ENEE-719D Transformers Lecture 4

Dr. Ching-Yi Lin





Transformer for Large-language Model

deepseek R1

Llama 4

unsloth | Meta



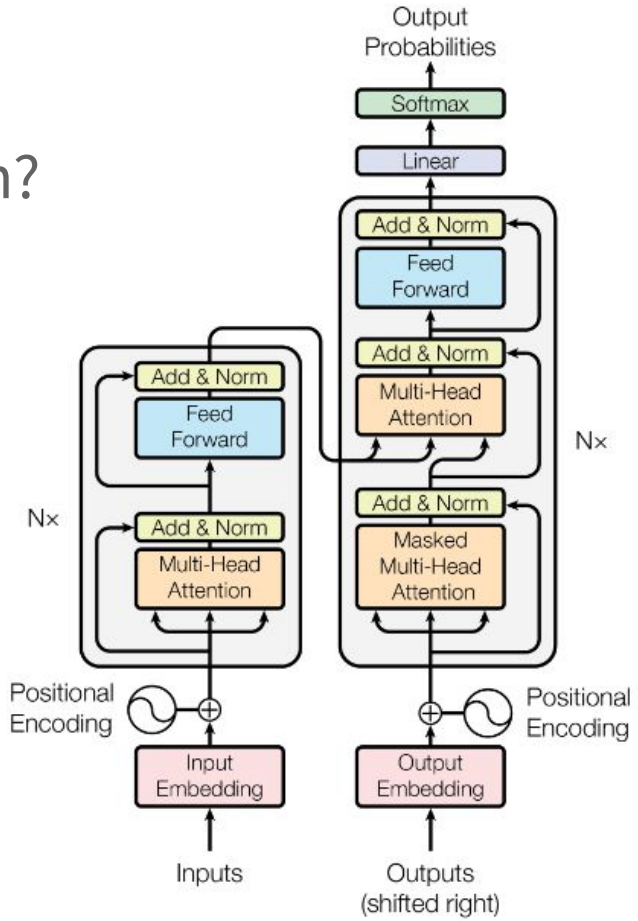




UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

Outline

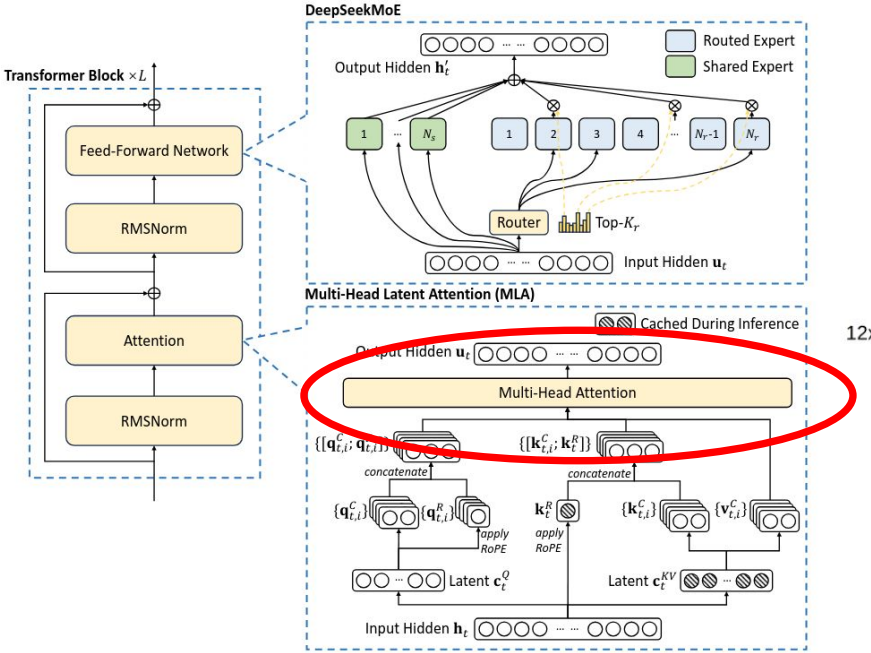
1. What is transformer? What is attention?
2. Transformer usage
3. Transformer in hardware
4. Generalized transformer

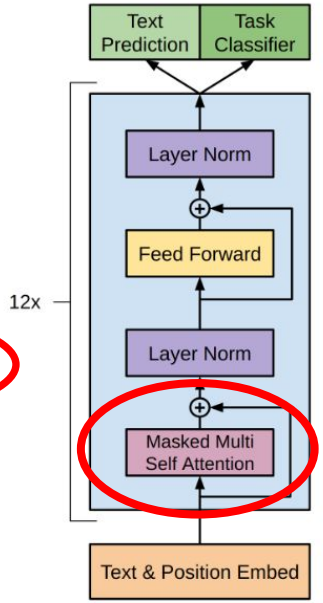


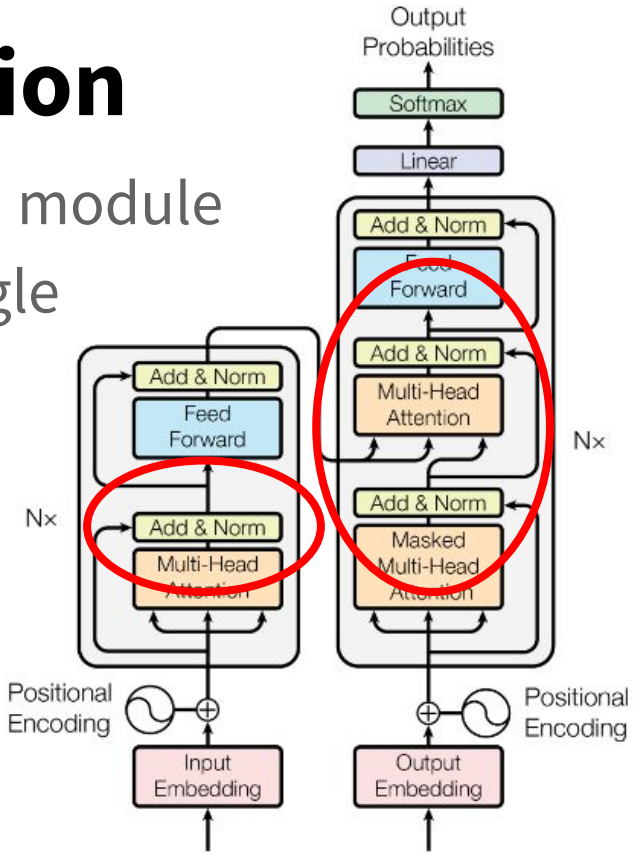
UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

Attention Attention Attention

- All transformer models have attention module
- (L->R) DeepSeek V3 -> Open AI -> Google

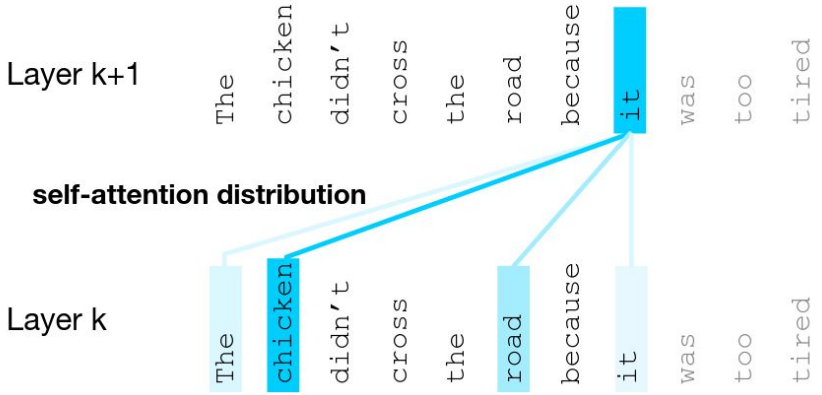






Motivation of Attention

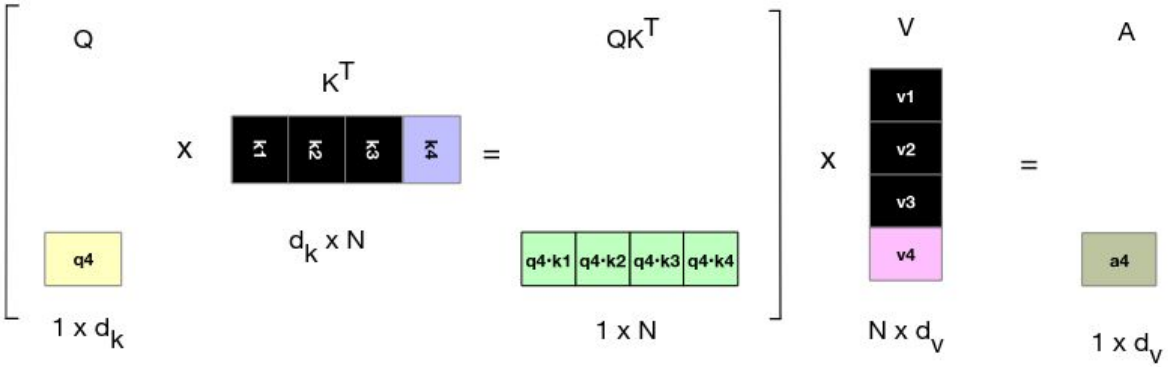
- Word-correspondence within a sentence
- The chicken didn't cross the road because it was too tired
  The chicken didn't cross the road because it was too wide
- Weighted-token relationship



Two-setp Attention Mechanism

- Each token (word) has their query (Q), key (K), and value (V)

1. Similarity calculated between query and key
   - Inner product between current query (q4) and all keys (k1, k2, k3, and k4)
   - Attention score is a matrix with size 1x4
2. Value (V) is weighted through attention score (QK^T)

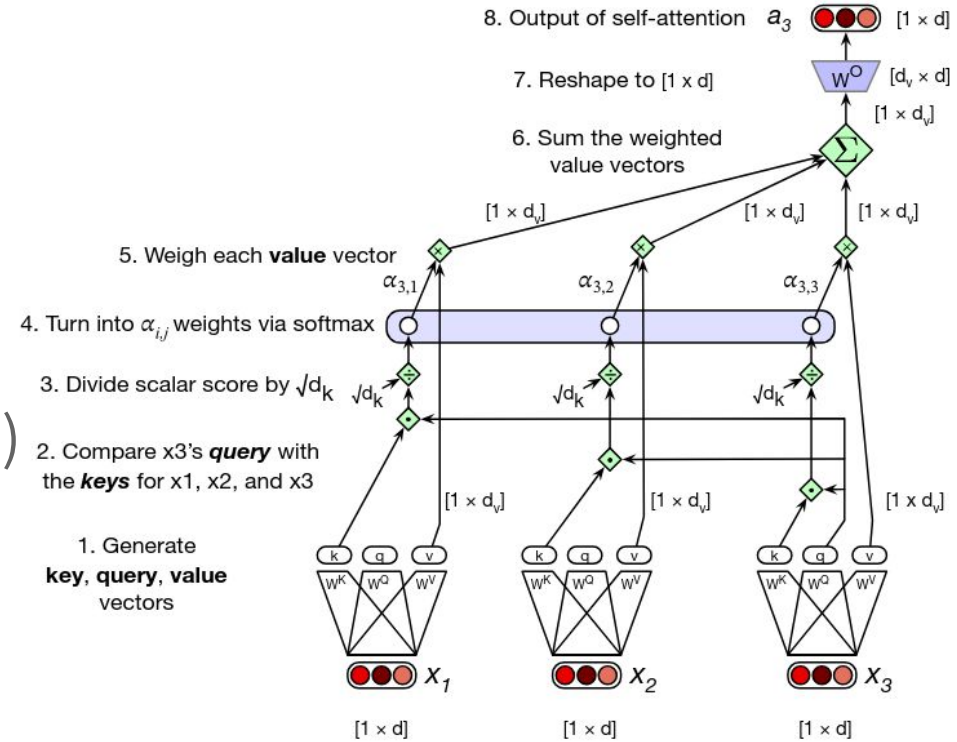


UNIVERSITY OF MARYLAND
FEARLESSLY FORWARD

Attention Computation

1. Q/K/V from linear layer
2. Q3 queries to K1, K2, K3
3. Attention matrix (Q3K1, Q3K2, Q3K3)
4. Softmax to prob matrix (α)
5. Weight v1, v2, v3
6. α1v1 + α2v2 + α3v3

