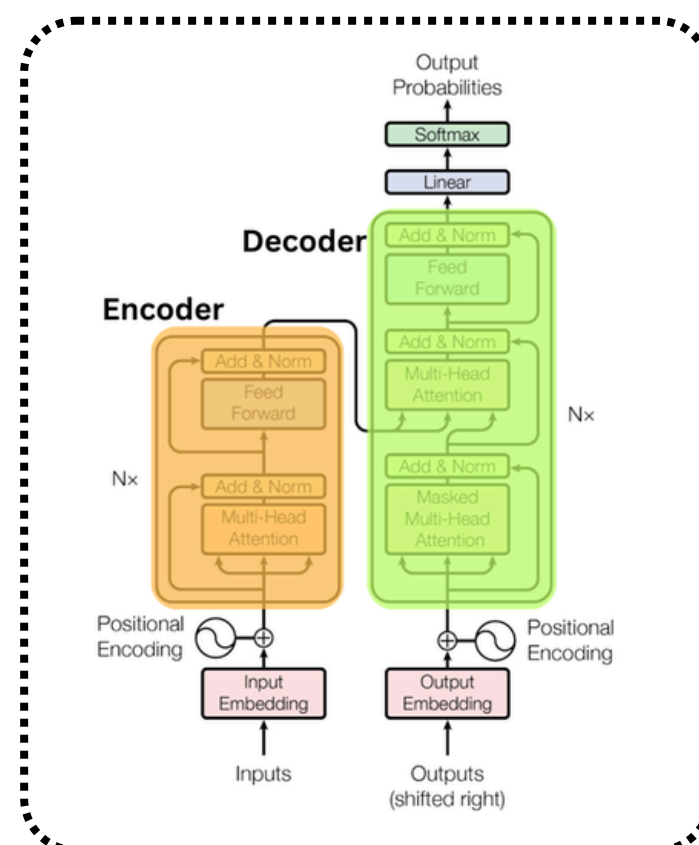
