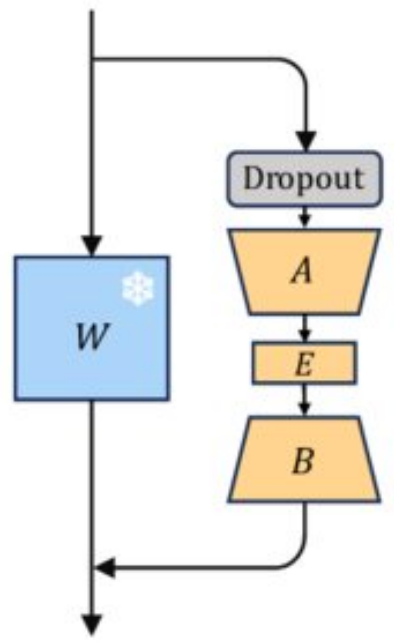
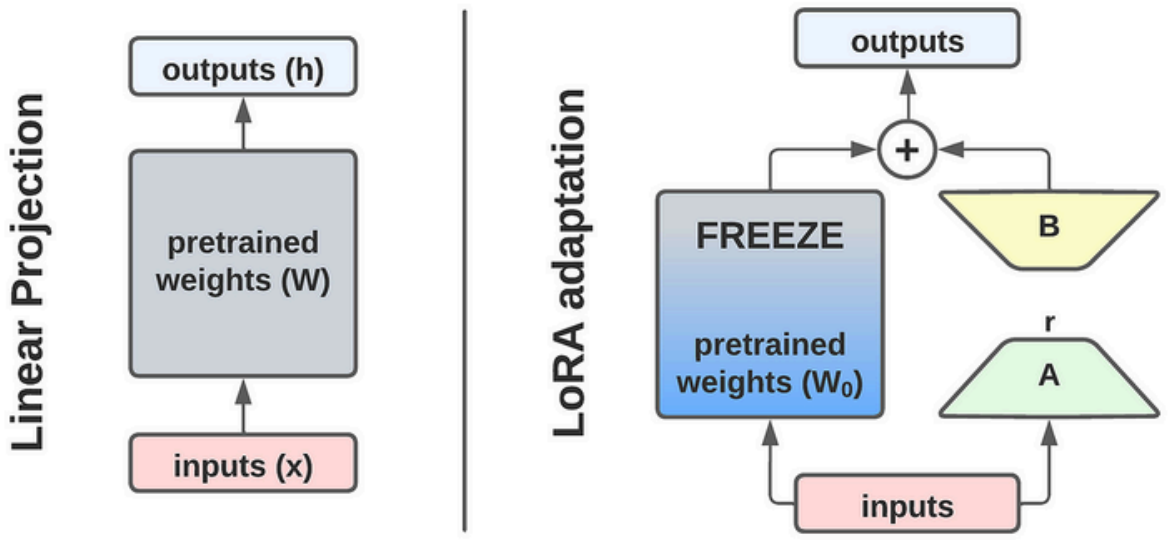
5 Techniques to FineTune LLMs

