VALDOM - NLP 2 LLM

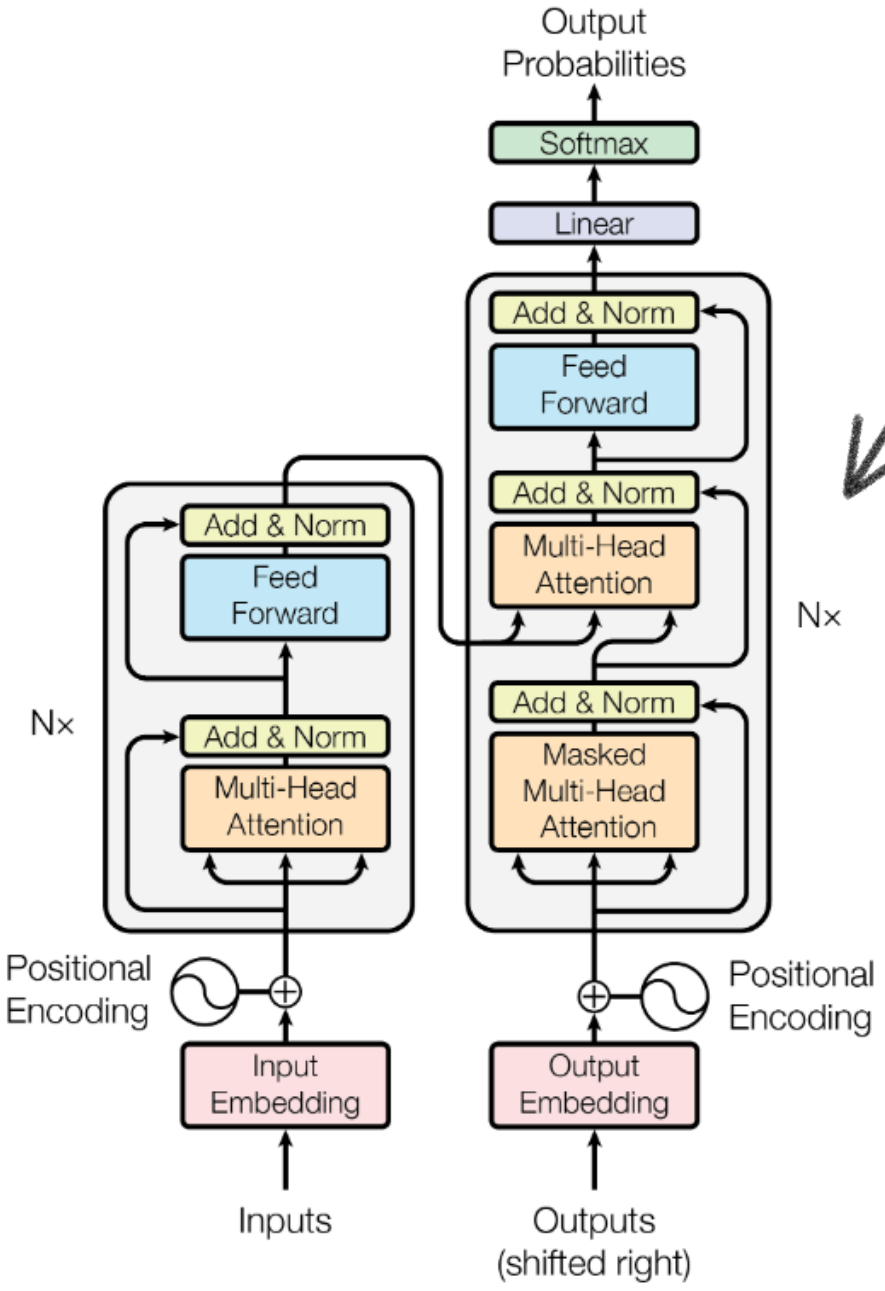
LARGE LANGUAGE MODELS

JOSEBA DALMAU

OBJECTIVE

Master the Transformer architecture for various different NLP tasks

TRANSFORMERS

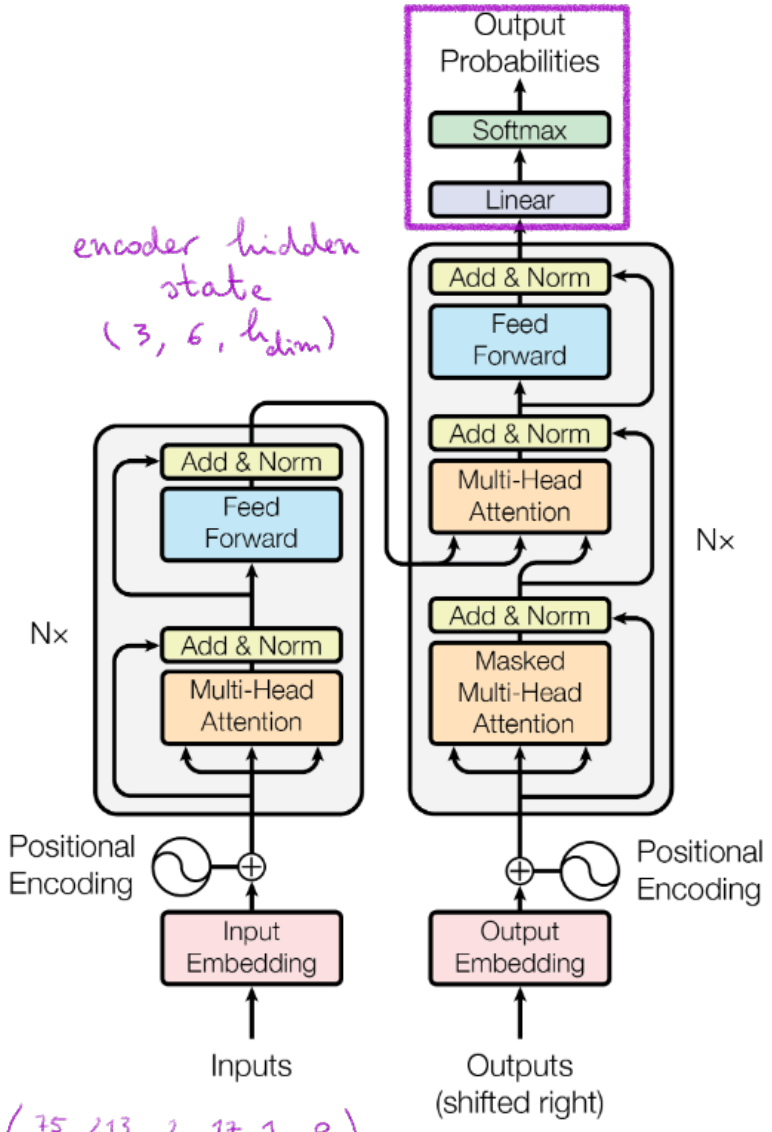


Main figure of the transformer architecture from the original paper "Attention is all you need"

Vaswani et al. 2017

First Week

$\begin{pmatrix} 63, 25, 7, 2, 1 \\ 213, 7, 1, 0, 0 \\ 17, 415, 55, 1, 0 \end{pmatrix}$



encoder hidden state $(3, 6, h_{dim})$

$\begin{pmatrix} 75, 213, 2, 17, 1, 0 \\ 25, 317, 33, 1, 0, 0 \\ 44, 17, 312, 2, 17, 1 \end{pmatrix}$

- Difference between inference / training
- Models/architectures:
  - → Encoder-only
  - → Decoder-only
  - → Encoder-Decoder

