What Are Transformers?

Transformers were first developed to solve the problem of sequence transduction, or neural machine translation, which means they are meant to solve any task that transforms an input sequence to an output sequence. This is why they are called “Transformers”.

But let’s start from the beginning.

What Are Transformer Models?

A transformer model is a neural network that learns the context of sequential data and generates new data out of it.

To put it simply:

> *A transformer is a type of artificial intelligence model that learns to understand and generate human-like text by analyzing patterns in large amounts of text data.*

Transformers are a current state-of-the-art NLP model and are considered the evolution of the encoder-decoder architecture. However, while the encoder-decoder architecture relies mainly on Recurrent Neural Networks (RNNs) to extract sequential information, Transformers completely lack this recurrency.

So, how do they do it?

They are specifically designed to comprehend context and meaning by analyzing the relationship between different elements, and they rely almost entirely on a mathematical technique called attention to do so.

TRANSFORMER
INPUT
OUTPUT

The Transformer Architecture

Overview

Originally devised for sequence transduction or neural machine translation, transformers excel in converting input sequences into output sequences. It is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution. The main core characteristic of the Transformers architecture is that they maintain the encoder-decoder model.

If we start considering a Transformer for language translation as a simple black box, it woul take a sentence in one language, English for instance, as an input and output its translation in English.

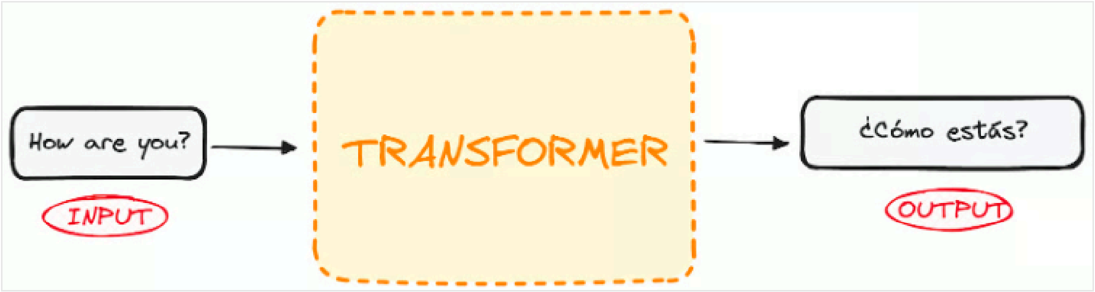
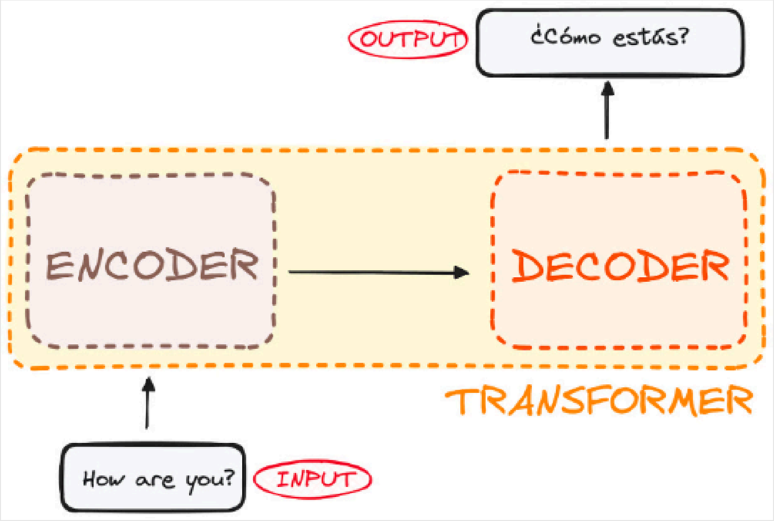


Image by the author.

