MIDDLE EAST TECHNICAL UNIVERSITY

SEMESTER I EXAMINATION 2024-2025

CENG 403 – Deep Learning - Transformers and Attention Mechanisms

January 2025 TIME ALLOWED: 3 HOURS

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains **SEVEN (7)** questions and comprises **TEN (10)** printed pages.
2. Answer all questions. The marks for each question are indicated at the beginning of each question.
3. Answer each question beginning on a **FRESH** page of the answer book.
4. This **IS NOT an OPEN BOOK** exam.
5. Calculators are allowed for numerical computations.
6. Show all mathematical derivations and computational steps clearly.
7. For matrix operations, clearly indicate dimensions and show intermediate steps.
8. Write pseudocode clearly with proper indentation and comments.
9. Draw clear diagrams where requested and label all components.

Question 1. Attention Mechanism Fundamentals (20 marks)
Based on Stanford CS224n and MIT 6.390 course materials, answer the following about attention mechanisms.

(a) Explain the four key components of an attention mechanism and describe how they interact to process sequential data. Include a discussion of how attention differs from traditional encoder-decoder approaches. (8 marks)

(b) Derive the mathematical formulation for scaled dot-product attention. Given query $Q \in \mathbb{R}^{n \times d_k}$, key $K \in \mathbb{R}^{m \times d_k}$, and value $V \in \mathbb{R}^{m \times d_v}$ matrices, show that: (12 marks)

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

Explain why scaling by $\frac{1}{\sqrt{d_k}}$ is crucial for gradient stability and provide the mathematical reasoning behind this choice.

Question 2. Computational Complexity Analysis (18 marks)
Compare the computational complexities of different sequence modeling approaches.

(a) Complete the complexity analysis table below for processing sequences of length n with dimensionality d: (10 marks)

Operation	**Self-Attention**	**Recurrent**	**Convolutional**
Sequential Operations	?	$O(n)$	?
Maximum Path Length	?	$O(n)$	?
Computational Complexity	$O(n^2 \cdot d)$	?	?

(b) For self-attention with sequence length $n = 1000$ and embedding dimension $d = 512$, calculate: (8 marks)

- Total number of attention parameters needed
- Memory complexity for storing attention weights
- Number of floating-point operations for one forward pass

Question 3. Multi-Head Attention Implementation (25 marks)
Based on UvA Deep Learning tutorial materials, implement and analyze multi-head attention.

(a) Write pseudocode for a multi-head attention layer that handles variable sequence lengths and optional masking. Your implementation should include: (15 marks)

- Input projection to multiple heads
- Parallel attention computation
- Output concatenation and projection
- Support for causal masking

```
# Your pseudocode here
function MultiHeadAttention(X, num_heads, mask=None):
    // Complete this implementation









    return output
```

(b) Prove that self-attention mechanisms are permutation-equivariant. That is, show that if P is a permutation matrix, then: (10 marks)

$$\text{SelfAttention}(PX) = P \cdot \text{SelfAttention}(X)$$

Question 4. Positional Encoding Design (22 marks)

Positional encoding allows transformers to understand sequence order without inherent positional bias.

(a) Design a positional encoding mechanism for sequences up to length 1000 using trigonometric functions. Write the mathematical formula and justify your design choices. (10 marks)

(b) Compare learnable positional embeddings versus trigonometric positional encoding: (8 marks)

- Generalization to sequences longer than training length
- Parameter efficiency
- Interpolation and extrapolation capabilities

(c) Given the trigonometric positional encoding formula:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

