data iku

LLM-Powered Applications

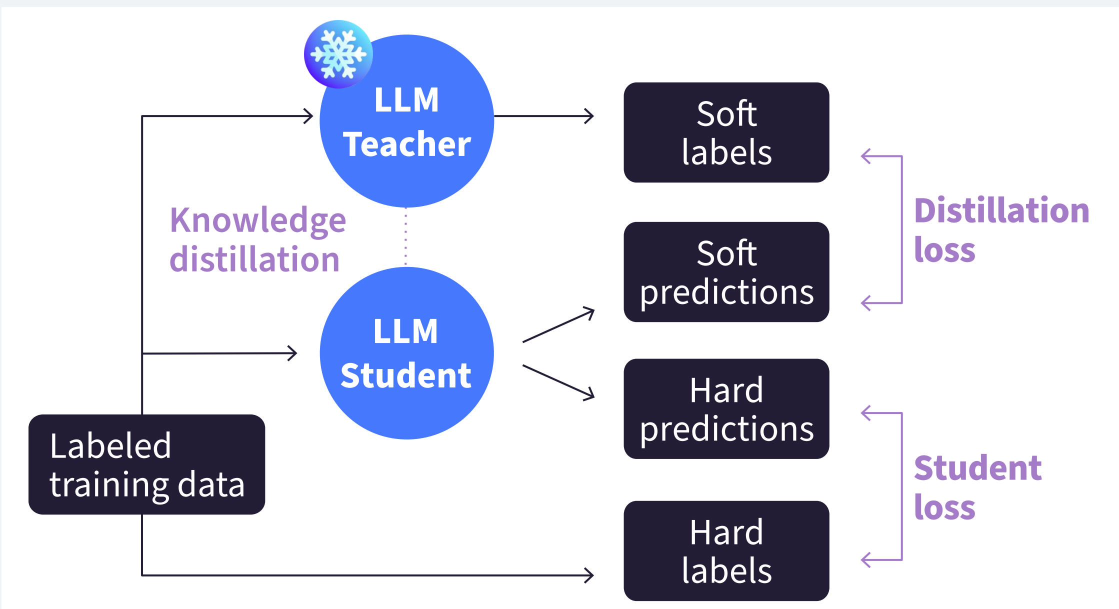
MODEL OPTIMIZATION FOR DEPLOYMENT

Inference challenges: High computing and storage demands

→ Shrink model size, maintain performance

Model Distillation

- Scale down model complexity while preserving accuracy.
- Train a small student model to mimic a large frozen teacher model.



- **Soft labels:** Teacher completions serve as ground truth labels.
- Student and distillation losses update student model weights via backpropagation.
- The student LLM can be used for inference.

Post Training Quantization (PTQ)

PTQ reduces model weight precision to 16-bit float or 8-bit integer.

- Can target both weights and activation layers for impact.
- May sacrifice performance, yet beneficial for cost savings and performance gains.

Model Pruning

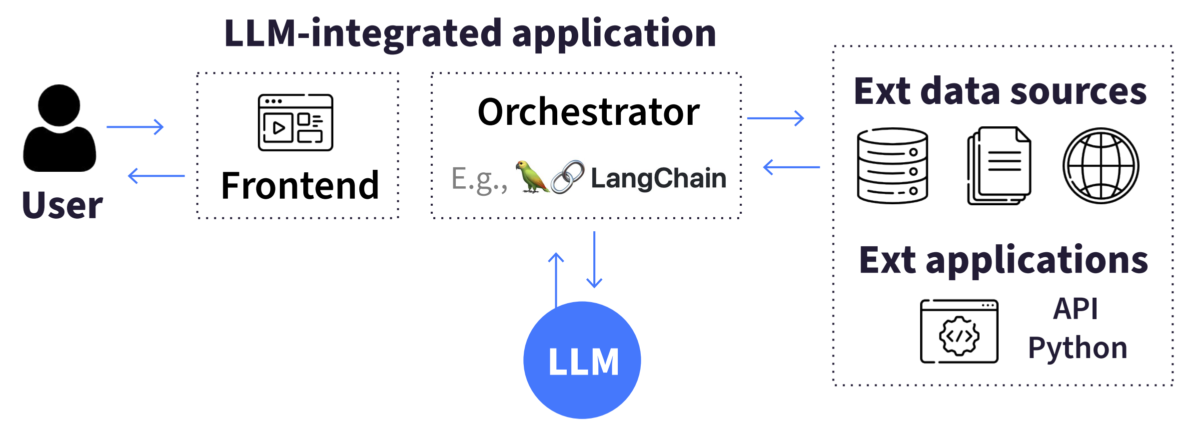
Removes redundant model parameters that contribute little to the model performance.

Some methods require full model training, while others are in the PEFT category (LoRA).