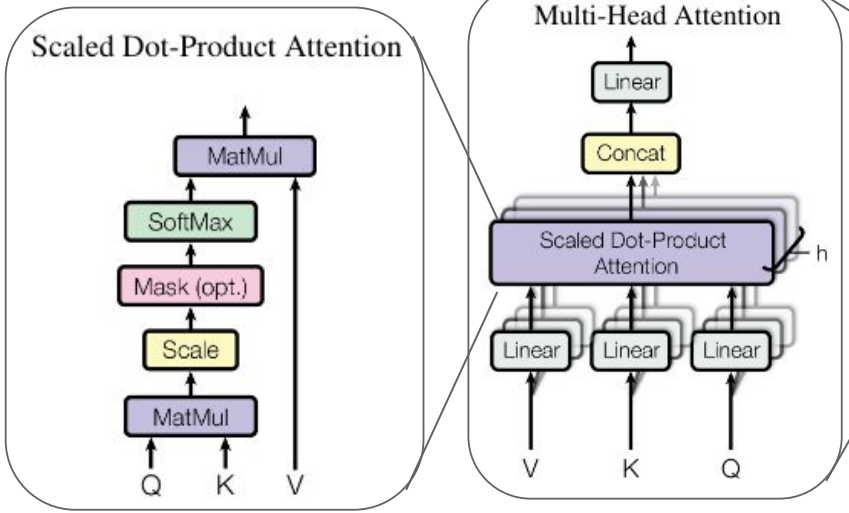
Two matrix multiplication

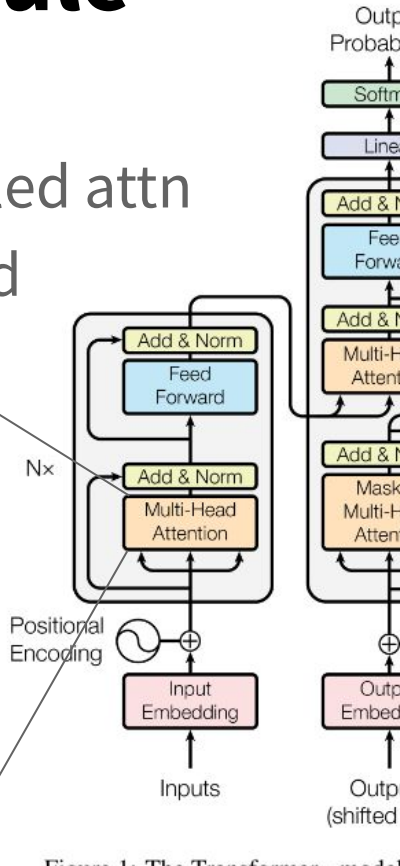


UNIVERSITY OF MARYLAND
FEARLESSLY FORWARD

Build Transformer from Attn Module

- Each scaled attention = Two matmul (QK and αV)
- Each multi-head attention (MHA) = h parallel scaled attn
- Each transformer blk = 1x/2x MHA + Feed-forward
  - Feed-forward = Multi-layer perceptron

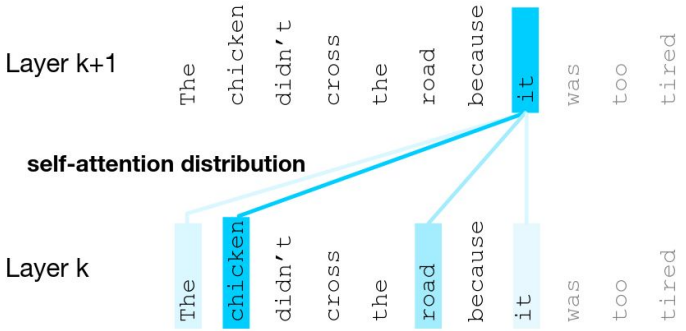


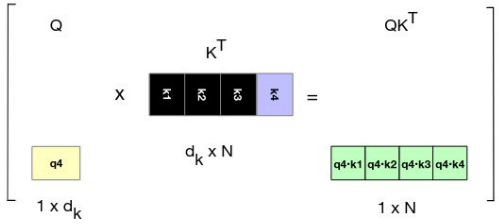


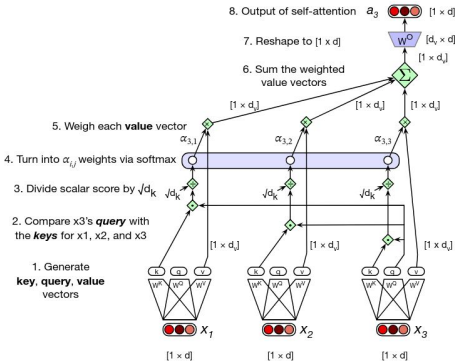
UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

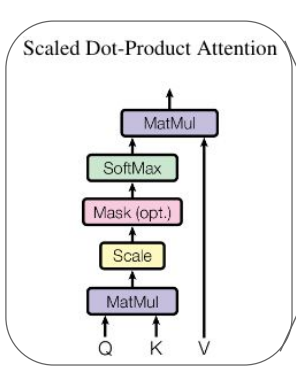
Quarter 1 Done. Any Question?

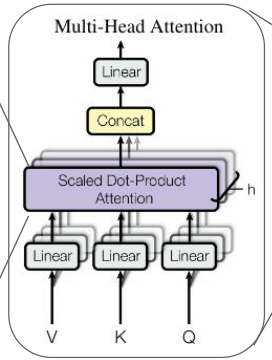
1. What is attention? What is transformer?
2. Transformer usage
3. Transformer in hardware
4. Generalized transformer











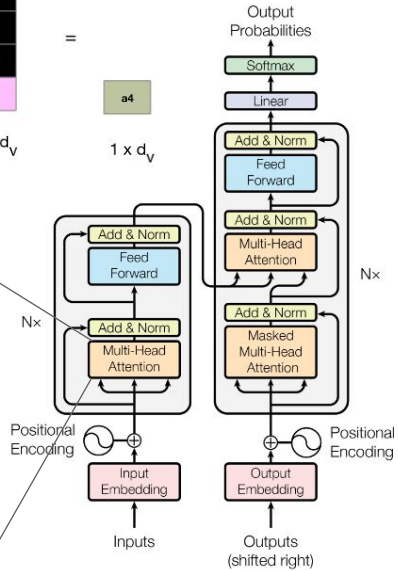
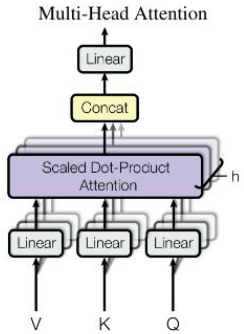


Figure 1: The Transformer - model architecture.

UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

Interpreting Transformer Model

- Stacking transformer layers
- Encoder (Left)
  - Encode context as keys and values for 2nd MHA
  - Keep reusing in the future inference
- Decoder (Right)
  - Feed input. Generate output
  - Provide query in 2nd MHA
  - 1st MHA = self-attention module



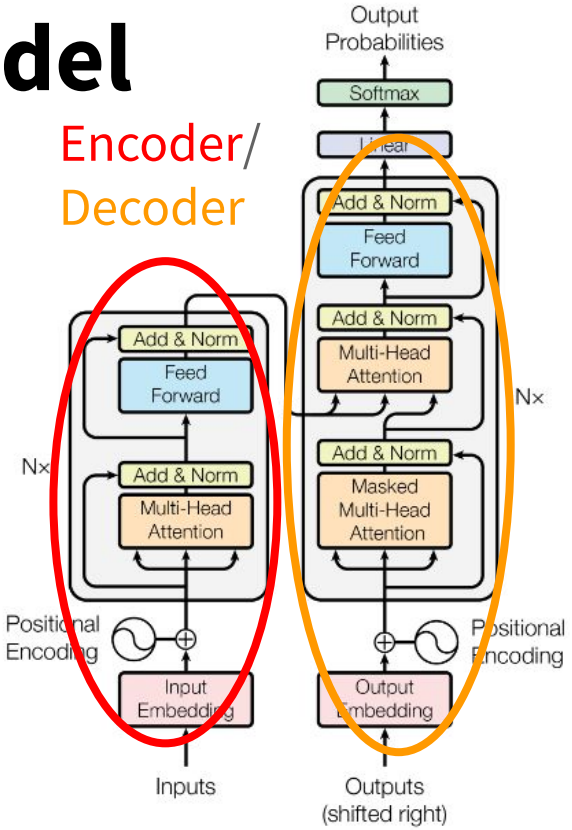
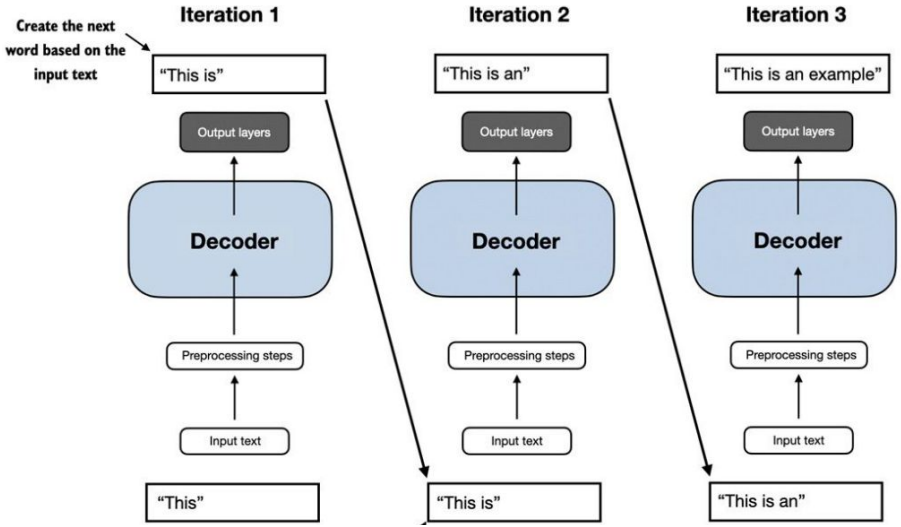


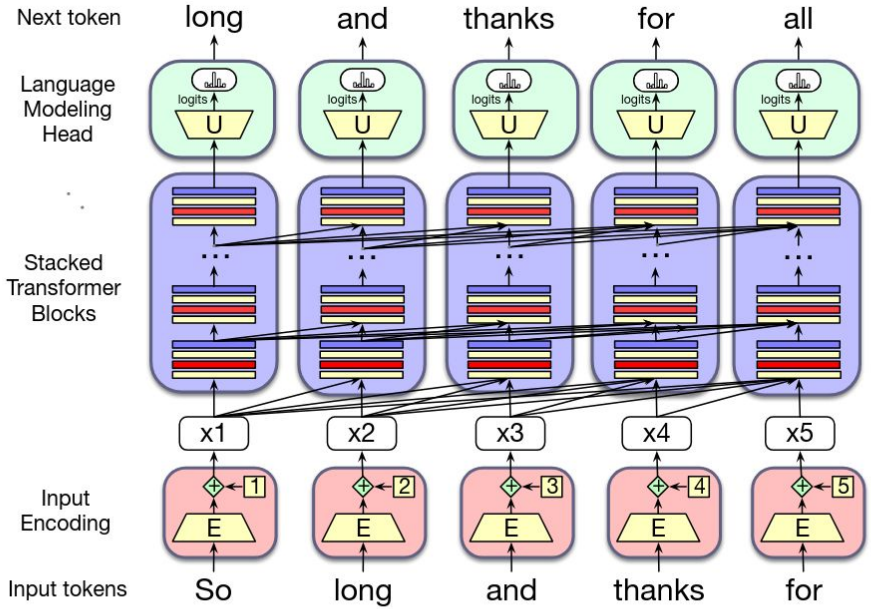
Figure 1: The Transformer - model architecture.

UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

Next-token Prediction

- Each transformer inference generates one token
- Decoder only

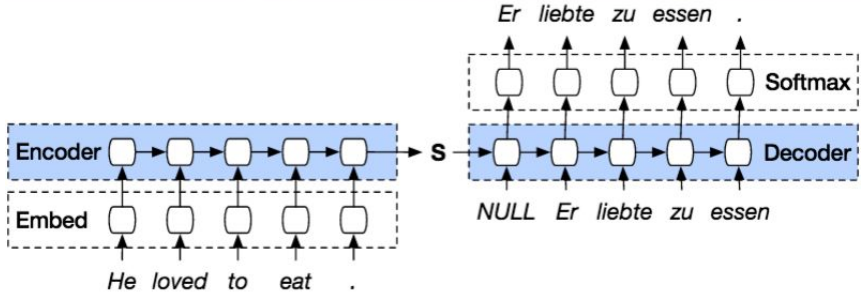


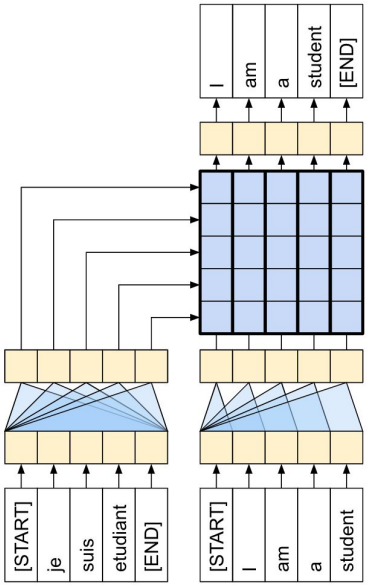


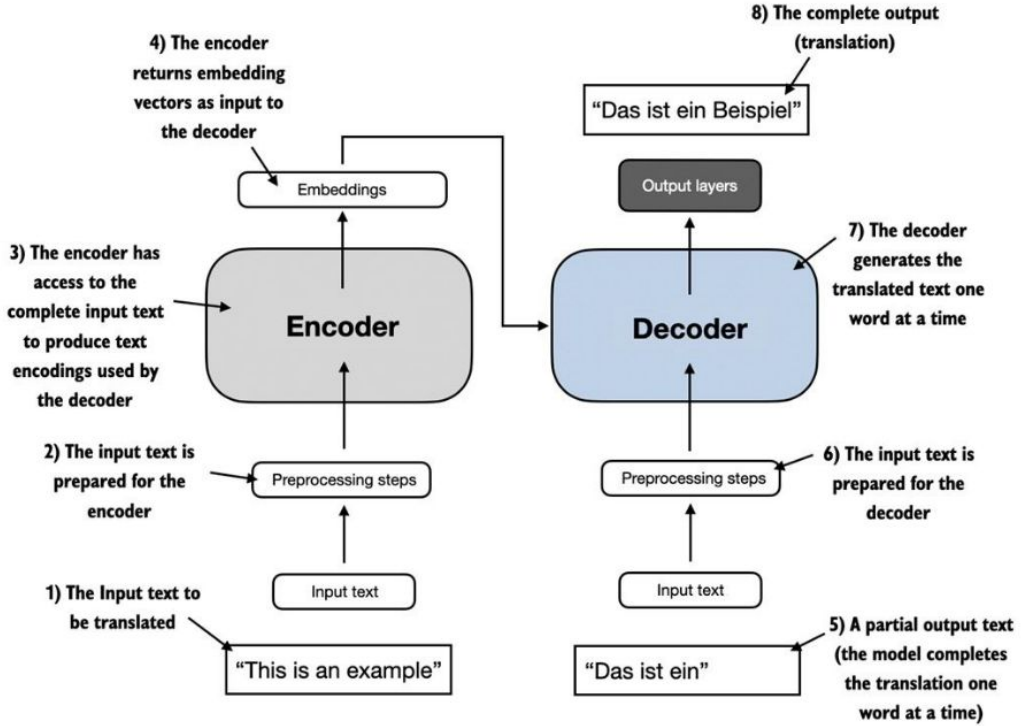
UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

Machine Translation

- Sequential token generation
- Original texts as context





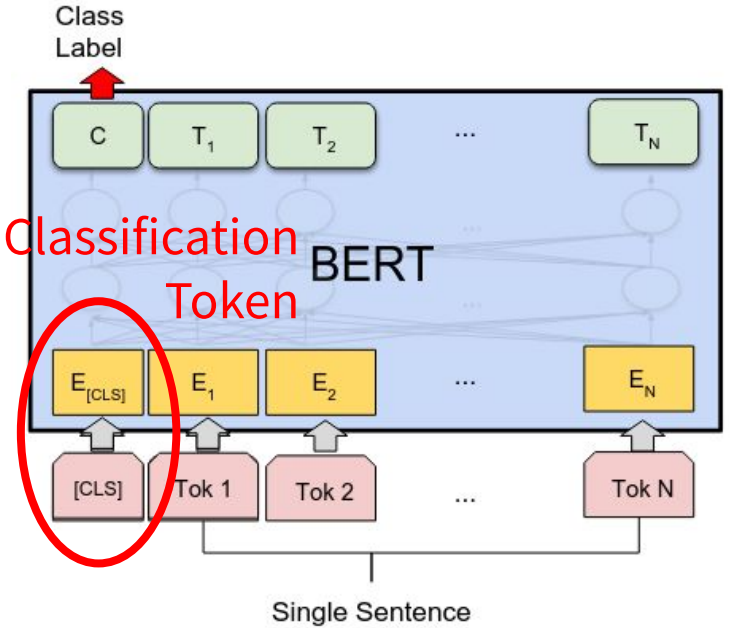


UNIVERSITY OF MARYLAND

Single Sentence Classification

- One special token: [CLS]
- Corpus of Linguistic Acceptability (CoLA)
  - 1: Acceptable, 0: Unacceptable

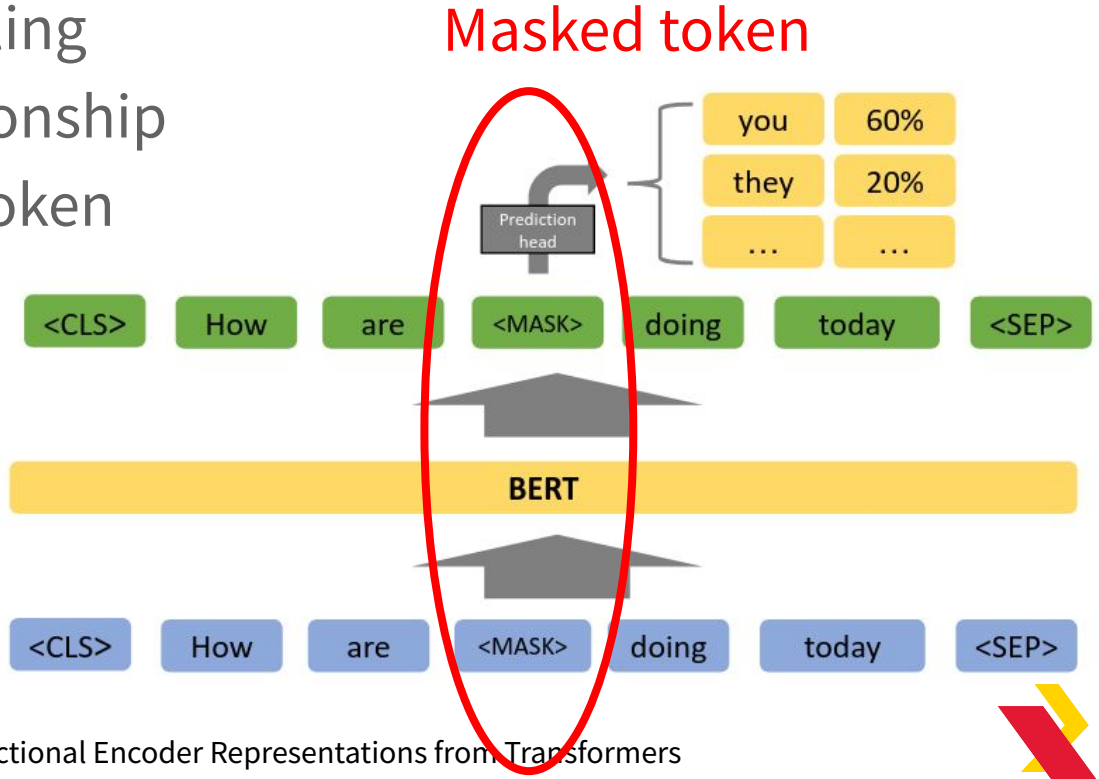
Label	Sentence
1	The weights made the rope stretch over the pulley.
1	The mechanical doll wriggled itself loose.
1	If you had eaten more, you would want less.
0	The more you would want, the less you would eat.



Model Training (1/2)

Masked Language Modeling

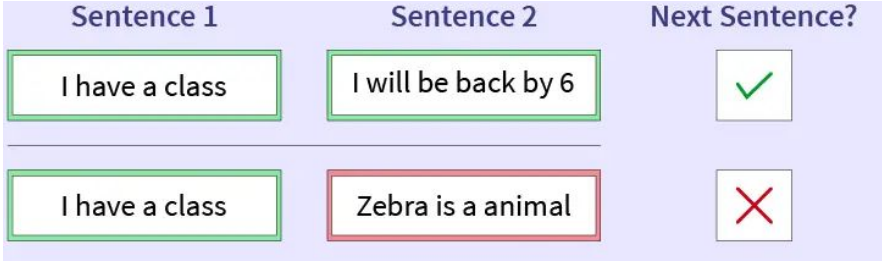
- Train for token relationship
- Use special [MASK] token
- Prediction head
  - Append to [MASK]

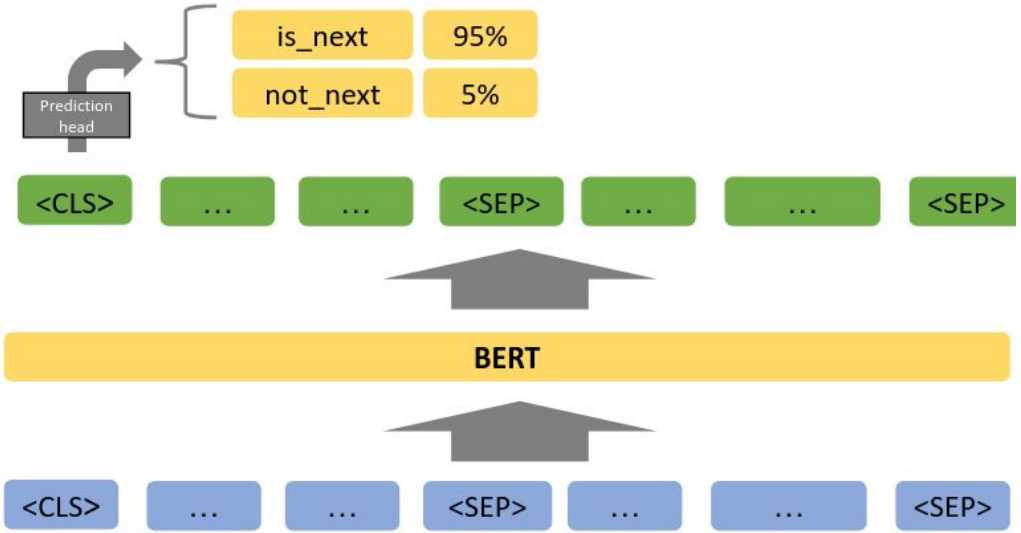


BERT: Bidirectional Encoder Representations from Transformers

Model Training (2/2)

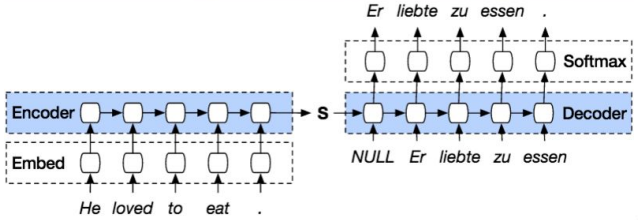
- Next Sentence Prediction
- Input two sentences
  - Separate with a [SEP] token
- Binary label
  - Is_next: Valid next sentence

