Large Language Models

Generative Pre-trained Transformer

GPT

Developed by OpenAI (participated by Microsoft)

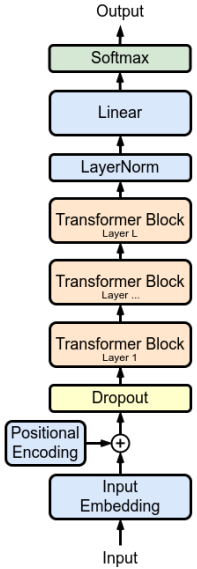
ChatGPT is the last of this family

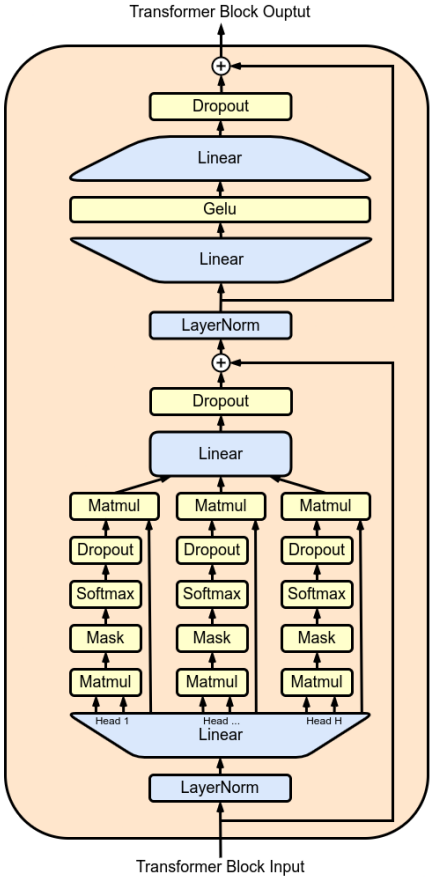
- Pre-trained with Language Modeling then fine-tuned with supervision

GPT

The original GPT model is a stack of 12 *transformer decoder* blocks

- Variant of the transformer without the encoder part
- As in BERT, the FFN at the end of each block uses GeLU activation





Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

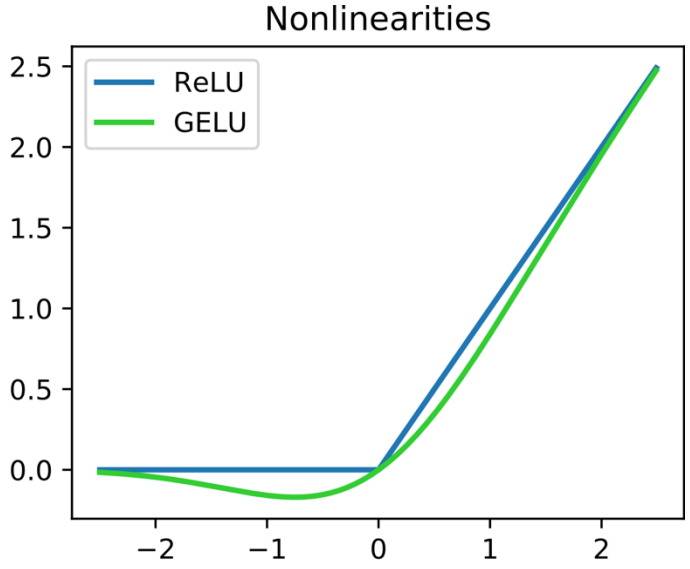
GeLU activation function

Gaussian Error Linear Unit is a variant of ReLU where the standard Gaussian cumulative distribution is used to modulate the linear activation

$$\mathrm{GELU}(x) = xP(X \leq x)$$

$$X \sim \mathcal{N}(0,1)$$

$$\mathrm{GELU}(x) \sim x\sigma(1.702x)$$



https://bit.ly/3sNBObx

GPT unsupervised pre-training

Given $\mathcal{U} = \{u_1, \ldots, u_n\}$ a set of tokens:

