

LLMs Under the Hood

An attempt to understand how LLMs work

Neural Networks and Transformer Architecture

Before diving into LLMs, it may be beneficial to learn about neural networks and an architecture for neural networks called 'Transformer Architecture'.

Neural networks are computational architectures that mimic biological neural networks, consisting of layers of interconnected nodes that process input data through weighted connections and non-linear activation functions to perform tasks. They adapt and learn by optimizing these weights using gradient descent during training.

Neural Networks and Transformer Architecture

The transformer architecture, introduced in the seminal paper "Attention Is All You Need", represents a groundbreaking approach in the field of natural language processing (NLP) and has since become the foundation for a wide range of state-of-the-art models.

In simpler terms, the transformer architecture is a way of organizing the neural networks. It utilizes self-attention mechanism to compute the output. There are two very important parts of this architecture, namely, encoder and the decoder.

Encoder is responsible for converting the input text into what is called embeddings. Decoder is responsible for converting embeddings to output text.

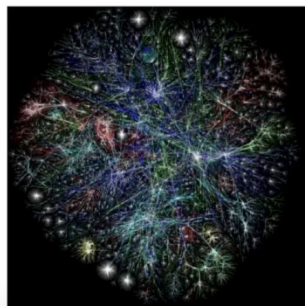
Back to LLMs

Large language models are an example of the transformer architecture. For simplicity, the LLM can be thought of as a neural network with a huge number of parameters.

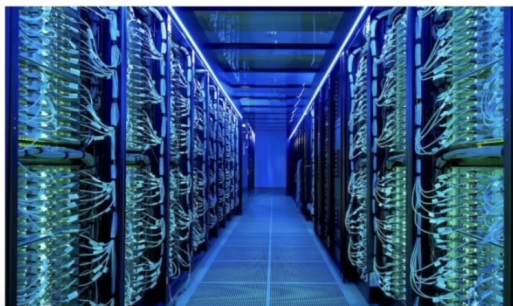
The construction of the parameters of the LLM can be split into two parts:
Pre-Training and Fine-Tuning.

Pre-Training

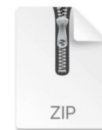
In the pre-training stage, the neural network of the LLM is trained on a huge amount of text (~10TB) to get the initial parameters.



Chunk of the internet,
~10TB of text



6,000 GPUs for 12 days, ~\$2M
~1e24 FLOPS



parameters.zip

~140GB file

*numbers for Llama 2 70B

Pre-training

After pre-training, a base-LLM is created with certain parameters. At this stage, the LLM is not very useful since it is only trained on unstructured and low quality data. For it to take on the form of an assistant, another step called fine-tuning is needed.

Fine-Tuning

Fine-tuning is performed on the base LLM model. The training step is almost identical except for the fact that dataset is swapped.

The new dataset contains high quality data that is in the form of a question and answer.

After this step, the parameters are finalized and the LLM can be used as an assistant.

Fine-Tuning

<USER>

Can you write a short introduction about the relevance of the term "monopsony" in economics? Please use examples related to potential monopsonies in the labour market and cite relevant research.

<ASSISTANT>

"Monopsony" refers to a market structure where there is only one buyer for a particular good or service. In economics, this term is particularly relevant in the labor market, where a monopsony employer has significant power over the wages and working conditions of their employees. The presence of a monopsony can result in lower wages and reduced employment opportunities for workers, as the employer has little incentive to increase wages or provide better working conditions...

~100K conversations

LLM Inference

Now that the LLM training process is done. LLM can be used for inference.

The constructed LLM will be used to predict the next word in a given sequence.