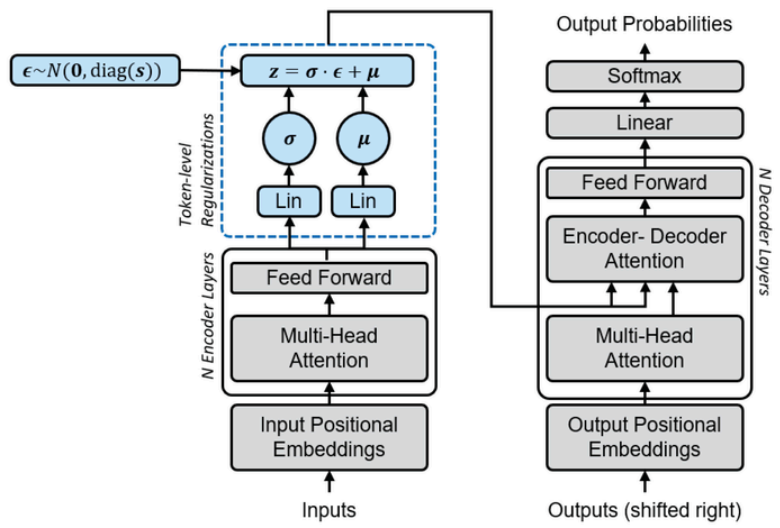
Delta LoRA



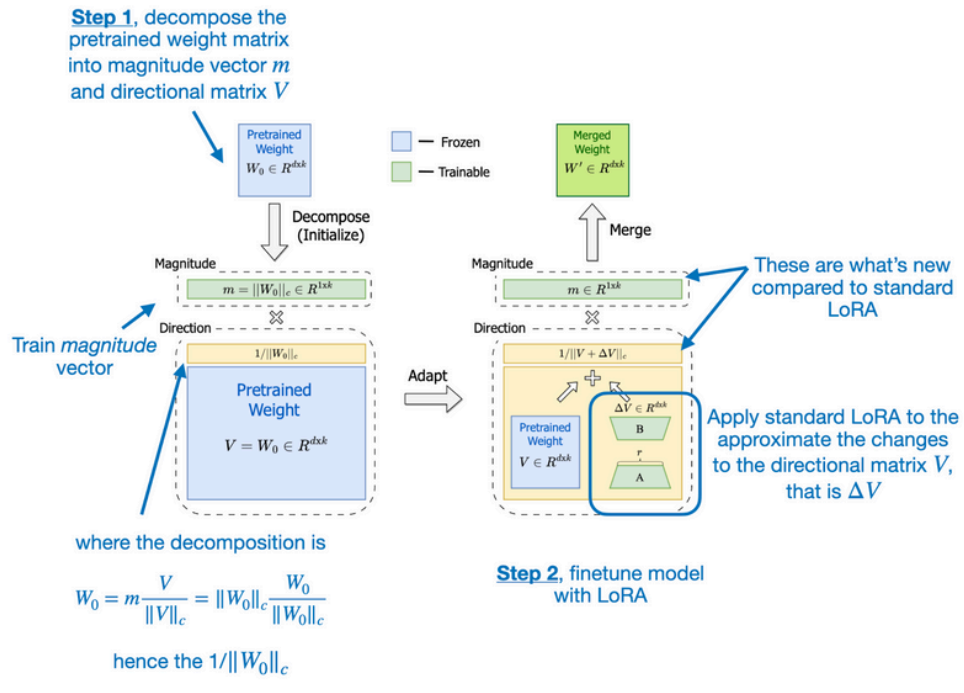
LoRA (Low-Rank Adaptation)



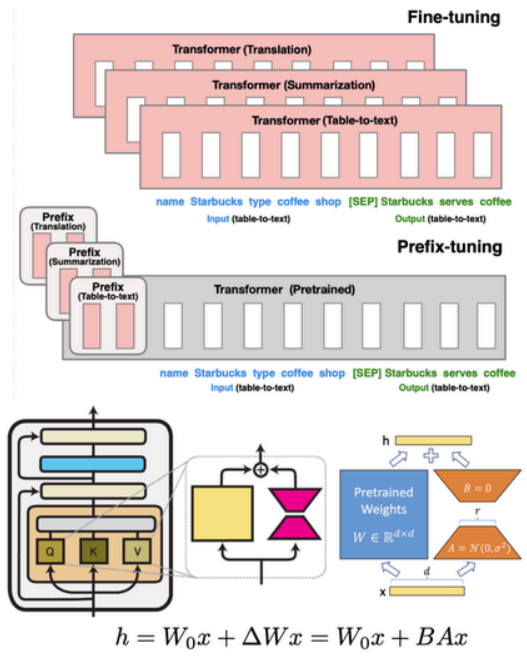
Vera



LoRA-FA(Feature Augmentation)



Prefix Tuning



LoRA

Overview

LoRA adds a low-rank adaptation matrix to existing layers of the model, reducing the number of trainable parameters.

How it works

Instead of updating the entire weight matrix, LoRA only updates a small rank-decomposed matrix, which significantly lowers computational costs while preserving performance.

Benefits

- Greatly reduces memory footprint.
- Improves training efficiency.
- Can be applied without changing the model’s core architecture.

LoRA-FA

Overview

LoRA-FA combines LoRA with Feature Augmentation, where external features or information are injected into the model along with low-rank adaptations.

How it works

In addition to the low-rank matrices, this approach augments the input with task-specific external features, improving the fine-tuning process by providing extra information.

Benefits

- Enhances model adaptability to specialized tasks.
- Improves performance with minimal added computational cost.
- Useful in domain-specific fine-tuning tasks where additional features are available.

Vera

Overview

Vera regularizes embeddings by adjusting them virtually during fine-tuning to prevent overfitting and improve generalization.

How it works

Vera applies regularization techniques to embeddings by fine-tuning them in such a way that they don’t deviate too much from the original, while still adapting to new data.

Benefits

- Helps maintain generalization while fine-tuning.
- Reduces the risk of catastrophic forgetting (when a model forgets previously learned tasks).
- Makes fine-tuning more robust across different domains.

Delta LoRA

Overview

Delta LoRA is an extension of LoRA, where delta updates are applied selectively, focusing on the layers that exhibit the most significant learning, further optimizing memory usage.

How it works

It adds another layer of granularity by fine-tuning only specific parts of the network, such as a subset of layers or blocks where the model shows the most learning gains, further reducing computational overhead.

Benefits

- Further reduces the number of parameters to fine-tune.
- Increases computational efficiency while maintaining model performance.
- Useful in scenarios where fine-tuning resources are limited.

Prefix Tuning

Overview

This technique prepends a trainable sequence of tokens (prefixes) to the input of each transformer layer during fine-tuning.

How it works

Instead of fine-tuning the entire model or even specific layers, prefixes are learned and optimized to guide the model’s attention and outputs towards the fine-tuning task.

Benefits

- Allows task-specific adaptation without modifying core model weights.
- Memory and computation efficient because only small amounts of data (prefixes) are trained.
- Flexible and effective for fine-tuning large-scale LLMs.