Self-Attention Mechanism, Multi-Head Attention, Transformer

Introduction

This document serves as a primer to help you understand the foundational components that make up the **Transformer** architecture such as the **Self-Attention Mechanism**, **Multi-Head Attention**, etc. These topics are at the heart of many recent advancements in AI, particularly in NLP models like **BERT**, **GPT**, and **T5** models.

- **BERT**: Bidirectional Encoder Representations from Transformers
- **GPT**: Generative Pre-trained Transformer
- **T5**: Text-to-Text Transfer Transformer

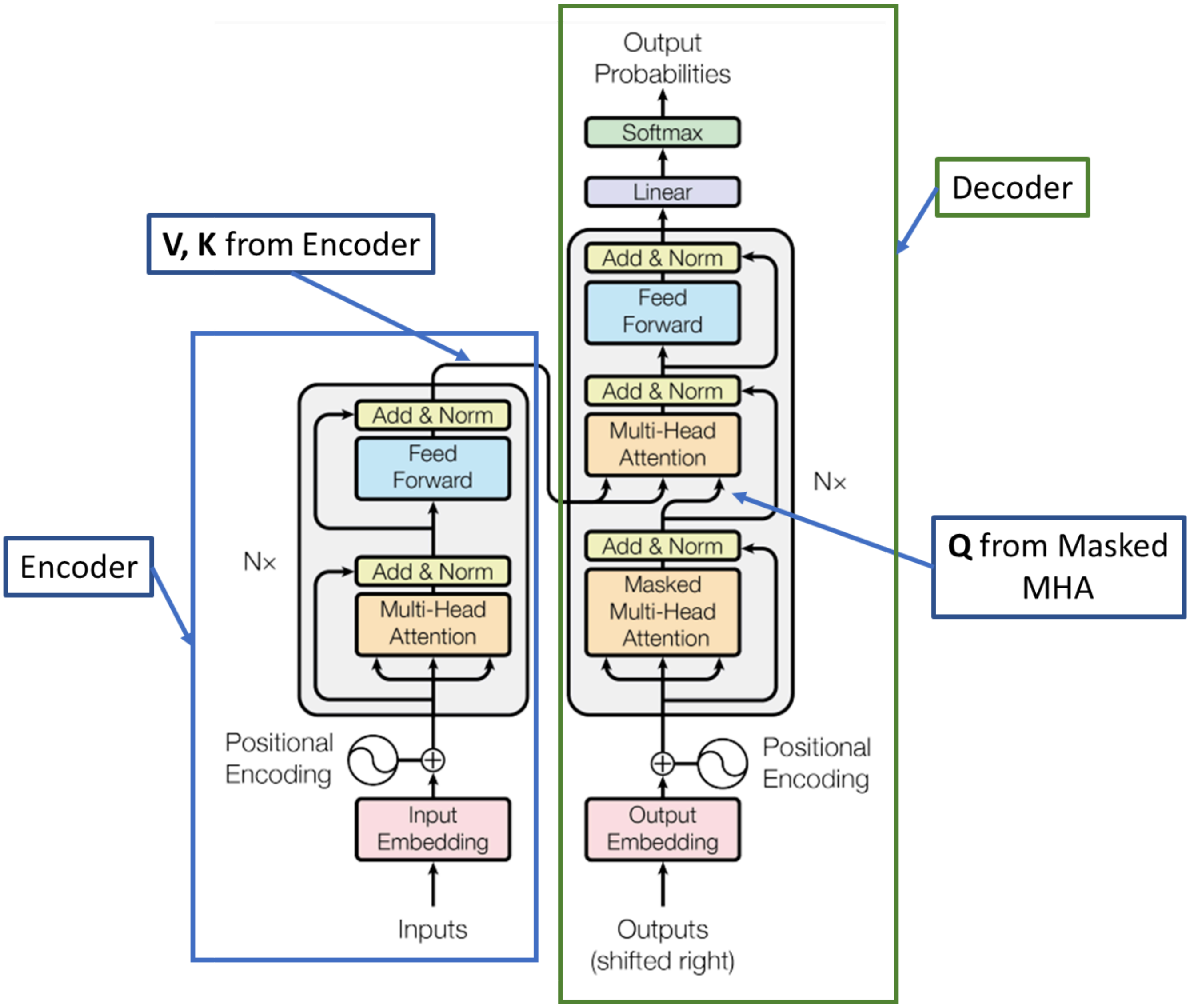
Key Concepts to Review:

1. What is a Transformer?

The Transformer model, introduced by **Vaswani et al.** in the 2017 paper *"Attention is All You Need"*, revolutionized the way we approach sequence modeling in machine learning. Unlike previous sequence models (like RNNs and LSTMs), the Transformer relies entirely on **self-attention mechanisms** and does not require sequential data processing, making it highly parallelizable and efficient.

The Transformer architecture consists of two main components:

- **Encoder:** Takes the input data (e.g., sentences) and transforms it into a set of representations.
- **Decoder:** Generates the output (e.g., translated sentence) using the encoder's representations.



Transformer model Architecture

2. Self-Attention Mechanism:

Self-attention allows the model to weigh the importance of different words in a sequence relative to one another. In essence, it helps the model determine which words to focus on when processing a particular word in the sequence. The key idea is that ***each word interacts with all other words in the sequence, and attention scores are computed to highlight these relationships***.

3. How does Self-Attention Work?

Self-attention operates in three steps:

- **Query (Q):** Represents the word we are currently focusing on.

- **Key (K):** Represents every other word in the sequence, to assess how much attention we should give them.
- **Value (V):** Encodes the information to be passed forward.

