CS324 – Large Language Models

Adaptation

HUMANE Lab

김태균

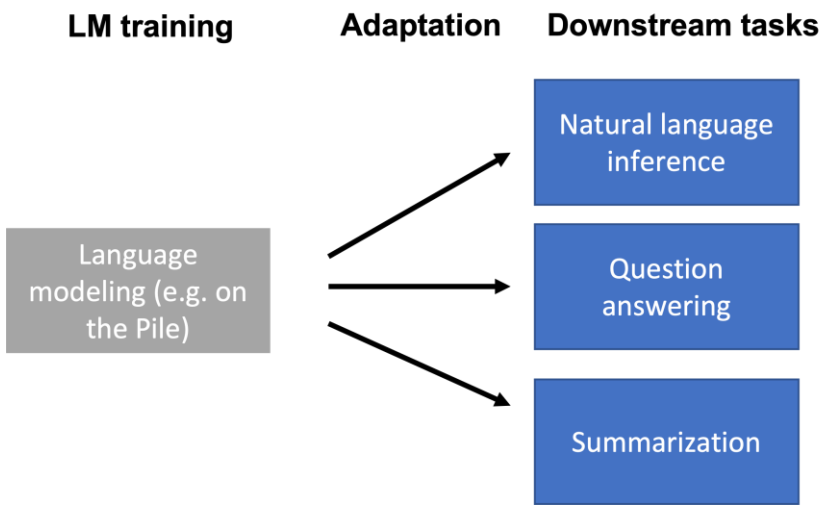
2025.02.07

Overview

- What is adaptation?
- Ways to utilize adaptation
  - Probing
  - Fine-tuning
  - Lightweight Fine-tuning

What is adaptation?

- Adaptation : The process of tuning a pre-trained model for a downstream task
- Why adapt the language model?
  - LMs are trained in a task-agnostic way
  - Downstream tasks can differ from LM training data in format and topic, or require updating new knowledge over time



What is adaptation?

- General adaptation setup
  - Given a downstream dataset, train a new model that depends on pre-trained LM
  - Optimize some parameters to minimize task loss, a subset of the existing parameters or introduce new parameters

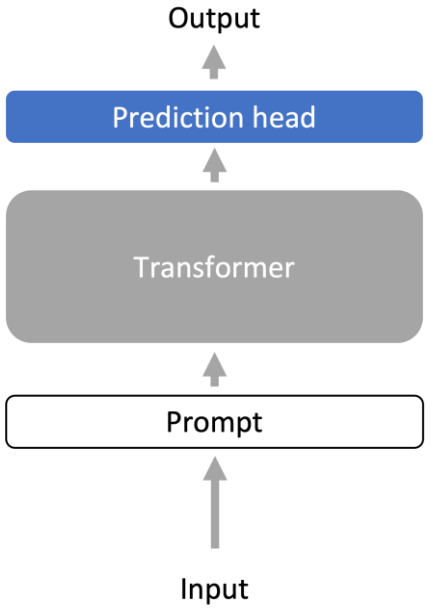
$$\gamma_{\text{adapt}} = \text{argmin}_{\gamma \in \Gamma} \frac{1}{n} \sum_{i=1}^{n} \ell_{\text{task}}(\gamma, \theta_{\text{LM}}, x_i, y_i)$$

Ways to utilize adaptation

1. Probing
2. Fine-tuning
3. Lightweight Fine-tuning (PEFT)

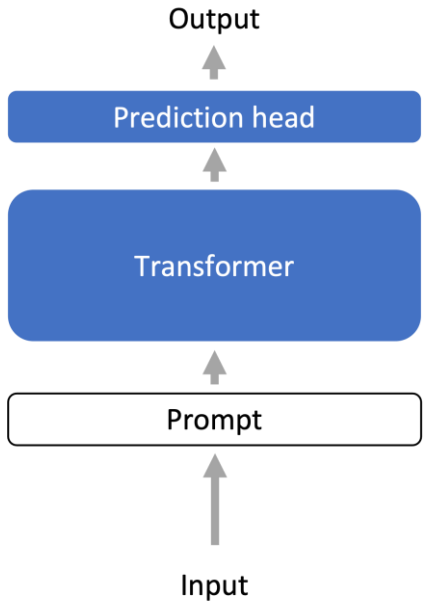
Probing

- Probing is usually for inspecting/understanding representations of the model
- For adaptation, we train a prediction head from the last layer representations of the LM to the output
  - e.g. sentiment analysis, POS tagging