LAST WEEK

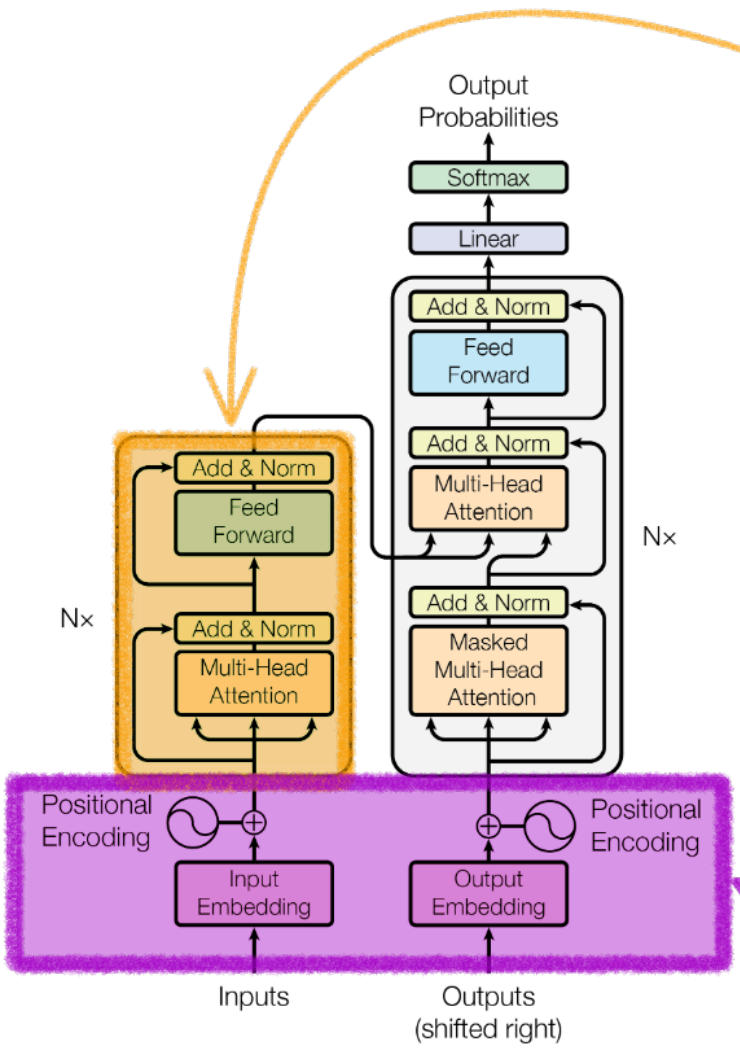


Figure 1: The Transformer - model architecture.

Encoder Block:

- Multi-head self attention
- LayerNorm
- FF network
- Residual Connections

- Input Embedding
- Positional Encoding

Objective

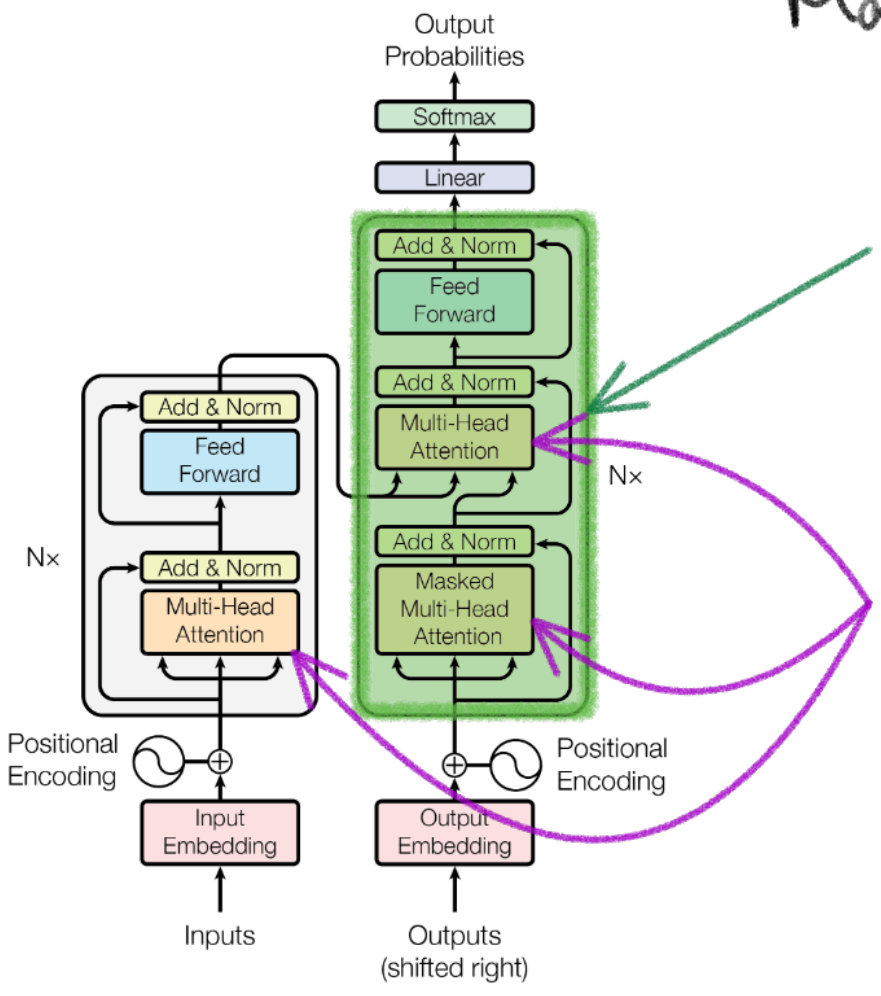


Figure 1: The Transformer - model architecture.

Master what happens:

