*Pre-training:*

1. Massive text data (Mainly in TBs)

2. Identify the model architecture: Once we have chosen the model architecture we need a

3. Tokenizer that can efficiently encode or decode our data

4. Dataset is pre-processed used the Tokenizer vocabulary using libraries like Sentence piece which ensures that the data is in a format which is suitable for training.

This step involves mapping tokens to the corresponding IDs and Incorporating any special tokens/Attention mask.

In the pre-training phase the model learns to predict the next word or fill-in missing words. It involves iterating training procedure which maximizes the likelihood of the generation of word we want.

Masked language model is used for filling-in the missing words where as Causal language model is used for Text generation based on context.

Since our model lacks knowledge about a particular task or domain we fine-tune the model to be able to generate the desired output.

*Fine-Tuning:*

1. Fine-Tuning allows us to specialist the LLMs capabilities and optimize its performance on a task specific dataset.

There are multiple methods to fine tune a model. These are some of the ways:

1. Instruction tuned model: Typically used for question answering, contains tuples of instruction-response.
2. Optimizes the task specific loss function.
3. The params of the pre-trained model are adjusted using the gradient based optimization algorithms like SGD or ADAM.
4. Learning rate scheduling and regularization method in order to prevent overfitting
5. The fine-tuned model should always generalize not memorize

*LoRA [Low Rank Adaptation]:*

1. Fine-Tuning is very expensive that where VRAMs and GPUs comes in.
2. LoRA can reduce the trainable prams by 10000 times and GPU memory by 3 times
3. QLoRA [Quantized LoRA] uses bitsandbytes to achieve near loss-less Quantization.

*Fine-Tuning a 7B model:*