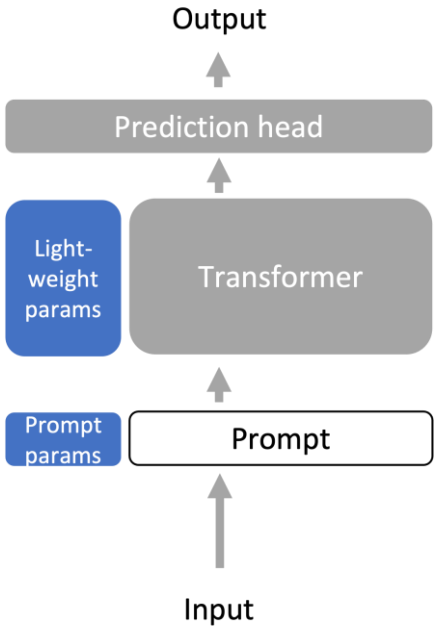
Fine-tuning

- Fine-tuning uses the LM parameters as initialization for further training all of the parameters on a downstream task
- Requires storing a LLM specialized for every downstream task, which can be expensive

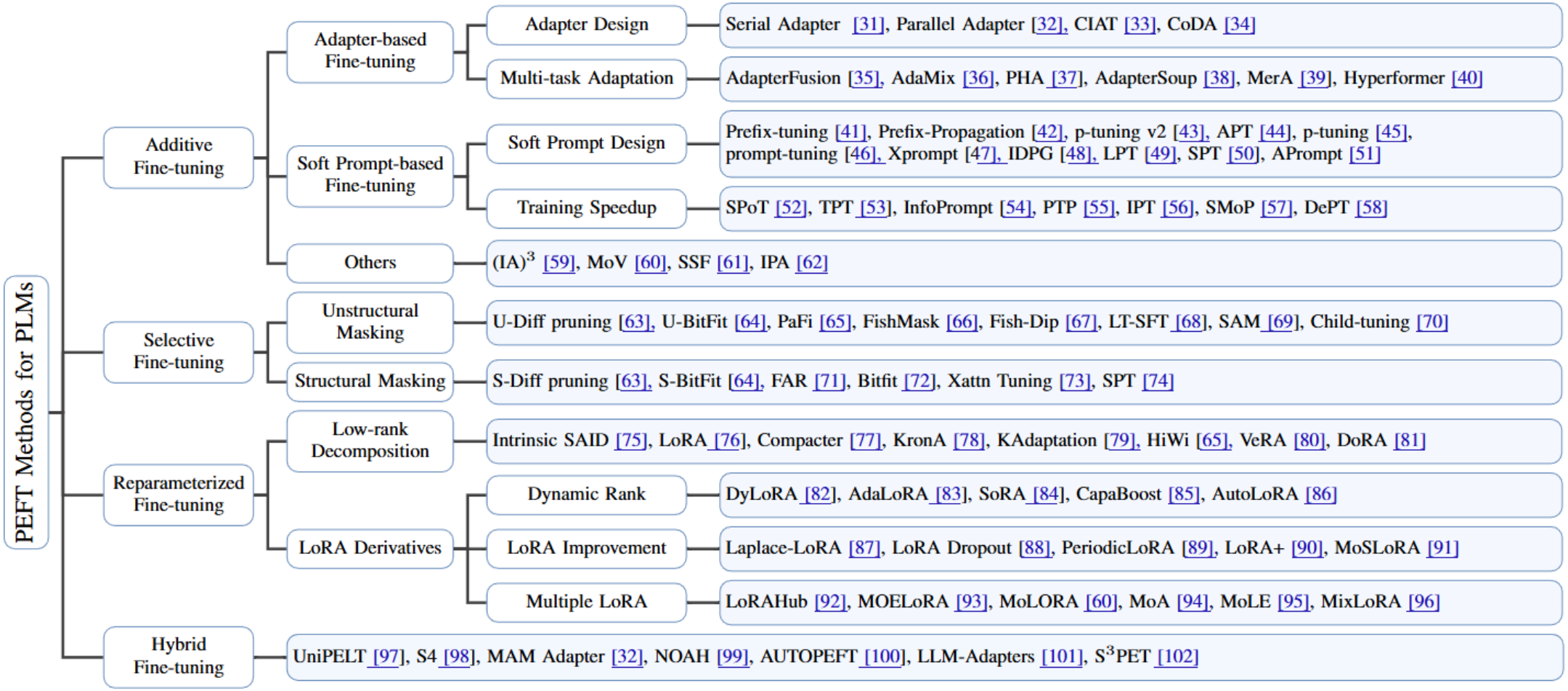


Lightweight Fine-tuning

- Lightweight fine-tuning aims to have the expressivity of full