

VALDOM - NLP 2 LLM

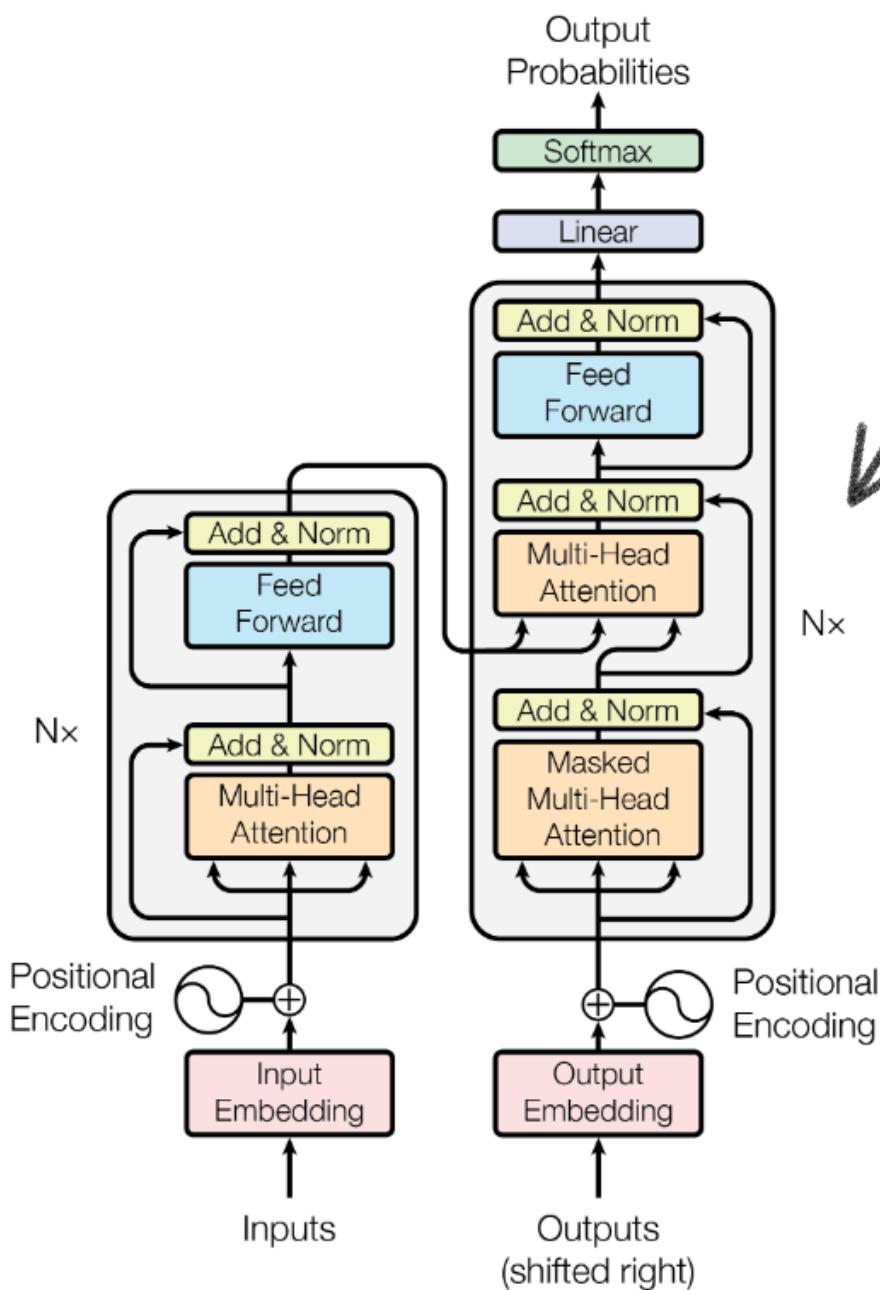
LARGE LANGUAGE MODELS

JOSEBA DALMAU

OBJECTIVE