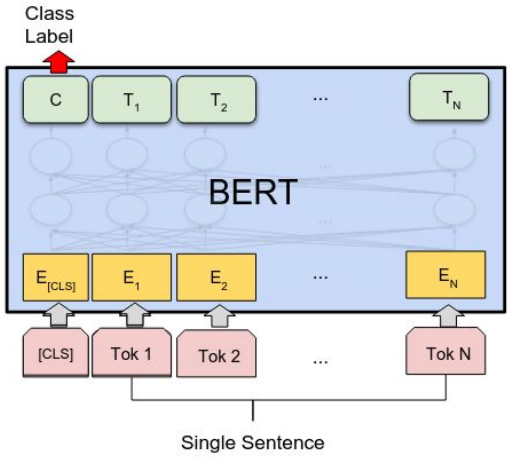


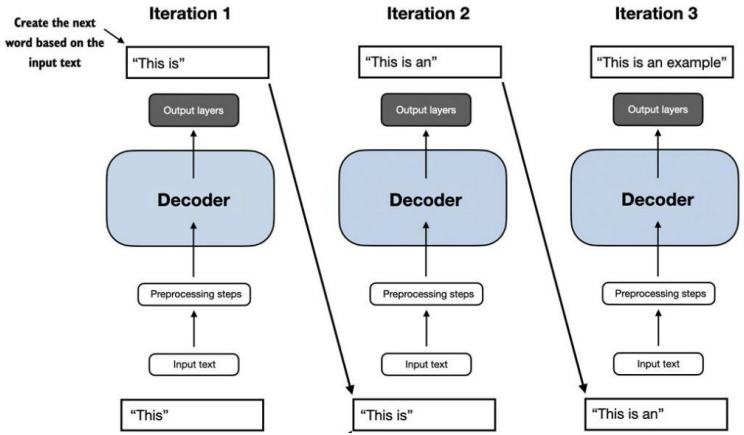


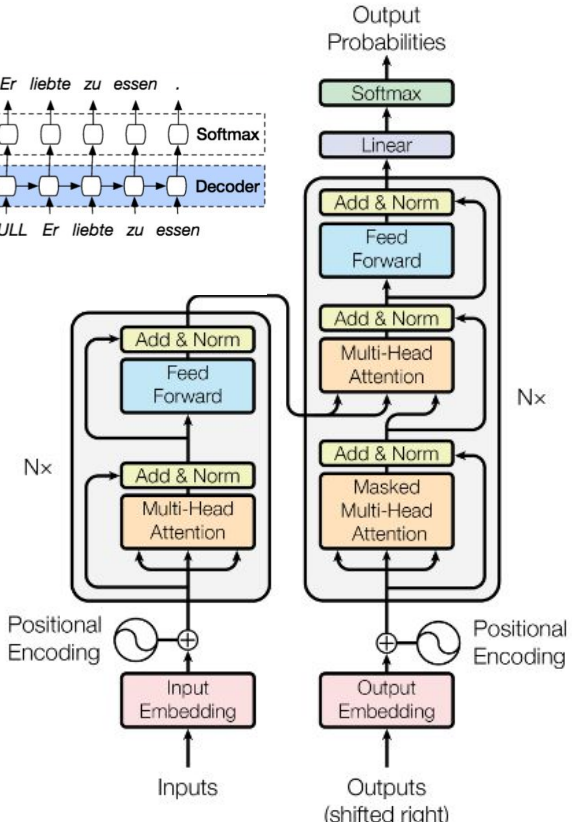
Quarter 2 Done. Any Question?

1. What is attention? What is transformer?
2. Transformer usage
3. Transformer in hardware
4. Generalized transformer









Transformer in Hardware

- Embedding
  - From texts to vectors
  - Positional encoding
- Architecture
  - Dynamic matrix multiplication
  - KV cache

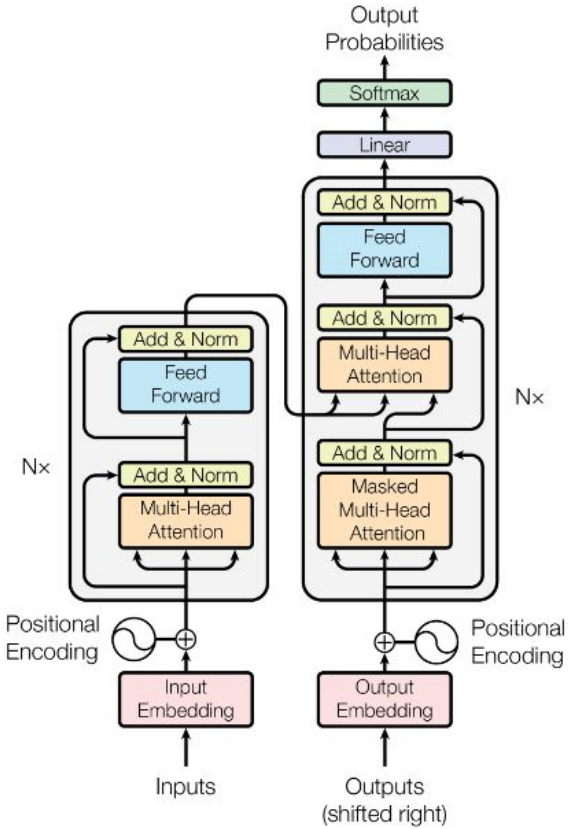
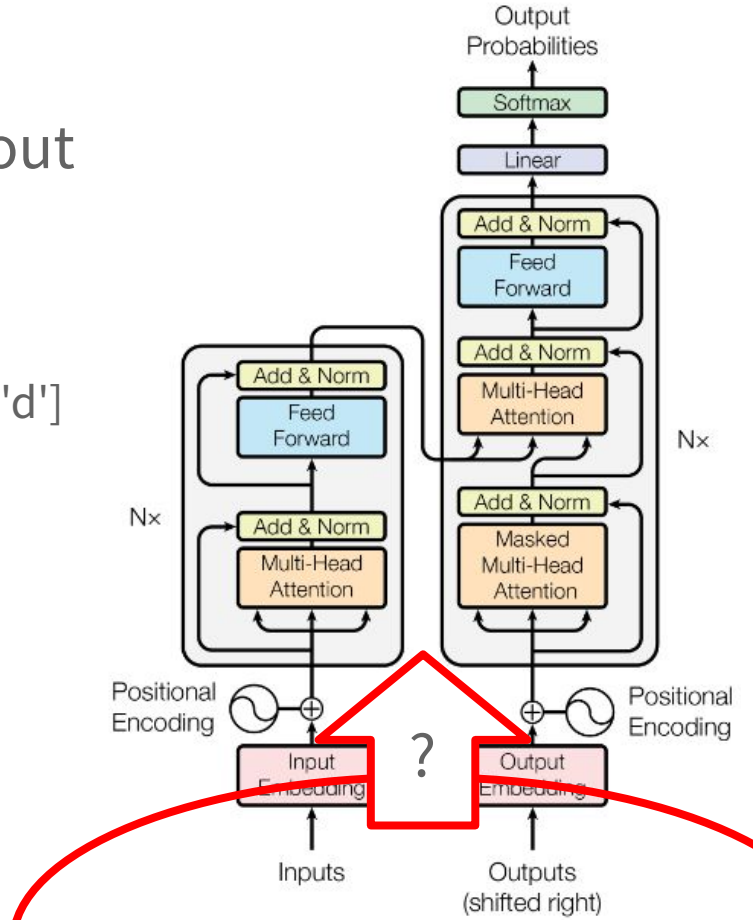


Figure 1: The Transformer - model architecture.

UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

From Texts to Vectors

- Transformers are matrix-in matrix-out
- How can texts convert to matrix?
- Available tokenization options
  - Character: ['H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd']
  - Word: [‘Tokenizer’, ‘is’, ‘tricky’]
  - Sub-word: [‘Token’, ‘##izer’, ‘is’, ‘tricky’]



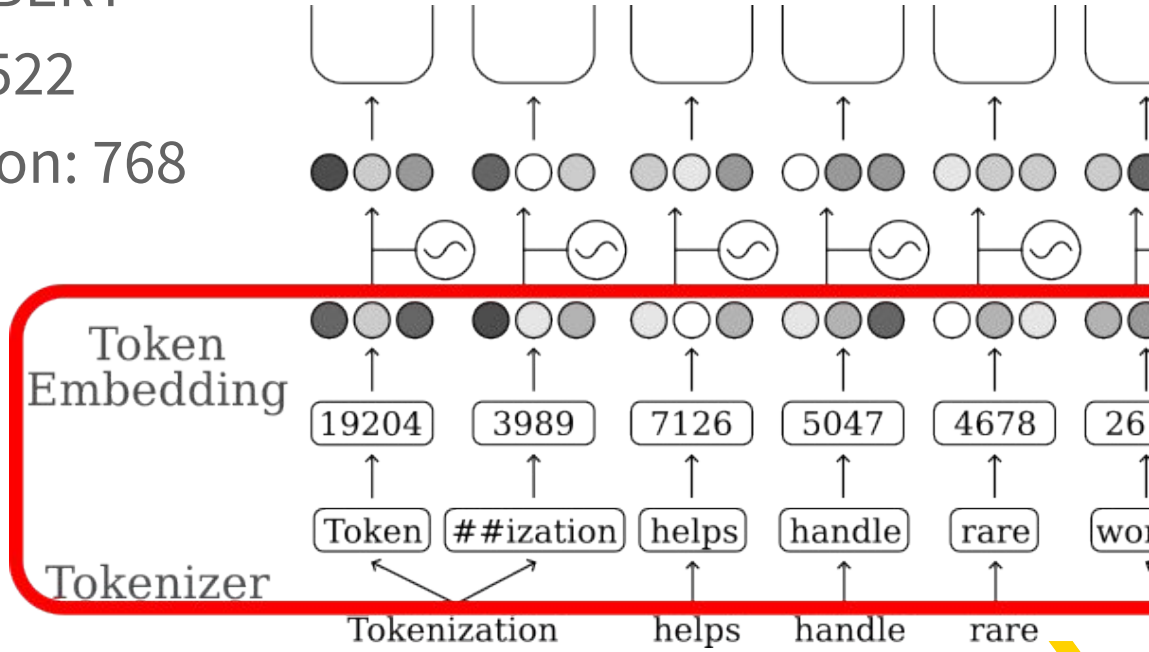
Token Embedding

WordPiece tokenizer in BERT

- Vocabulary size: 30,522
- Embedding dimension: 768

Embedding Lookup

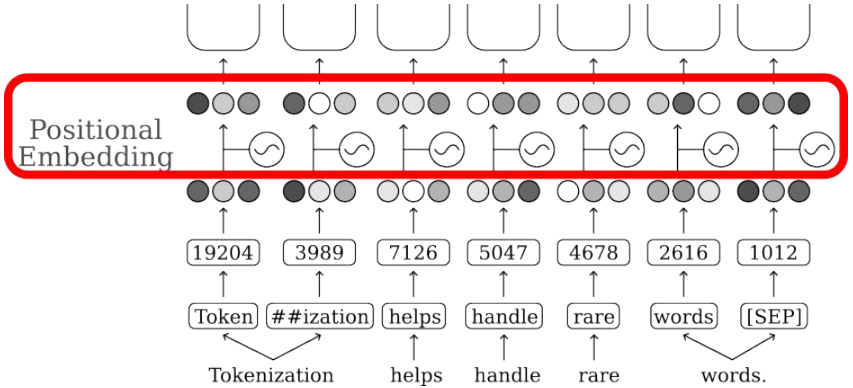
- Token ID -> token

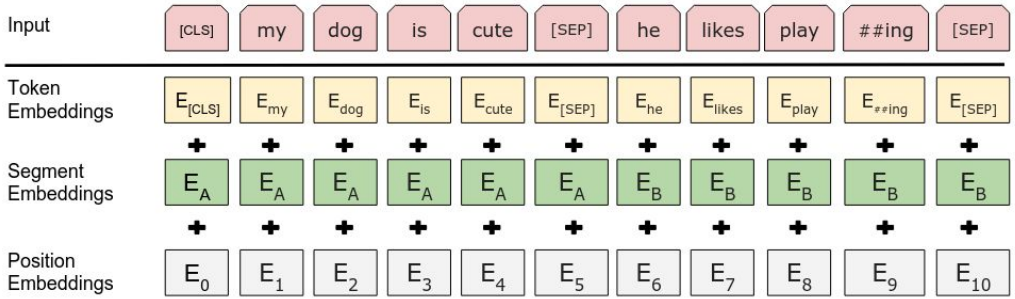


Positional Encoding – Position Matters

To encode positional information

- Each position maps to a token
  - Called positional embeddings
- Add/Mul to token emb.
- Good positional embedding
  - Non-cyclic
  - Bounded