fine-tuning while not requiring us to store the full language model for every task
- Parameter efficient fine-tuning (PEFT)

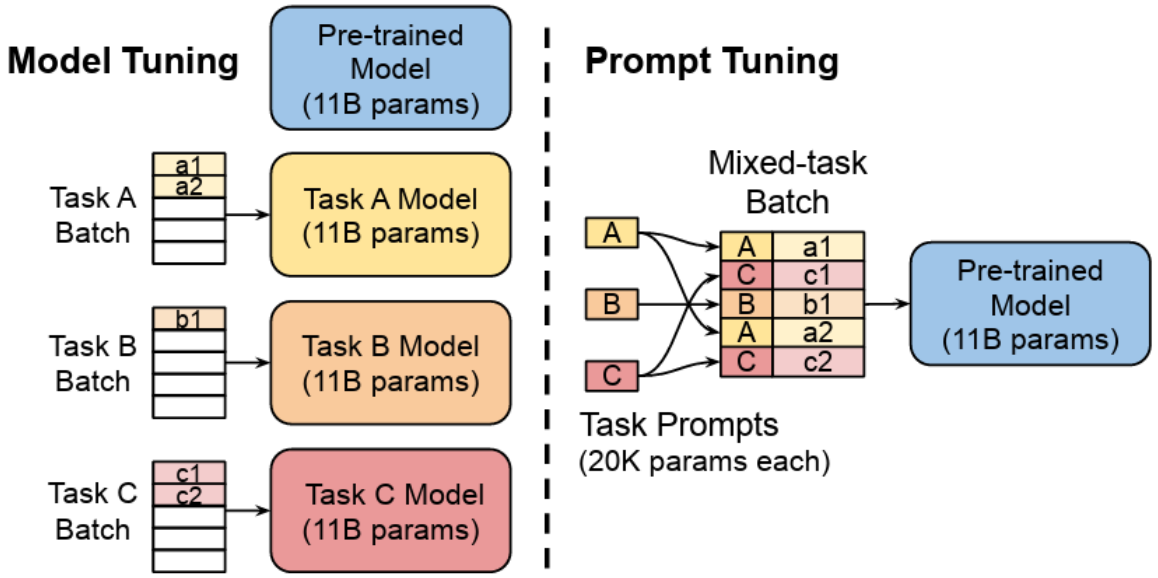


PEFT taxonomy



Prompt Tuning

- Prompting is the approach of adding extra information for the model to condition
- Prepends k learnable, continuous token embeddings to the input and trains this on the labeled task data