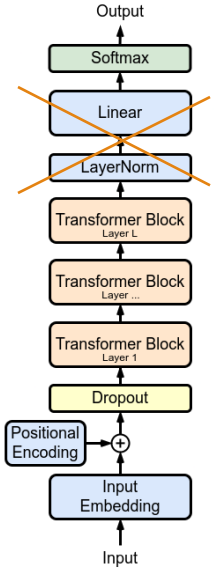
Token embedding

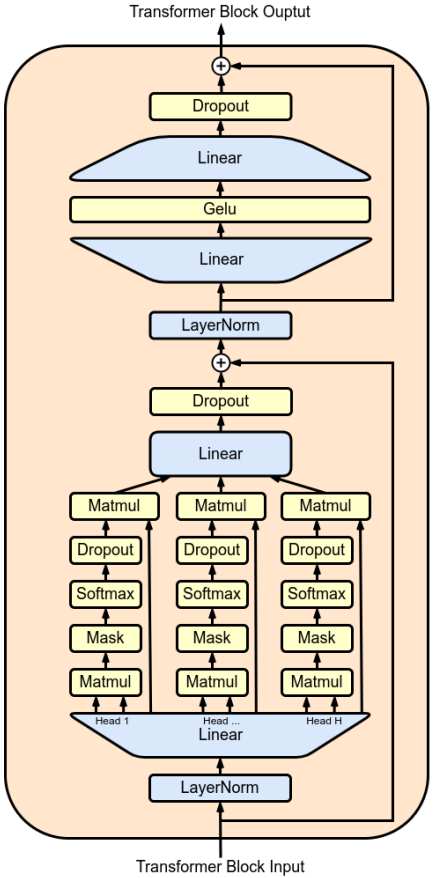
position embedding

$$h_0 = UW_e + W_p$$

$$h_l = \texttt{transformer\_block}(h_{l-1}) \forall i \in [1, n]$$

$$P(u) = \texttt{softmax}(h_n W_e^T)$$



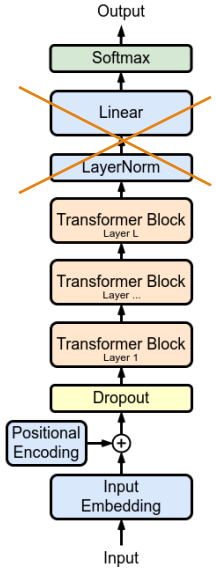


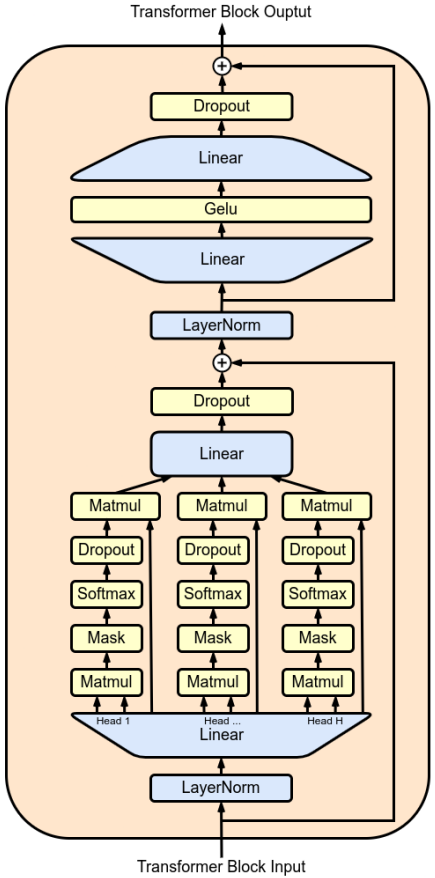
where $U = (u_{-k}, \ldots, u_{-1})$ is the context

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

GPT unsupervised pre-training

$$h_0 = UW_e + W_p$$
$$h_l = \texttt{transformer\_block}(h_{l-1}) \forall i \in [1, n]$$
$$P(u) = \texttt{softmax}(h_n W_e^T)$$





Language Modeling objective function:

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

GPT supervised fine-tuning

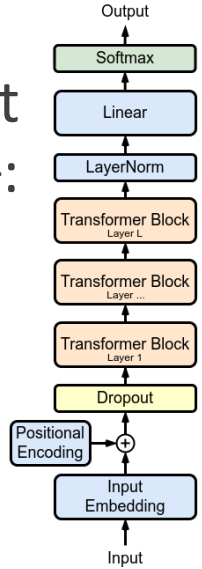
A linear layer is added on top to predict a label *y* from a set of tokens $\{x_1, \ldots, x_n\}$:

