Experiment No. : Group 1 - ④

- **Title :**

Transformer.

- **Problem statement :**

Create a transformer from scratch using the pytorc library.

- **Objective :**

To create transformer.

- **Course outcome :**

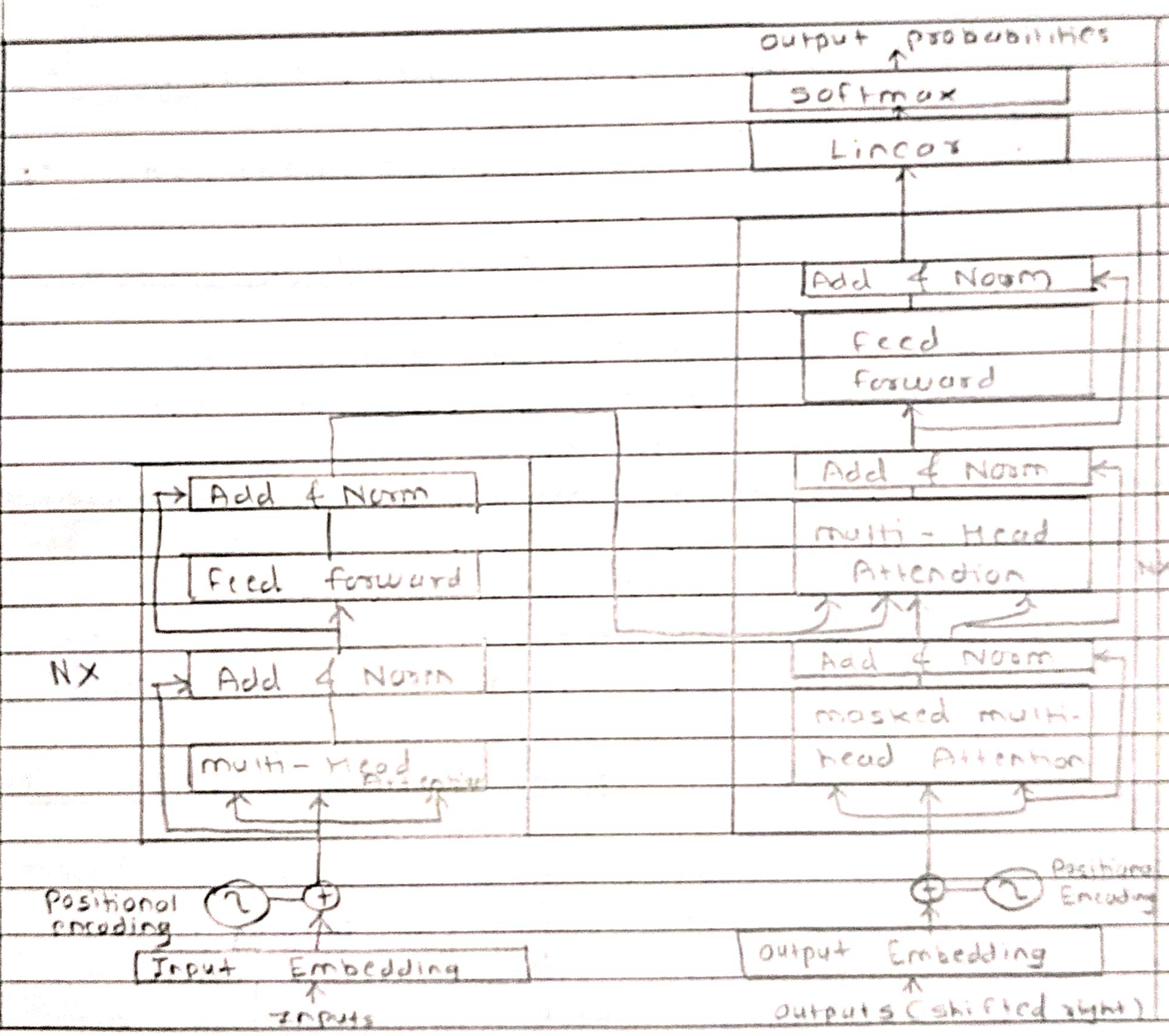
CO3 : Design and develop applications on subjects of their choice.

- **Softwares and hardware requirements :**

Sr. no.	Softwares / Hardwares	Specifications
1.	Laptop / Desktop	64-bits, 8 GB RAM. windows OS.
2.	Jupyter notebook	Version 7.3.3.

Theory:

The diagram is shown over view of the model of the Transformer model. Their inputs to the encoder will be the english sentence, and the 'output' entering the decoder will be the French sentence. In effect, their are five processes we need to understand to implement this model



There are five processes we need to understand to implement this model.

- Embedding the inputs
- The positional Encodings
- creating Masks
- The multi-Head Attention layer.
- The feed-forword layer.

1. Embedding the inputs:

Embedding words has become standard practice in NMT, feeding the Network with far more information about words than a one hot encodin would. when each word is fed into the networ this code will perform a look-up and retrieve its embedding vector. These vectors will then be learnt asaparameters bythe model, adjusted with each iteration of gradient descent

2. The positional Encodings:

The positional encoding matrix is a constant whose values are defined by the above eq^{n}. when added to the ~~m~~ embedding matrix. each word embedding is altered in a way specific to its position.

3. creating our masks :

masking plays an important role in the transformer. it serve purpuse. the initial input into the decoder will be the target sequence (the french translation). the way the decuder predict each output word is by making use of all the encoder outputs and the french sentence only up until the point of each word its predicting.

4. multi-headed Attention :

once we have our embedded values (with pusitional encodings) and our masks. we can start building the layers of our model. Here is an overview of multi-headed attaint on layer.

5. The Feed forward Network:

This layer just consists of two linear operations, with are drop out operation between them The feed-forward layers simply deepen souonnetwork. employing linearly layer to analysis patterns in the attention layers output.

Conclusion :

In this practical, I have build a transformer from scratch using pytorch demonstration the core principles of attending mechanisms and deep learning model architecture for the sequen modelling.