5. Misc. post-2022 bits

Welcome in post-Nov. 2022…

How GPT-4 is probably made

Model

- Decoder-only architecture
- Very large models (1.8T, v3 was 175B)
- Mixture of Experts (MoE) of 8×220 B or 16×110B
- Distilled/quantized, at least for the mini & turbo versions
- Vision model: an adapter which projects image into input tokens for the decoder

Training

- Pretraining on TB of data. https://commoncrawl.org/, Wikipedia, books, Stackoverflow, Reddit…
- Probably finetuned on some various datasets.
- Aligned using RLHF at first, now maybe DPO or any other RL technique.
- Cost: 60-500 M$ using 100,000s of GPUs
    - but DeepSeek r1 training cost was apparently a fraction of this!

Inference

- Probably tricks like FlashAttention
- Prompt caching, KV caching…
- Advanced decoding strategy (MCTS?), sampling for novelty
- Array of 100's of GPUs

Decoding strategies

- Greedy: argmax P at each step
- Bream search: keep n beams of best options
- Sampling among best options
  - Top k: k best
  - Top p (nucleus sampling): options such as P(x) > p
- Monte-Carlo Tree-Search (MCTS) and variants: tree search with budget and heuristic
- Speculative decoding: use a larger network to check the decoded seq, which it can perform in parallel

What is a “foundation model”

Simply a model (pre)trained on enough data to generalize well to new, unseen tasks (not only data).

Great performance in zero/few-shots regime.

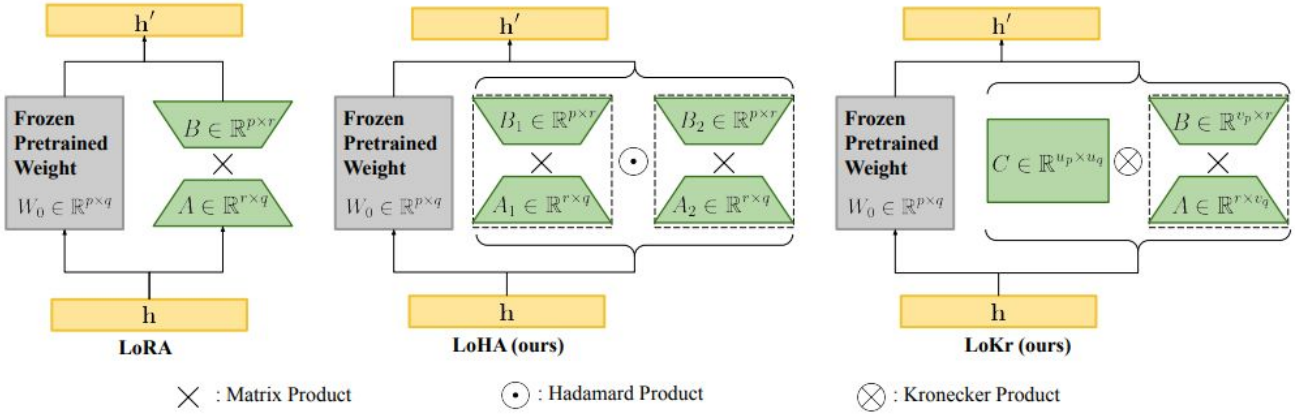
Can usually be fine-tuned to specific tasks for improved performance.

Efficient fine-tuning of transformers

Transformers are not like CNNs. Think of “circuits” rather than “filters”.

All layers must be updated, not only the last ones.

Use LoRa or recents variants of it thanks to the PEFT library to train them efficiently.



Efficient inference

Flash attention: P(x1 | x0), P(x2 | x1, x0), P(x3 | x2, x1, x0)… Scan pattern!

KV cache: store and reuse the intermediate computations (key and value pairs) in the attention layers

…

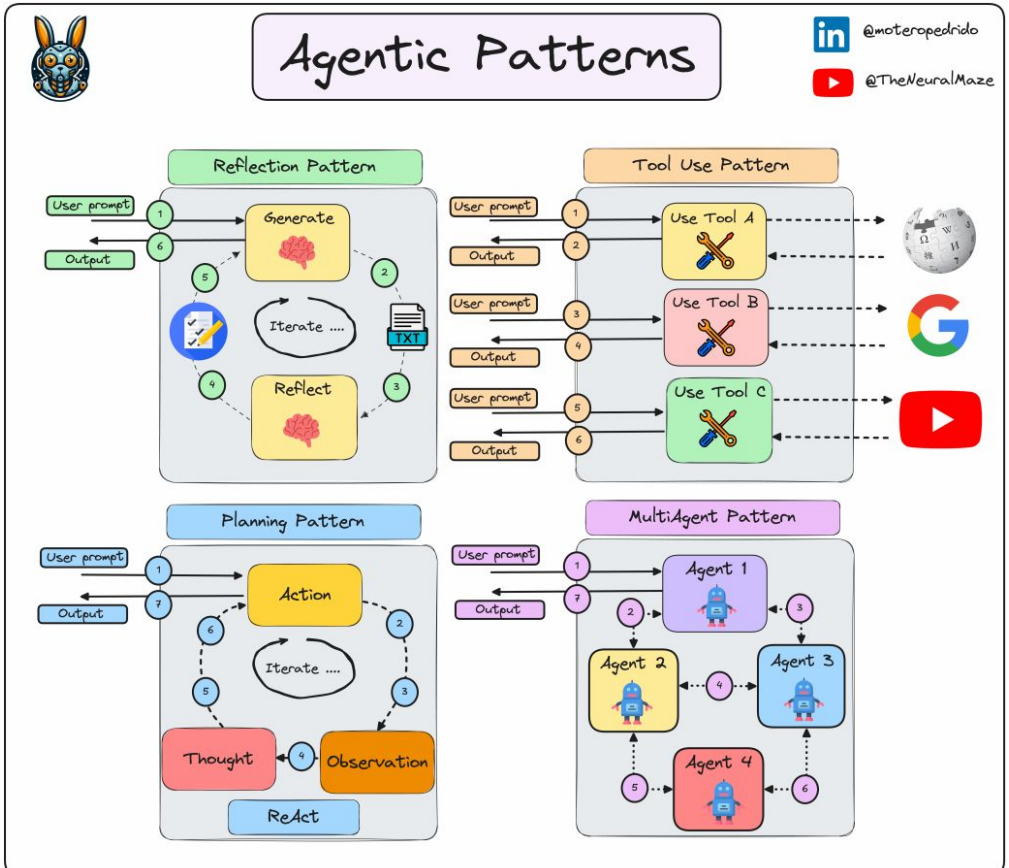
Chain-of-thoughts (CoT)

The model is encouraged to break down the problem posed, possibly adding some <thinking> special tokens to describe what it is doing.

It can be implemented using

- prompt engineering
- specific alignment
- the “planning” agentic pattern (auto-prompting loop)

Agentic patterns



HuggingFace code: some specificities

labels & loss integrated to base model

So, are there only Transformers now?

No.

Diffusion models.

ResNet variants.

State-Space Models like Mamba.

RNN mutants like RWKV.