- inside the decoder transformer blocks

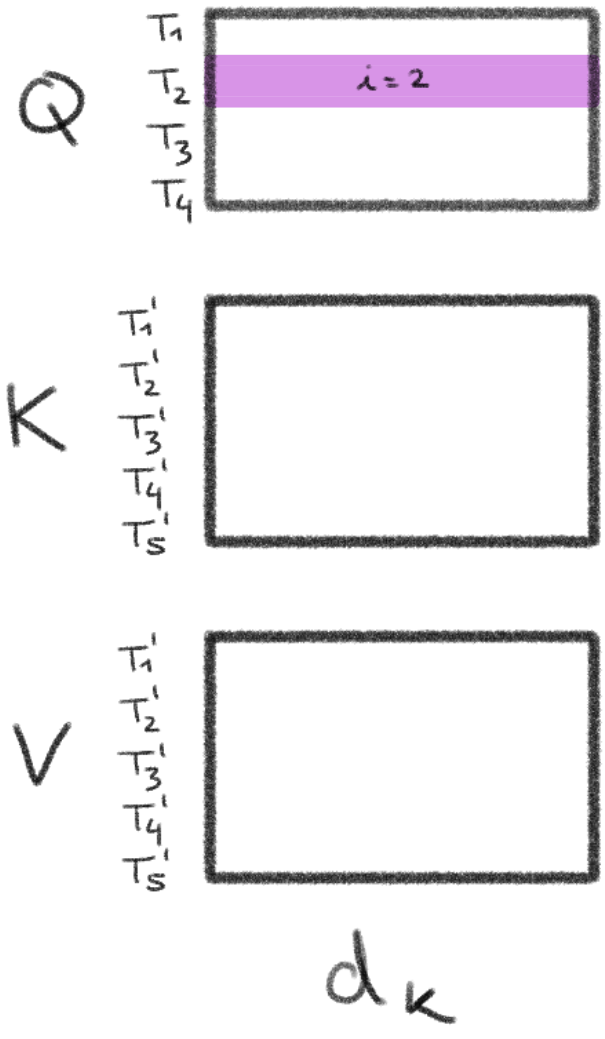
- masking

Scaled Dot-Product Attention

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

Scaled Dot-Product Attention

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



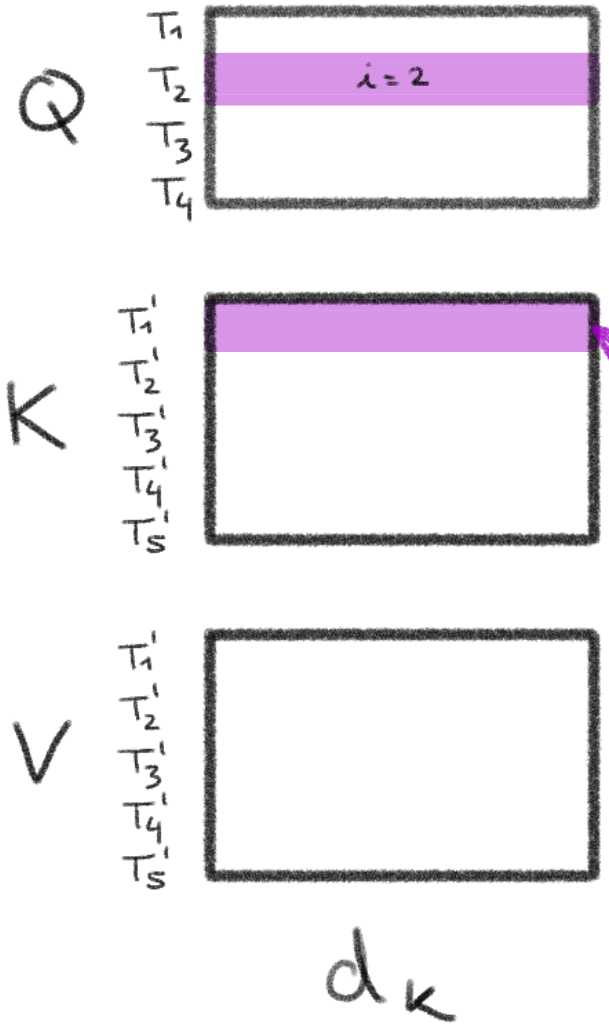
Attention weights

$\Rightarrow$ i-th row

$$[s_{i1} \; s_{i2} \; \cdots \; s_{il'}]$$

Scaled Dot-Product Attention

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



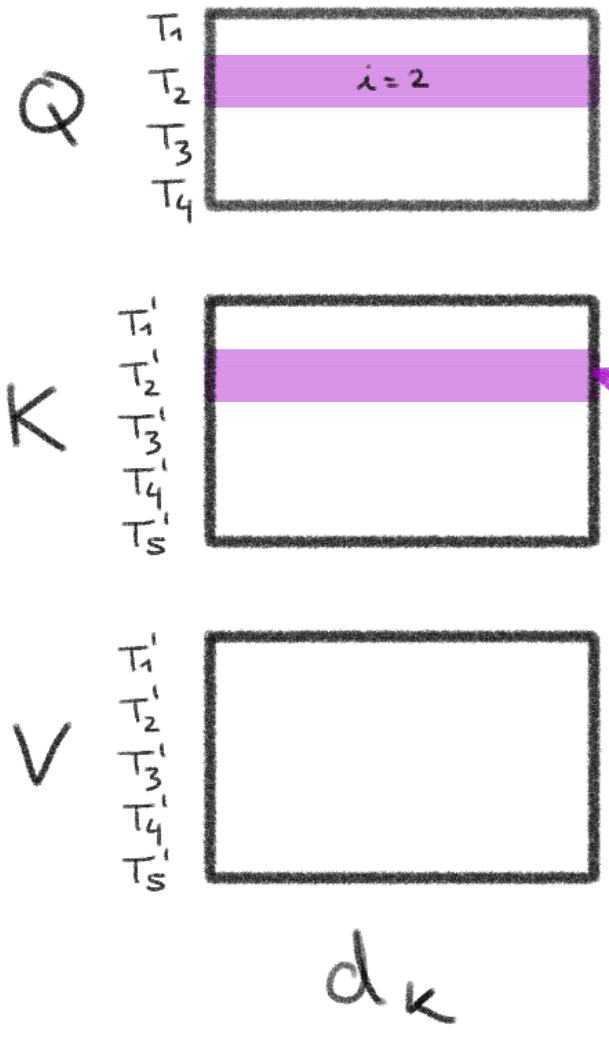
Attention weights

$\Rightarrow$ i-th row

$$[\, s_{i1} \; s_{i2} \; \cdots \; s_{il'} \,]$$

Scaled Dot-Product Attention

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



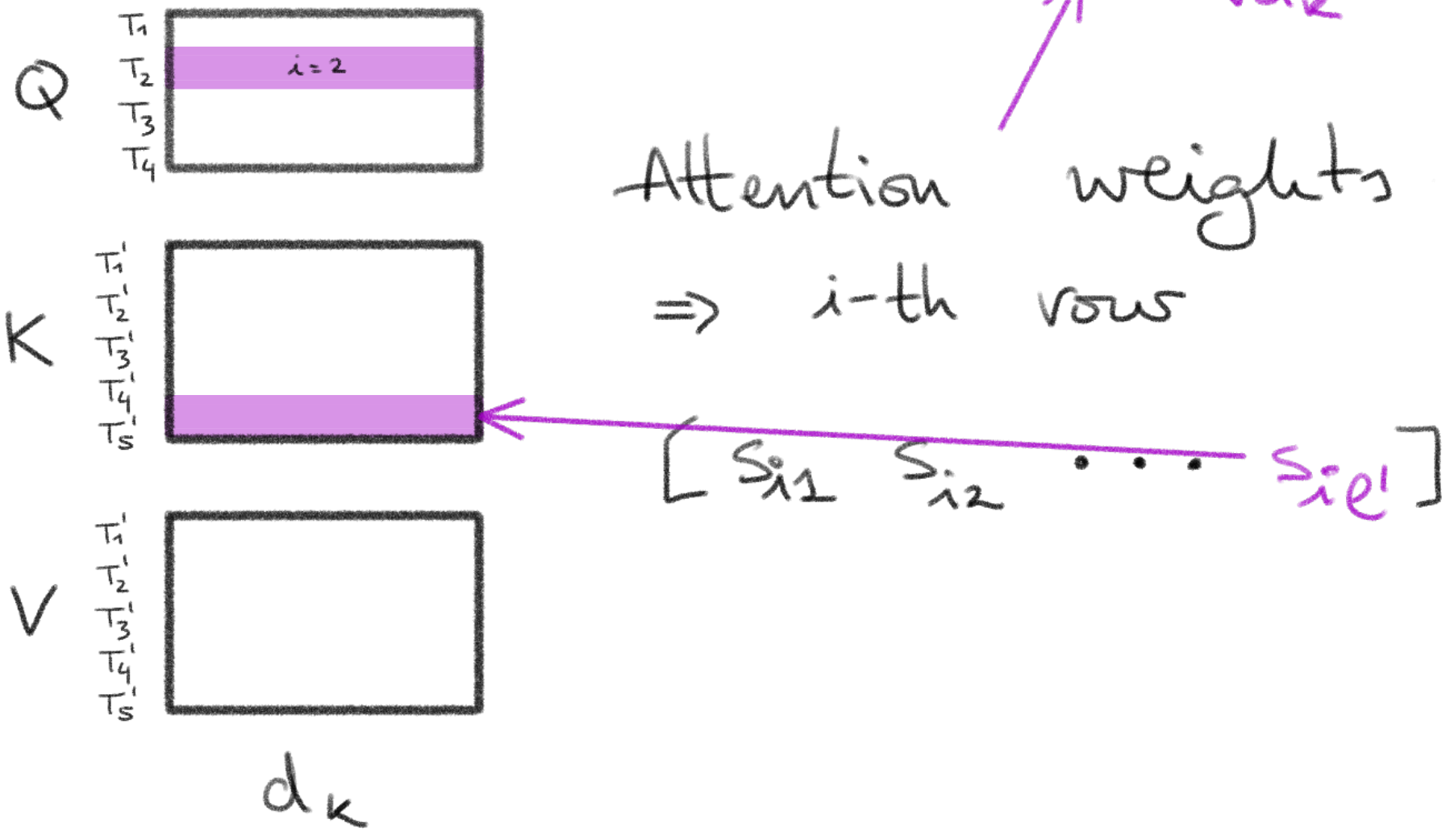
Attention weights

$\Rightarrow$ i-th row

$$[\, s_{i1} \; s_{i2} \; \cdots \; s_{iL'} \,]$$