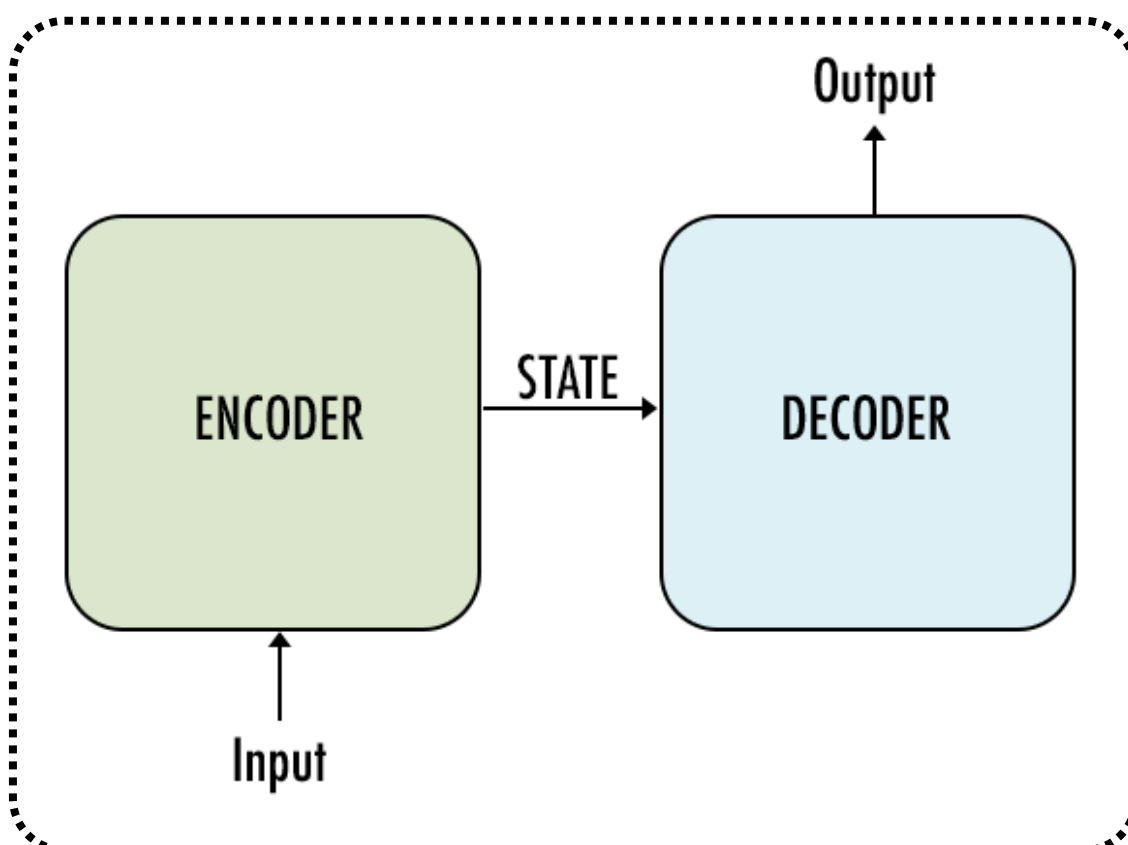
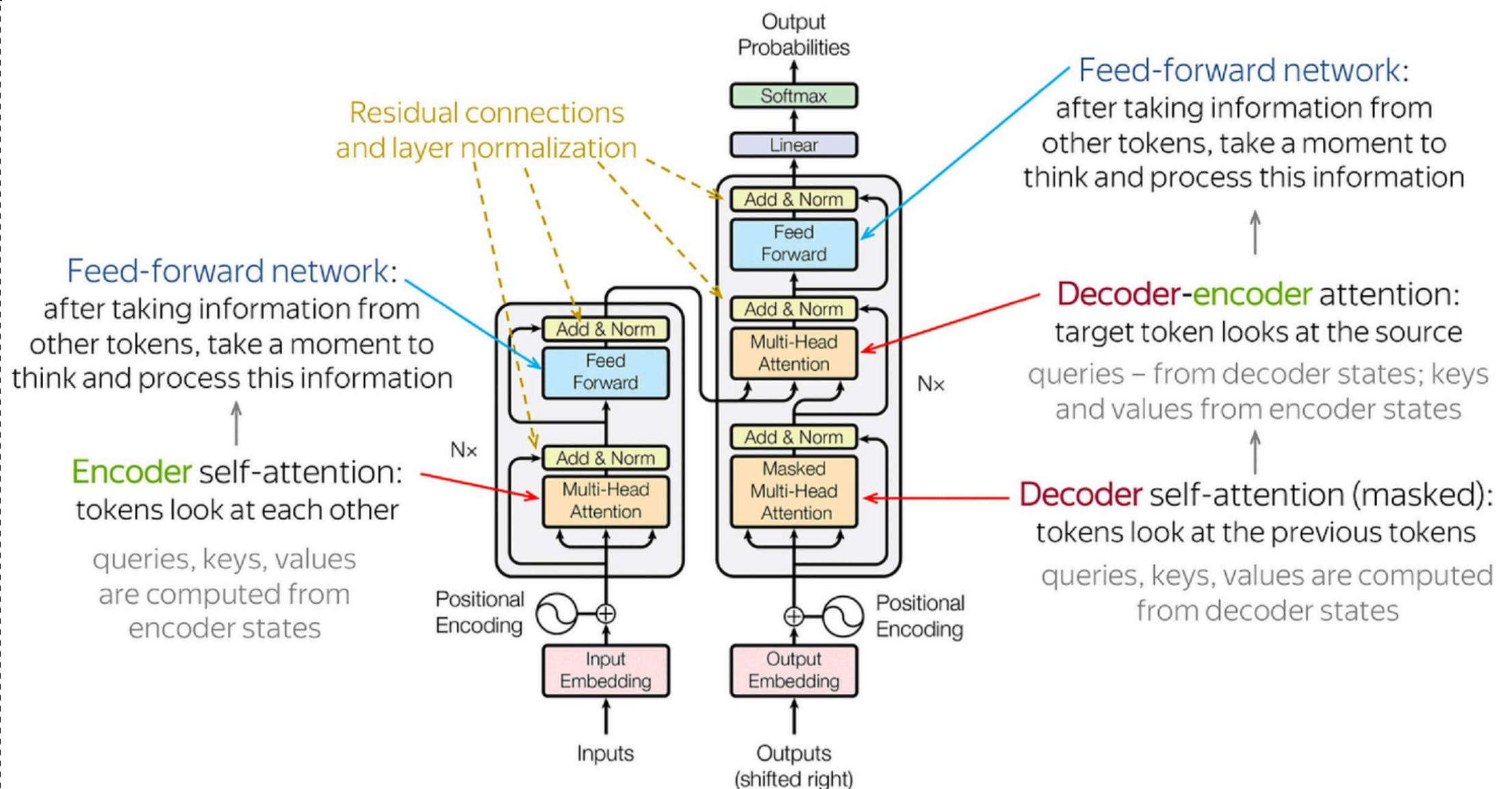
