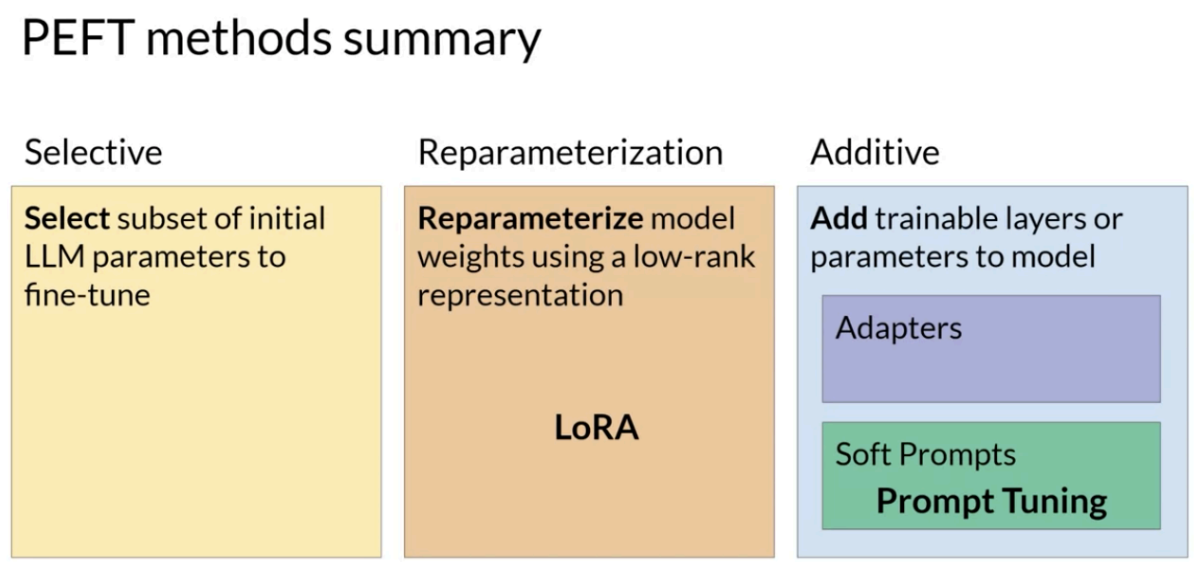
Parameter Efficient Fine Tuning

Training Language Models (LMs) poses significant computational challenges, especially with the proliferation of large-scale models. Parameter Efficient Fine-Tuning (PEFT) methods offer a solution by updating only a subset of parameters, reducing memory requirements while maintaining performance. In this article, we delve into the intricacies of PEFT, its advantages, and explore various techniques within this paradigm.

Training LMs demands extensive memory allocation not only for storing model weights but also for optimizer states, gradients, and temporary memory. Full fine-tuning, where all model weights are updated, exacerbates these challenges, leading to prohibitive memory requirements, particularly on consumer hardware.

PEFT methods aim to mitigate memory constraints by updating only a fraction of model parameters. This approach significantly reduces the memory footprint, making training feasible even on single GPUs. Moreover, PEFT minimizes the risk of catastrophic forgetting associated with full fine-tuning, as it preserves the majority of the original model weights.



Selective Methods:

Selective PEFT methods fine-tune specific components or layers of the original LM. While offering flexibility, they often involve trade-offs between parameter efficiency and computational efficiency. However, these methods are not the focus of this article.

Reparameterization Methods:
Reparameterization techniques retain the original LM parameters but reduce the number of trainable parameters through low-rank transformations. One such method is LoRA (Low-Rank Adaptation), which efficiently compresses model parameters while preserving performance.

Additive Methods:
Additive PEFT approaches maintain the frozen original LM weights and introduce new trainable components. Adapter methods add new layers to the LM architecture, while soft prompt methods focus on manipulating input for performance enhancement. Prompt tuning, a soft prompt technique, is explored further in this article.

Exploring Prompt Tuning:
Prompt tuning, a specific soft prompt method, entails modifying prompt embeddings or retraining embedding weights to optimize model performance. By keeping the LM architecture fixed, prompt tuning achieves efficient fine-tuning while enhancing task-specific performance.