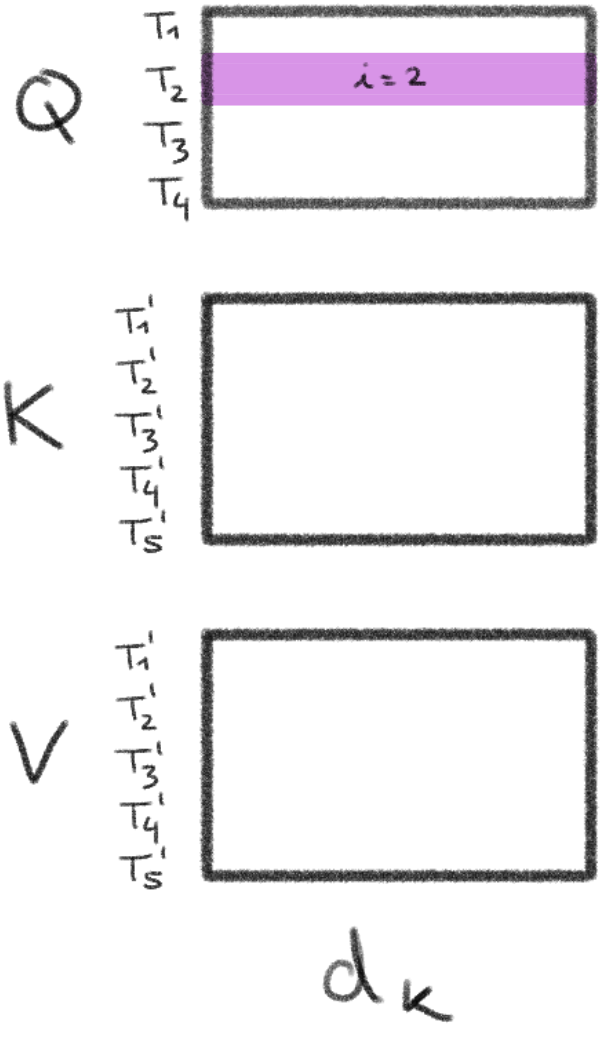
Scaled Dot-Product Attention

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



Masked Attention

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



Attention weights

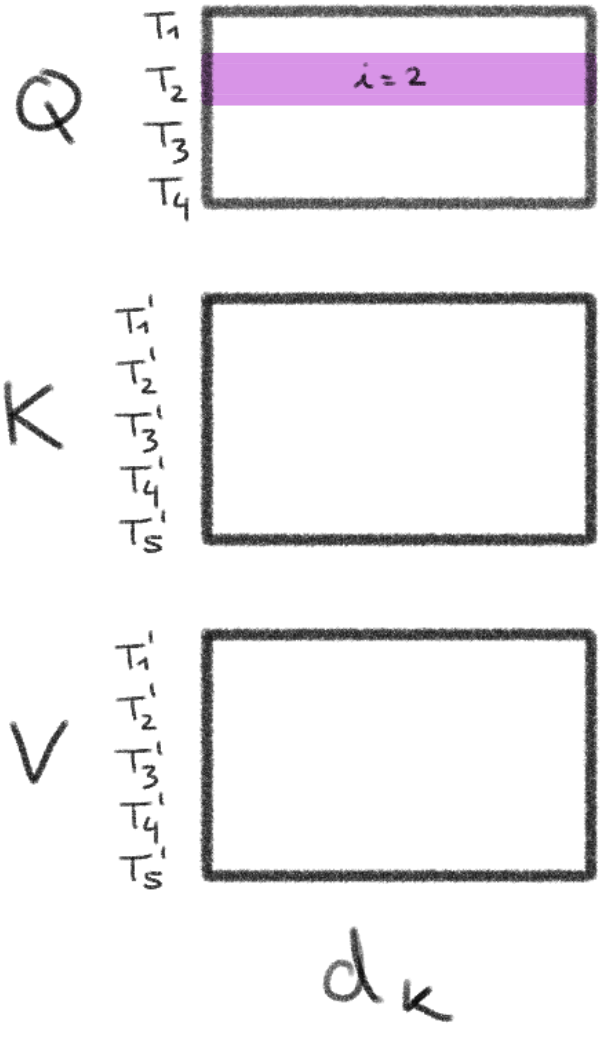
$\Rightarrow$ i-th row

$$[s_{i1} \quad s_{i2} \quad \cdots \quad s_{iT'}]$$

We want: the i-th query NOT to attend to the j-th Key!

Masked Attention

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



Attention weights

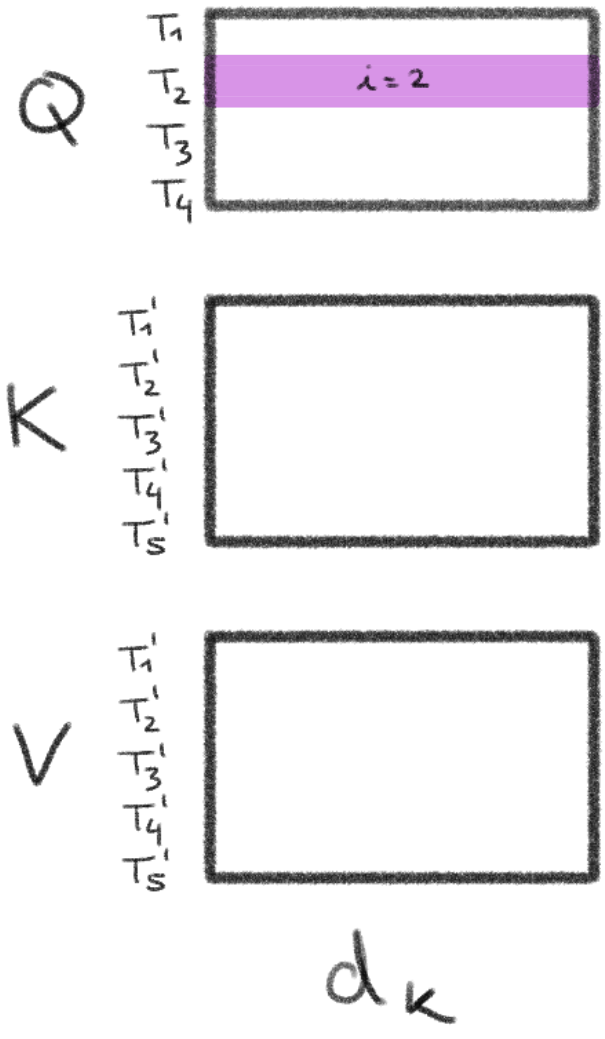
$\Rightarrow$ i-th row

$$[\, s_{i1} \;\; s_{i2} \;\; \cdots \;\; s_{iL'} \,]$$

Sol: set $s_{ij} = 0$

MASKED ~~ATTENTION~~

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



Attention weights

$\Rightarrow$ i-th row

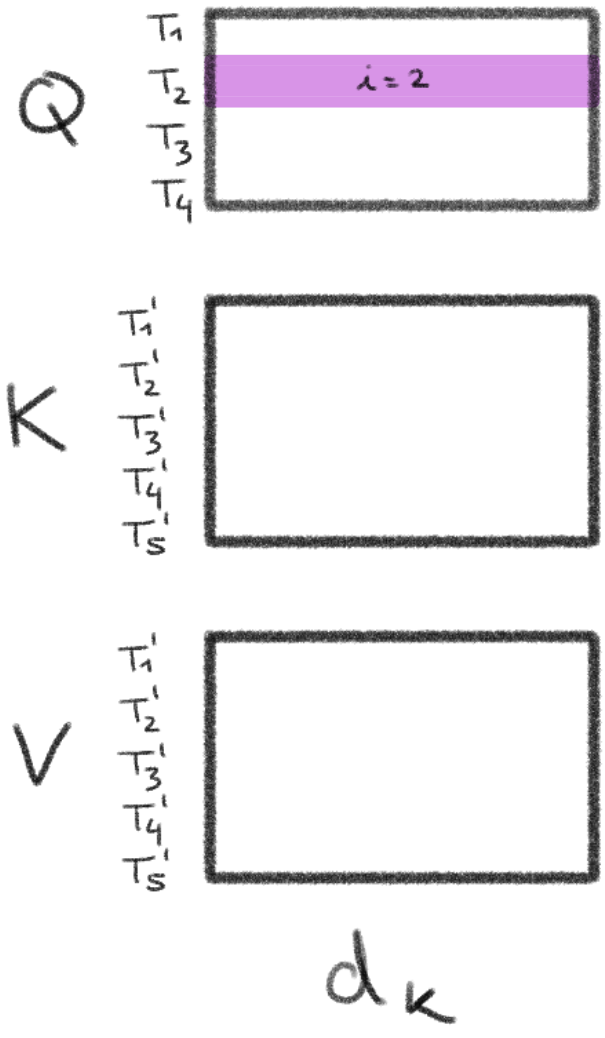
$$[\, s_{i1} \;\; s_{i2} \;\; \cdots \;\; s_{il'} \,]$$