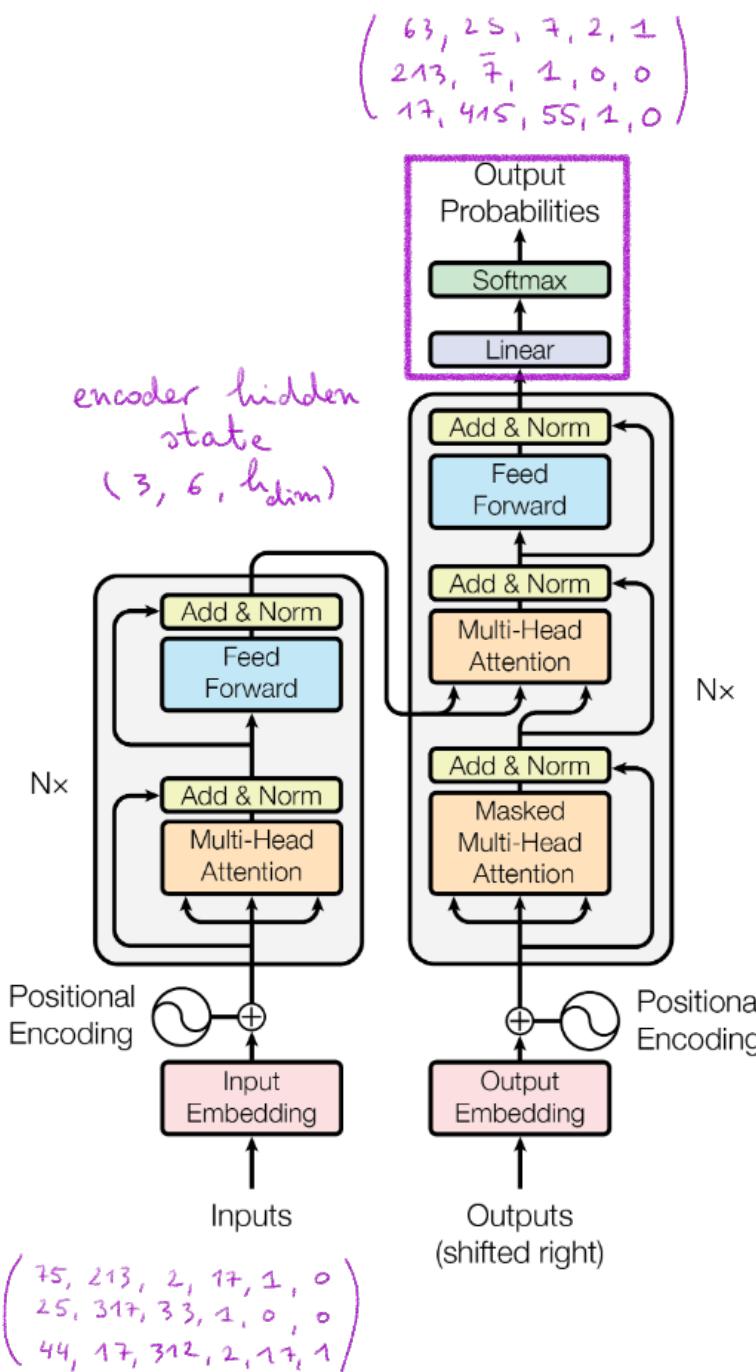
Master the Transformer
architecture for various
different NLP tasks

