|  |  |  |
| --- | --- | --- |
|  |  | Research notes  NAZMUS SAMMO-103512692 |

Date: 20 Aug, 2023

# **Large language model**

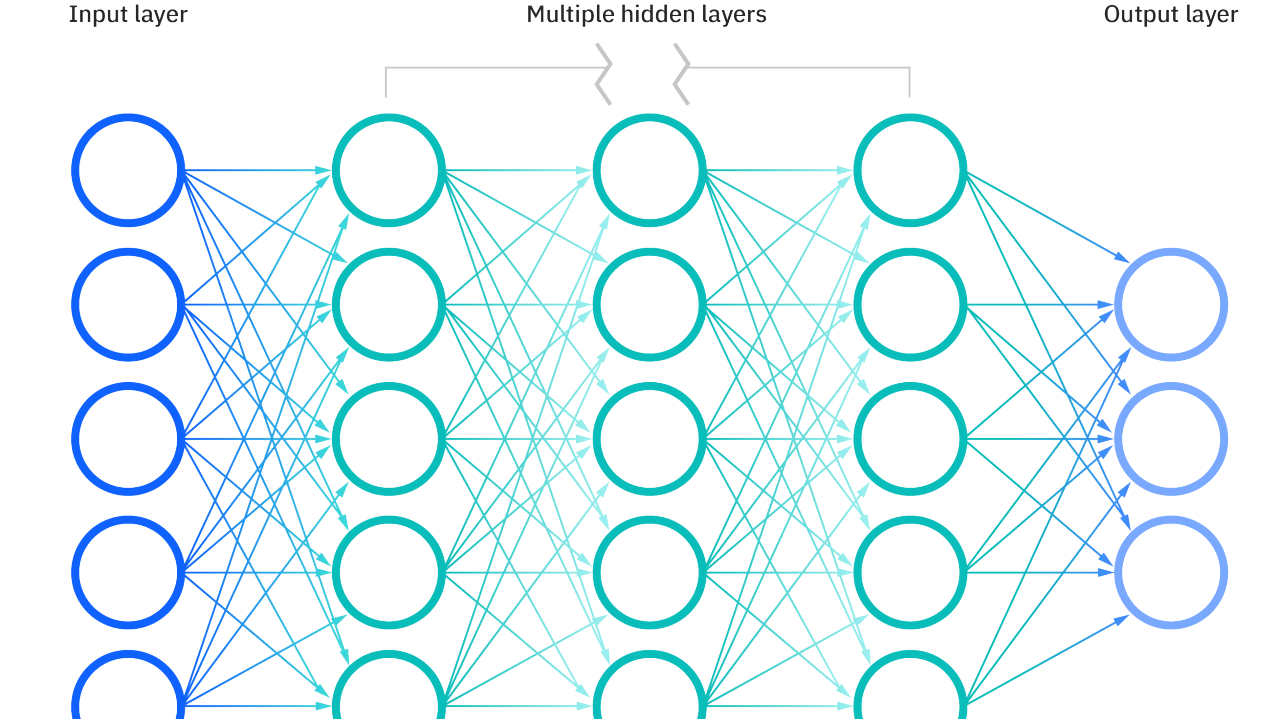
LLM’s are a part of generative AI which imitates human intelligence mainly language and speech. In the backend it uses the deep learning techniques like massive artificial neural networks and natural language processing tasks combined huge amounts of data and then aligned to human values to create a reasoning engine. LLM’s are basically trained on 2 states- Pre-training and Fine tuning.

In pre training phase the model is trained on huge datasets like we can have a look at Meta’s open source LLM, Llama-2 which is trained on 2 trillion token and the parameter of the model is 70 billion. The objective of the pre training phase is to teach the model about facts, reasoning, context, grammar etc.

And the next step is to fine tune the model. Why do we need fine tuning? After the pre training of the model, we call the model as a Base model, which preforms average to all the tasks. To understand it more clearly, we can take an example of the student. When he goes through the school he got to learn about basic and wide range of thing, and when he goes to the university, he learns specific topic and become good it at it. So, we can compare the students school phase with model’s pre training phase and the university with the fine tuning, where the model becomes expert on specific tasks. So, to summarize, Fine-tuning allows the model to leverage the broad language understanding it gained during pre-training and specialize it for narrower tasks, making it more useful for practical applications.