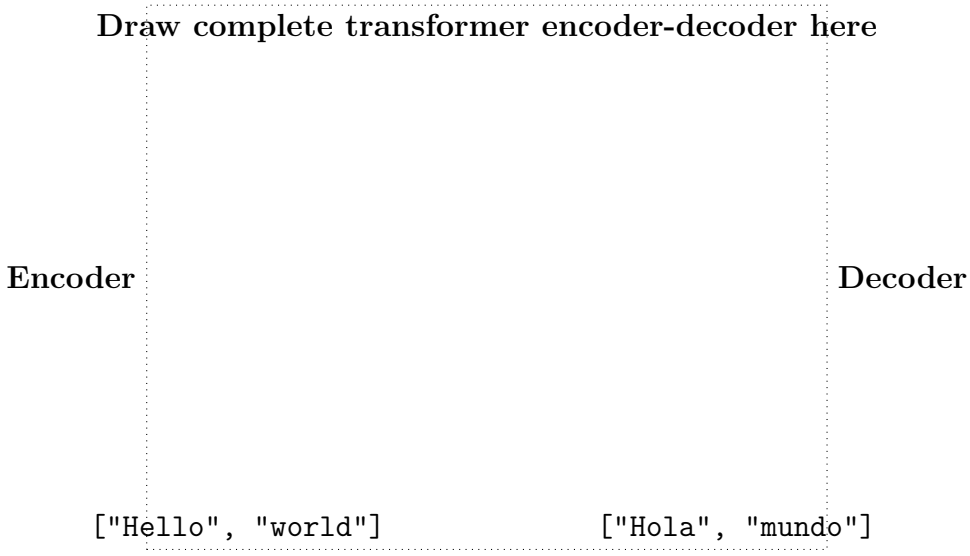
$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

Calculate the positional encoding for position $pos = 3$ and dimensions $i = 0, 1$ when $d_{model} = 512$. (4 marks)

Question 5. Transformer Architecture and Information Flow (28 marks)
Analyze the complete transformer architecture and its information processing capabilities.

(a) Draw a complete transformer encoder-decoder architecture processing the translation task "Hello world" → "Hola mundo". Show: (12 marks)

- Input and output embeddings with positional encoding
- Multi-head self-attention in encoder
- Masked multi-head self-attention in decoder
- Cross-attention between encoder and decoder
- Feed-forward networks and residual connections
- Layer normalization placements



(b) Explain why causal masking is essential in the decoder during training. What would happen if we didn't use masking? Provide a mathematical justification. (8 marks)

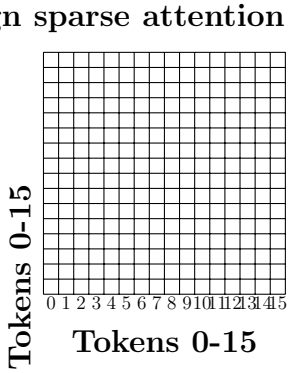
(c) Derive the gradient flow through a residual connection in a transformer layer. Show how residual connections help mitigate vanishing gradients in deep transformer networks. (8 marks)

Question 6. Advanced Attention Mechanisms (20 marks)
Explore advanced concepts in attention mechanisms and their applications.

(a) Sparse attention patterns can reduce computational complexity. Design a sparse attention pattern for sequences of length 16 where each token attends to: (10 marks)

- Itself
- Previous 2 tokens
- Next 2 tokens
- Every 4th token globally

Draw the 16×16 attention mask matrix and calculate the computational savings compared to full attention.



(b) Analyze the trade-offs between different attention mechanisms: (10 marks)

- Full self-attention vs. local attention windows
- Single-head vs. multi-head attention
- Additive vs. multiplicative attention

Discuss computational complexity, expressiveness, and practical considerations.

Question 7. Practical Implementation and Optimization (17 marks)
Address practical considerations in transformer implementation and optimization.

(a) Design a learning rate schedule for transformer training. Implement the warm-up scheduler with cosine decay used in many transformer models: (8 marks)

$$lr(step) = \begin{cases} lr_{max} \cdot \frac{step}{warmup\_steps} & \text{if } step \leq warmup\_steps \\ lr_{max} \cdot 0.5 \cdot \left(1 + \cos\left(\frac{step - warmup\_steps}{total\_steps - warmup\_steps}\pi\right)\right) & \text{otherwise} \end{cases}$$

Plot this schedule for $lr_{max} = 0.001$, $warmup\_steps = 4000$, $total\_steps = 100000$.

(b) Explain three key optimization techniques for transformer training: (9 marks)

- Gradient clipping and why it's necessary
- Layer normalization placement (pre-norm vs. post-norm)
- Weight initialization strategies for attention layers

END OF PAPER