Sol: set $s_{ij} = 0$

⚠ Pb: $\sum_j s_{ij} \neq 1$

Masked Attention

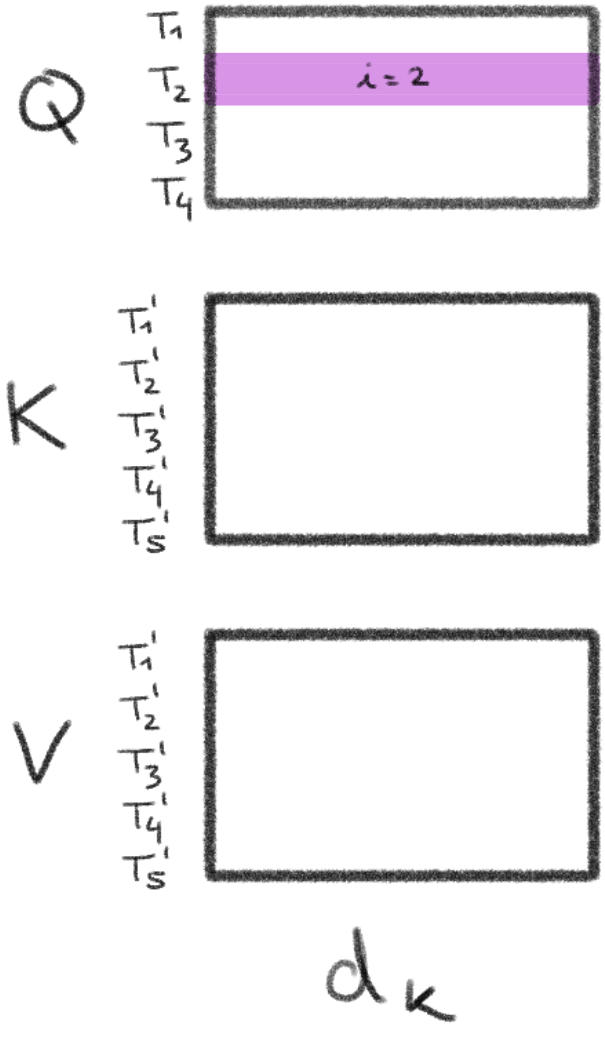
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



Sol: Modify the matrix QK^T instead!

Masked Attention

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

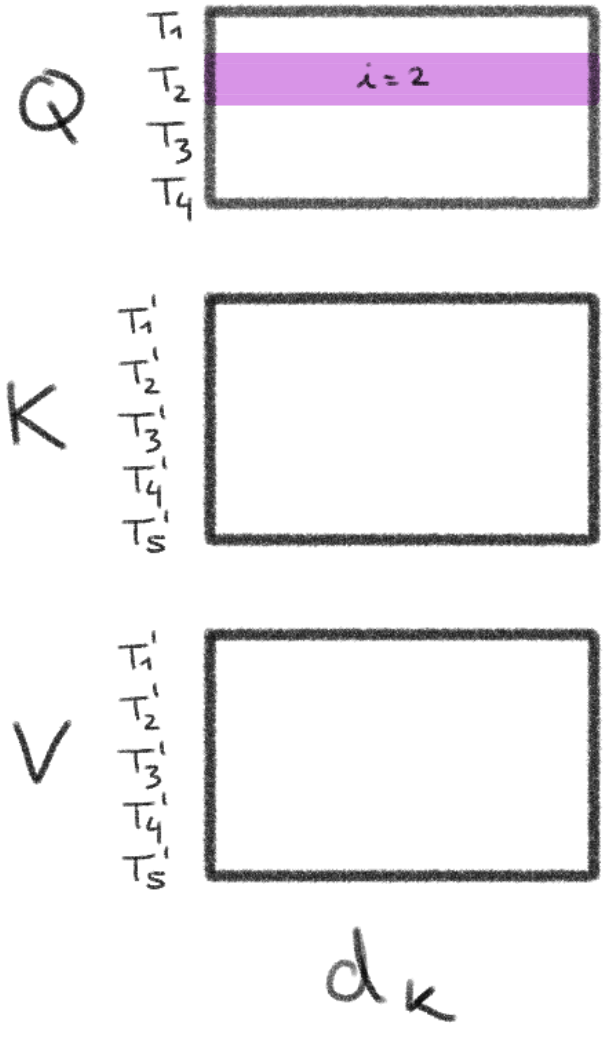


Sol: Modify the matrix QK^T instead!

Qn:- What value should we change $(QK^T)_{ij}$ to? (so that $s_{ij} = 0$)

MASKED ~~ATTENTION~~

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



Sol: Modify the matrix QK^T instead!

Answer: set

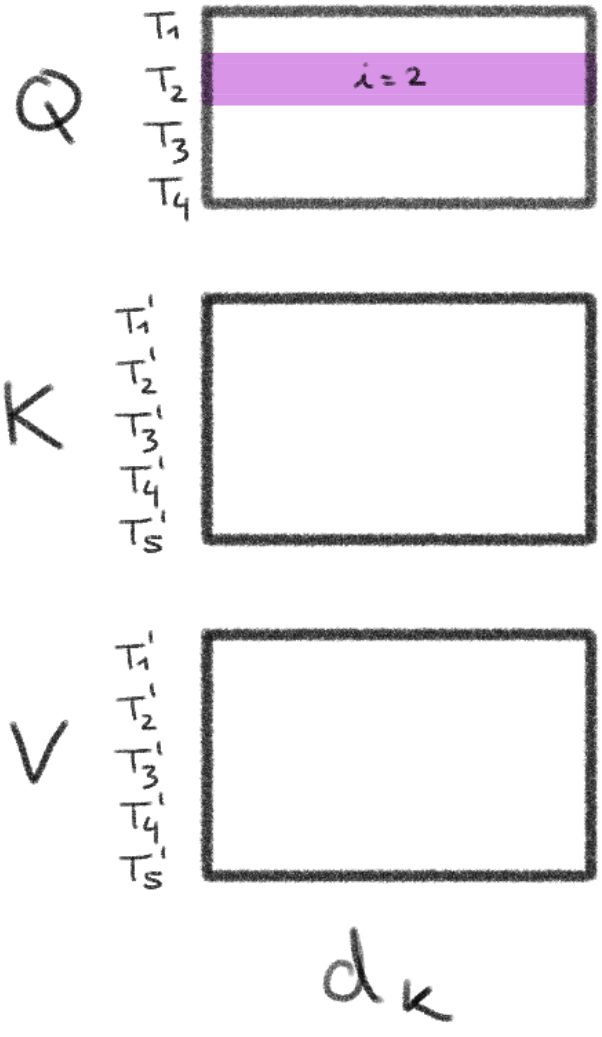
$$(QK^T)_{ij} = -\infty$$

Masked Attention

$$\text{Attention}(Q, K, V, M) =$$

MASKED ~~ATTENTION~~

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



Sol: Modify the matrix QK^T instead!

Answer: set

$$(QK^T)_{ij} = -\infty$$

Masked Attention

$$\text{Attention}(Q, K, V, M) =$$

$$\text{softmax}\left(\frac{QK^T + \tilde{M}}{\sqrt{d_k}}\right) V$$

Masked Attention

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

$$\tilde{M}_{ij} = \begin{cases} -\infty & \text{if } M_{ij} = 0 \\ 0 & \text{if } M_{ij} = 1 \end{cases}$$

query i does NOT attend Key j

query i DOES attend Key j

Padding Masks

The sandwich is delicious <eos>

I am bored <eos> <pad> <pad> <pad>

Padding Masks

The sandwich is delicious <eos>

I am bored <eos> <pad> <pad> <pad>

Padding Masks

The sandwich is delicious <eos>

I am bored <eos> <pad> <pad> <pad>

(M_1, M_2)

for seq 1 for seq 2

Padding Masks

The sandwich is delicious <eos>

I am bored <eos> <pad> <pad> <pad>

(M_1, M_2)

$$M_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Padding Masks

In code:

Tensor of shape (b, H, ℓ, ℓ)

nb. of seqs in batch

nb. of heads

nb. of tokens

Padding Masks

In code:

Tensor of shape (b, H, ℓ, ℓ)

same mask for all heads

same mask for all rows

Padding Masks

In code:

Tensor of shape (b, H, ℓ, ℓ)

same mask for all heads

same mask for all rows

$\Rightarrow$ input mask of shape (b, ℓ)

containing 1s for tokens $\neq$ <pad>

0s for tokens = <pad>

Padding Masks

In code:

