Lecture 6 – Transformers

Provide short answers (max. 25 words each) to the following questions.

- What is the key mechanism in Transformers?
- Why is attention important in language tasks?
- What does a self-attention block compute?
- What are the three main components in self-attention?
- Why is attention computation nonlinear?
- Why do we scale the dot product in attention?
- Why do we need positional encodings in Transformers?
- What is absolute positional encoding?
- What is relative positional encoding?
- What is multi-head attention?
- Why use multiple heads in self-attention?
- How are outputs from multiple heads combined?
- What are the main components of a Transformer layer?
- What is tokenization?
- Why is tokenization challenging?
- What is a word embedding?
- How are embeddings learned in Transformers?
- What is the role of an encoder in Transformers?
- What is the role of a decoder in Transformers?
- What is an encoder-decoder architecture used for?
- What type of Transformer model is BERT?
- What is the main training strategy for BERT?
- What type of Transformer model is GPT?
- Why does GPT use masked self-attention?
- What is KV-cache in GPT?
- What is cross-attention in encoder-decoder models?
- Why is self-attention expensive for long sequences?
- What is Nyströmformer?

Multiple-choice questions

What key mechanism enables Transformers to handle long-range dependencies?

A) Convolution
B) Pooling
C) Self-attention
D) Dropout

What is the main advantage of attention in NLP tasks?

A) Reduces vocabulary size
B) Focuses on relevant parts of the input
C) Removes positional encoding
D) Speeds up tokenization

What does a self-attention block output?

A) A single scalar
B) Weighted sums of values for each input
C) Only queries
D) Only keys

Which components define attention?

A) Queries, Keys, Values
B) Tokens, Layers, Heads
C) Embeddings, Dropout, Normalization
D) Encoder, Decoder, Feedforward

How are attention weights computed?

A) Softmax of query-key dot products
B) Sigmoid of values
C) ReLU of embeddings
D) Max pooling

Why is scaling applied in dot-product attention?

A) To increase variance
B) To prevent large values destabilizing softmax
C) To normalize embeddings
D) To reduce sequence length

What is the complexity of self-attention with respect to sequence length?

A) Linear
B) Quadratic
C) Cubic
D) Logarithmic

What ensures attention weights sum to one?

A) Normalization layer
B) Dropout
C) Softmax function
D) Batch normalization

Why do Transformers need positional encoding?

A) To reduce parameter count
B) To normalize embeddings
C) To compute attention faster
D) Because self-attention ignores input order

Which positional encoding uses sinusoidal patterns?

A) Absolute encoding
B) Relative encoding
C) Learned encoding
D) Attention-based encoding

What does relative positional encoding capture?

A) Absolute token positions
B) Vocabulary size
C) Offsets between query and key positions
D) Embedding normalization

What is multi-head attention?

A) Multiple feedforward layers
B) Parallel attention computations with different parameters
C) Multiple positional encodings
D) Multiple tokenizers

Why use multiple heads?

A) To reduce memory usage
B) To speed up tokenization
C) To eliminate positional encoding
D) To capture diverse relationships between tokens

How are outputs from multiple heads combined?

A) Averaged
B) Concatenated and linearly transformed
C) Summed directly
D) Normalized only

Which components form a Transformer layer?

A) Multi-head attention, feedforward network, normalization
B) Convolution, pooling, dropout
C) RNN, GRU, LSTM

D) Tokenizer, embedding, decoder only

What is the role of the feedforward network?

A) Adds positional encoding
B) Applies nonlinearity and increases capacity
C) Computes attention weights
D) Normalizes embeddings

Why is layer normalization used?

A) To reduce vocabulary size
B) To stabilize training
C) To compute attention faster
D) To eliminate Dropout

What is tokenization?

A) Splitting text into smaller units
B) Normalizing embeddings
C) Computing attention weights
D) Adding positional encoding

Why is tokenization challenging?

A) Due to handling punctuation and word variations
B) Requires convolution
C) Needs positional encoding
D) Requires dropout

What is a word embedding?

A) One-hot representation
B) Dense vector representation of a token
C) Attention weight
D) Positional encoding

How are embeddings learned?

A) Fixed during training
B) Computed by softmax
C) Through trainable parameters
D) Generated by dropout

What does an encoder do?

A) Generates next token
B) Applies Dropout only

C) Converts input tokens into contextual representations
D) Normalizes embeddings

What does a decoder do?

A) Computes attention weights only
B) Removes embeddings
C) Adds positional encoding
D) Generates output tokens based on previous outputs and encoder outputs

What is encoder-decoder architecture used for?

A) Image classification
B) Tokenization
C) Sequence-to-sequence tasks like translation
D) Dropout regularization

What type of Transformer is BERT?

A) Decoder-only
B) Encoder-only
C) Encoder-decoder
D) Convolutional

What is BERT's training strategy?

A) Supervised only
B) Pre-training with self-supervision and fine-tuning
C) Reinforcement learning
D) Dropout-based training

Name one pre-training task for BERT.

A) Next token prediction
B) Image patch prediction
C) Masked language modeling
D) Speech synthesis

How is BERT adapted for downstream tasks?

A) By adding task-specific layers and fine-tuning
B) By freezing all layers
C) By removing embeddings
D) By using convolution

What type of Transformer is GPT?

A) Encoder-only

B) Decoder-only
C) Encoder-decoder
D) Convolutional

What is GPT's main objective?
A) Masked token prediction
B) Image patch prediction
C) Autoregressive next-token prediction
D) Sentence classification

Why does GPT use masked self-attention?
A) To prevent access to future tokens
B) To allow future tokens
C) To compute embeddings
D) To normalize attention

What is KV-cache used for?
A) Storing embeddings
B) Storing previous key-value pairs for faster generation
C) Computing positional encoding
D) Normalizing attention

What is cross-attention?
A) Decoder attends to encoder outputs
B) Encoder attends to decoder outputs
C) Attention between tokens in same layer
D) Attention between positional encodings

Why is self-attention expensive for long sequences?
A) Linear complexity
B) Quadratic complexity
C) Cubic complexity
D) Logarithmic complexity

What is Nyströmformer?
A) A convolutional Transformer
B) A vision Transformer
C) A low-rank approximation of self-attention
D) A recurrent Transformer