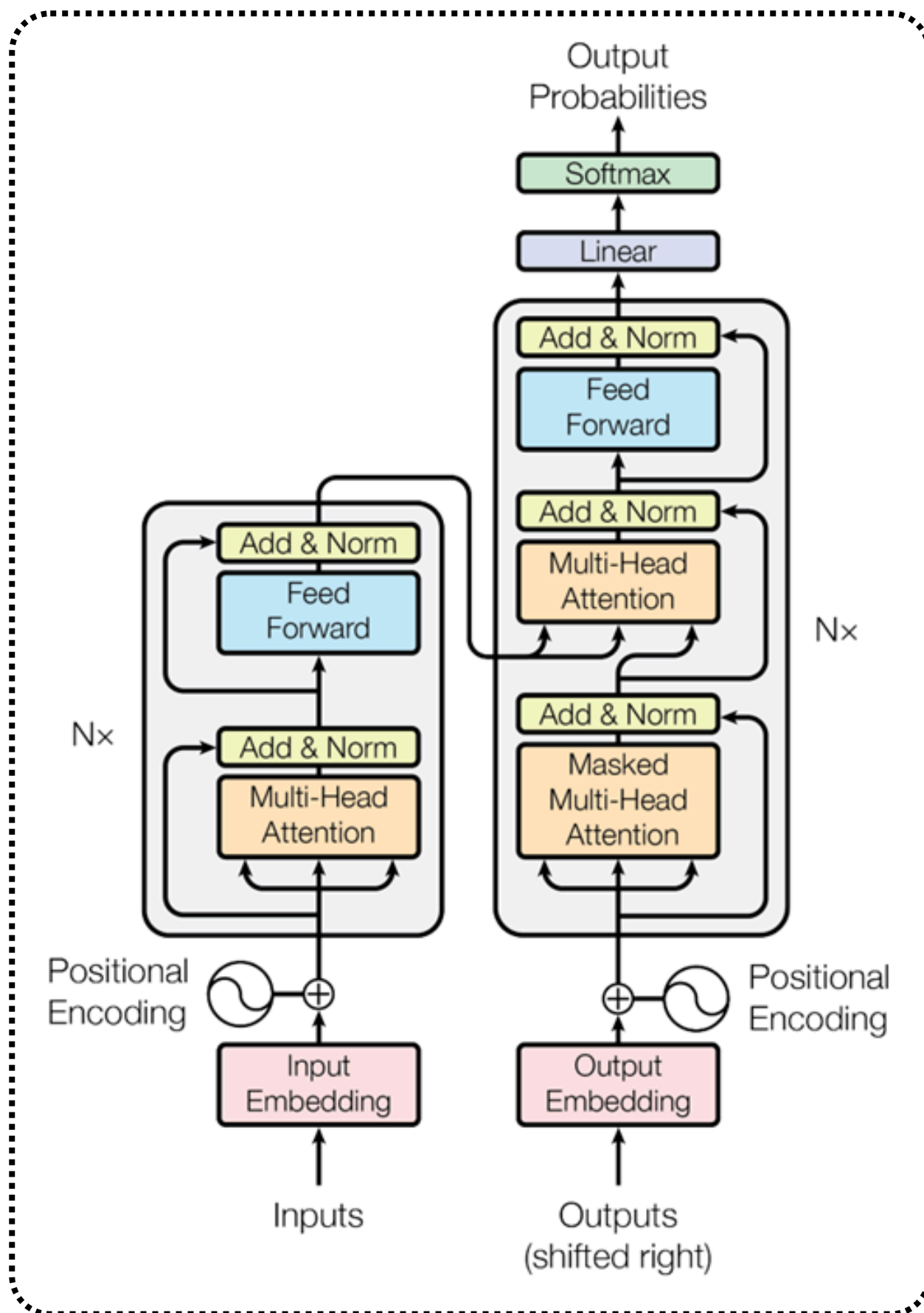