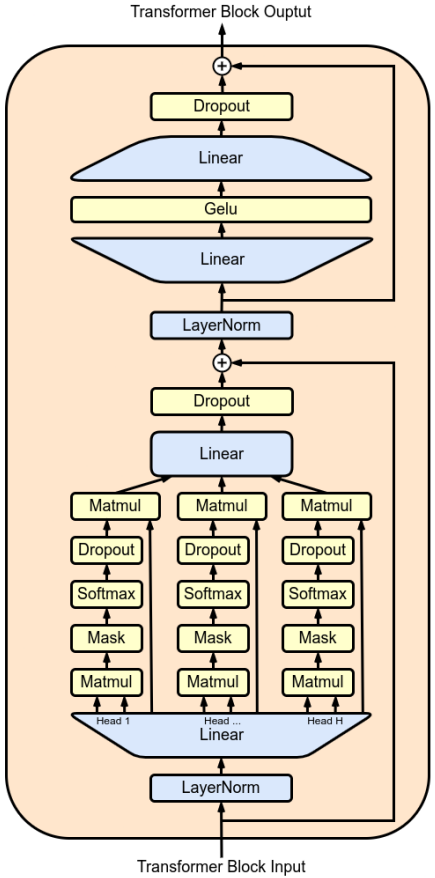
$$P(y|x^1, \ldots, x^m) = \texttt{softmax}(h_l^m W_y).$$

with a composite objective function:

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$

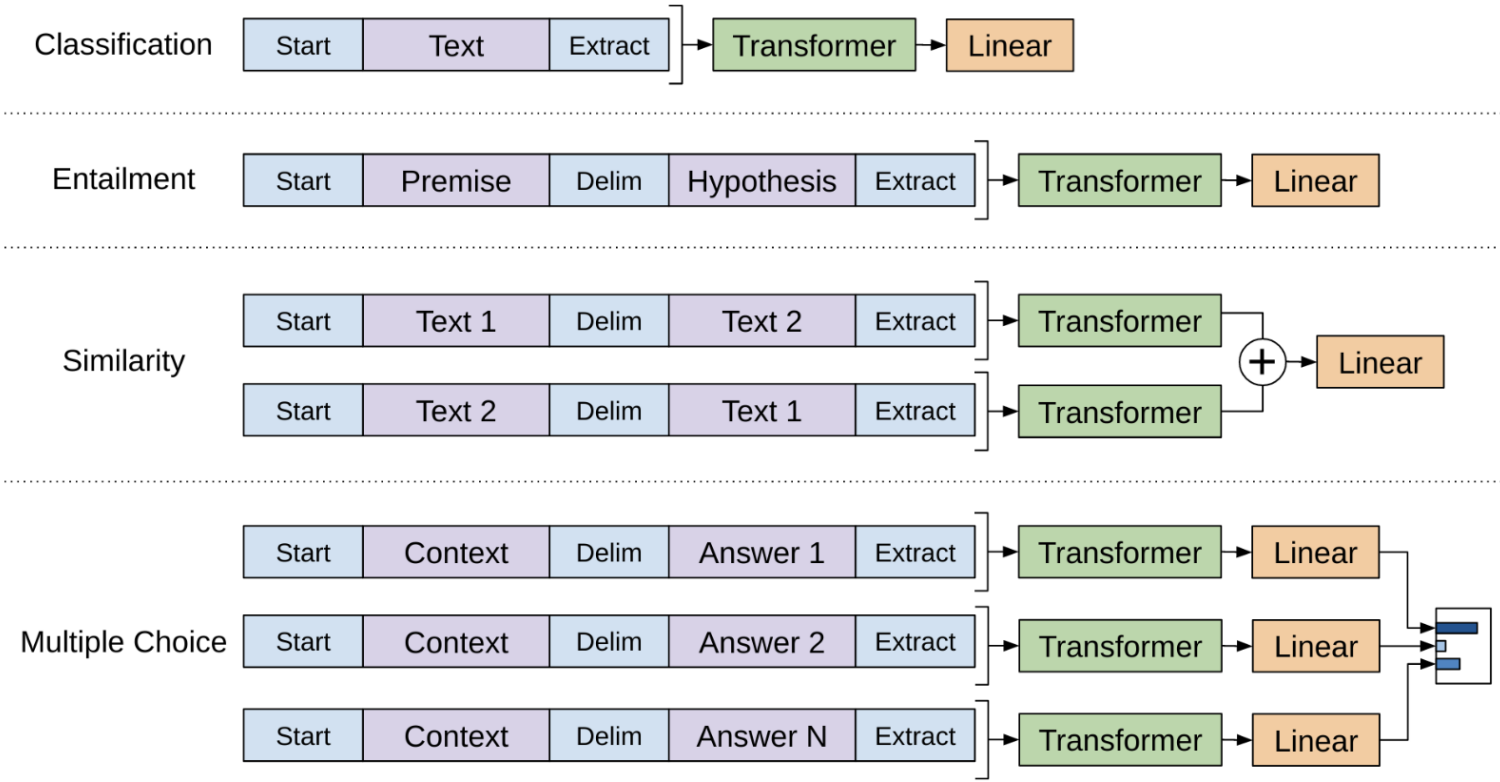
$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m)$$





Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

GPT fine-tuning tasks



Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018).
Improving language understanding by generative pre-training.

GPT families

Model	Architecture	Parameter count	Training data	Release date	Training cost
GPT-1	12-level, 12-headed Transformer decoder (no encoder), followed by linear-softmax.	117 million	BookCorpus:[27] 4.5 GB of text, from 7000 unpublished books of various genres.	June 11, 2018[8]	30 days on 8 P600 GPUs, or 1 petaFLOP/s-day.[8]
GPT-2	GPT-1, but with modified normalization	1.5 billion	WebText: 40 GB of text, 8 million documents, from 45 million webpages upvoted on Reddit.	February 14, 2019 (initial/limited version) and November 5, 2019 (full version)[28]	"tens of petaflop/s-day",[29] or 1.5e21 FLOP.[30]
GPT-3	GPT-2, but with modification to allow larger scaling	175 billion[31]	499 billion tokens consisting of CommonCrawl (570 GB), WebText, English Wikipedia, and two books corpora (Books1 and Books2).	May 28, 2020[29]	3640 petaflop/s-day (Table D.1 [29]), or 3.1e23 FLOP.[30]
GPT-3.5	Undisclosed	175 billion[31]	Undisclosed	March 15, 2022	Undisclosed
GPT-4	Also trained with both text prediction and RLHF; accepts both text and images as input. Further details are not public.[26]	Undisclosed. Estimated 1.7 trillion[32]	Undisclosed	March 14, 2023	Undisclosed. Estimated 2.1e25 FLOP.[30]

Wikipedia

ChatGPT

Based on GPT-3.5 and GPT-4

- Fine-tuned to target conversational usage
- Uses the so called *Reinforcement Learning With Human Feedback* (RLHF)

RLHF

Makes use of human trainers to improve model performance

- Human trainers rank the response provided by the model in a previous conversation
- Ranks are then used to create a reward model used in the iterations of the *Proximal Policy Optimization* (PPO) reinforcement learning algorithm

LLaMA

Large Language Model Meta AI was released by Meta in February 2023

- 7, 16, 33, and 65B parameters versions
- 13B parameters was reported to outperform GPT-3
- LLaMA-2 released in July 2023 with 7, 13, and 70B versions

https://bit.ly/3Rf4K5C

LLaMA-2

Based on transformer decoder stack

Minor architectural differences with GPT-3:

- SwiGLU activation function instead of ReLU
- Rotary positional embeddings
- Root-mean-squared layer-normalization instead of standard layer normalization
- Increases context length to 4K tokens

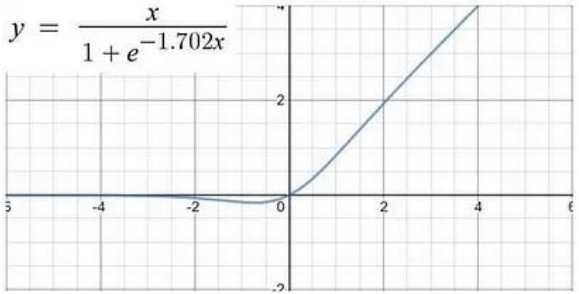
https://bit.ly/3Rf4K5C

SwiGLU activation function

Swish Gated Linear Unit has an activation function that is the combination of Swish and Gated LUs