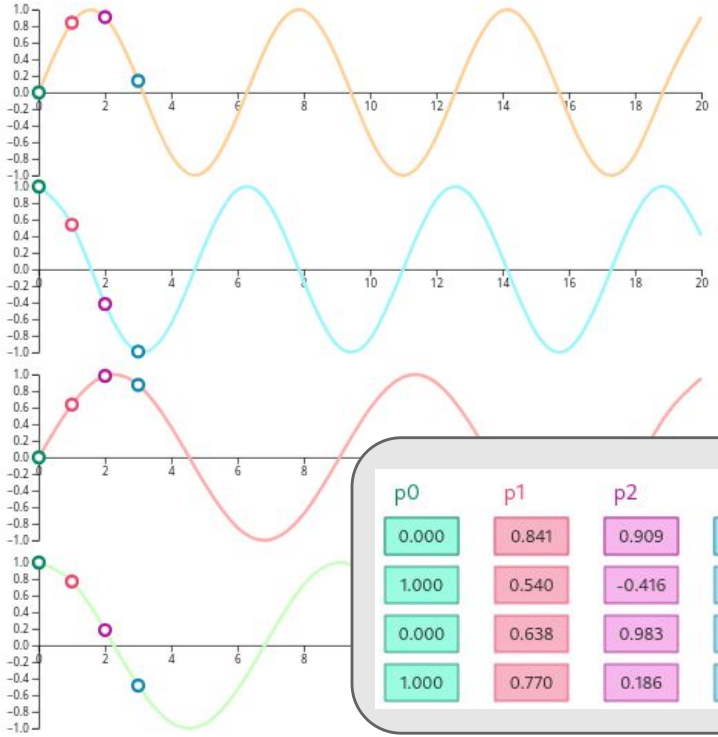
https://erdem.pl/2021/05/understanding-positional-encoding-in-transformers

Rotary Positional Encoding (RoPE)

- PE(pos,2i) = sin(pos×θi)

  PE(pos,2i+1) = cos(pos×θi)
  - $\theta i = 10000^{-2(i-1)/d}$

- RoPE properties
  - Cyclic but very-large period
  - Linear independent
  - Bounded (PE$\in$[-1,1])
  - Fix-L2 (ΣPE_i^2 is const)

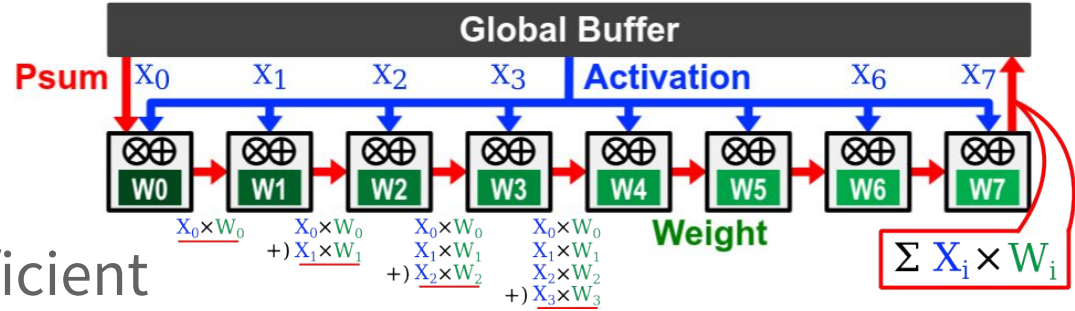


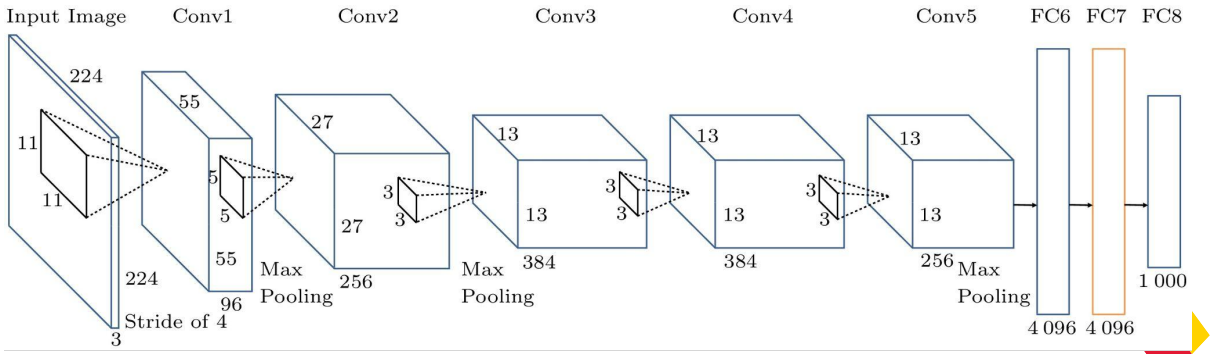
	p0	p1	p2	p3	
	0.000	0.841	0.909	0.141	i=0
	1.000	0.540	-0.416	-0.990	i=1
	0.000	0.638	0.983	0.875	i=2
	1.000	0.770	0.186	-0.484	i=3

https://erdem.pl/2021/05/understanding-positional-encoding-in-transformers

Matrix Multiplication: Conv HW Recap

- Conv-based model
  - Small filter size (< 5x5)
  - Large input size
- Weight reuse will be efficient
  - Weight-stationary dataflow
  - Example $\Sigma X_i W_i$

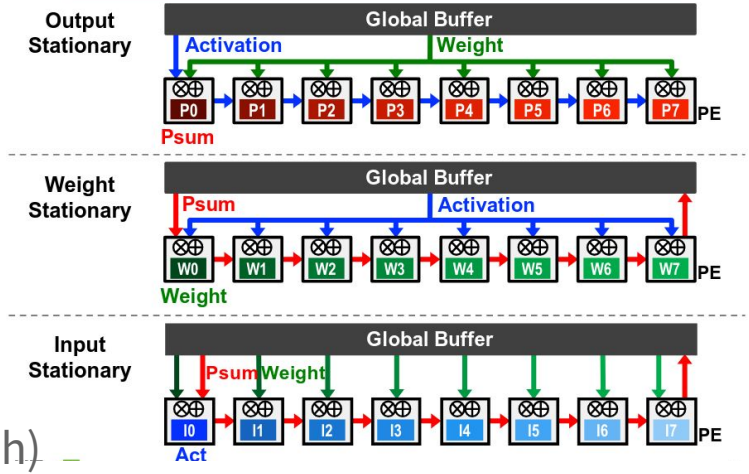


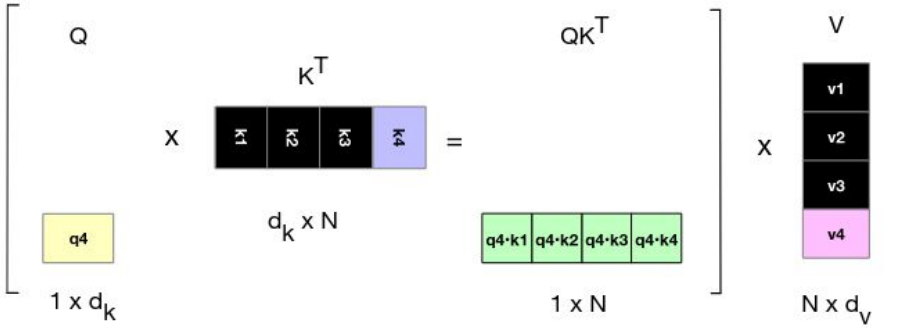


UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

Matmul in Transformer

- Attention
  - Matmul between Q(1xd_k) and K(N x d_k)
  - Output has shape (1xN)
  - d_k^2xN MACs to generate N outputs (d_k^2 each)
- d_k (embedding dim) vs N (attention scope)
  - Large d_k and small N
    - Image-based
    - SwinT-T: d_k= 96 and N = 16
  - Small d_k and large N
    - Llama3-70B: d_k= 8192, N = 8000

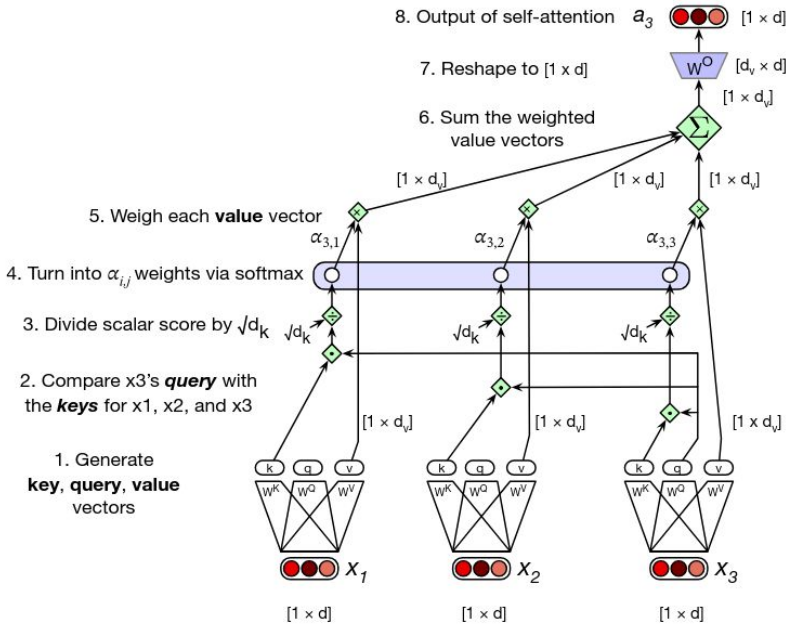


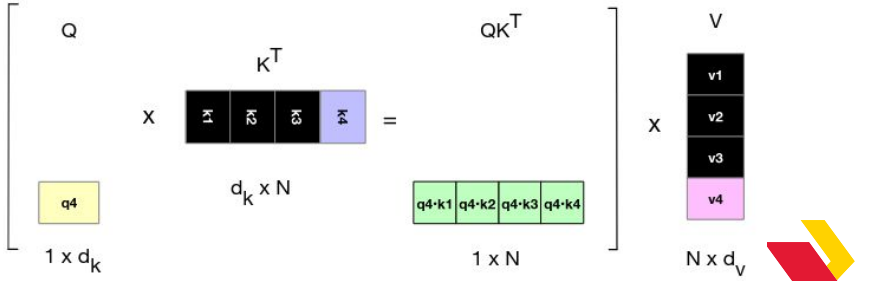


UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

KV Caching

- Step 1 generates Q/K/V from X
- Step 2 use current Q and all Ks
  - One (X→Q) + N× (X→K)
    - (N-1)× (X→K) are computed before
    - Prev N-1 (X→K) were computed
- Step 5 use current α and all Vs
  - N× (X→V)
    - Prev N-1 (X→V) were computed

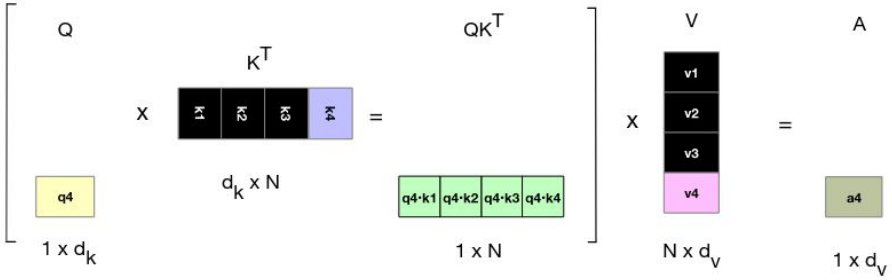


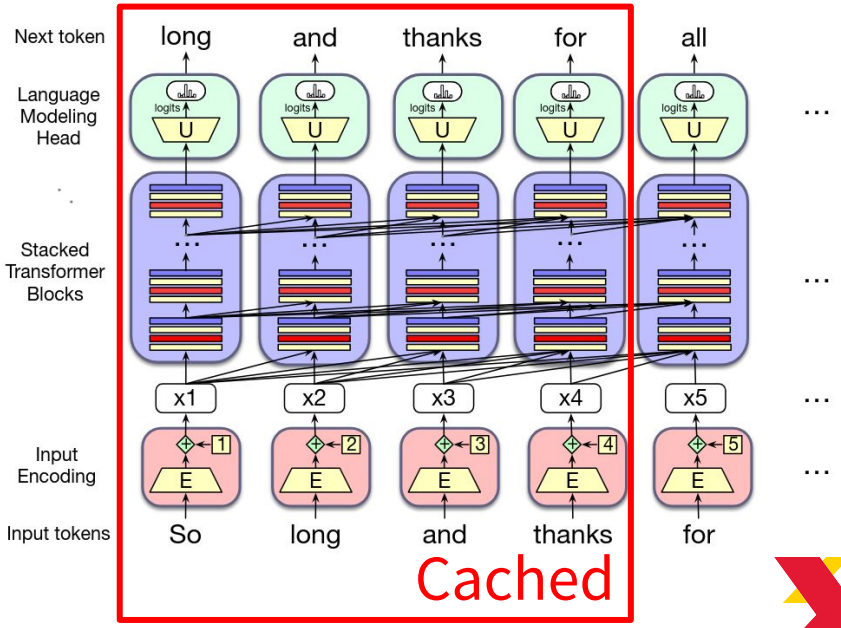


UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

KV Caching

- Cache all K and V from prev N-1 rounds
- Same number of Q/K/V generations every round
- KV cache can be large
  - 2.1GB in LLaMa-7B
  - 19GB in LLaMa-70B