Mastering LLMs

Day 2: Transformers and The Attention Mechanism



In 2017, “**The Attention Is All You Need**” paper introduced the ground breaking **Transformer model**.



The Transformer model was introduced to address key **limitations** of **traditional architectures** in **handling sequential data**, particularly in natural language processing (NLP) tasks.

Here's why it was introduced →

Limitations of RNNs and CNNs

Recurrent Neural Networks (RNNs)

- **Sequential Bottleneck:** RNNs process input sequentially, leading to slow training and difficulty parallelizing computation.
- **Vanishing Gradient Problem:** They struggle to capture long-range dependencies effectively due to gradient decay during backpropagation.
- **Limited Context Awareness:** Standard RNNs have difficulty accessing distant context in sequences.

Convolutional Neural Networks (CNNs)

- **Fixed Context Size:** While CNNs are better suited for parallelism, their receptive fields are limited, making them less effective for long-range dependencies.
- **Task-Specific Design:** They are more naturally suited to image data and require significant adaptation for sequential tasks.

Transformers overcame this issue with the **Self-Attention Mechanism**

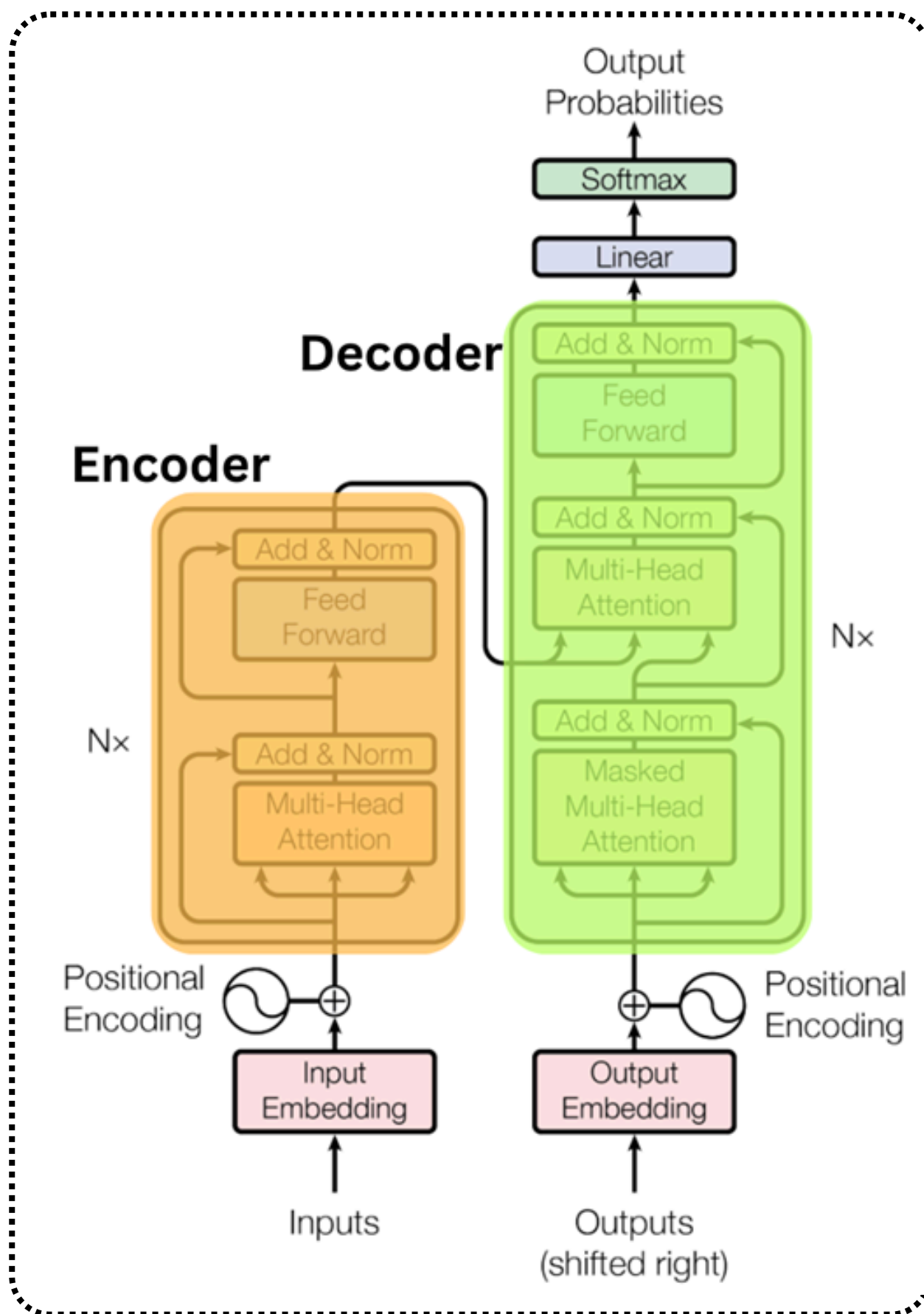
- Models relationships between all tokens in a sequence at once, regardless of their distance.
- Captures both short- and long-range dependencies effectively.
- Handles context flexibly, allowing it to prioritize important parts of the sequence.