Prefix Tuning

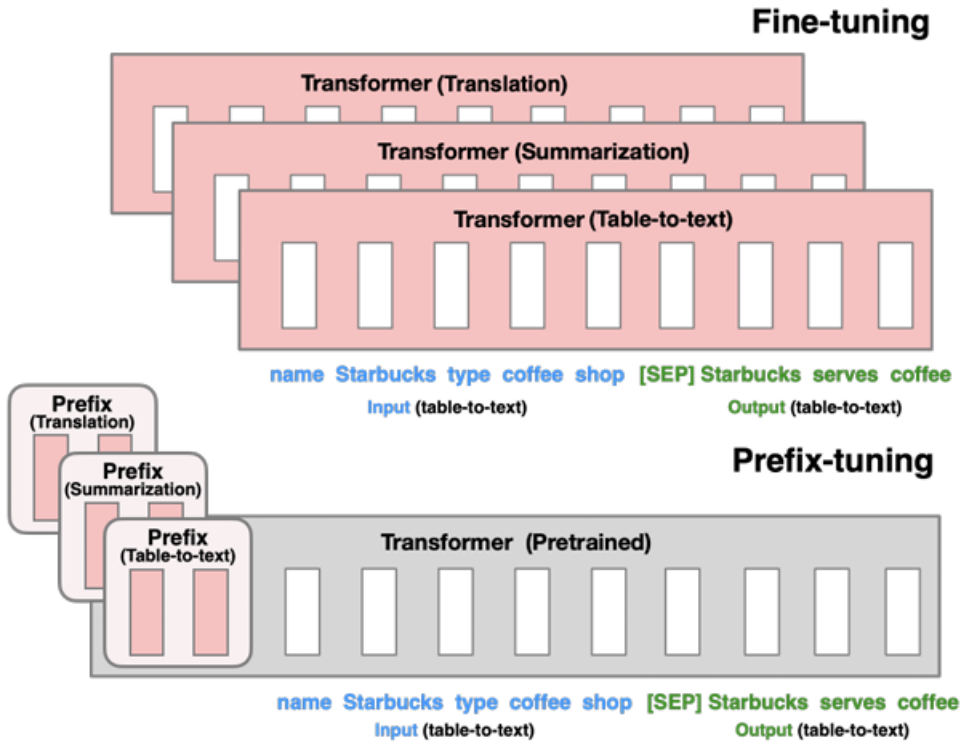
- For k positions prepended to the input, concatenate additional learnable weights for keys and values at every attention layer

$$\text{Attn-op}(Q, K, V) = V\text{softmax}(K^\top Q/\sqrt{d})$$

$$K_{\text{prefix}} = [P^{(i)}_{\text{key}}, K]$$

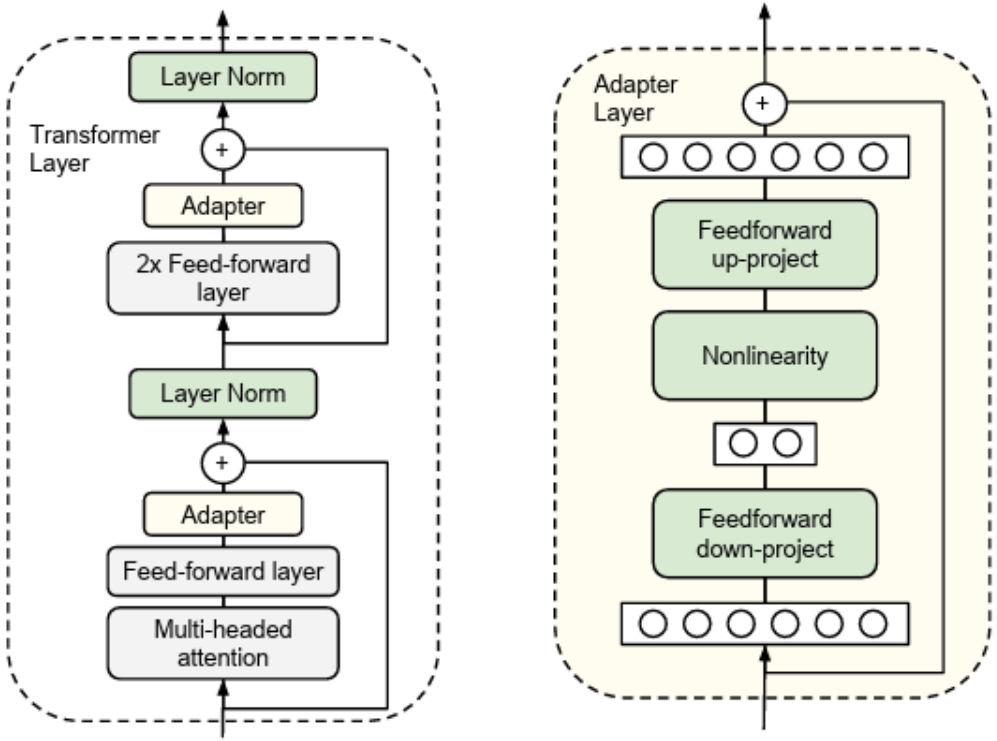
$$V_{\text{prefix}} = [P^{(i)}_{\text{value}}, V]$$