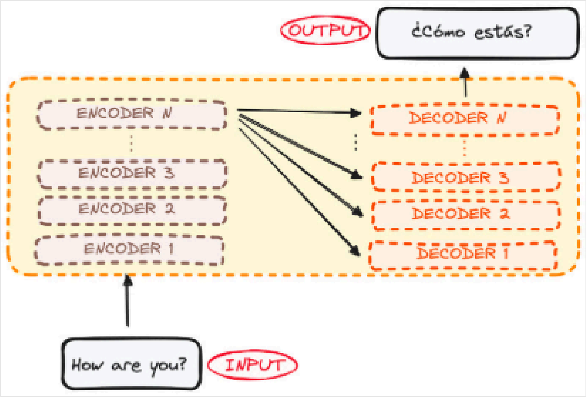
If we dive a little bit, we observe that this black box is composed of two main parts:

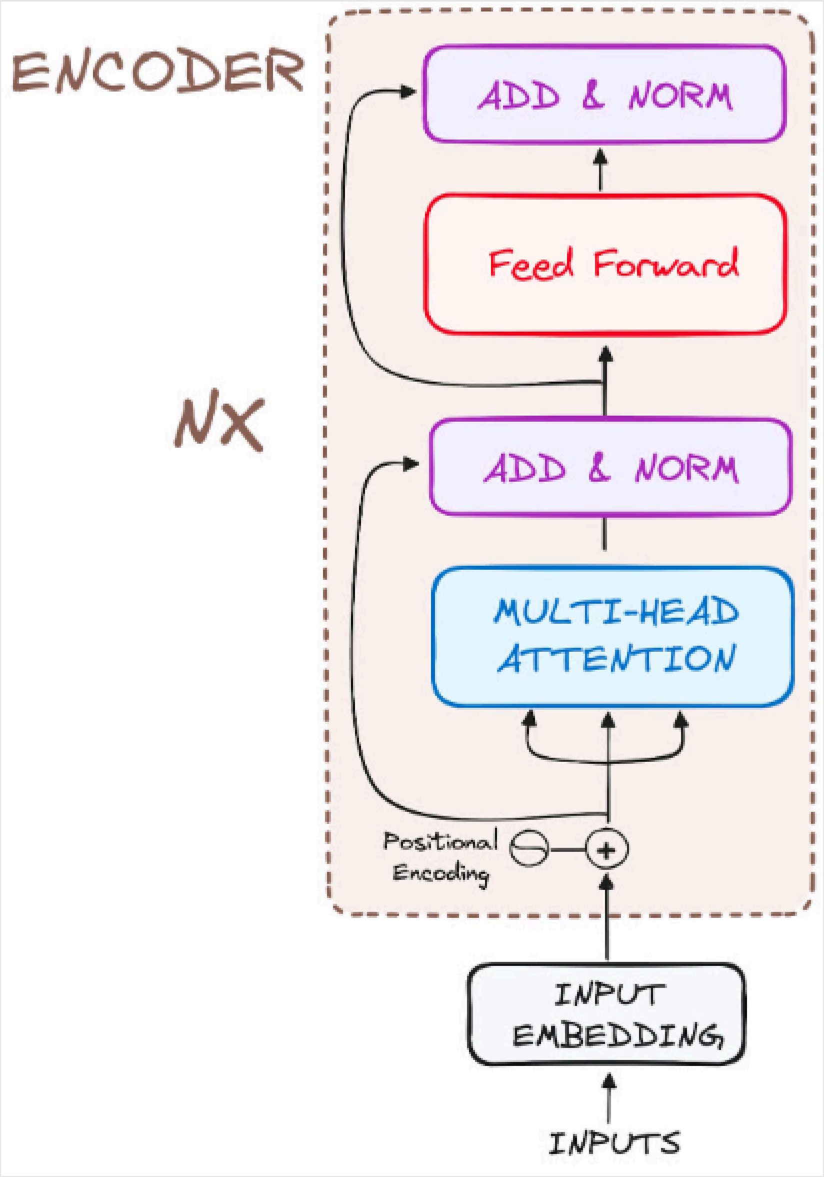
- The encoder takes in our input and outputs a matrix representation of that input. For instance, the English sentence “How are you?”
- The decoder takes in that encoded representation and iteratively generates an output. In our example, the translated sentence “¿Cómo estás?”