Transformers process multiple tasks simultaneously using **Parallelization**

The Transformer architecture, based on self-attention mechanisms, allows for the entire sequence to be processed simultaneously. This is a significant improvement over **RNNs**, which process tokens sequentially.

Let's take a high level view of

Transformer Architecture




In the Transformer architecture, the **Encoder** and **Decoder** are the two main components that work together to handle tasks like language translation, summarization, and more.

Interaction Between Encoder & Decoder

- Encoder Processes Input
 - Converts the input sequence into a sequence of contextualized embeddings.
- Decoder Generates Output
 - Uses these embeddings along with self-attention to predict the output sequence token by token.
- Shared Knowledge
 - The Encoder provides a holistic understanding of the input, while the Decoder leverages this understanding to produce coherent outputs.

Attention Mechanism

- The attention mechanism is a foundational component of the Transformer architecture, enabling the model to focus on the most relevant parts of the input sequence when processing data. It addresses the challenge of capturing long-range dependencies and relationships within data efficiently.
- The attention mechanism allows a model to dynamically assign importance (or weights) to different parts of an input sequence when processing a specific token.
- The **attention mechanism** comes in various types, each designed to handle specific tasks or improve performance in different scenarios. These types can broadly be classified based on their structure, purpose, or data processing methods. Here's an overview of the different types of attention mechanisms. 

Self-Attention

- **Definition:** The attention mechanism where a sequence attends to itself, allowing each token to consider all other tokens in the same sequence.
- **Usage:** Widely used in Transformers (e.g., BERT, GPT).
- **Example:** In a sentence like “The cat sat on the mat,” the word “cat” attends to other words in the same sentence, such as “sat” and “mat,” to understand its role.

Multi-Head Attention

- **Definition:** Splits the queries, keys, and values into multiple subspaces, computes attention in each subspace independently, and combines the results.
- **Usage:** Widely used in Transformers to capture diverse relationships.
- **Advantage:** Enhances the model’s ability to focus on different aspects of the data simultaneously.

Encoder-Decoder Attention

- **Definition:** Enables the decoder to focus on relevant parts of the input sequence processed by the encoder.
- **Usage:** Crucial for tasks like machine translation, where the target sequence depends heavily on the input sequence.

January

2025

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
			1	2	3	4
5	6 ✓	7 ✓	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Stay Tuned for **Day 3** of

Mastering LLMs