Swish LU are a generalization of the GELU approximation

$$\text{Swish}(x) = x \cdot \sigma(\beta x)$$



https://bit.ly/40T9npg

SwiGLU activation function

Gated LUs embed a linear activation inside the sigmoid function

$$\mathrm{GLU}(x) = \sigma(Wx + b)$$

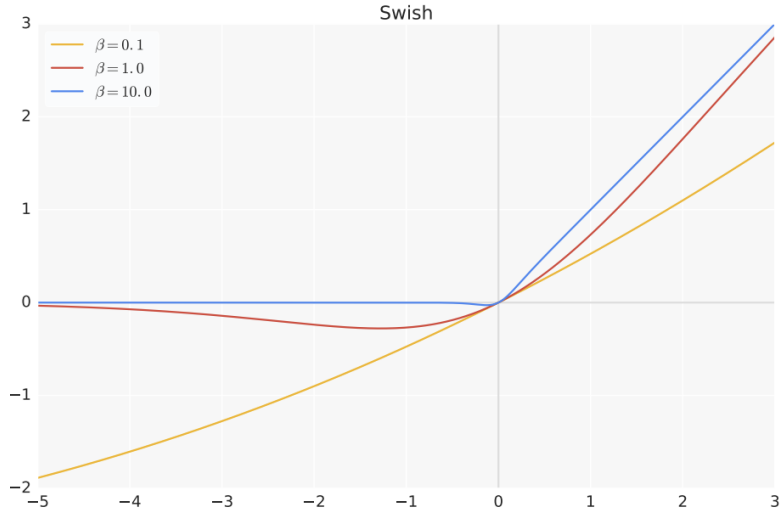
Gating mechanism: the neuron is activated based on the input it receives

https://bit.ly/40T9npg

SwiGLU activation function

SwiGLU embeds the previous activations

$$\mathrm{SwiGLU}(x) = x \cdot \sigma(\beta x) + (1 - \sigma(\beta x)) \cdot \sigma(W \cdot x + b)$$



https://bit.ly/40T9npg

Rotary Positional Embeddings (RoPE)

A particular type of *relative position embeddings*

- We can represent **q**, **k**, and **v** self-attention vectors in terms of their embeddings as

$$\boldsymbol{q}_m = f_q(\boldsymbol{x}_m, m)$$
$$\boldsymbol{k}_n = f_k(\boldsymbol{x}_n, n)$$
$$\boldsymbol{v}_n = f_v(\boldsymbol{x}_n, n),$$

- The element (m, n) of the attention matrix $\mathbf{q}_m\mathbf{k}_n$ is formulated in terms of relative position $m - m$ as a function $g(\boldsymbol{x}_m, \boldsymbol{x}_n, m - n)$

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., & Liu, Y. (2021). Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Rotary Positional Embeddings (RoPE)

A particular type of *relative position embeddings*

- RoPE expresses *g* as

$$f_q(\boldsymbol{x}_m, m) = (\boldsymbol{W}_q \boldsymbol{x}_m) e^{im\theta}$$
$$f_k(\boldsymbol{x}_n, n) = (\boldsymbol{W}_k \boldsymbol{x}_n) e^{in\theta}$$
$$g(\boldsymbol{x}_m, \boldsymbol{x}_n, m-n) = \mathrm{Re}[(\boldsymbol{W}_q \boldsymbol{x}_m)(\boldsymbol{W}_k \boldsymbol{x}_n)^* e^{i(m-n)\theta}]$$

- So the **q** and **k** vectors, and their inner product are

$$f_{\{q,k\}}(\boldsymbol{x}_m, m) = \boldsymbol{R}^d_{\Theta,m} \boldsymbol{W}_{\{q,k\}} \boldsymbol{x}_m$$
$$\boldsymbol{q}_m^\intercal \boldsymbol{k}_n = (\boldsymbol{R}^d_{\Theta,m} \boldsymbol{W}_q \boldsymbol{x}_m)^\intercal (\boldsymbol{R}^d_{\Theta,n} \boldsymbol{W}_k \boldsymbol{x}_n) = \boldsymbol{x}_m^\intercal \boldsymbol{W}_q R^d_{\Theta,n-m} \boldsymbol{W}_k \boldsymbol{x}_n$$
$$\Theta = \{\theta_i = 10000^{-2(i-1)/d}, i \in [1, 2, ..., d/2]\}$$

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., & Liu, Y. (2021). Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