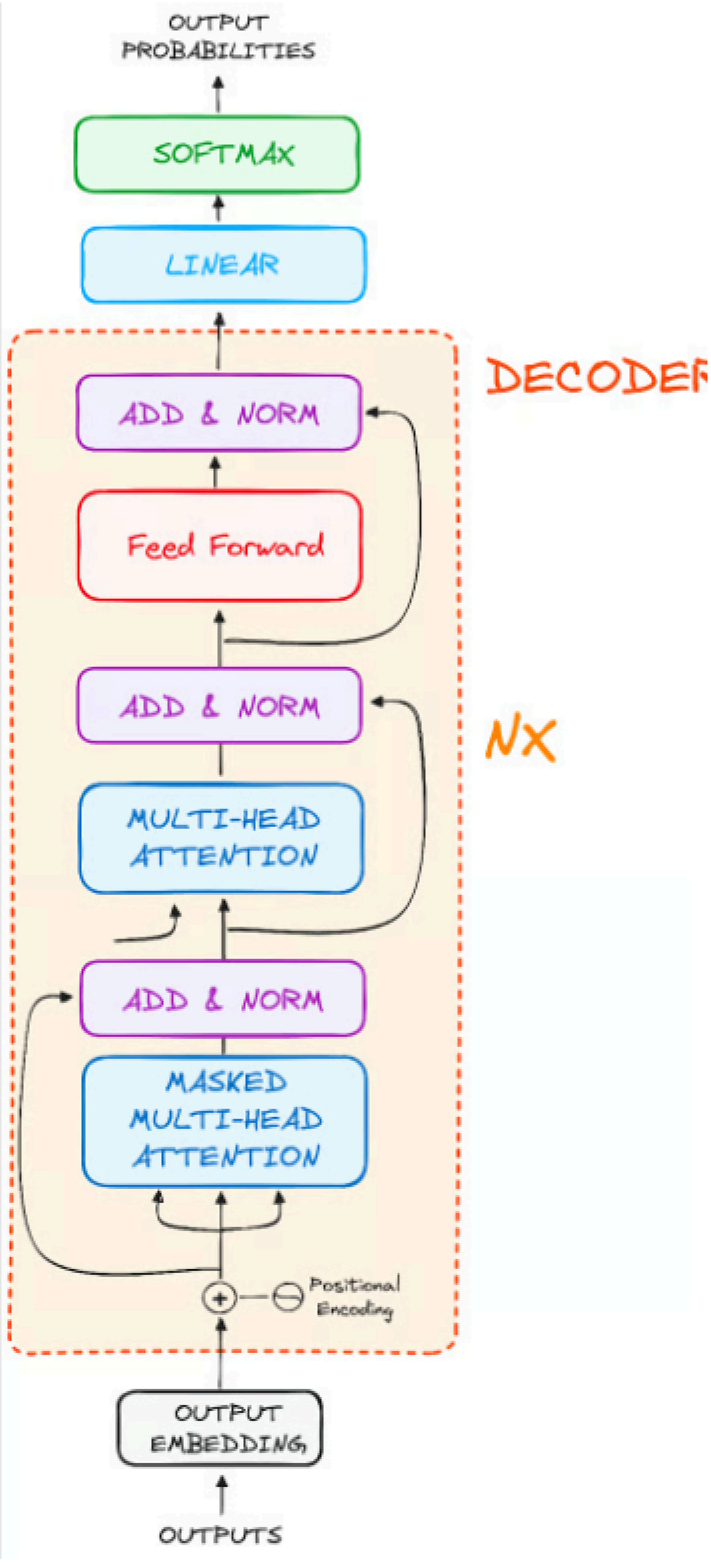


However, both the encoder and the decoder are actually a stack with multiple layers (same number for each). All encoders present the same structure, and the input gets into each of them and is passed to the next one. All decoders present the same structure as well and get the input from the last encoder and the previous decoder.

The original architecture consisted of 6 encoders and 6 decoders, but we can replicate as many layers as we want. So let’s assume N layers of each.




ENCODER
NX
ADD & NORM
Feed Forward
ADD & NORM
MULTI-HEAD ATTENTION
Positional Encoding
INPUT EMBEDDING
INPUTS


OUTPUT
PROBABILITIES
SOFTMAX
LINEAR
DECODER
ADD & NORM
Feed Forward
ADD & NORM
NX
MULTI-HEAD
ATTENTION
ADD & NORM
MASKED
MULTI-HEAD
ATTENTION
Positional
Encoding
OUTPUT
EMBEDDING
OUTPUTS