Humayra Rahman Nipa
Id- 2121128042

Attention in Transformers: Concepts and Code

Early Translation Methods:

- Early translation methods faced issues with word order and sentence length.
- They used a single vector to represent whole sentences, which wasn't effective.
- The introduction of the attention mechanism improved translation quality.
    - It allows words to be considered individually, maintaining their meanings.
    - Helps in better understanding the context, making translations more accurate.
    - Contextual embeddings show how word meanings can change depending on surrounding words.

ChatGPT and the Transformer Architecture:

- ChatGPT is an AI model that answers questions and engages in conversations, based on transformer architecture.
    - It involves three main parts: word embedding, positional encoding, and attention.
    - Word embedding turns words into numbers for neural networks.
    - Positional encoding helps track the word order in sentences.
    - Attention connects words in a sentence, helping to understand their relationships.

Self-Attention Mechanism:

- The self-attention mechanism uses three components: Query (Q), Key (K), and Value (V).
    - Query (Q): What you're searching for.
    - Key (K): What you're searching against.
    - Value (V): The result of the search.

- Similarity between queries and keys is calculated, and the softmax function is applied to turn these into probabilities.
- The model computes a weighted sum of the values, focusing on the most relevant keys.

Implementation of Self-Attention in Code:

- Libraries: PyTorch is imported to work with tensors and neural network modules.
- SelfAttention Class: Defines the model's architecture and initializes the dimensions of the model.
- Weight Matrices: Used to create the queries, keys, and values for each token.
- Forward Method: Computes the attention values by applying weight matrices and calculating the similarities between tokens.

Masking in Self-Attention:

- Masking modifies the self-attention mechanism to prevent a token from attending to tokens that come after it in a sequence (used in text generation).
- For example, during generation, a word like "Write" can only attend to itself and previous tokens.

Encoder-Only and Decoder-Only Transformers:

- Encoder-Only Transformers: Used for understanding and generating contextual embeddings, effective in tasks like classification.
- Decoder-Only Transformers: Use masked self-attention to predict the next token, ideal for text generation tasks.
- Cross-Attention: A method combining both encoder and decoder features, still relevant in multimodal models.

Multi-Head Attention:

- Multi-head attention involves multiple attention heads operating independently, allowing the model to capture various relationships between words in a sentence.
- The outputs from all attention heads are combined to form a comprehensive representation.
- This technique enhances the model’s ability to understand complex relationships in data.

Attention Mechanisms in PyTorch:

- The `Attention` class is created to implement the attention mechanism.
- The `MultiHeadAttention` class incorporates multiple attention heads, each focusing on different aspects of the input.

Conclusion:

- Self-Attention: Excellent for capturing long-range dependencies but computationally expensive for large sequences.
- Massive Self-Attention: Optimized for longer sequences using sparse attention or kernel methods.
- Encoder-Decoder Attention: Best for aligning input and output sequences, such as in machine translation.
- Researchers must choose the appropriate attention mechanism based on task requirements.