## horizontal line

LLMs

31.05.2025

# Epic History of LLMs

LLMs basically used Seq2Seq type RNN i.e. Many to Many RNNs Architecture that are Asynchronous (of variable length).

Seq2Seq tasks used in various applications like Machine translation , Q/A , Chatbots , Speech-to-text , Chatgpt etc.



## Stage 1 : Encoder Decoder Architecture

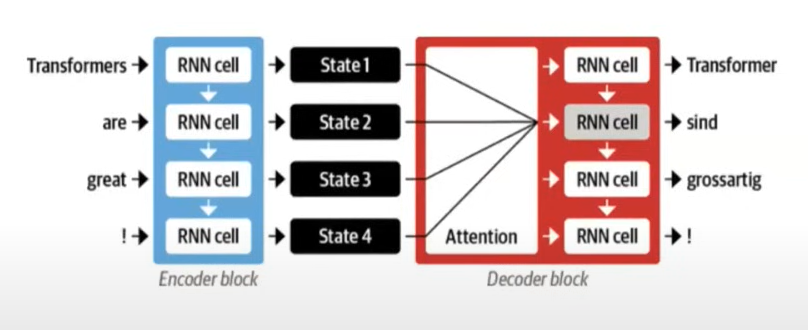
Encoder as well as Decoder both consist of RNN/ LSTM / GRU.

At each time stamp word by word input is provided where each arrow at the encoder represents the updated Ct / Ht at each time stamp .

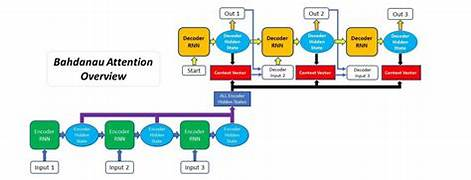
At last time stamp the final Ht/Ct are provided as compressed input (State) and at the decoder end step by step output is generated.

Problem with this architecture is Long term Dependency (Vanishing Gradient Problem).

## Stage 2 : Attention mechanism

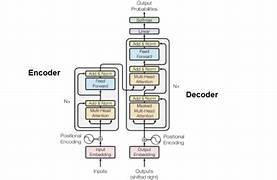


Attention Mechanism also follows Encoder Decoder but the difference is the output generated at Decoder can use Ht/Ct of all time stamps and finds out which context is important for it to predict correctly. This removes the issue of Long term Dependency , all words keep their semantic meaning.

Here we have provided an attention layer that identifies which context (Hts and Cts) are useful for the particular output.

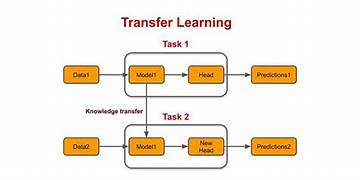
Problem with this is it has LSTM that takes input in sequential order word by word at each time stamp would lead to (nXm) computational complexity.

## Stage 3 : Transformers

Transformer is ideally solving the problem just by using Attention Mechanism instead of RNN / LSTM for parallel processing.

But the main problem is from where to get such a huge amount of data to train the model.

## Stage 4 : Transfer Learning

Provide Pretrained models that can be used to get outputs on any new test data. 

E.g. Sentiment Analysis cannot predict NER.

NLP previously do not use Transfer learning because :

1. Problem with it is that a model trained on one type of data cannot be used to predict output on any other new data.
2. Then it is decided to use Language models i.e. make model an understanding of language so that it can do all tasks related to Sentiment Analysis , NER , POS , Q/A , Speech to text , Machine translation.But to make a model like this we need a lot of labeled data e.g. {Eng : correctly translated Hindi} , not possible to gather.

That time ULMFit Research paper came and introduced language modelling i.e. to train models on Next Word Predictor . This gives the model the overall understanding of all NLP tasks. Language modelling succeeded because :

1. Rich feature learning
2. Huge availability of data (data all over internet).

## Stage 5 : LLMs

In 2018 based on transformers , we will have BERT (Encoder only) from Google and GPT(Decoder only) from OpenAI. These both models are very good at transfer learning and can perform all NLP tasks.

GPT —--> GPT2 —--> GPT3 —--> GPT4

Qualities of LLMs :

* Data - billions e.g. GPT3 - 45TBs of data
* Hardware - Clusters of GPU e.g GPT3 trained on Supercomputer , 1000s NVIDIA GPUs
* Training - days to weeks
* Cost - millions
* Energy Consumption - a small town light up a month

# ChatGPT

ChatGPT is an application made on the GPT model.

How is ChatGPT made ?

1. RLHF (Reinforcement Learning from Human Feedback)

* Labeled data on human conversation
* Ranking of prompt input responses by experts.

1. Incorporate safety and ethical issues
2. Improvement in contextual points. Maintain context for dialogue conversation.
3. Dialogue specific training that’s why it is Chat–GPT.
4. Continuous improvement by human feedbacks.