TRANSFORMERS



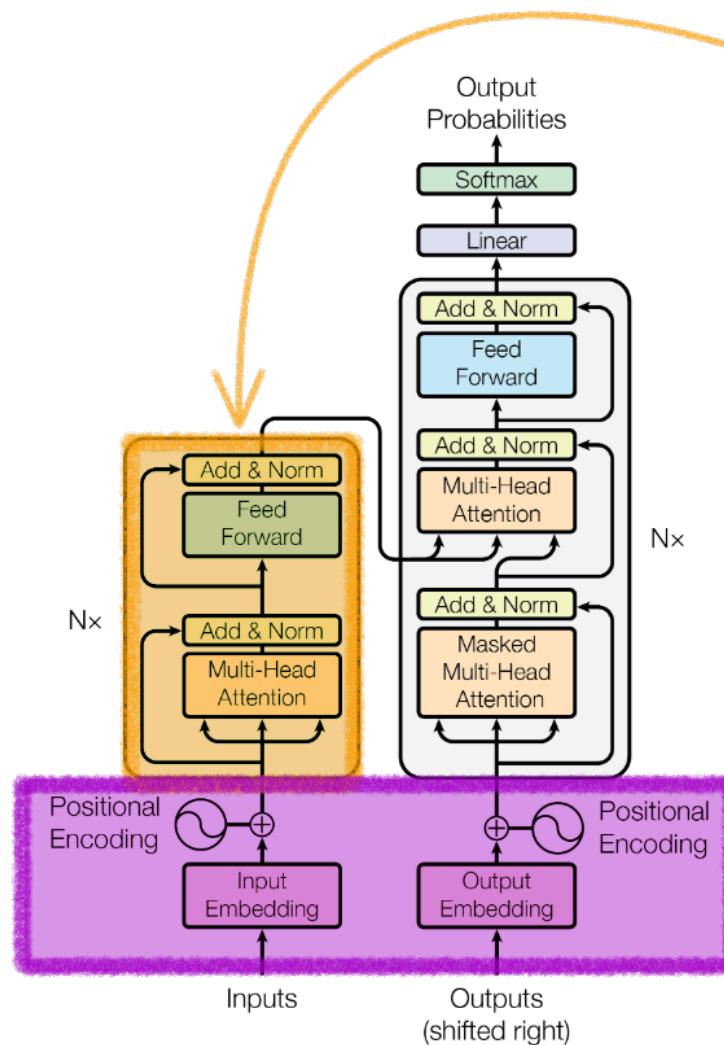
Main figure of
the transformer
architecture from
the original paper
"Attention is all
you need"
Vaswani et al. 2017

FIRST LECTURE



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

LAST LECTURE



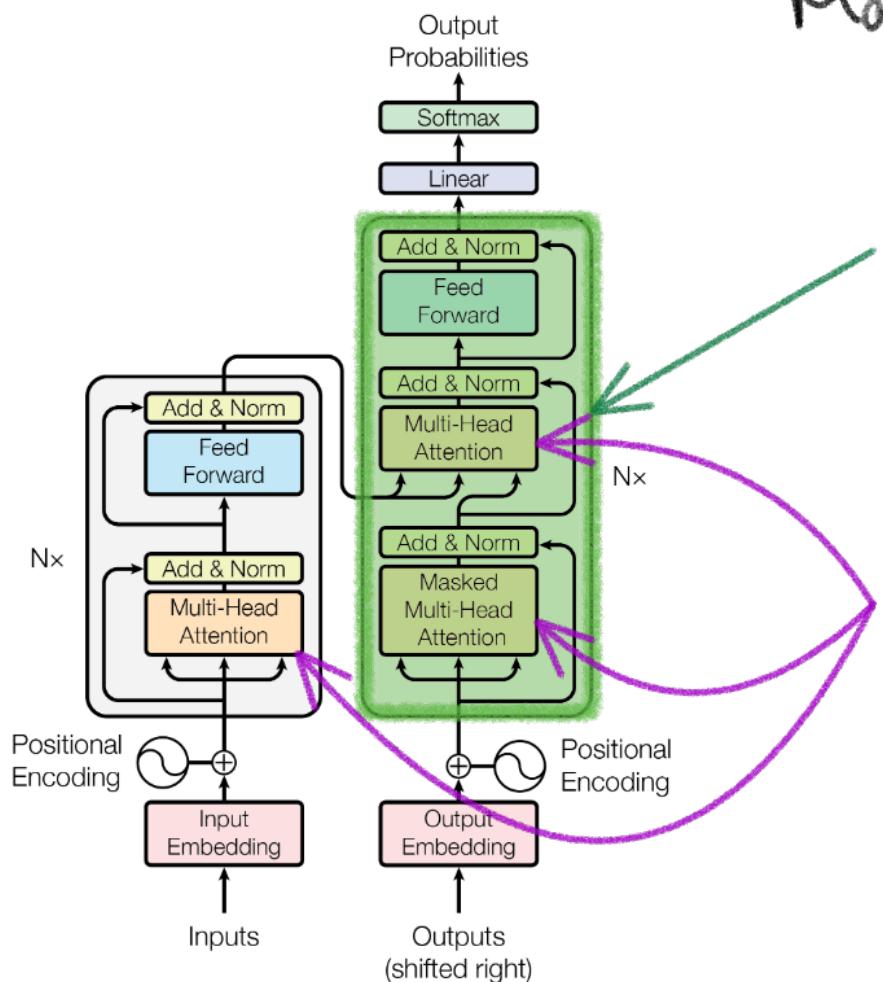
Encoder Block:

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

→ Input Embedding
→ Positional Encoding

Figure 1: The Transformer - model architecture.