LLM-INTEGRATED APPLICATIONS

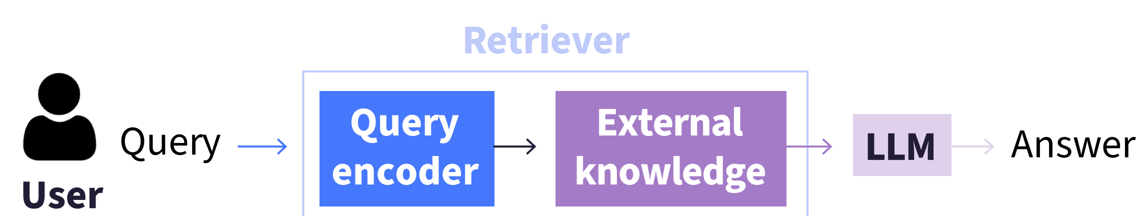
- Knowledge can be out of date.
- LLMs struggle with certain tasks (e.g., math).
- LLMs can confidently provide wrong answers ("hallucination").

→ Leverage **external app** or **data sources**



Retrieval Augmented Generation (RAG)

AI framework that integrates **external data sources** and **apps** (e.g., documents, private databases, etc.). *Multiple implementations exist, will depend on the details of the task and the data format.*



- We retrieve **documents most similar to the input query** in the external data.
- We combine the **documents with the input query** and **send the prompt to the LLM to** receive the **answer**.

Size of the context window can be a limitation.

→ Use multiple **chunks** (e.g., with LangChain)

Data must be in format that allows its relevance to be assessed at inference time.

→ Use **embedding vectors** (vector store)

Vector database: Stores vectors and associated metadata, enabling efficient nearest-neighbor vector search.

LLM REASONING WITH CHAIN-OF-THOUGHT PROMPTING

Complex reasoning is challenging for LLMs.
E.g., problems with multiple steps, mathematical reasoning

→ LLM should serve as a **reasoning engine.**
The prompt and completion are important!

1. Plan actions	2. Format outputs	3. Validate actions
Set of instructions *Step1: Get customer ID* *Step2: Reset password*	Requires formatting for applications to understand actions	Collect information that allows validation of an action

Chain-of-Thought (CoT)

- Prompts the model to **break down problems into sequential steps.**
- Operates by integrating **intermediate reasoning steps** into examples for one-or few-shot inference.

Prompt
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5+6=11. The answer is 11.
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Completion
A: The cafeteria had 23 apples. They used 20 to make lunch. 23-20=3. They bought 6 more apples, so 3+6=9. The answer is 9. ✓

In the completion, the whole prompt is included.

→ Improves performance but struggles with precision-demanding tasks like tax computation or discount application.

Solution: Allow the LLM to communicate with a proficient math program, as a Python interpreter.

PROGRAM-AIDED LANGUAGE & REACT

Program-Aided Language (PAL)

Generate scripts and pass it to the interpreter.

Prompt
Q: Roger has 5 tennis balls. [...]
A: CoT reasoning
# Roger started with 5 tennis balls
tennis_balles=5 PAL execution
# 2 cans of tennis balls each is
bought_balls=2*3
# tennis balls. The answer is
answer = tennis_balls + bought_balls
Q. [...]

Completion is handed off to a Python interpreter.
↓
Calculations are accurate and reliable.

ReAct

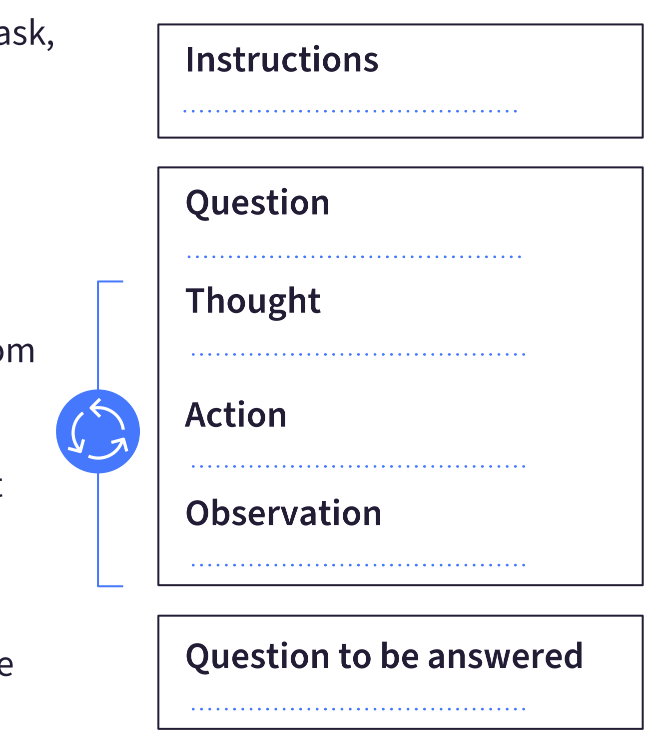
Prompting strategy that combines CoT reasoning and action planning, employing **structured examples** to guide an LLM in **problem-solving** and decision-making for **solutions**

Instructions: Define the task, what is a thought and the actions

Thought: Analysis of the current situation and the next steps to take

Action: The actions are from a predetermined list and defined in the set of instructions in the prompt
The loop ends when the action is finish []

Observation: Result of the previous action



→ **LangChain** can be used to connect multiple components through agents, tools, etc.

Agents: Interpret the user input and determine which tool to use for the task (LangChain includes agents for PAL & ReAct).

ReAct reduces the risks of errors.