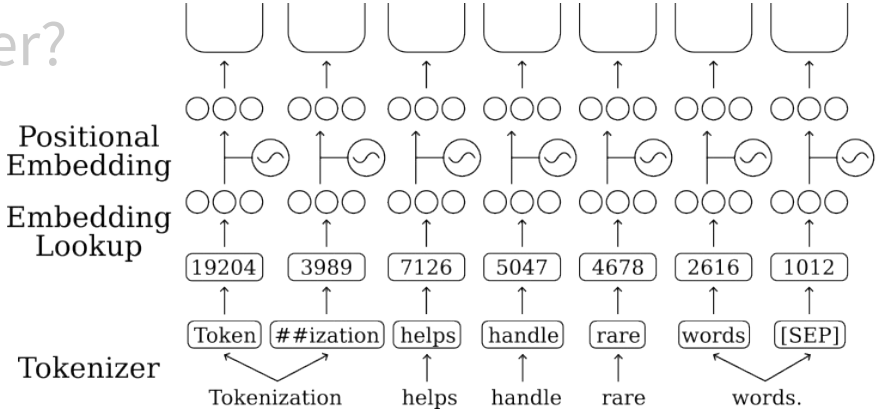


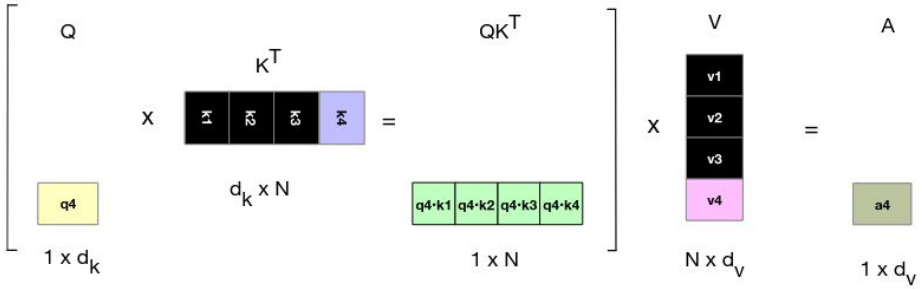
UNIVERSITY OF MARYLAND

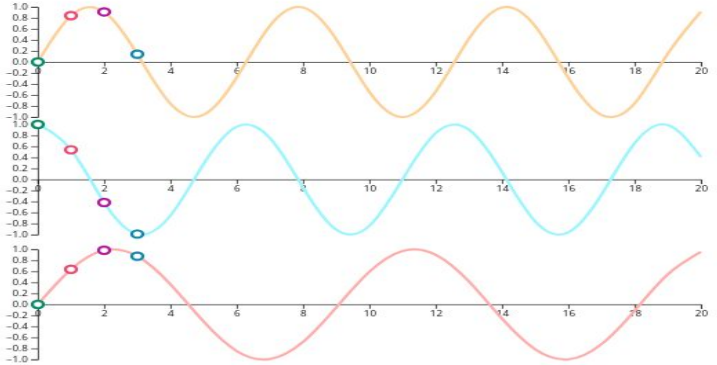
FEARLESSLY FORWARD

Quarter 3 Done. Any Question?

1. What is attention/transformer?
2. Transformer usage
3. Transformer in hardware
4. Generalized transformer



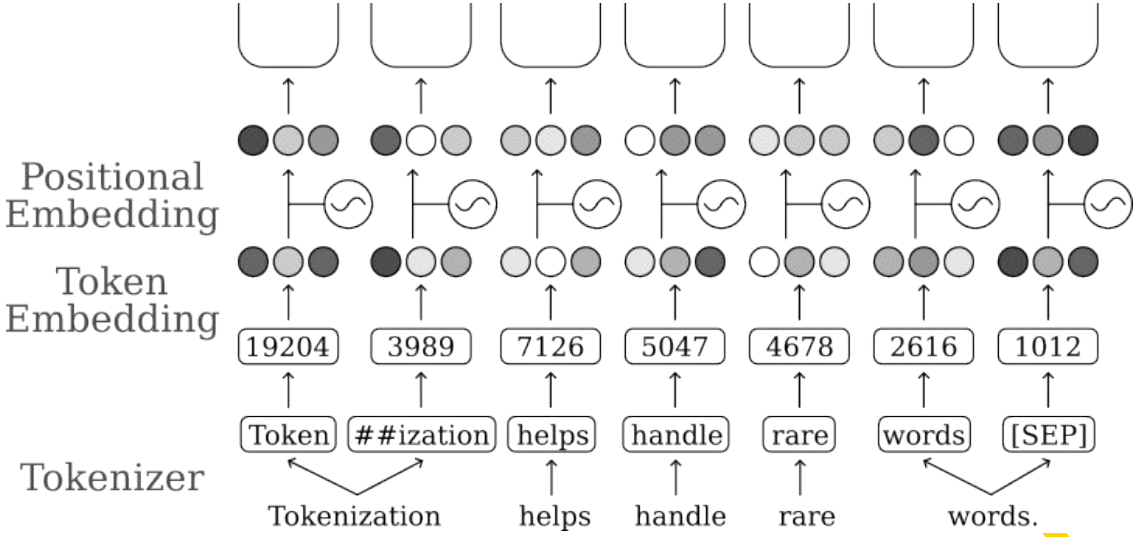




UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

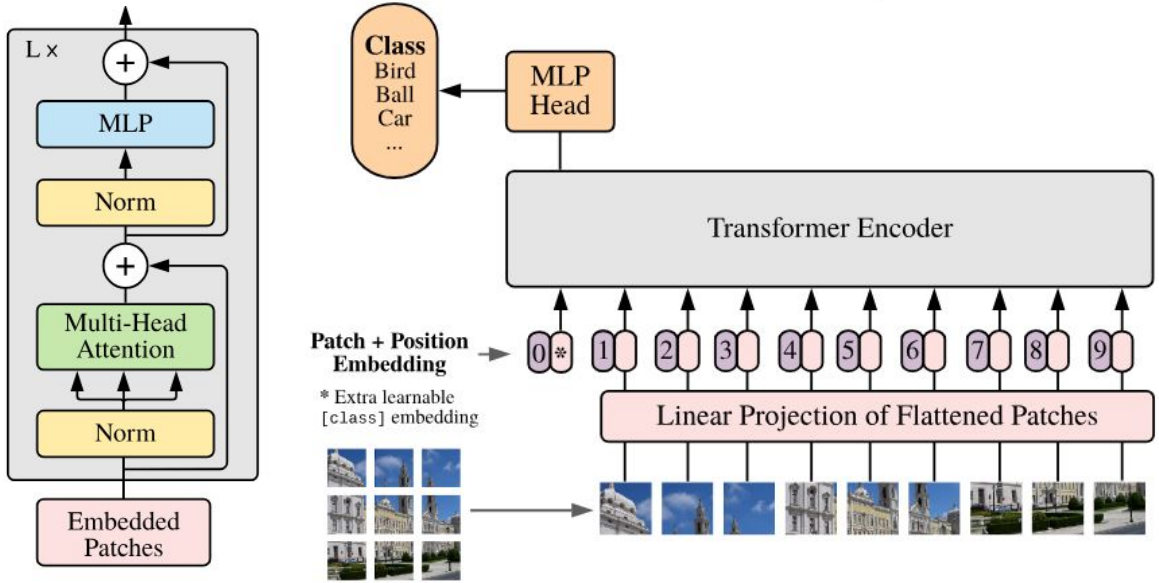
Recap: Transformer in NLP

- Token embedding + Positional embedding
- Transformer model
- How to generalize to other applications
  - Vision
  - Audio
  - Multimodal



Vision Transformer (ViT)

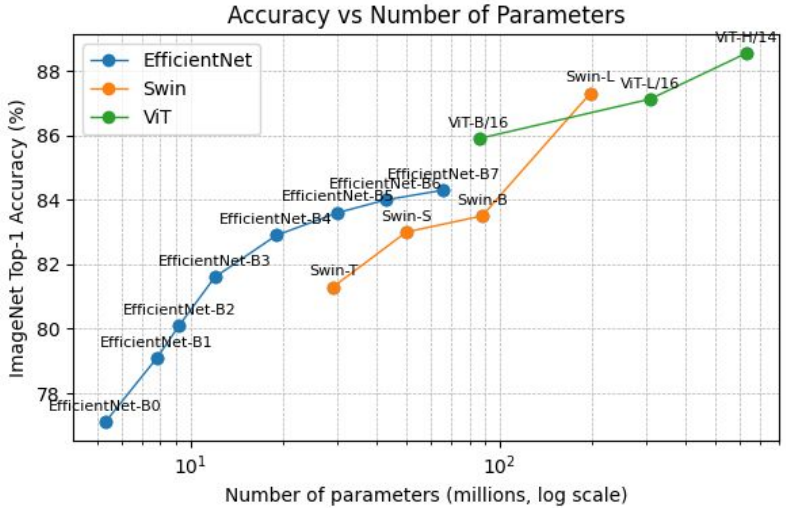
- Token embedding:
  - Image patch -> flattened patch -> linear projection
- Same position embedding
- Model
  - Encoder-only



UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

Results of ViT

- Beat state-of-the-art (SOTA)
- But requires more data and more parameter



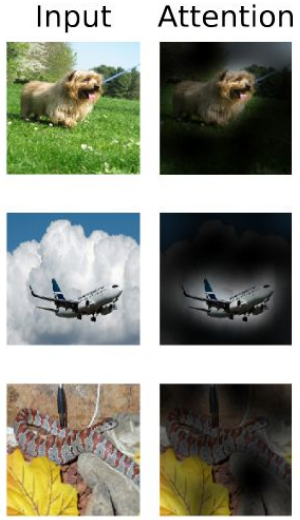
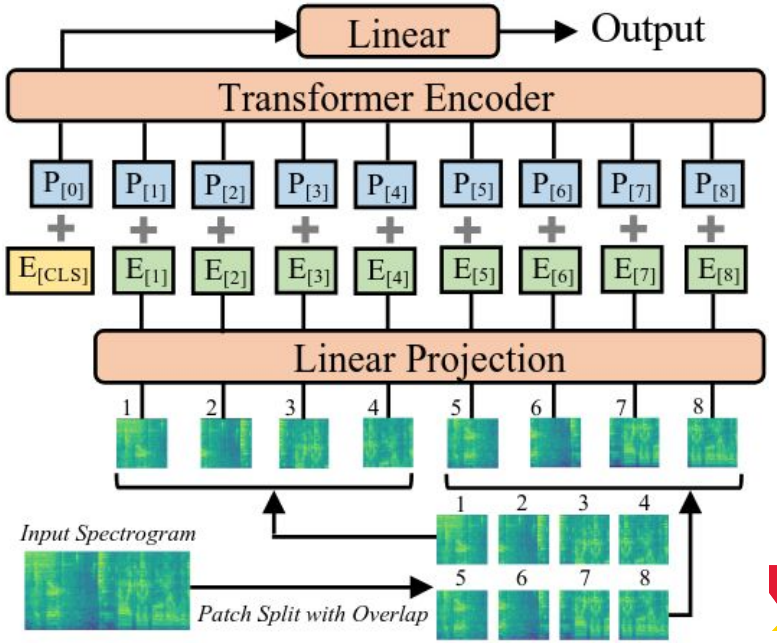


Figure 6: Representative examples of attention from the output token to the input

UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

Audio Spectrogram Transformer

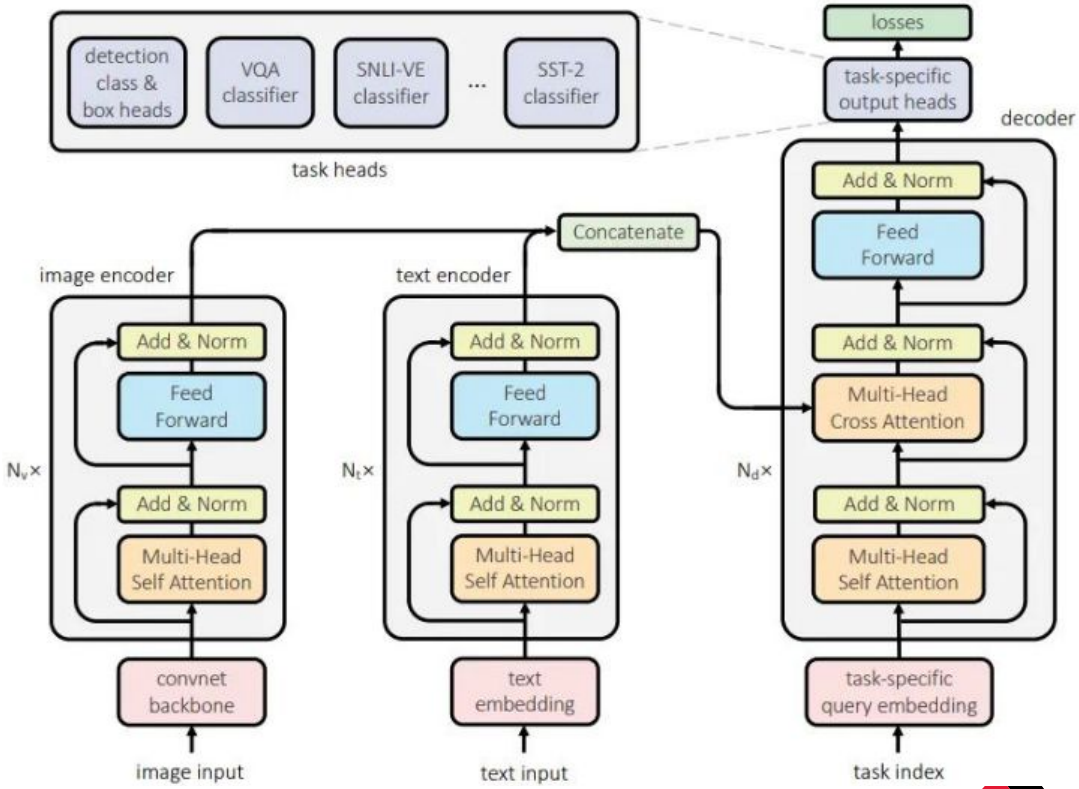
- Token embedding:
  - Audio spectrogram -> linear projection
  - Spectrogram (x: time, y:freq)
- Same position embedding
- Model
  - Only transformer encoder
  - With a classification token



UNIVERSITY OF MARYLAND FEARLESSLY FORWARD

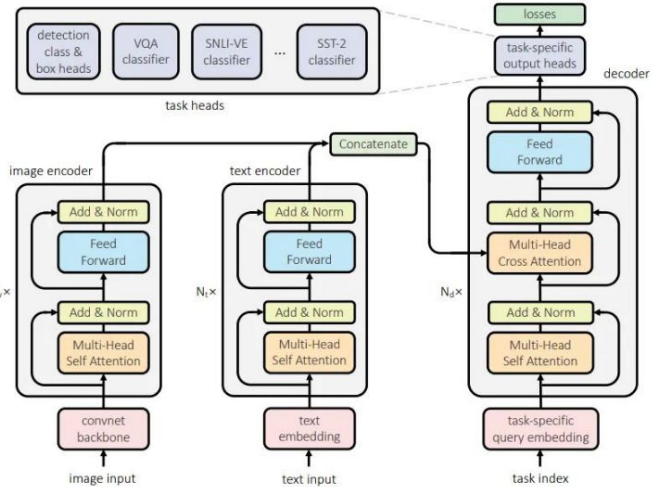
Multimodal Transformer

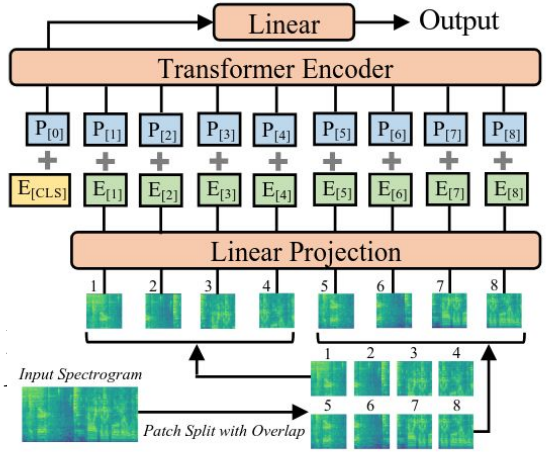
- Token embedding:
  - Image embedding
    + Text embedding
- Model
  - Image encoder
  - Text encoder
  - Joint decoder
    - Concatenated context

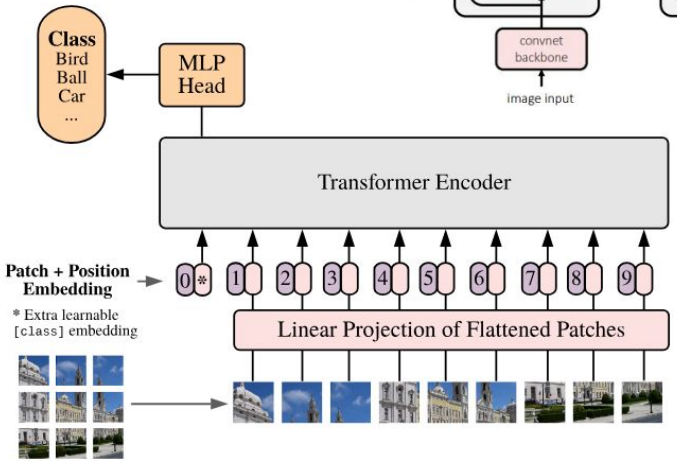


Quarter 4 Done. Any Question?

1. What is attention/transformer?
2. Transformer usage
3. Transformer in hardware
4. Generalized transformer







Questions?



UNIVERSITY OF MARYLAND FEARLESSLY FORWARD


UNIVERSITY OF
MARYLAND
FEARLESSLY
FORWARD



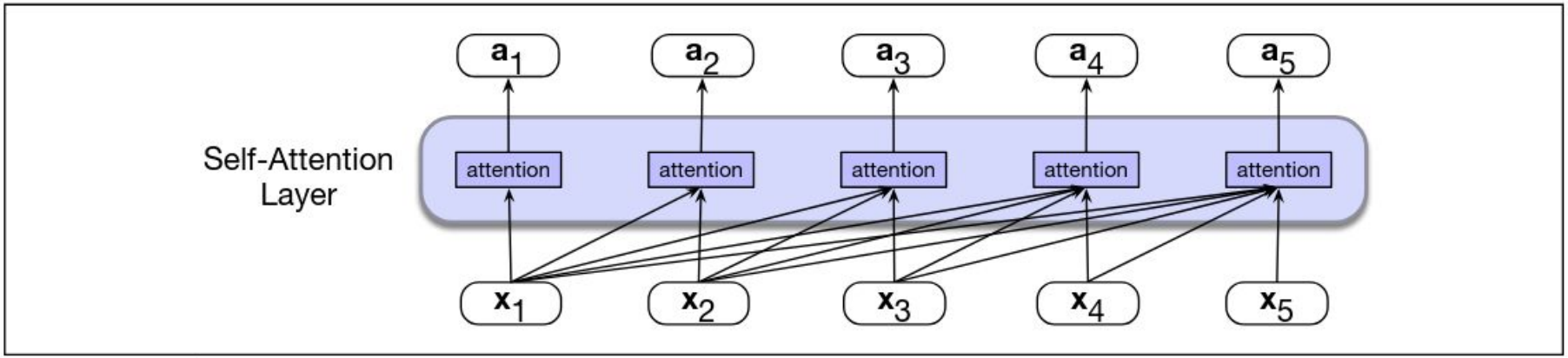


Figure 8.3 Information flow in causal self-attention. When processing each input $\mathbf{x}_i$, the model attends to all the inputs up to, and including $\mathbf{x}_i$.

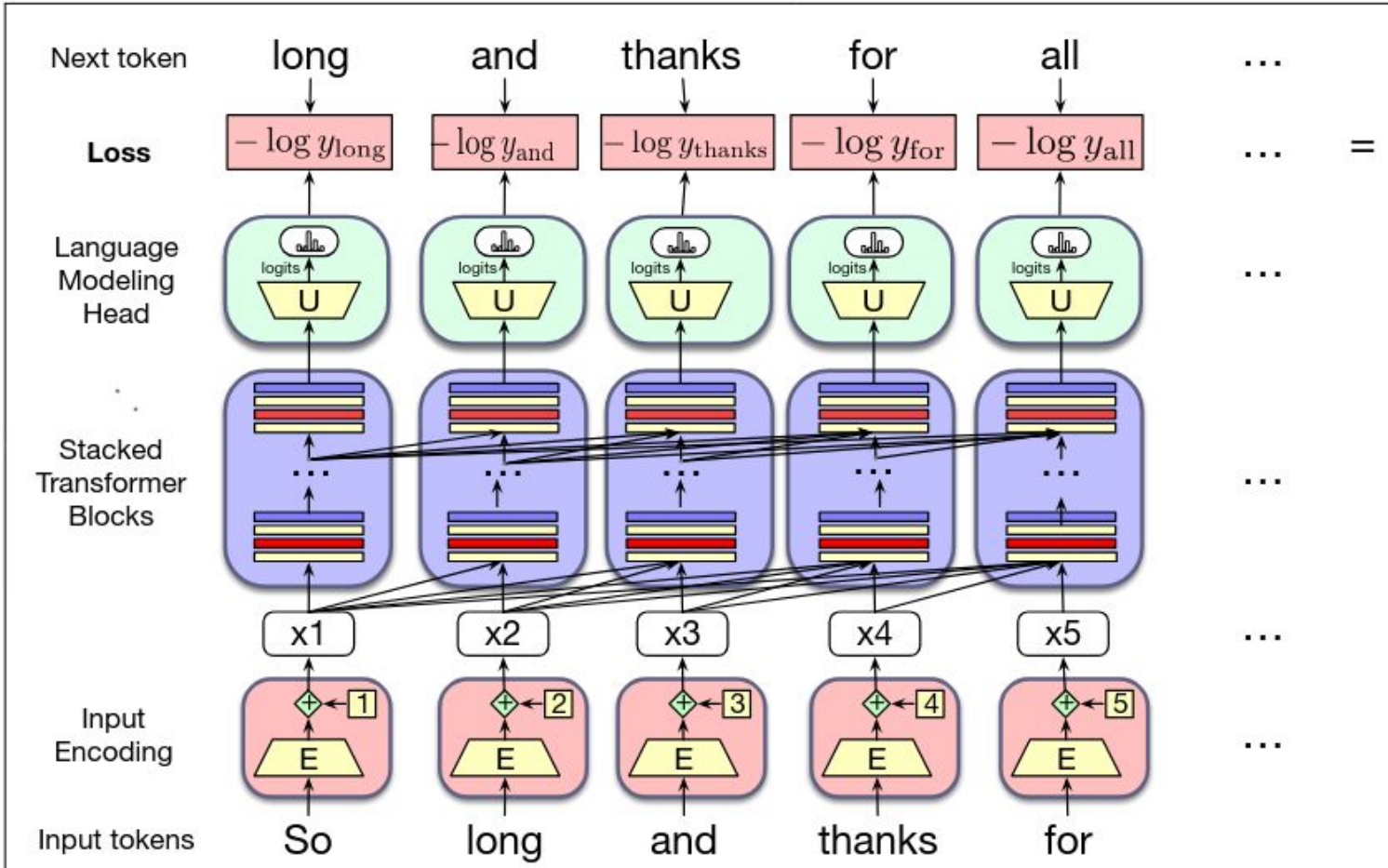


Figure 8.16 Training a transformer as a language model.