The attention score between words is computed by comparing the Query and Key vectors, followed by a weighted sum of the Value vectors.

4. Mathematics Behind Self-Attention:

$$\text{Attention}(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- d_k: dimension of K

Self-attention can be computed through the following steps:

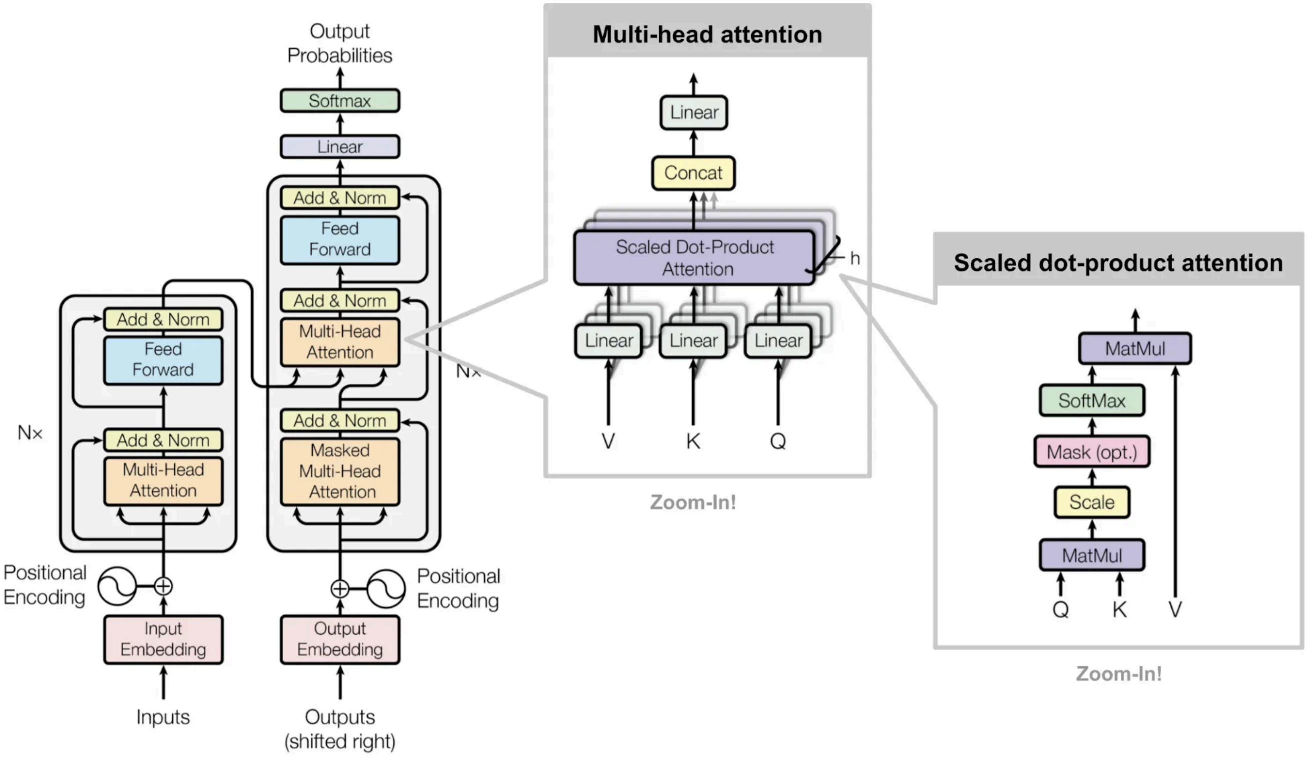
- Compute **attention scores** by taking the dot product of Query (Q) and Key (K) vectors.
- Scale these scores to avoid excessively large values by dividing by the square root of the dimension of the Key vector.
- Apply a **softmax** function to normalize the scores.
- Multiply these normalized scores by the Value (V) vector to generate the weighted sum.

This process is performed for each word in the sequence, allowing every word to "attend" to every other word.

5. Multi-Head Attention (MHA):

While self-attention computes the relationship between a word and other words in the sequence, Multi-Head Attention enhances this mechanism by allowing the model to focus on different aspects of the sequence simultaneously. Instead of using a single set of Query, Key, and Value vectors, Multi-Head Attention uses several sets (called "heads") to capture various relationships between words.

- **Why Multi-Head Attention?** Multiple heads allow the model to learn different representations of the input sequence at different positions in the input, leading to better performance in tasks like translation, question answering, and text generation.



6. Positional Encoding:

Since the Transformer model processes all words simultaneously (in parallel), it lacks the notion of word order, which is inherent in sequential models like RNNs. To address this, **Positional Encoding** is added to the input embeddings to inject information about the relative or absolute position of words in the sequence.

Position encodings are typically added to the input embeddings before they enter the encoder and decoder.

7. Why are Transformers Important?

Transformers have drastically improved the performance of models across several NLP tasks, such as:

- **Machine translation** (e.g., translating text from one language to another),
- **Text generation** (e.g., creating human-like text),
- **Sentiment analysis**, **Named Entity Recognition (NER)**, and more.

The success of models like **BERT** and **GPT-3** is built on the Transformer architecture, which enables them to handle vast amounts of data efficiently and achieve state-of-the-art results on many tasks.

Resources for Further Reading:

1. **"Attention Is All You Need"** (Vaswani et al., 2017) – The original paper introducing the Transformer model.
    - Link to paper
2. **The Illustrated Transformer** by Jay Alammar – A visual and intuitive explanation of Transformers.
    - Link to article