**Transformers**

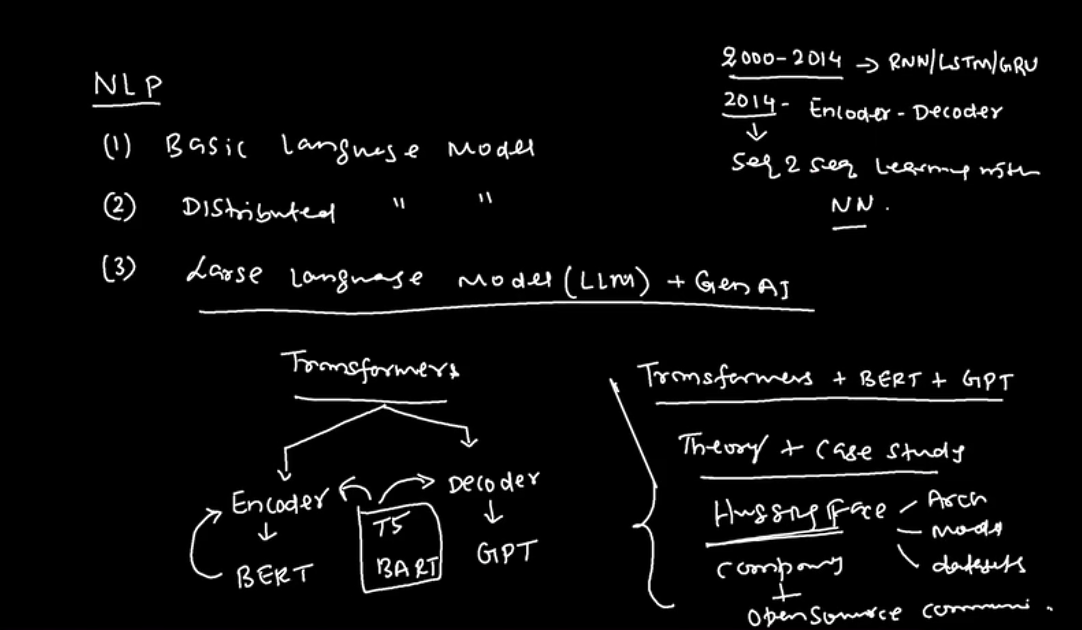
Transformers has two parts:

Encoder – developed by BERT (by Google)

Decoder – developed by GPT (by OpenAI)

In the middle we have T5 (BART) which focuses both on encoder and decoder unlike BERT and GPT which focuses on encoder and decoder respectively.

We can solve tabular dataset, image dataset and sequential dataset problems by the help of Transformer model (which is also a neural network) but mainly we use it for Seq 2 Seq data or sequential data also RL problems.



When we talk about transformers, the Self-Attention Model comes into the picture. Transformers model makes uses of Parallel Processing and hence are very fast unlike algorithms like RNN which process the data one by one.

Note:- Encoder means to understand language and decoder means to Generate text.

In encoder-decoder architecture (Sequence to sequence model), whenever we see the ‘Start’ written, it should be clear that the decoder operation has started and then terminated.

Inside the architecture of sequence to sequence models, they use LSTMs which generates the consolidated output and pas it to the other cell or entity. However, the biggest challenge with LSTM is that they are generate good result with only small vocab size (around 30) beyond that its performance starts to fall.

The second research paper published in 2015 by Bahdanau. The paper was called Neural machine translation. This was the foundation of the Attention Model.