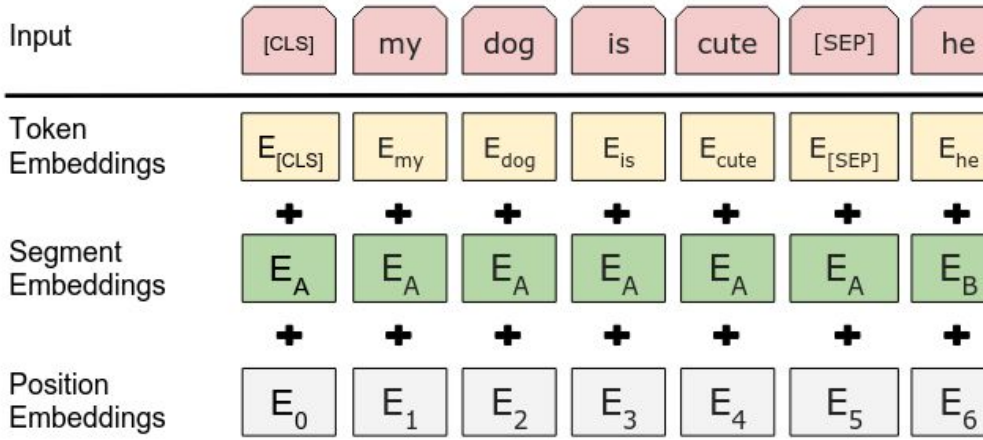


Figure 2: BERT input representation. The input embeddings are the s
tion embeddings and the position embeddings.

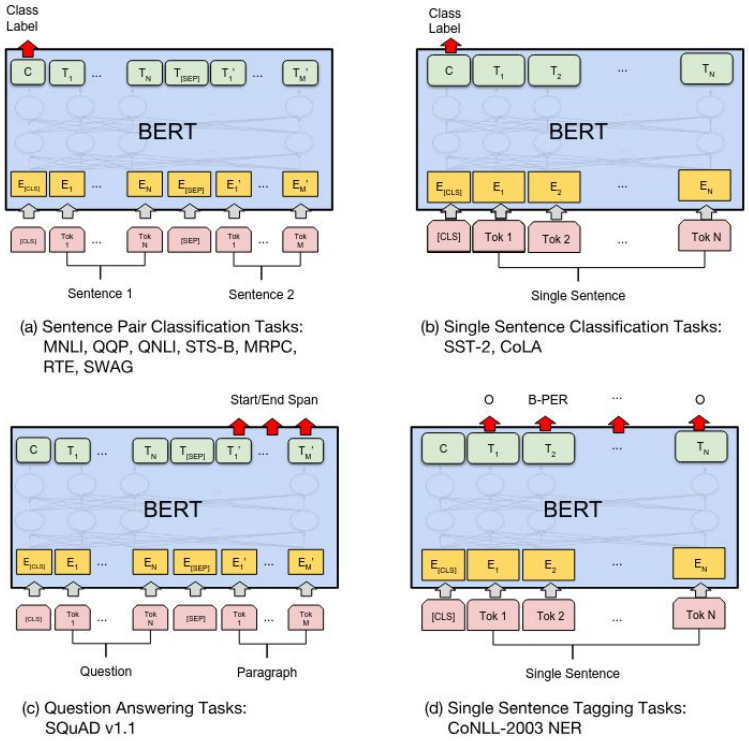


Figure 4: Illustrations of Fine-tuning BERT on Different Tasks.

UNIVERSITY OF MARYLAND
FEARLESSLY FORWARD

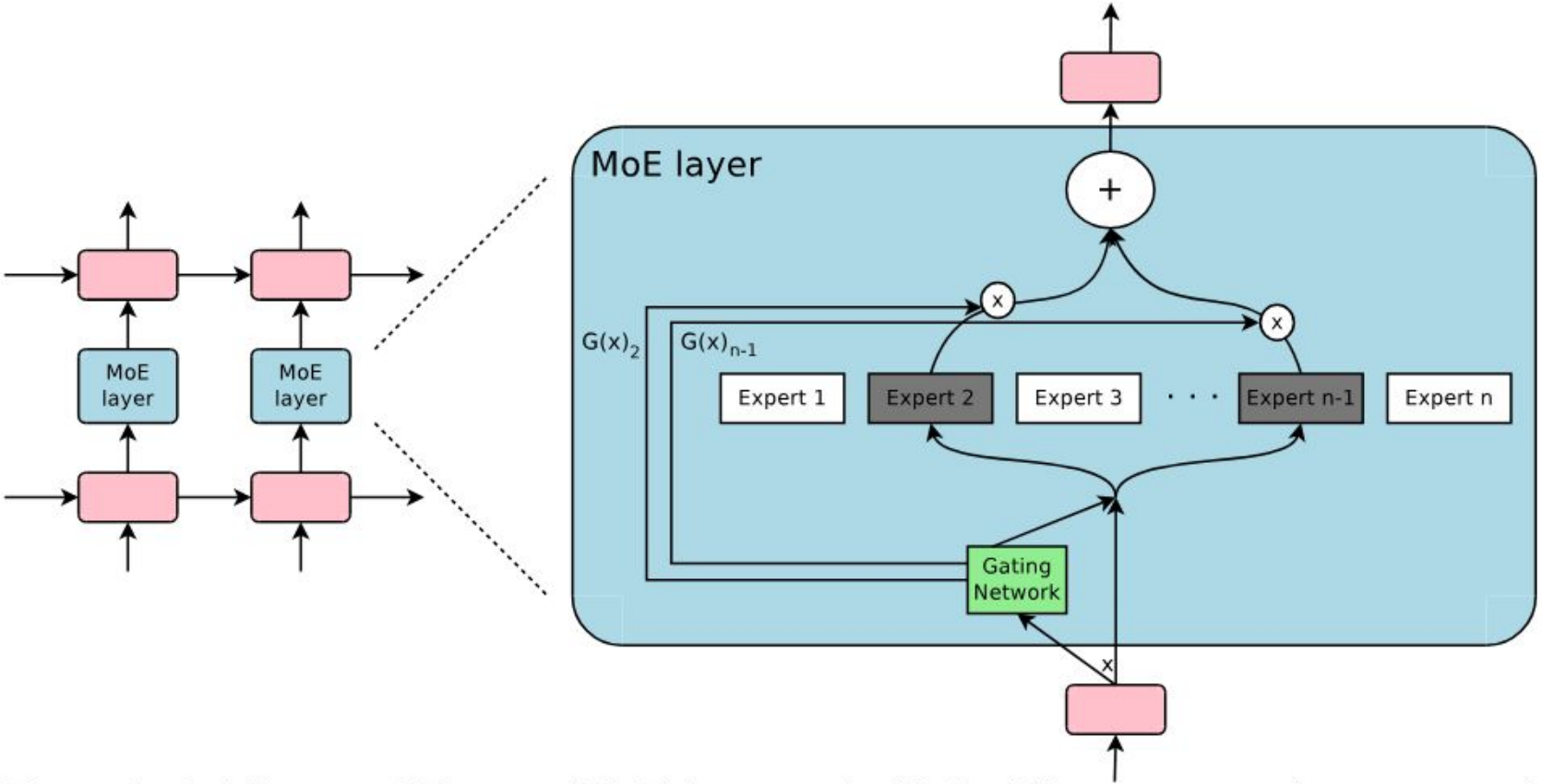
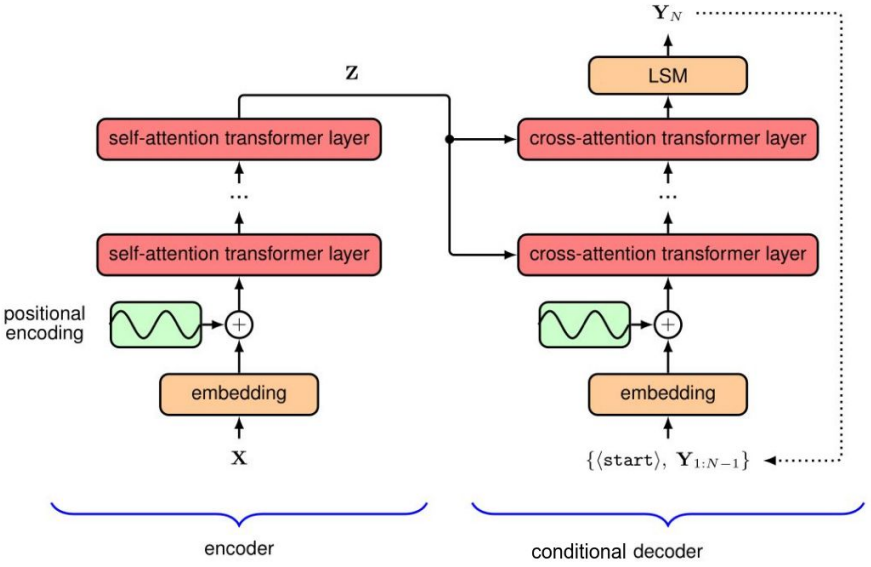
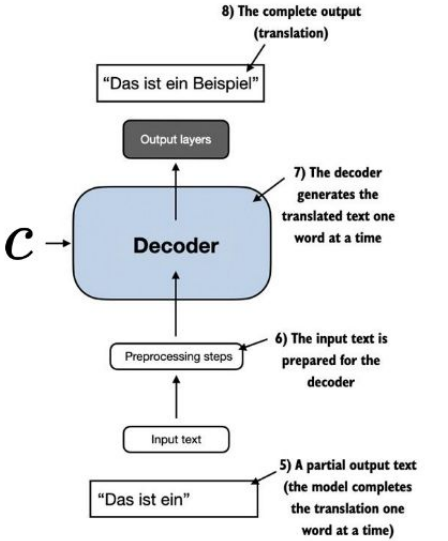


Figure 1: A Mixture of Experts (MoE) layer embedded within a recurrent language model. In this case, the sparse gating function selects two experts to perform computations. Their outputs are modulated by the outputs of the gating network.





Approximate

$$p(x_n | x_1, \ldots, x_{n-1}, c)$$


UNIVERSITY OF
MARYLAND
FEARLESSLY
FORWARD



Multi-token Attention Mechanism

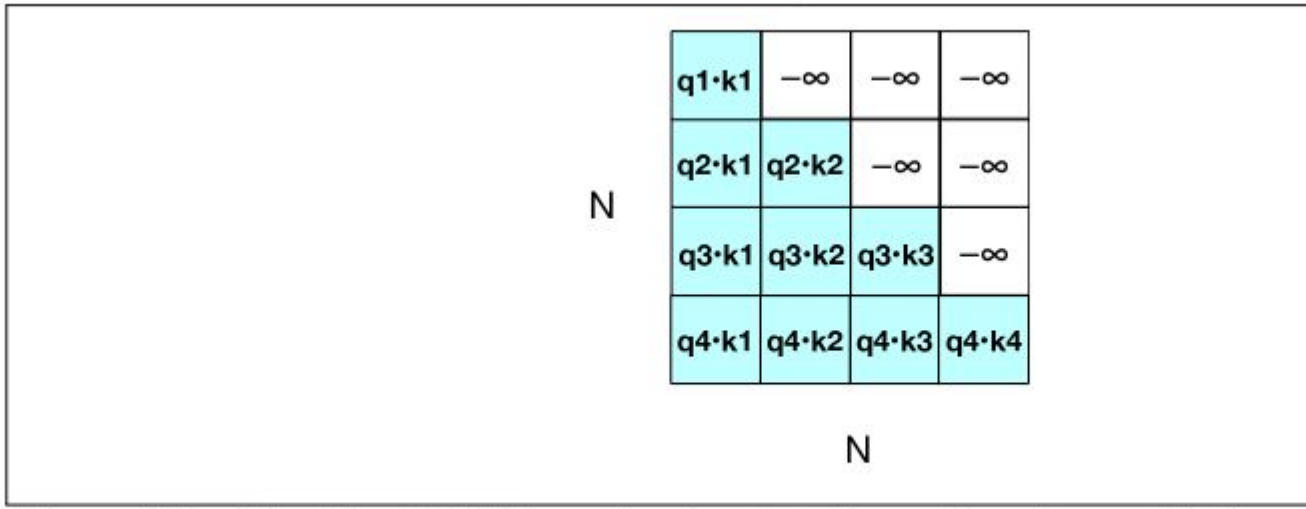


Figure 8.9 The $N \times N$ $\mathbf{QK}^\intercal$ matrix showing the $q_i \cdot k_j$ values, with the up
tion of the comparisons matrix zeroed out (set to $-\infty$, which the softmax wi

UNIVERSITY OF MARYLAND FEARLESSLY FORWARD


UNIVERSITY OF
MARYLAND
FEARLESSLY
FORWARD

