Lecture Notes on Transformer

Hong QIn

March 28, 2025

The transformer architecture splits into an encoder and a decoder, and while both use the scaled dot-product attention mechanism, they differ in how and where this mechanism is applied.

Scaled Dot-Product Attention (Common to Both)

At the core of both components is the scaled dot-product attention defined as:

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

where:

- Q (queries),
- K (keys),
- V (values), and
- d_k is the dimensionality of the keys.

Encoder

In the **encoder**, every layer performs self-attention on the input sequence. Here, the queries, keys, and values are all derived from the same input x:

$$Q = xW^Q, \quad K = xW^K, \quad V = xW^V.$$

Thus, the encoder's self-attention is computed as:

$$\text{SelfAttention}_{\text{enc}}(x) = \text{softmax}\left(\frac{(xW^Q)(xW^K)^T}{\sqrt{d_k}}\right)(xW^V).$$

This mechanism allows each token in the input to attend to all other tokens, integrating contextual information across the entire sequence.

Decoder

The **decoder** is more complex because it has to generate an output sequence while incorporating information from the encoder. It uses two main attention mechanisms:

1. Masked Self-Attention

The decoder first applies self-attention to its own previous outputs. To maintain the autoregressive property (i.e., ensuring a token only depends on earlier tokens), a mask M is applied. This mask typically sets the upper triangular part of the attention matrix to a very negative value, so that the softmax zeroes out any attention weights corresponding to future tokens. Formally:

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

Here, Q, K, and V are derived from the decoder's own input (previously generated tokens).

2. Encoder-Decoder (Cross) Attention

After the masked self-attention, the decoder incorporates information from the encoder. In this step, the queries come from the decoder (from the output of the masked self-attention), while the keys and values come from the encoder's final output. The formula is:

$$\text{EncDecAttention}(Q_{\text{dec}}, K_{\text{enc}}, V_{\text{enc}}) = \text{softmax}\left(\frac{Q_{\text{dec}}K_{\text{enc}}^T}{\sqrt{d_k}}\right)V_{\text{enc}}.$$

This step allows the decoder to "look" at the input sequence and incorporate context from the encoder into the output generation.

Summary

- **Input Sources:**
  - **Encoder:** Uses self-attention where Q, K, and V are all derived from the same input x.
  - **Decoder:** Uses two attention mechanisms: masked self-attention on its own output (with a mask M) and cross-attention that uses the encoder's outputs for K and V.
- **Masking:**
  - **Encoder:** No masking is necessary; all tokens attend to each other.
  - **Decoder:** Uses a mask in the self-attention to prevent future tokens from being attended to, preserving the autoregressive property.
- **Attention Layers:**
  - **Encoder:** A single self-attention layer per encoder block.
  - **Decoder:** Two sequential attention layers (masked self-attention followed by encoder-decoder attention) in each decoder block.

These differences in the attention formulas are key to enabling the decoder to generate coherent output sequences while leveraging the complete context provided by the encoder.