OBJECTIVE



Master what happens:

- inside the decoder transformer blocks
- masking

Figure 1: The Transformer - model architecture.

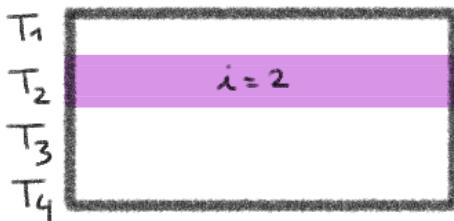
SCALED DOT-PRODUCT ATTENTION

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

SCALED DOT-PRODUCT ATTENTION

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

Q



K



V



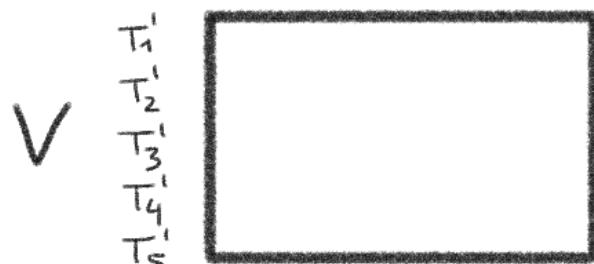
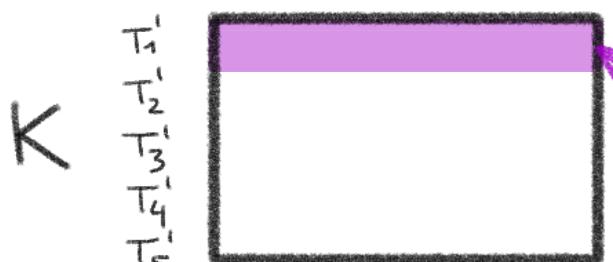
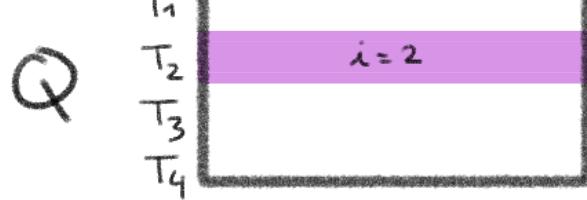
d_K

Attention weights
⇒ i-th row

$$[s_{i1} \ s_{i2} \ \dots \ s_{ie}]$$

SCALED DOT-PRODUCT ATTENTION

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



d_k

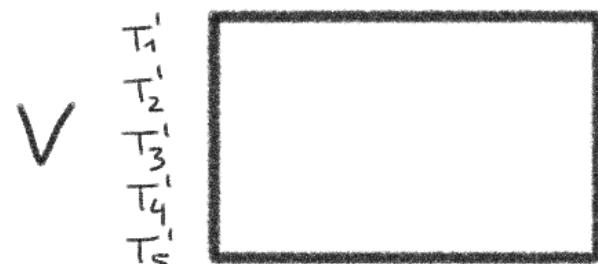
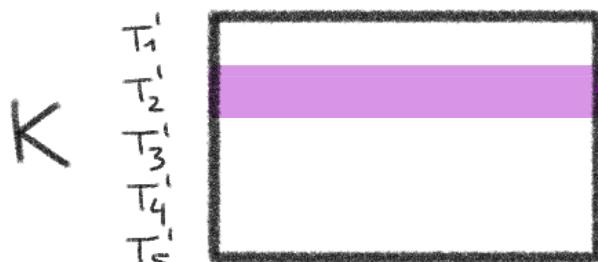
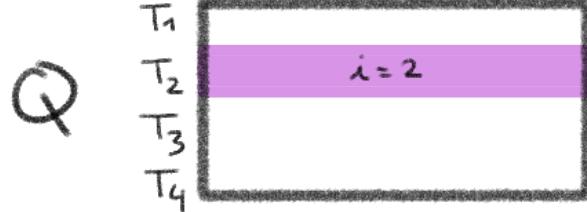
Attention weights