LLaMA-3

Name	Release date	Parameters	Training cost (petaFLOP-day)	Context length (tokens)	Corpus size (tokens)	Commercial viability?
LLaMA	February 24, 2023	• 6.7B • 13B • 32.5B • 65.2B	6,300[33]	2048	1–1.4T	No
Llama 2	July 18, 2023	• 6.7B • 13B • 69B	21,000[34]	4096	2T	Yes
Code Llama	August 24, 2023	• 6.7B • 13B • 33.7B • 69B				
Llama 3	April 18, 2024	• 8B • 70.6B	100,000[35][36]	8192	15T	
Llama 3.1	July 23, 2024	• 8B • 70.6B • 405B	440,000[32][37]	128,000		
Llama 3.2	September 25, 2024	• 1B • 3B • 11B • 90B[38][39]		128,000[40]		

https://bit.ly/3Rf4K5C

LLaMA-3

	8B	70B	405B
Layers	32	80	126
Model Dimension	4,096	8,192	16,384
FFN Dimension	14,336	28,672	53,248
Attention Heads	32	64	128
Key/Value Heads	8	8	8
Peak Learning Rate	3×10^{-4}	1.5×10^{-4}	0.8×10^{-4}
Activation Function	SwiGLU		
Vocabulary Size	128,000		
Positional Embeddings	$\mathrm{RoPE}(\theta = 500,000)$		

https://bit.ly/3Rf4K5C

LLaMA training

LLaMA-1 trained with 1.4 trillion tokens

- Webpages scraped by CommonCrawl
- Open source repositories of source code from GitHub
- Wikipedia in 20 different languages
- Public domain books from Project Gutenberg
- The LaTeX source code for scientific papers uploaded to ArXiv
- Questions and answers from Stack Exchange websites

https://bit.ly/3Rf4K5C

LLaMA training

LLaMA-2 trained with 2 trillion tokens

LLaMA-2 chat was fine-tuned on 27,540 prompt-response pairs created for this project

RLHF was used with rejection sampling and PPO

https://bit.ly/3Rf4K5C

LLaMA training

LLaMA-3 trained using mainly English data, with over 5% in over 30 other languages.

Its dataset was filtered by a text-quality classifier, and the classifier was trained by text synthesized by Llama 2

https://bit.ly/3Rf4K5C

Gemini

Gemini is a family of multimodal large language models developed by Google DeepMind, serving as the successor to LaMDA and PaLM 2

They are decoder-only transformers, with modifications to allow efficient training and inference on TPUs. They have a context length of 32,768 tokens, with *multi-query attention*.

https://bit.ly/3Z1PkVk

Gemini

Gemini 1.0 versions:

- Ultra, designed for "highly complex tasks";
- Pro, designed for "a wide range of tasks";
- Nano, designed for "on-device tasks".

Gemini 1.5 Pro is a multimodal MoE, with a context length in the millions of tokens

https://bit.ly/3Z1PkVk

Gemini

Gemma (Gemini) 2 27B is trained on web documents, code, science articles.

Gemma 2 9B was *distilled* from 27B.

Gemma 2 2B was *distilled* from a 7B model that remained unreleased.

Mistral

Mistral AI was co-founded in April 2023 by people coming from Google DeepMind and Meta Platforms

It releases both open-weights and open-AI models

https://bit.ly/3Z1Vsgi

Mistral

Open-weights models

Mistral 7B: a 7.3B parameter language model using the transformers architecture and Grouped Query Attention (GQA)

Mixtral 8x7B sparse mixture of experts architecture. Eight distinct groups of "experts", giving the model a total of 46.7B usable parameters

Mixtral 8x22B uses an architecture similar to that of Mistral 8x7B, but with each expert having 22 billion parameters instead of 7

https://bit.ly/3Z1Vsgi

Claude

Claude is a family of large language models developed by Anthropic. The first model was released in March 2023.

Claude models are generative pre-trained transformers that have been also fine-tuned, notably using constitutional AI and RLHF

https://bit.ly/3CAzrxt

Claude

Constitutional AI

Training AI systems, particularly LLMs, to be harmless and helpful without relying on extensive human feedback

The method involves two phases: supervised learning and reinforcement learning

https://bit.ly/3CAzrxt

Claude

Constitutional AI

In the supervised learning phase, the model generates responses to prompts, self-critiques these responses based on a set of guiding principles (a "constitution"), and revises the responses.

Then the model is fine-tuned on these revised responses

https://bit.ly/3CAzrxt

Claude

Constitutional AI

In the Reinforcement Learning from AI Feedback (RLAIF) phase, responses are generated, and an AI compares their compliance with the constitution

This dataset of AI feedback is used to train a preference model that evaluates responses based on how much they satisfy (*align* to) the constitution

https://bit.ly/3CAzrxt