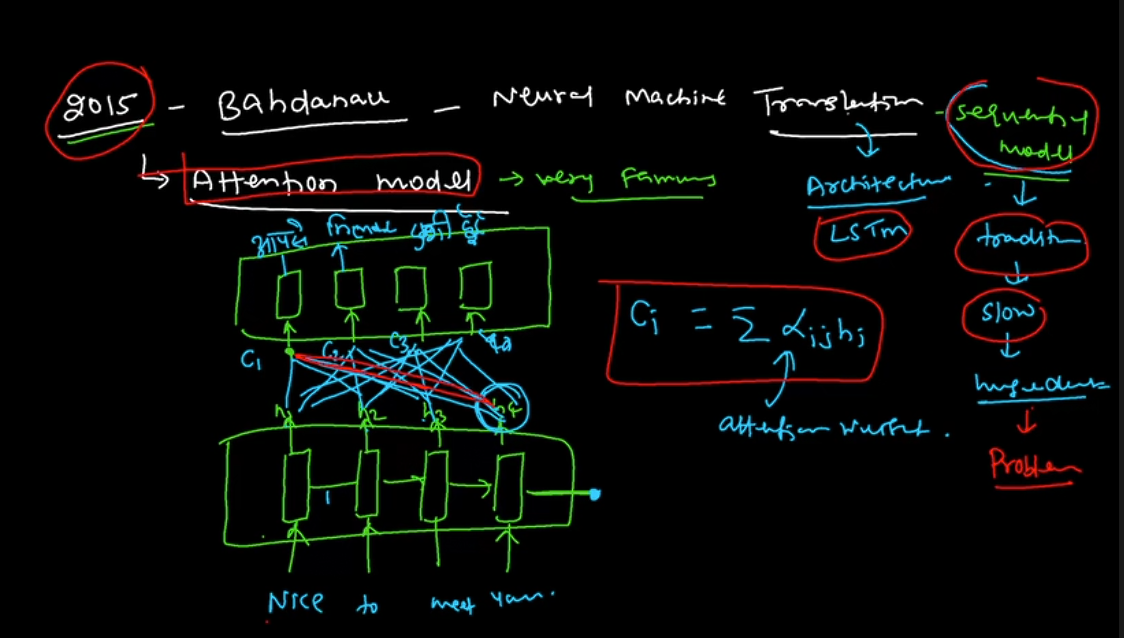
This person also gave the concept of Attention model which also came in 2015.

The encoder-decoder model and Attention models are both Sequential models and they both use LSTMs as their internal architecture. But the main difference is that in encoder-decoder model, the encoder part generates a summary which gets fed to the decoder part to generate the output.

However, in Attention model, it makes use of the dense neural network which is connected to each neurons of the encoder and decoder cells and it generate the output basis the attention weight which is calculate by them.

Here, the output is not dependent on the last cell output but is dependent on each cell as they all are connected and final decoding is done basis the maximum weight generated by these hidden layers. Hence, new weights also called as Attention weights get calculated which determines the significance of passed inputs

This Attention model is what we will be using in the Transformer model.



Now, in 2017 came Transformer (V.V. Significant topic) with a popular research paper called – Attention is all you need.

What is Model?

Relationship captured between input and output. Model is used to predict the output of the unseen input

What is Language Model?

Model that captures the sequential relationship between words and sentences of a language is called Language Model.

How do we build a language model?

When we have a huge amount of data. We have to build a powerful architecture. This architecture is nothing but Language Modelling Technique which is called a Transformers.

What are the Language Modelling Techniques?

Auto Regressive Task (Uni-directional model) – Here, we predict the output word by word (Predicting one token at a time). It is not commonly used.

Auto Encoder Task (Bi-directional model) – Generate the entire token together or the entire target sequence together.

Auto regressive task aims to maximize the likelihood of the next token given the previous token.

Auto encoder task aims to minimize the reconstruction error.

In a summary, in 2014, we have learnt a model called Encoder Decoder architecture.

In 2015, we got to learn Attention Mechanism by Bahdanau.

In 2017, we have a paper published on the Transformer Architecture(AIAYN). This is a universal model

In 2018, came LLMs like fine-tuning model, etc.,

Attention is All you need:- this paper proposed the Transformer Architecture which used the Encoder Decoder architecture.

They introduced the term called ‘Self-Attention mechanism’.