Tensor of shape (b, H, ℓ, ℓ)

same mask for all heads

same mask for all rows

$\Rightarrow$ input mask of shape (b, ℓ)

containing 1s for tokens $\neq$ <pad>

0s for tokens $=$ <pad>

$\Rightarrow$ replicate to shape (b, H, ℓ, ℓ)

Causal Mask

The window is closed <eos>

Causal Mask

Causal Mask

Causal Mask

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Causal Mask

In code:

Tensor of shape (b, H, ℓ, ℓ)

same mask for all seqs

same mask for all heads

Causal Mask

In code:

Tensor of shape (b, H, ℓ, ℓ)

same mask for all seqs

same mask for all heads

$\Rightarrow$ input mask of shape (ℓ, ℓ)

Causal Mask

In code:

Tensor of shape (b, H, ℓ, ℓ)

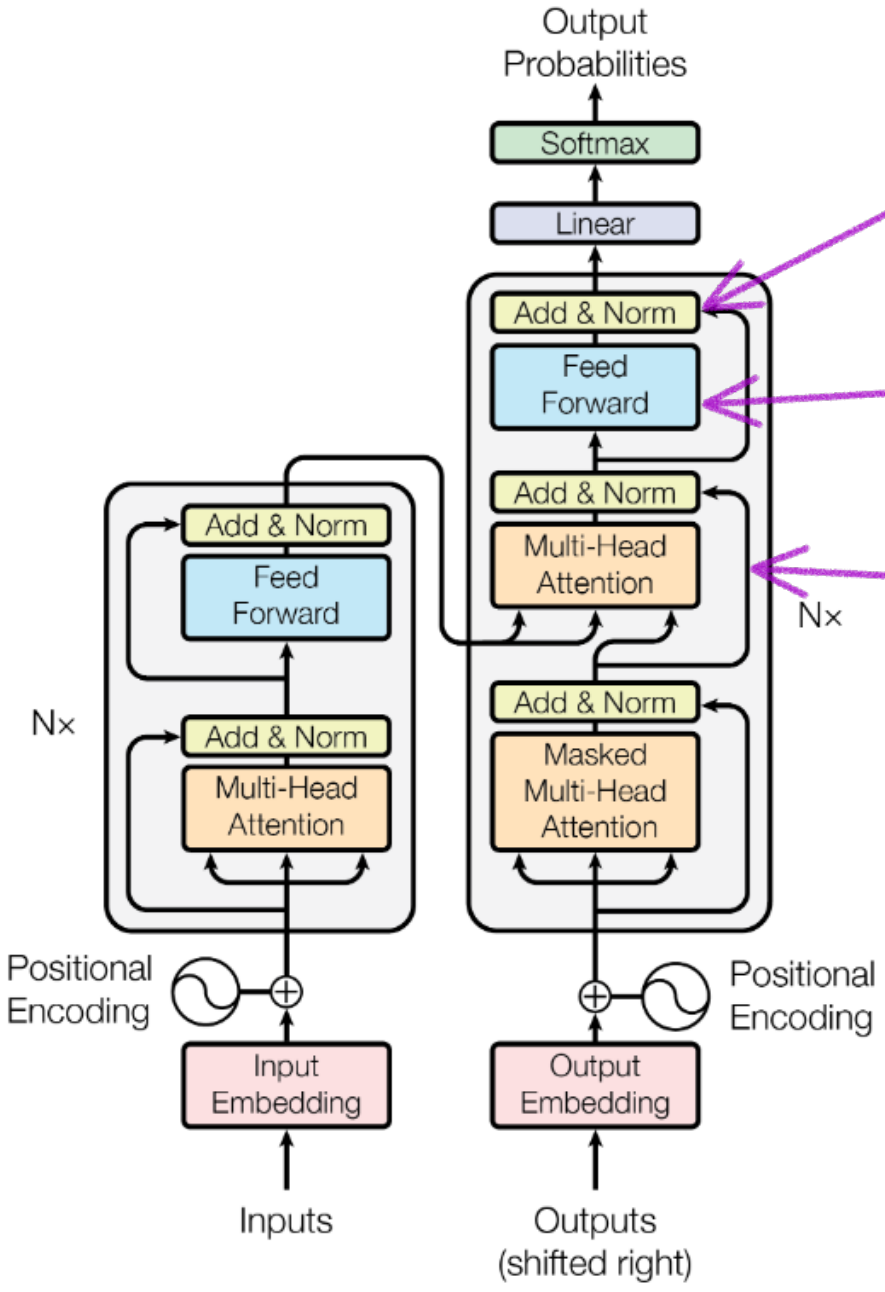
same mask for all seqs

same mask for all heads

$\Rightarrow$ input mask of shape (ℓ, ℓ)

$\Rightarrow$ replicate to shape (b, H, ℓ, ℓ)

Decoder Block



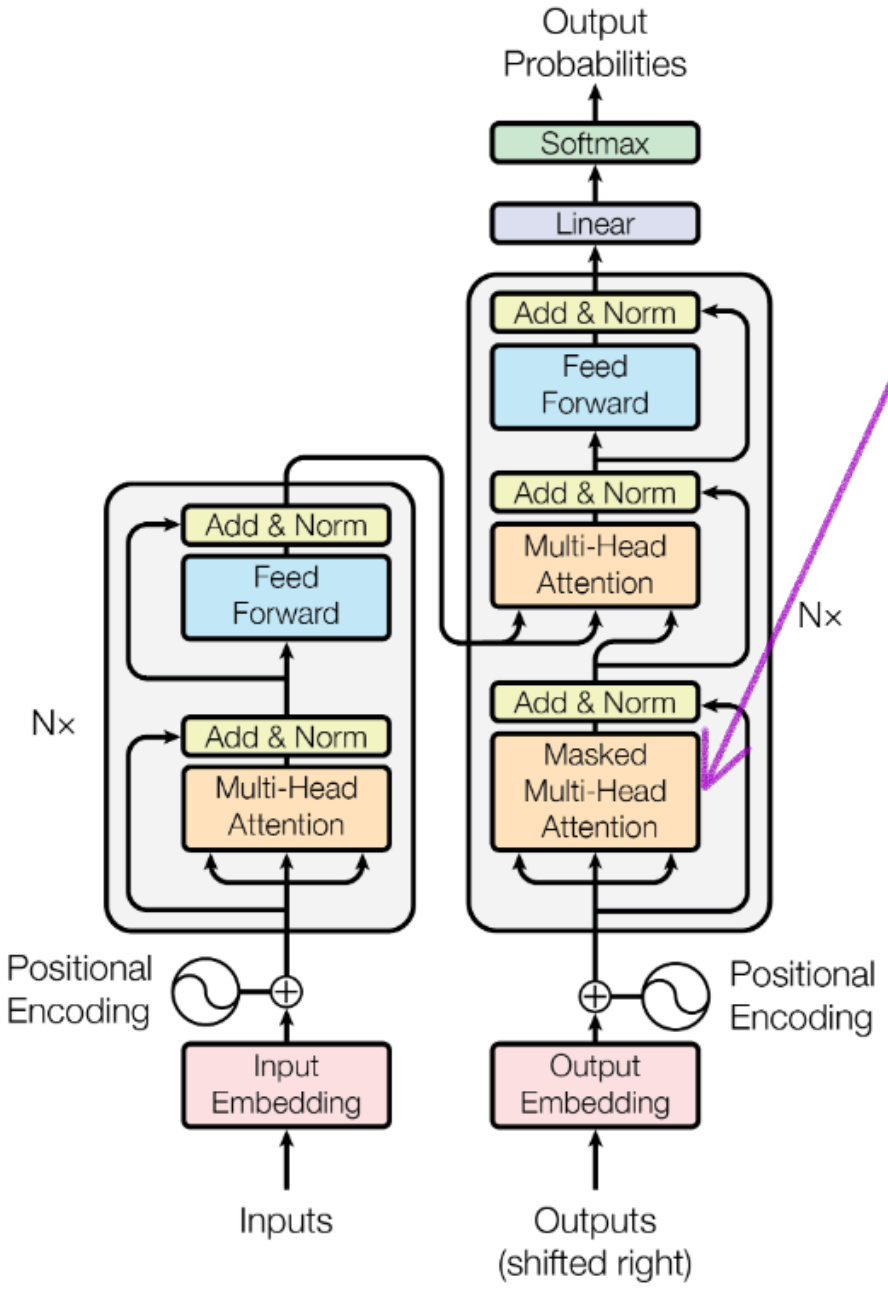
Layer Norm

Feed-forward module

Residual Connections

Exactly the same as for the decoder!