\Rightarrow i-th row

$$[s_{i1} \ s_{i2} \ \dots \ s_{ie}]$$

SCALED DOT-PRODUCT ATTENTION

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



d_K

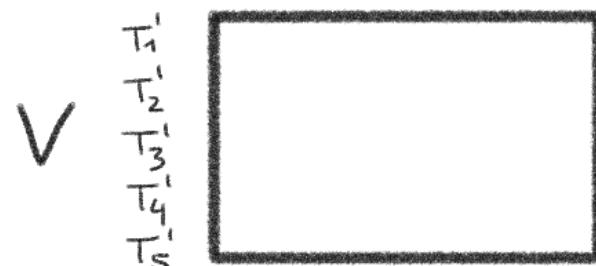
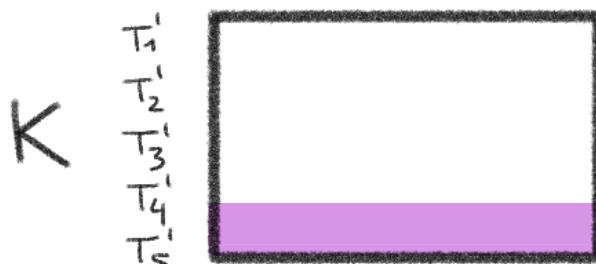
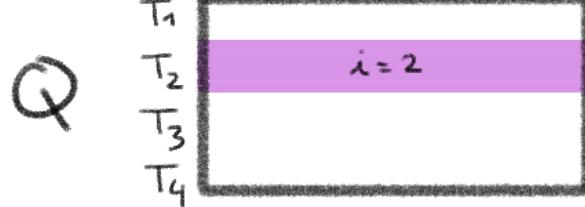
Attention weights

\Rightarrow i -th row

$$[s_{i1} \ s_{i2} \ \dots \ s_{ie}]$$

SCALED DOT-PRODUCT ATTENTION

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



d_k

Attention weights

\Rightarrow i -th row

$$[s_{i1} \ s_{i2} \ \dots \ s_{ie}]$$

MASKED ATTENTION

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

Q

T ₁	
T ₂	i=2
T ₃	
T ₄	

Attention weights

⇒ i-th row

K

T ₁	
T ₂	
T ₃	
T ₄	
T ₅	

V

T ₁	
T ₂	
T ₃	
T ₄	
T ₅	

d_K

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

MASKED ATTENTION

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

Q

T ₁	
T ₂	i=2
T ₃	
T ₄	

K

T ₁	
T ₂	
T ₃	
T ₄	
T ₅	

V

T ₁	
T ₂	
T ₃	
T ₄	
T ₅	

d_K

Attention weights

⇒ i-th row

$$[s_{i1} \ s_{i2} \ \dots \ s_{ie}]$$

Sol: set s_{ij} = 0

MASKED ATTENTION

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

Q

T ₁	
T ₂	i = 2
T ₃	
T ₄	

Attention weights

⇒ i-th row

K

T ₁	
T ₂	
T ₃	
T ₄	
T ₅	

V

T ₁	
T ₂	
T ₃	
T ₄	
T ₅	

d_K

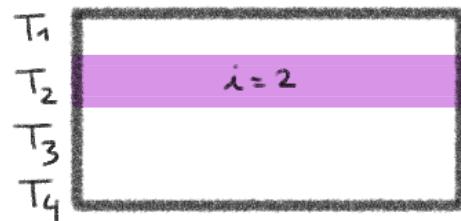
Sol: set s_{ij} = 0

⚠ Pb: $\sum_j s_{ij} \neq 1$

MASKED ATTENTION

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

Q



K



V



d_K

Sol: Modify the matrix
 QK^T instead!

MASKED ATTENTION

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

Q

T ₁	
T ₂	i=2
T ₃	
T ₄	

K

T ₁	
T ₂	
T ₃	
T ₄	
T ₅	

V

T ₁	
T ₂	
T ₃	
T ₄	
T ₅	

d_k

Sol: Modify the matrix

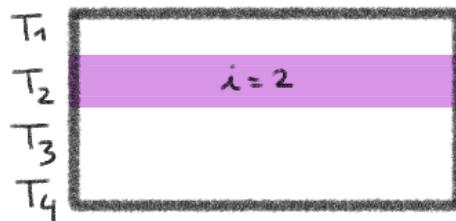
QK^T instead !

