1 Paper Walkthrough: Attention is All You Need

In this discussion, we will walk through the paper that introduced transformers, which can be found at this link. The goal is to improve your conceptual understanding of the attention mechanism and transformer model, while also giving practice in breaking down and digesting academic computer science papers, which is an important skill.

We will be using a three-pass approach to reading the paper, as suggested here. These three passes progress from a high-level overview of the main ideas to a low-level deep-dive of the implementation details.

(a) **Pass 1:** Read the title, abstract, introduction, and conclusion sections. Discuss to make sure you know the answers to the following questions:

i. What are the main benefits of the transformer model as it's presented in the paper?

ii. What task is the transformer created to solve? What is it being tested on?

iii. What is the paper's main contribution? What makes it unique compared to previous work?

(b) **Pass 2:** Now, we'll dive deeper into the specifics of the ideas presented. The second pass involves taking note of key figures, equations, and results, as well as finding other papers to read in the background/relevant work section.

i. Let's take a closer look at Equation (1):

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}})\mathbf{V} \tag{1}$$

1. Given an input $\mathbf{X} \in \mathbb{R}^{T \times d_{model}}$ representing a sequence of T inputs stacked as rows, how do we compute $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ corresponding to the input sequence?
What are the shapes of $\mathbf{Q}, \mathbf{K}, \mathbf{V}$? Let d_v be the value dimension and let d_k be the key and query dimension.
2. Which part of this equation is calculating the attention weights?
3. Why do we perform *scaled* dot-product attention: why do we divide by $\sqrt{d_k}$ within the softmax?

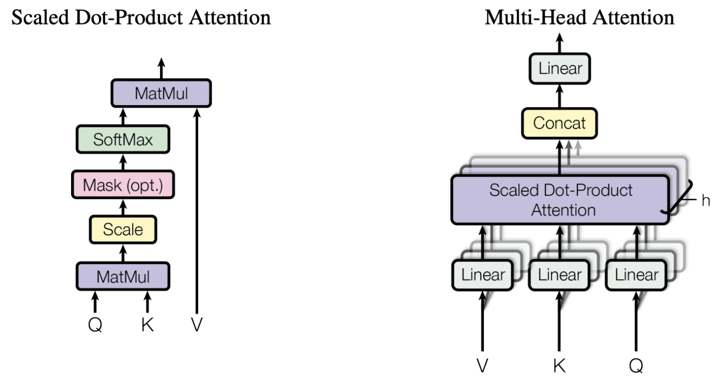

Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

ii. Next, let's examine how to add multiple attention heads:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$$
$$\text{where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

1. Show how we can achieve this multi-head attention by splitting our input into h chunks and passing them into separate self-attention mechanisms.
2. What's the benefit of adding more attention heads in parallel?

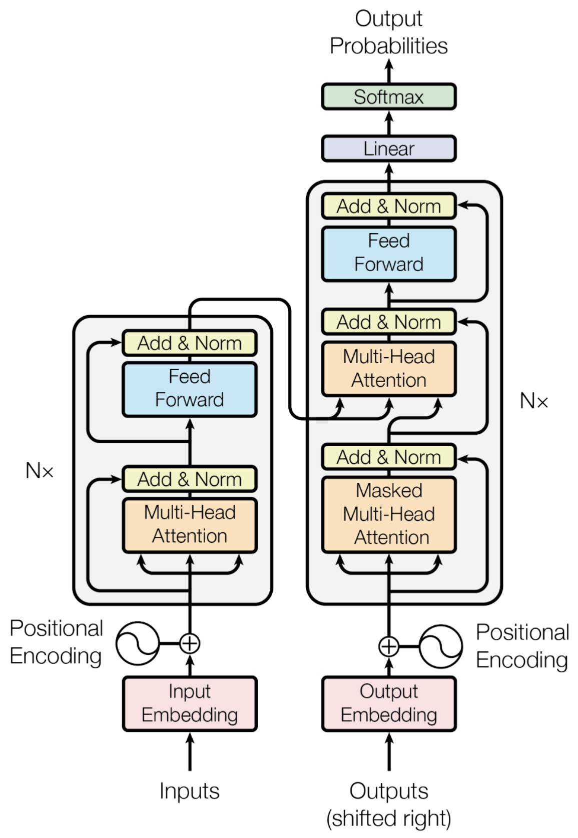

Figure 1: The Transformer - model architecture.

iii. The figure above shows the transformer encoder-decoder architecture. The encoder's job is to condense a source text input (for example, in German) into a real-valued representation which gets fed into the decoder to generate a desired output sequence (for example, an English translation).

   1. How exactly does the encoder's representation get used in the decoder?
   2. Why does the decoder include a *masked* multi-head attention unit?

iv. A few other details addressed in sections 3.3-3.5:

   1. What is the reason for positional encoding? How is this typically implemented?
   2. How is this position-wise feedforward network placed between self-attention layers different from a regular feedforward layer?
   3. For a machine translation task, how is the source text string fed into the encoder (or target text into the decoder)?
   4. Lastly, take a look at the results in section 6. How does the transformer model compare against the baseline models? What tasks and metrics are used?

(c) **Pass 3:** The last step is to implement the paper to try and reproduce the results. You will do this in the homework!