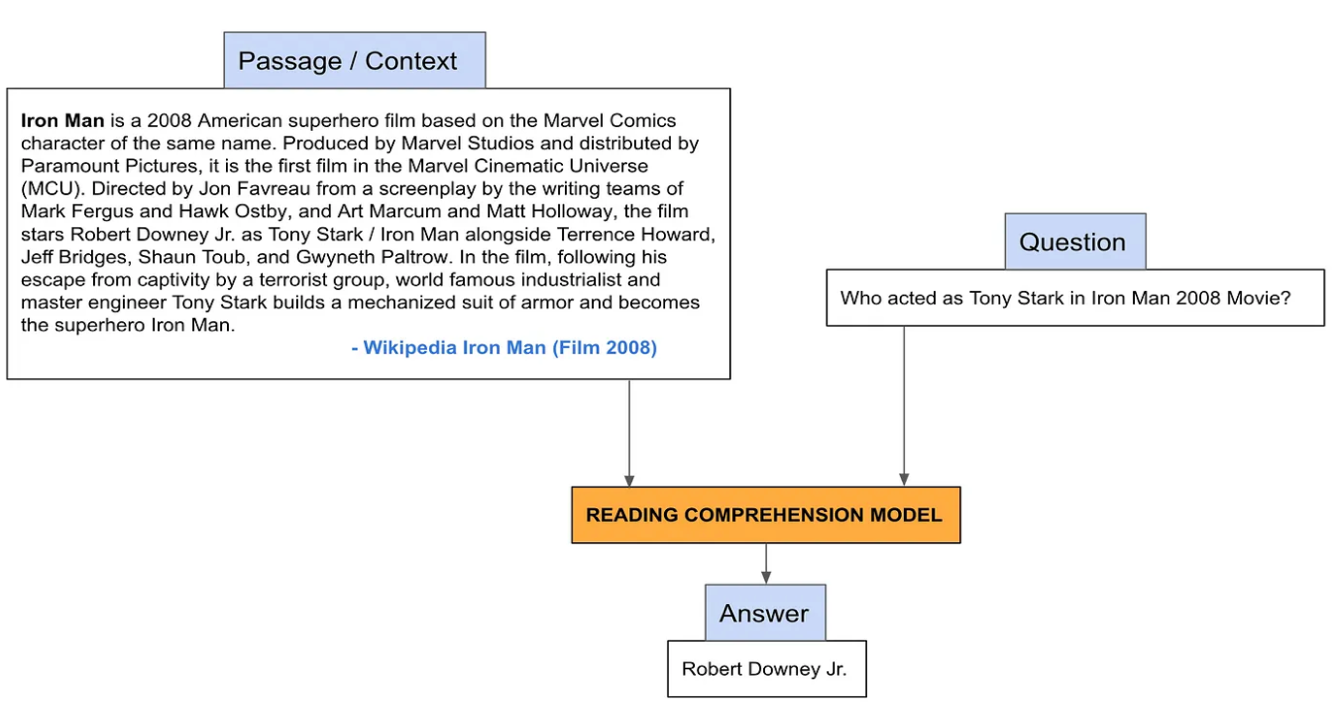
**NLP and ANN**

The process of giving the ability to a machine to understand human like text and spoken words is called natural language processing, NLP. For a human being understanding is much easier but when we try to provide the same power to a machine it’s more complex than we can understand. So in this process the NLP engineers try to replicate the same architecture as human brains which is billions of inter connection of brains so that the machine can think like a human, and they name it same as Neural network. It’s a type of machine learning process called deep learning which uses billions of interconnected nodes.



## **Machine Reading Comprehension**

MRC is an NLP task which involves training computer systems to understand text paragraphs and answer the question which is asked from the context. To train a machine which can perform this task have to go through a lot of training which encompasses various aspects of learning such as information extracting, logical reasoning and language understanding. In simple words if we want to understand what MRC is, we can point to the process where we will provide the large amount of dataset to the MRC model and ask a question from that passage, and an MRC will give the exact answer.



**Fine Tuning LLM**

We have already discussed about LLM and the 2 parts of LLM- Pre training and fine tuning the pre trained LLM. In this part we will dive deep into the process of how to fine tune a pre trained LLM which is the main objective of our project.

There are many types of fine tuning depending on the needs of our project for example if we want to use a pre trained LLM which can generate text and we want to use it solve classification problems that time we can repurpose the model by making tiny changes to its architecture before fine tuning it with the dataset. For this application we only need to use the embeddings of transformer part. We will connect the model’s embeddings layer to a classifier model that maps the embeddings to class probabilities. Here the attention layer of the pre trained LLM will be frozen, so we don’t need much computing power.