They also introduced Positional Encoding words

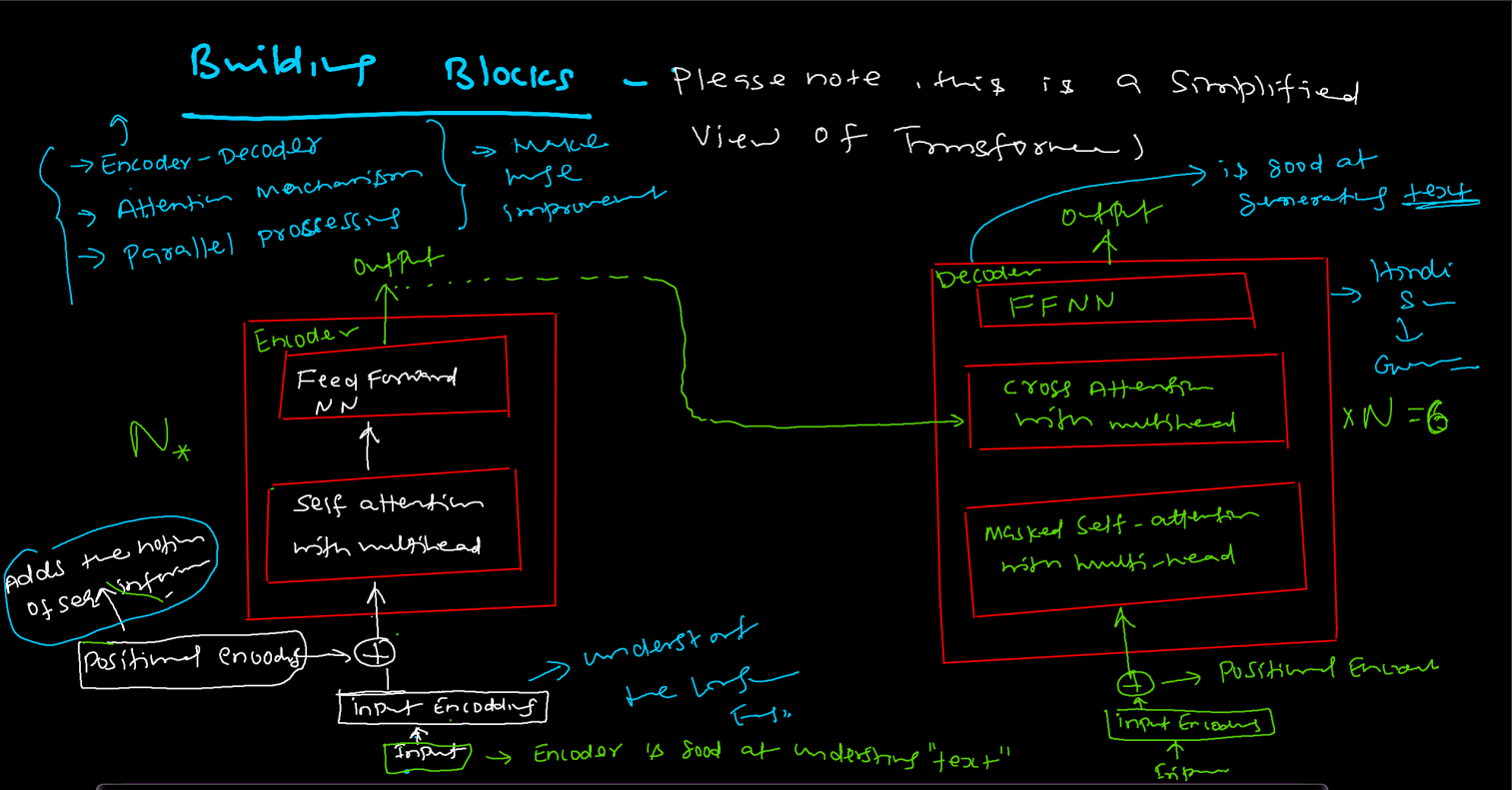
They have also used Multi-head self-attention model (self-attention model)

They also used Cross Attention using Multi-head.

