + b) \otimes (xV + c)$$

Architecture

- **Rotary Embeddings [GPTNeo]**

  - Rotary Positional Embeddings (RoPE) 사용. (원래는 absolute positional embeddings 사용했었음.)

  - 각 위치마다 임베딩 벡터에 회전을 적용하여, 절대 위치 정보를 유지하면서도 Attention score가 상대 위치에만 의존하도록

    만드는 위치 인코딩 기법