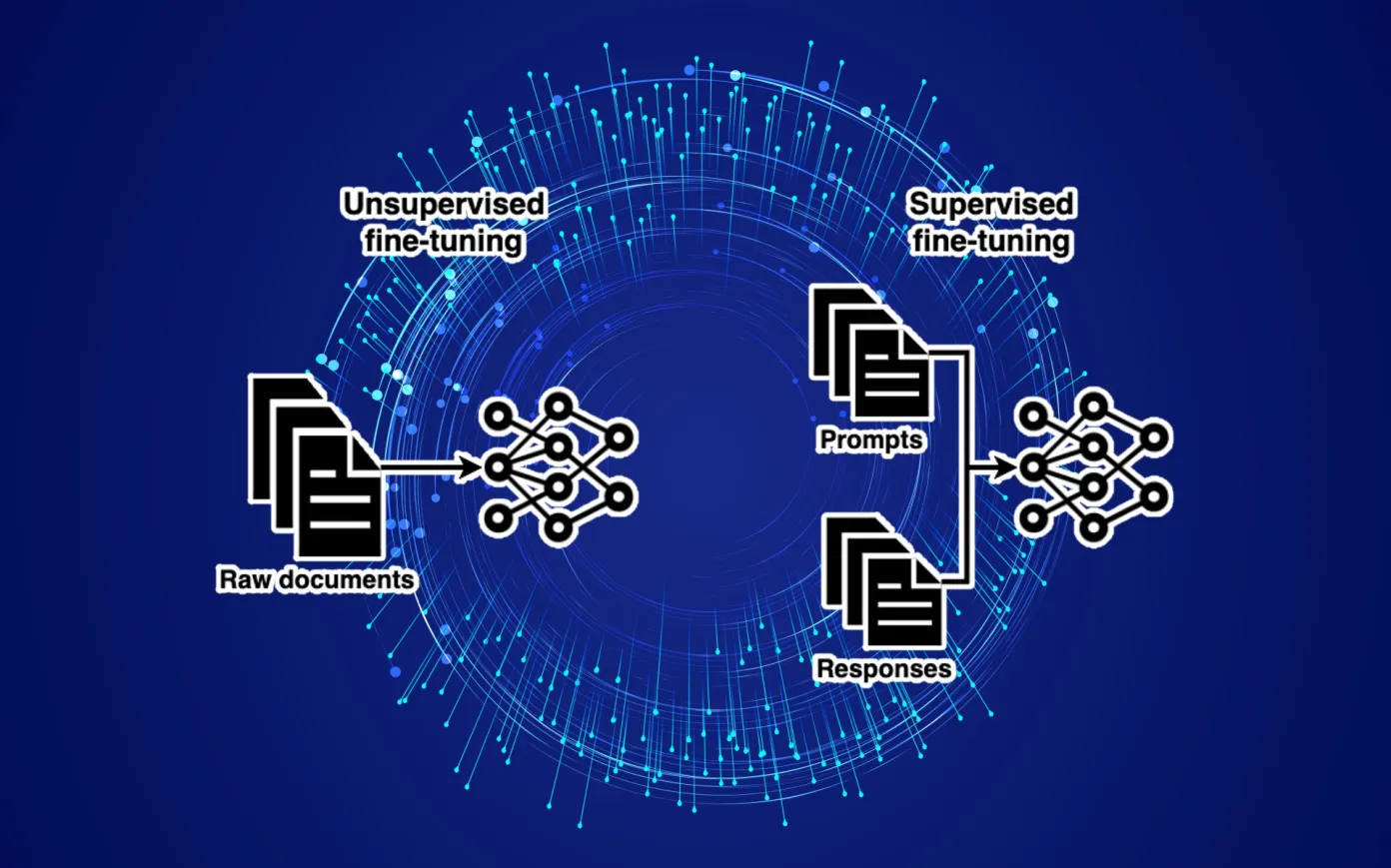
But it’s not always the case, sometimes we need to fully fine tune a LLM depending on our need which is complex and need much more computing power because that time the attention layer of the transformer model needs to be unfrozen.

A diagram of a machine

Description automatically generated

**Supervised and Unsupervised fine tuning:**

On the pre training of a LLM, most of the time the model is trained with unstructured data because it is scalable, and the model can be trained through unsupervised or self-supervised learning. But if we use unsupervised method the model is learning the things by itself and produces the result from there. In some cases, we want to control or change the behavior of the pre trained LLM, that time the Supervised Fine Tuning (SFT) comes into the picture. That time we need to use the structured dataset. What do we mean by structured dataset? Let’s say we want to fine tune a pre trained LLM for medical question answering, that time we can’t use the pre trained LLM because we don’t know for question how the existing model will react. So basically, for this problem we want to control the behavior of the existing dataset. That time we take a collection of prompts and their corresponding responses. And use the SFT method to train the model. SFT is really important for those LLM models which are designed to follow user instruction and stay on specific task across long stretches of text.