$$\text{head}_i = \text{Attn-op}(Q, K_{\text{prefix}}, V_{\text{prefix}})$$



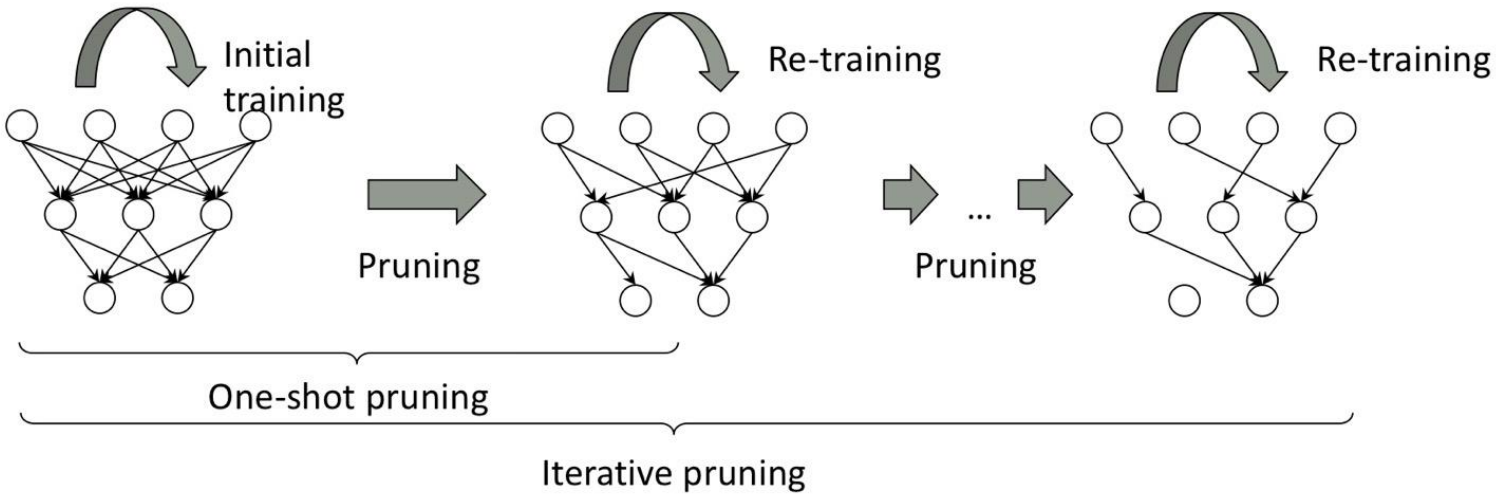
Adapter Tuning

- Adding a adapter layer to each layer of the Transformer model
- Adapter layer learns only a very small number of parameters while keeping the original model's weights fixed
  - Bottleneck structure



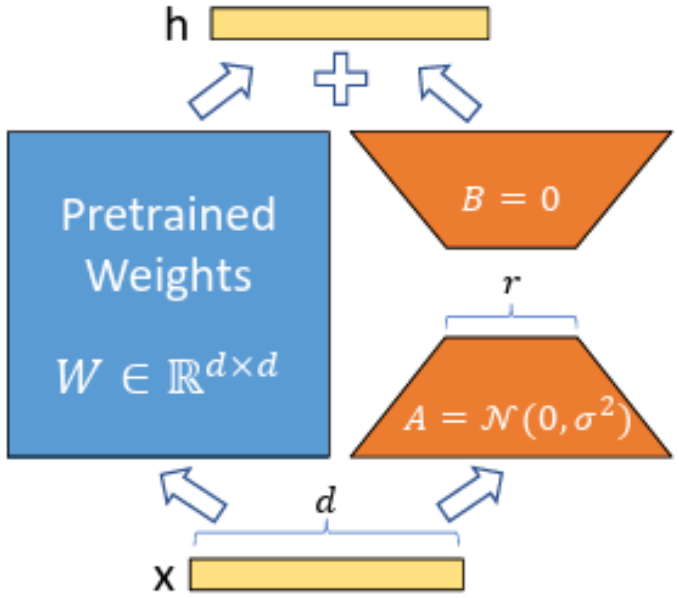
Pruning

- During pruning, a fraction of the lowest-magnitude weights are removed
  - e.g. S-diff pruning, U-diff pruning



LoRA (Low-Rank Adaptation)

- The process of fixing the existing weights of a model and using low-rank approximation to update the model



Conclusion

- We need to adapt LLMs to the diverse array of downstream tasks