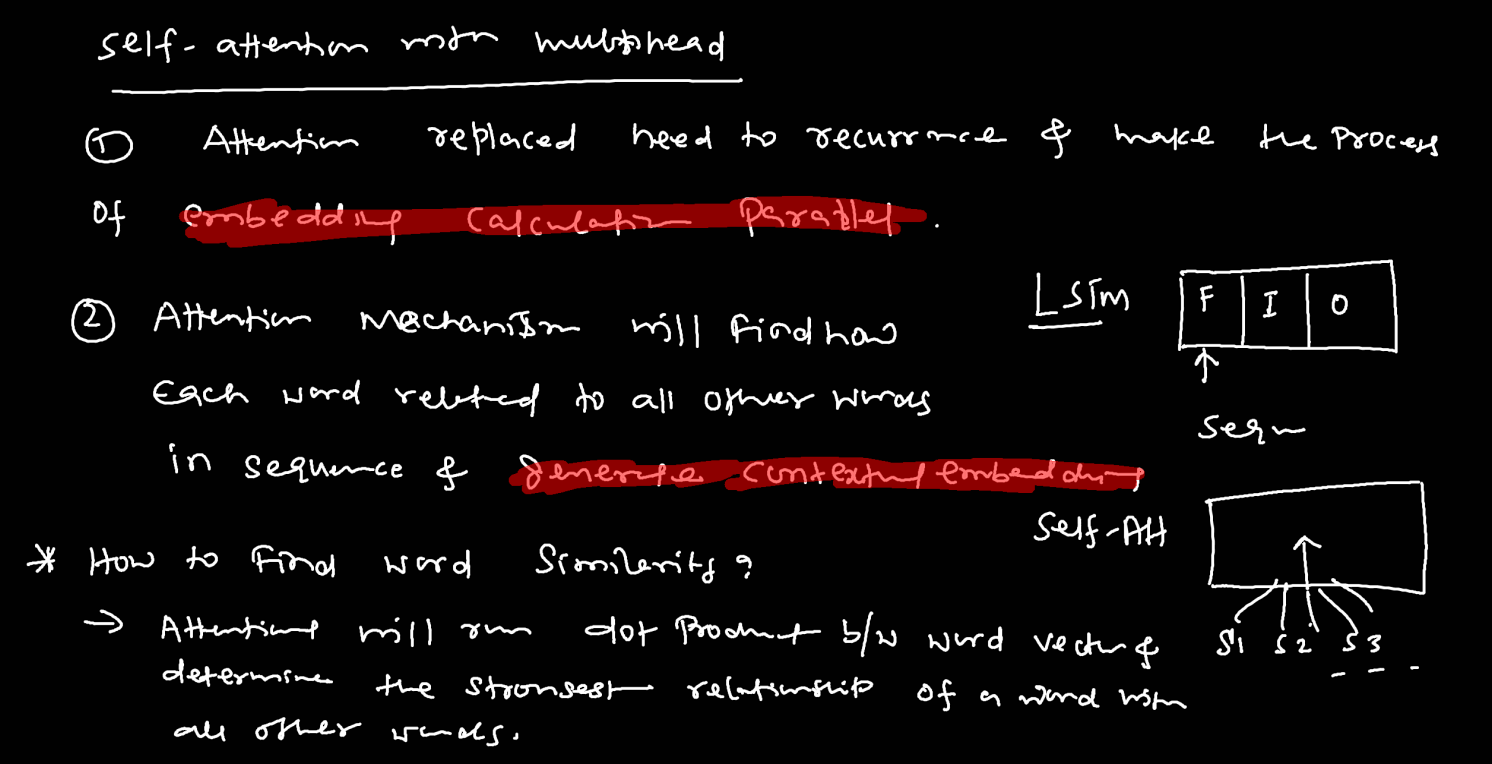
**Building Blocks of the Transformer Model (this is a simplified view of the transformer):-**

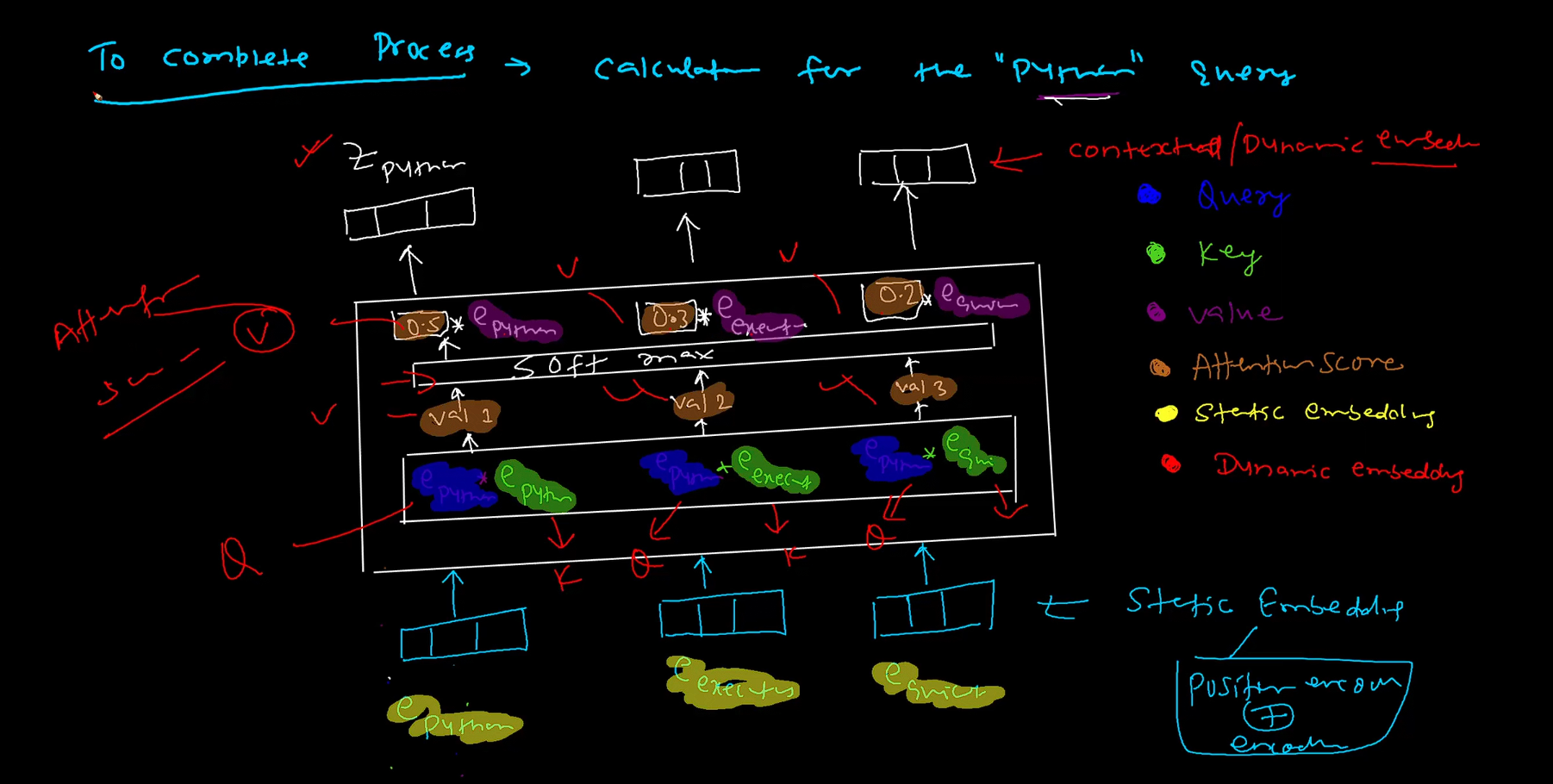
Positional encoding:- Adds the notion of sequence information to the Input encoding. It also means giving the contextual meaning of the sentence by reading the whole sentence (feature of transformer model that it also see the context of the words)