**RLHF**

There are one fine tuning method called, Reinforcement Learning From Human Feedback (RLHF), which is very expensive and complicated process to implement. The main idea behind this training model is adding human feedback. If we look at a LLM model which is trained on SFT or UFT, these are basically billion of tokens and after training it can generate token which is meaningful and there can be many solutions to the instruction. So, in the RLHF method what we do is, we include humans to the process and Humans give rating to all the answer the model generates. Finally, by using the deep reinforcement learning the model loop, the reward model rates the model, and the model updates it’s parameters in a way that maximizes its reward.

**Parameter Efficient Fine Tuning**

This technique tries to reduce the number of parameters which needs to fine-tune. When we look at all the LLM models, here we can take the example of Llama-2 with 70B parameter architecture, if we want to fine tune this massive LLM model it’s impossible to do it our local computer also storing deploying the fine-tuned model for each downstream task is very expensive. With PEFT technique we can solve both problems. So, what this technique does is, it freezes most of the most of the parameters of pre rained model and only fine tune a small number of model parameters. With this technique it overcomes the issue of increased storage and computational power also the issue of catastrophic forgetting when we want the fine tune the whole model.

**Low-Rank Adaption (LoRA)**

Before we get to know about LoRA, we first need to understand how a LLM model gets fine-tuned. Basically, what is does it, when the model is provided with fine tuning dataset it try to predict the next the token and then compare the predicted token with the actual token in the dataset. Then it adjusts the weights to correct its prediction. And the model keeps repeating the process and the model becomes fine-tuned for the downstream task.

But in LoRA it applies different approach, it freezes the pre trained weights and with the fine-tuning dataset the model applies the modification to a separate set of weights and it add their new values to a new sets of parameters. In LoRa the model creates 2 downstream weight matrices. One transforms the input parameters from original dimension to the low rank dimension and another one transforms the low rank data to the output dimension of the original model. During training, modifications are made to the LoRA parameters, which are now much fewer than the original weights. So, this fine-tuning approach is affordable and time efficient

## **Prompt Engineering**

Prompt is the way we send the text into a llm model, the way we adjust the text that goes into the llm is basically prompt engineering. Language models are the statistical models of text, and they assign a probability to every suffix of a prompt token by token. Statistical model – So can we take any text from the source and tokenize then and feed it through a model and it can predict what the next token is going to be, so this is called an auto regressive model like predicting on itself. But statistical program tend to give you bad intuition.

Reference:

*What are neural networks?* *IBM*. Available at: https://www.ibm.com/topics/neural-networks (Accessed: 19 August 2023).

*What is natural language processing?* (no date) *IBM*. Available at: https://www.ibm.com/topics/natural-language-processing#:~:text=Natural%20language%20processing%20(NLP)%20refers,same%20way%20human%20beings%20can. (Accessed: 19 August 2023).

*NLP tutorial*. YouTube. Available at: https://www.youtube.com/playlist?list=PLeo1K3hjS3uuvuAXhYjV2lMEShq2UYSwX (Accessed: 19 August 2023).

*LLM bootcamp By The Full Stack*. YouTube. Available at: https://www.youtube.com/playlist?list=PL1T8fO7ArWleyIqOy37OVXsP4hFXymdOZ (Accessed: 19 August 2023).

(2023). YouTube. 18 August. Available at: https://www.youtube.com/watch?v=xZDB1naRUlk (Accessed: 19 August 2023).

Shakkari, K. (2023) *Introduction to machine reading comprehension*, *Medium*. Available at: https://medium.com/analytics-vidhya/open-domain-question-answering-series-part-1-introduction-to-reading-comprehension-question-1898c8c9560e (Accessed: 19 August 2023).

Heyer, S. (2023) *LLM prompt engineering*, *Medium*. Available at: https://medium.com/google-cloud/generative-ai-best-practices-for-llm-prompt-engineering-2a0131c805cc#:~:text=Prompt%20Engineering%20refers%20to%20crafting,ChatGPT%20to%20perform%20desired%20tasks. (Accessed: 19 August 2023).

Ben Dickson *et al.* (2023) *LLM Fine-Tuning*, *TechTalks*. Available at: https://bdtechtalks.com/2023/07/10/llm-fine-tuning/ (Accessed: 19 August 2023).