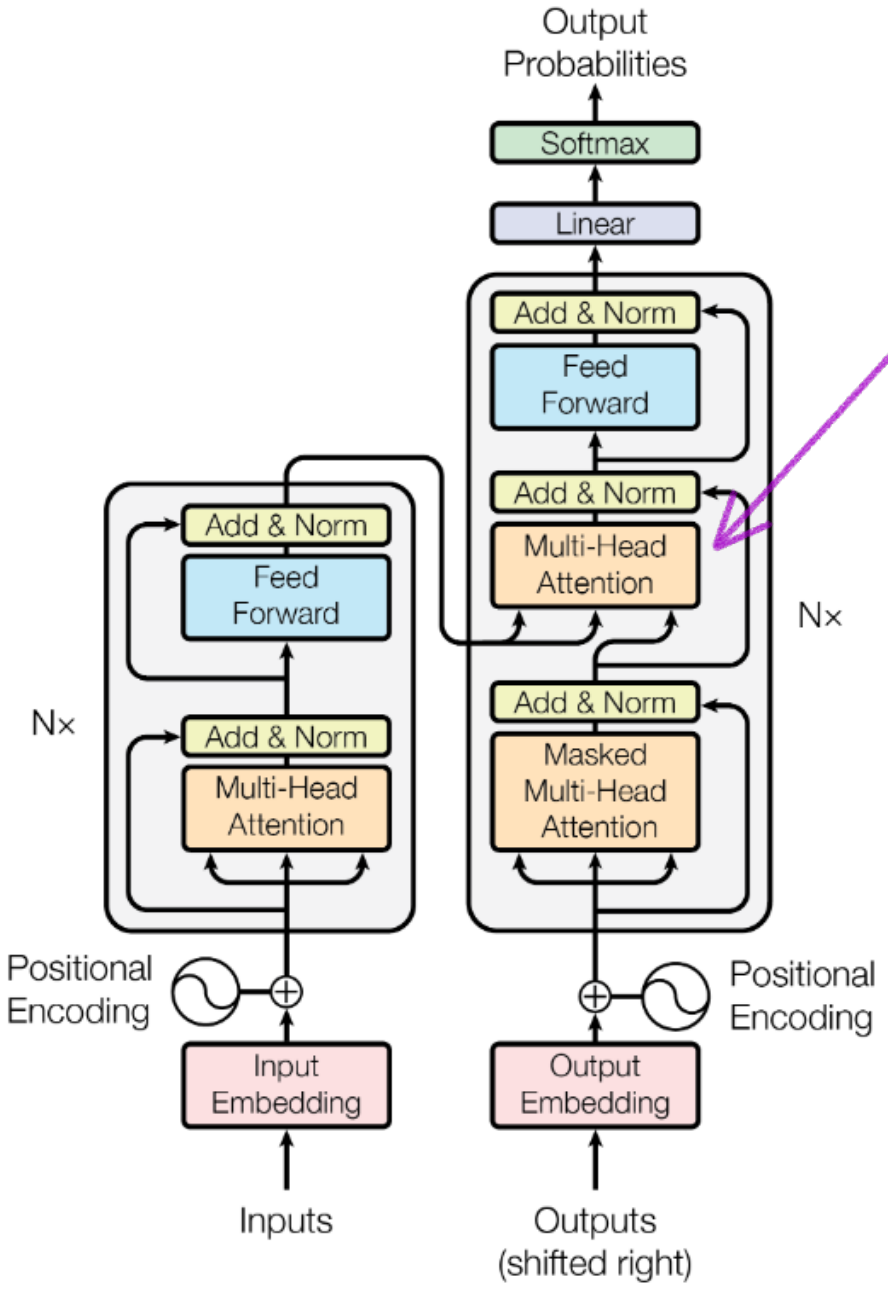
Decoder Block



Masked Multi-head (self) attention:

Self attention with a causal mask

Decoder Block

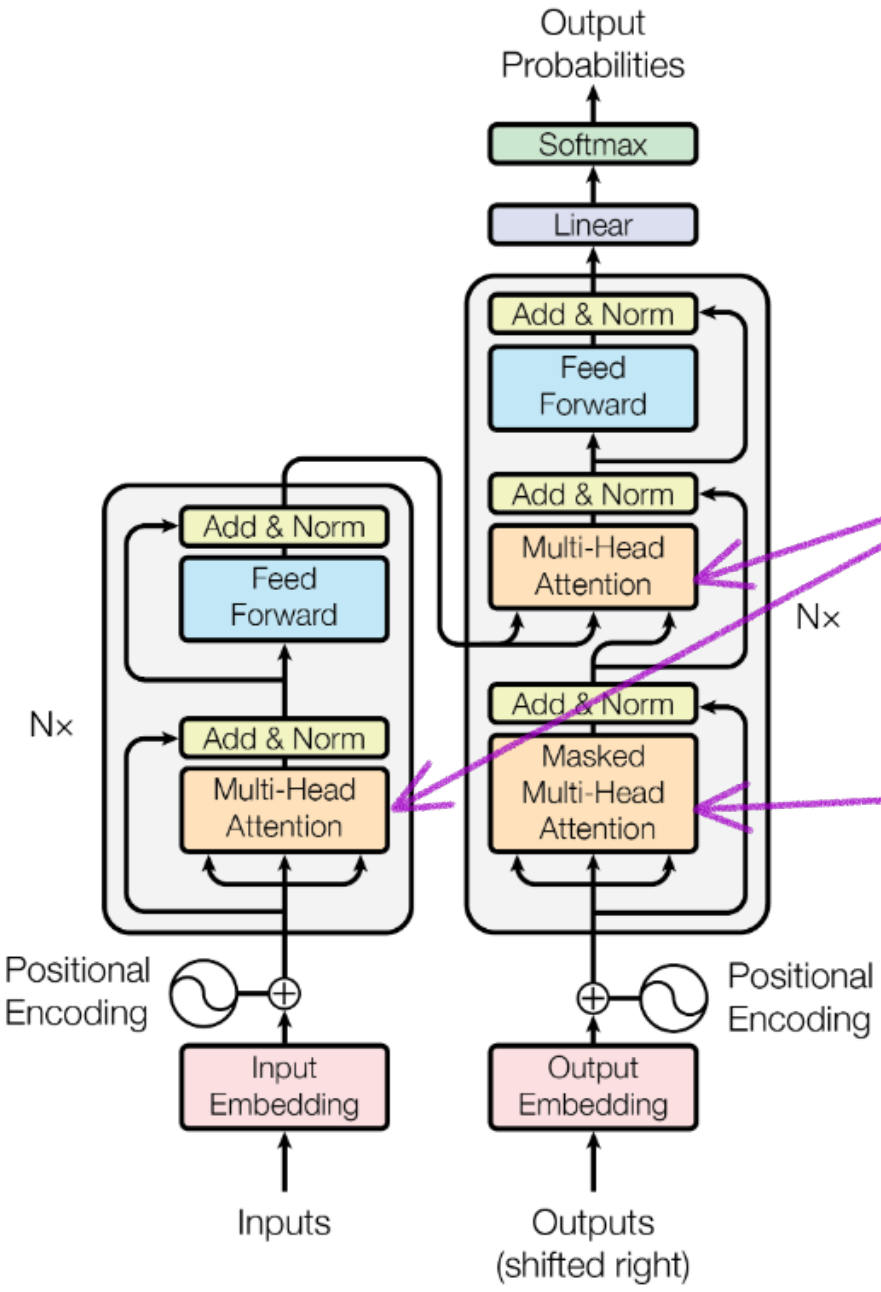


Multi-head (cross) attention:

Queries → from decoder input

Keys, Values → from encoder hidden state

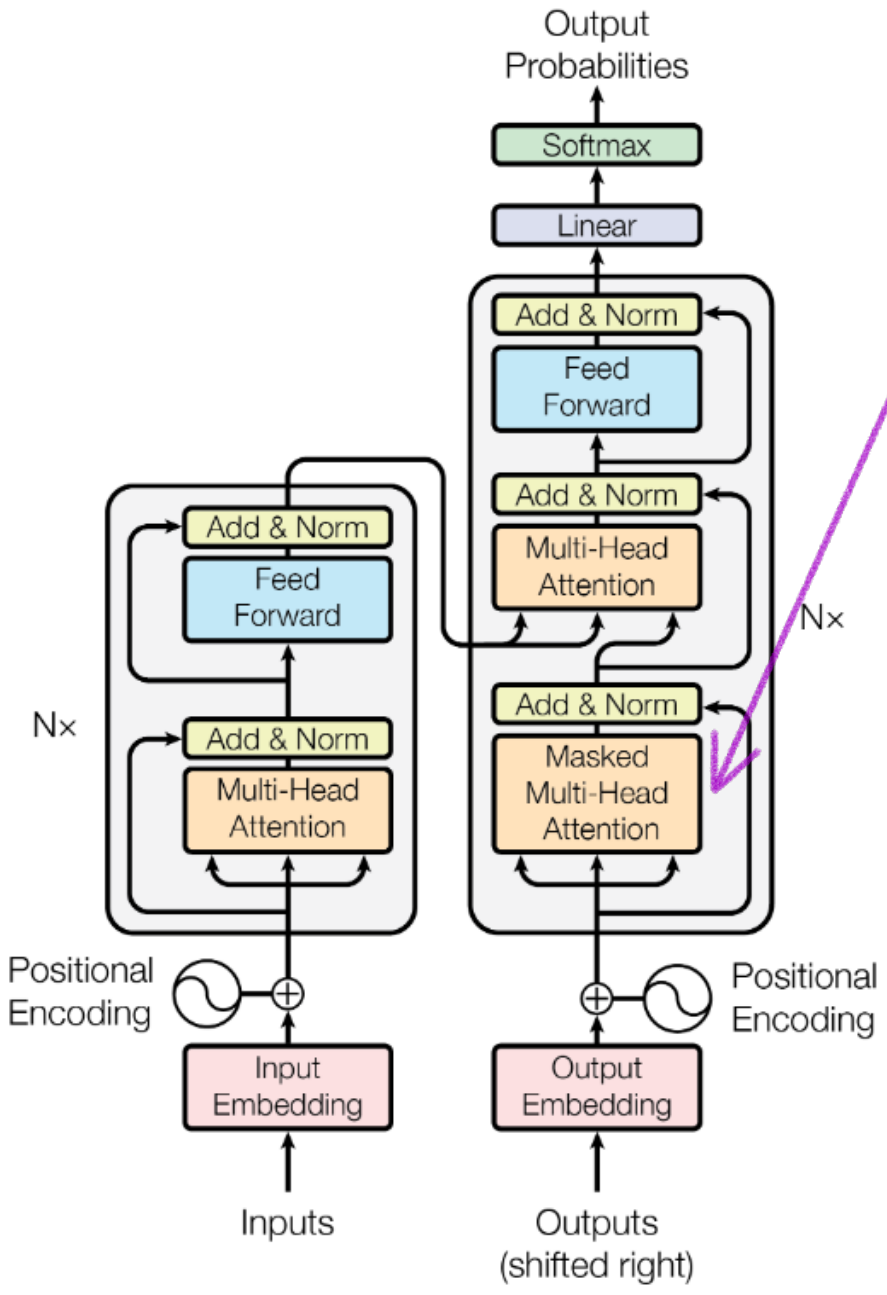
Padding Masks Everywher



encodder input padding mask

decoder input padding mask

Combining Masks



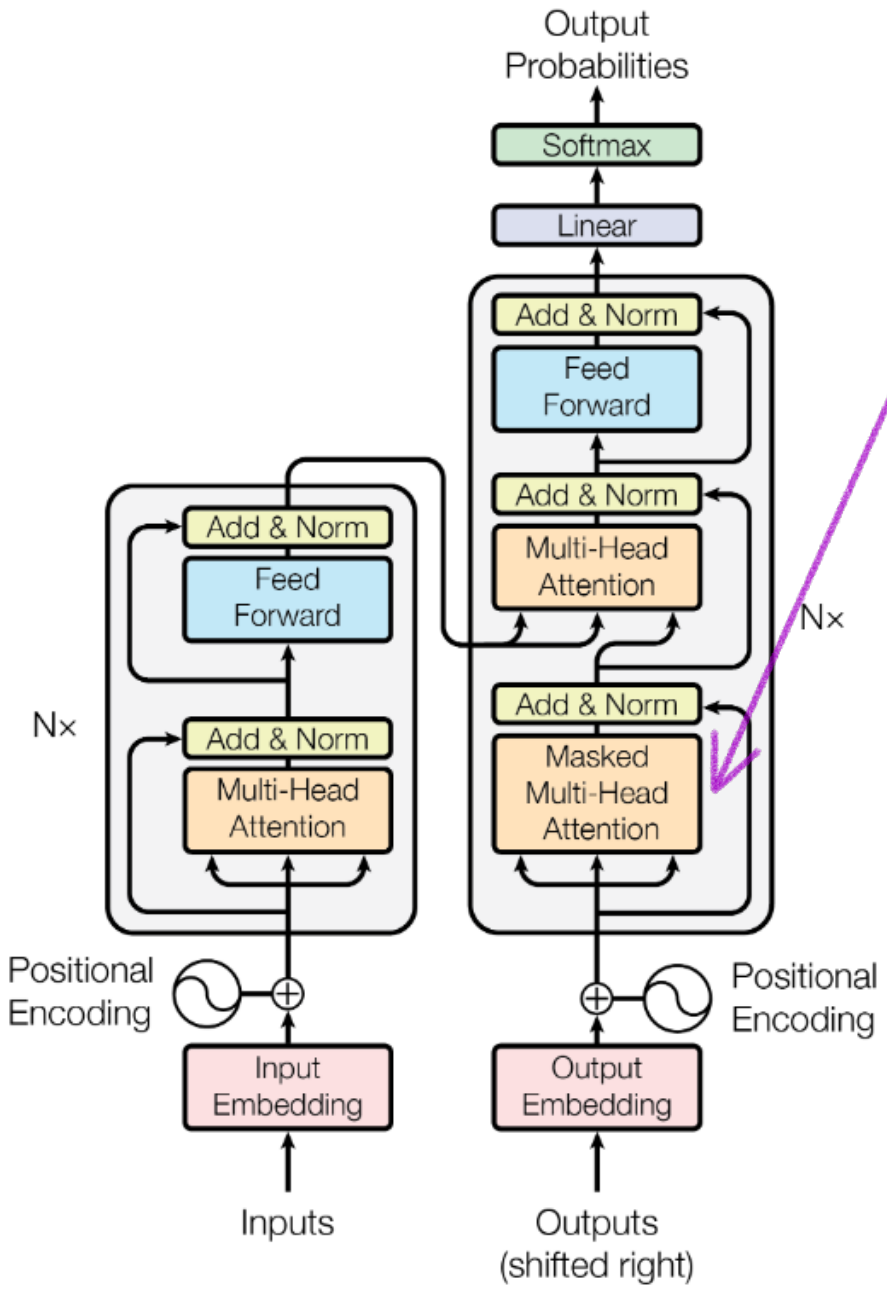
Padding mask
+
Causal mask

$$M_1 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$M_1 \,\&\, M_2 =$

attend if both masks = 1

Combining Masks



Padding mask
+
Causal mask

$$M_1 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$M_1 \,\&\, M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

attend if both masks = 1

Kahoot time!

Question Time

- Write down something you understood well

- Write down something you did not fully understand

- Write something you would like to know more about

SUMMARY

- What is the purpose of a padding mask? And of a causal mask?

- In what layers do we use padding masks? How are those padding masks obtained?

- In what layers do we use causal masks? How are those causal masks obtained?

- How do we combine two different attention masks?

Extra Resources:

→ "The illustrated transformer" J. Alammar

→ "The annotated transformer"

→ "Understanding the attention mechanism in sequence models" J. Jordan

→ "Understanding the transformer architecture for neural networks" J. Jordan

→ LLM Visualization by B. Bycroft