Note:- the transformer is also an encoding decoding architecture. It also has Attention Mechanism. It is not a sequential model but makes use of Parallel processing hence very fast.



Self Attention with Multi-head:-

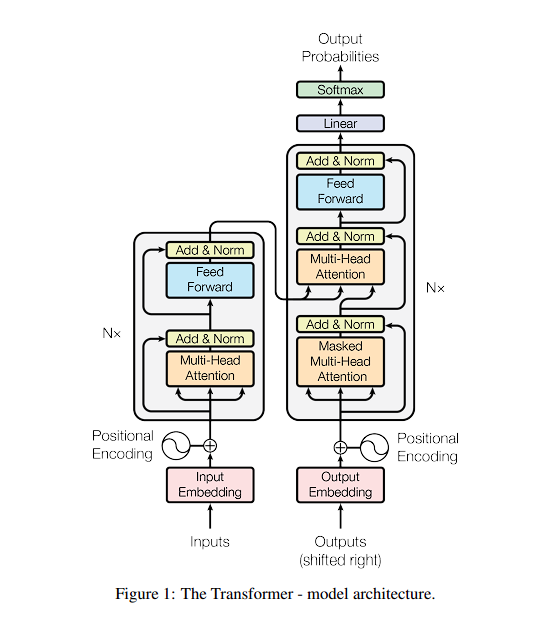




The encoder used the auto encoder method (Output is generated for entire sequence) to generate the output which is to be fed to the decoder, while the decoder use the auto regressive (Output is generate word by word) method while generating the output.

[**https://chatgpt.com/c/676c3ac9-087c-800d-a486-5ea41977b6ad**](https://chatgpt.com/c/676c3ac9-087c-800d-a486-5ea41977b6ad)

[**https://chatgpt.com/c/67701e6c-3ba8-800d-88a4-ad67af4d8fe2**](https://chatgpt.com/c/67701e6c-3ba8-800d-88a4-ad67af4d8fe2)



Very very very important questions for Interview purpose:-  
  
What is Q, k and v (Query, Key, Value)?

Q (Query) = What information the model is trying to extract.

Key = Represents the contents or features the model can pay attention to. We either do Dot product or Cosine Similarity on the features.

Value or Attention Score = Represents the actual information that gets passed along once the attention has been calculated (Dot product between Query and Key and then applying the Softmax Activation)

Why Query, Key and Value?

If we apply token embedding directly without transforming into Q, k and V, the model will be limited to just one fixed way of computing relationship between tokens.

Suppose in a text we are talking about a river bank and RBI bank then, without Q, K and V the embedding for bank will be same for both sentences say (0.5). Hence, the model will treat them as a same word, which is not the case, hence Positional encoding comes into the picture.