Question 1
What is self-attention in a Transformer? What problem does it address? [no more than half a page] [2 points]

Solution
Self-attention in a Transformer is a mechanism that allows the model to weigh the importance of different words in a sequence against each other. It computes a representation for each word based on the other words in the sequence, enabling the model to capture dependencies and relationships within the input sequence.

The problem self-attention addresses is the challenge of efficiently capturing long-range dependencies in sequences, such as those found in natural language. Traditional recurrent neural networks (RNNs) and convolutional neural networks (CNNs) struggle with processing long sequences due to issues like vanishing gradients in RNNs and fixed-size receptive fields in CNNs. These limitations hinder their ability to effectively model relationships between distant words in a sequence.

Self-attention allows a Transformer model to attend to different positions in the input sequence to varying extents, depending on the relevance of each position to the current word being processed. This attention mechanism helps the Transformer capture contextual information effectively across long distances in the input, leading to improved performance in tasks like language modeling, machine translation, and other sequence-to-sequence tasks.

Overall, self-attention in Transformers addresses the need for a more robust and scalable method of capturing contextual relationships in sequences, which is essential for understanding and generating natural language with high accuracy and efficiency.

Question 2
Give an example for encoder-only Transformer, decoder-only Transformer, and encoder-decoder Transformer. [2 points]

Solution

1) Encoder-Only Transformer (e.g., BERT):

   Example:
   BERT (Bidirectional Encoder Representations from Transformers)

   Description:
   BERT is a pre-trained model that uses an encoder-only Transformer architecture. It consists of a stack of transformer encoder layers. BERT is designed for tasks like masked language modeling and next sentence prediction, where the model learns bidirectional representations of input sequences. The encoder processes the entire input

sequence independently to capture contextual embeddings for each token.

2) Decoder-Only Transformer (e.g., GPT):

   Example:
   GPT (Generative Pre-trained Transformer)

   Description:
   GPT is a language model based on a decoder-only Transformer architecture. It uses a stack of transformer decoder layers. GPT is designed for autoregressive language modeling tasks, where the model generates one token at a time based on previously generated tokens. The decoder attends to the entire context up to the current token to generate the next token in the sequence.

3) Encoder-Decoder Transformer (e.g., Transformer for Machine Translation):

   Example: Transformer for Machine Translation (e.g., Google's Neural Machine Translation model)

   Description:
   This type of Transformer architecture combines both encoder and decoder components. It is commonly used for sequence-to-sequence tasks such as machine translation. The encoder processes the input sequence and generates a contextual representation, while the decoder uses this representation to generate an output sequence. The encoder-decoder attention mechanism allows the model to focus on relevant parts of the input sequence during decoding, enabling effective translation between languages.