|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Scope** | **Select** | **Adapt & Align Model** | | **Application Integration** | |
| Define the Problem | Choose the model | Prompt Engineering | Evaluate | Optimize & Deploy model for inference | Augment model & build LLM powered application |
| Fine Tuning |
| Align with human feedback |

**Fine Tuning**

**Fine tune and LLM with Instruction Prompts:**

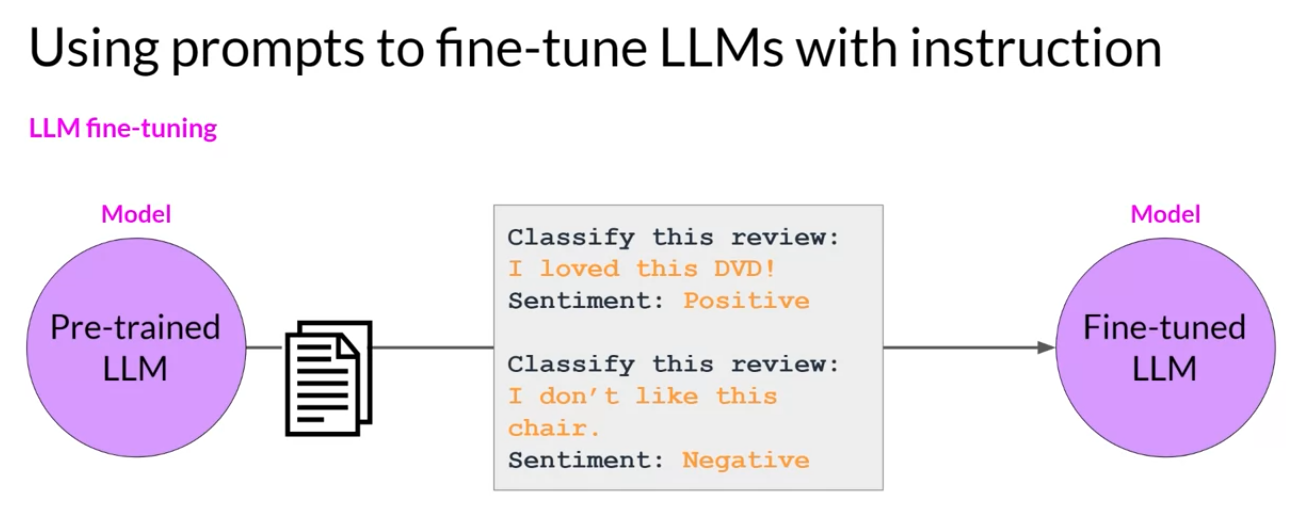
There are limitations of in context learning (one shot, few shot) because:

* For smaller models it doesn’t work even with five or six examples are included.
* Examples take up space in context window (where prompt is written).
* Instead try fine tuning the model.

In pre training where we train the model using unstructured textual data via self-supervised learning.

Fine tuning is supervised learning process where you use a dataset of labeled examples to update the weights of the LLM. The labeled examples are prompt, completion pair.

**Instruction fine tuning:** Trains the model using example that demonstrates how it should respond to a specific instruction.

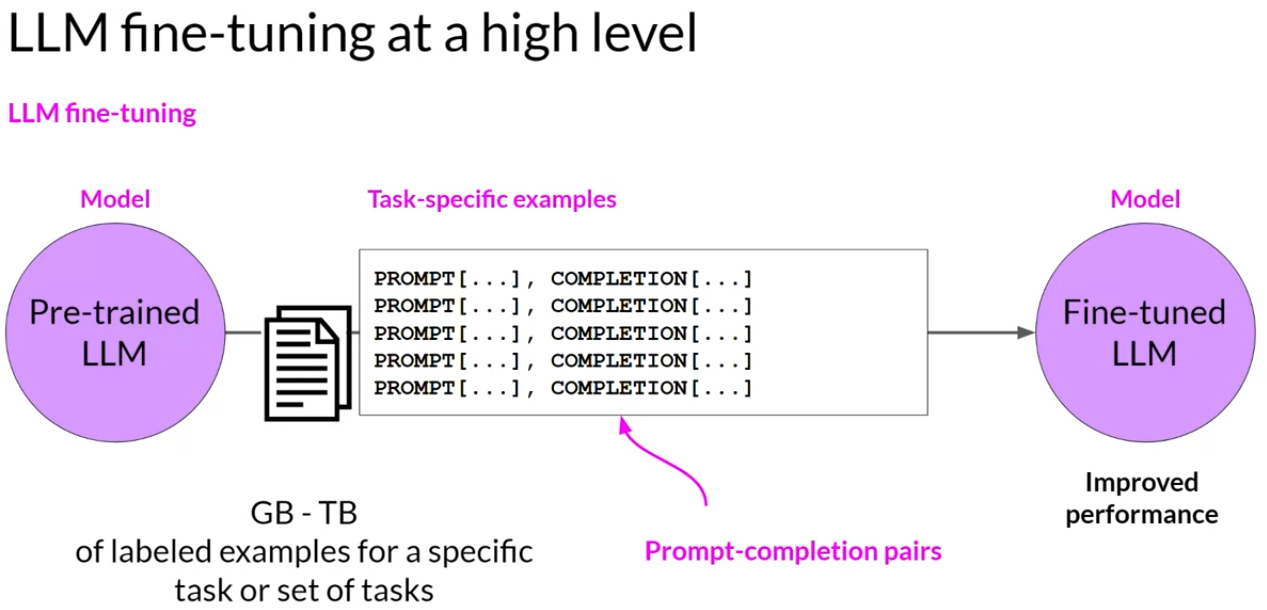


Where all the model weights are updated is known as full fine tuning.

How do you go about Instruction fine tuning an LLM:

* Prepare your training data.
* Divide the dataset into Training [prompt, completion], validation [prompt, completion], and test set [prompt, completion].

The fine tuning results in new version of base model often called instruct model that is better at task you are interested in. Fine tuning with instruct prompt is the most common way to fine tune LLMs these days. Fine tuning is also called instruct fine tuning.



**Fine Tuning on Single task:**

Down side is that model can be affected by Catastrophic forgetting, when fine tuning the LLM on single task it can perform better but it can degrade performance on other tasks.

**To avoid catastrophic forgetting:**

Fine tune the model on multiple tasks.

Consider parameter efficient Fine tuning (PEFT) instead of full fine tuning. PEFT preserves the weights of original LLM and trains only a small number of task specific layers.

**Multitask fine tuning:**

It is extension of single task fine tuning where dataset is comprised of example inputs and outputs for multiple tasks.