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

MASKED ATTENTION

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

Q



K



V



d_k

Sol: Modify the matrix
 QK^T instead !

Answer: set

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

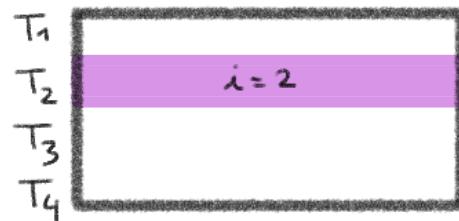
MASKED ATTENTION

Attention (Q, K, V, M) =

MASKED ATTENTION

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

Q



K



V



d_k

Sol: Modify the matrix
 QK^T instead !

Answer: set

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

MASKED ATTENTION

Attention (Q, K, V, M) =

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

MASKED ATTENTION

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

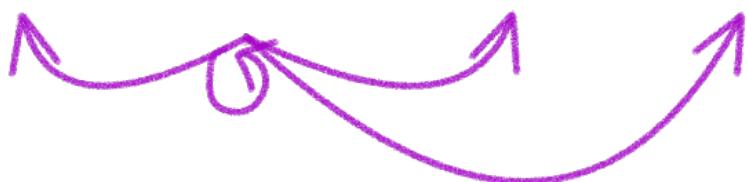
The sandwich is delicious <eos>

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

PADDING MASKS

The sandwich is delicious <eos>

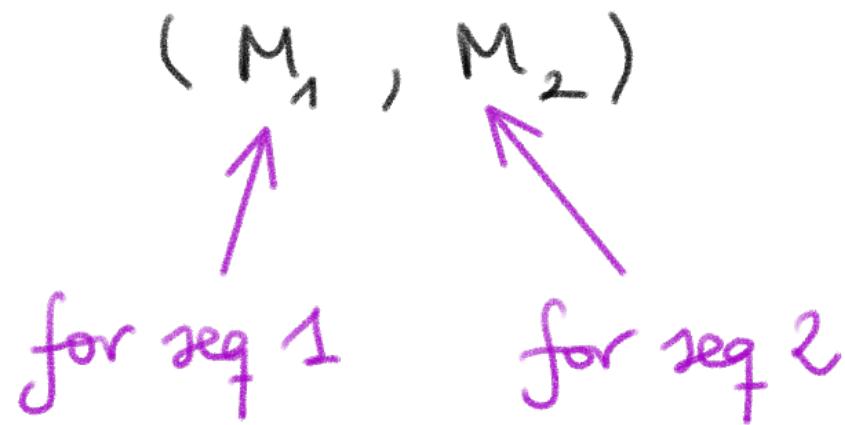
I am bored <eos> <pad> <pad> <pad>



PADDING MASKS

The sandwich is delicious <eos>

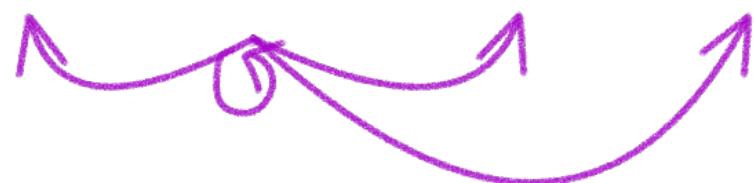
I am bored <eos> <pad> <pad> <pad>



PADDING MASKS

The sandwich is delicious (eos)

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



(M₁, M₂)

PADDING MASKS

In code :

Tensor of shape (b, H, ℓ, ℓ)

nb. of segs 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 $\neq \langle \text{pad} \rangle$
0s for tokens $= \langle \text{pad} \rangle$

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 $\neq \langle \text{pad} \rangle$
0s for tokens $= \langle \text{pad} \rangle$

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

CAUSAL MASK

The window is closed <eos>
⑤

CAUSAL MASK

The window is closed <eos>



CAUSAL MASK

The window is closed <eos>



CAUSAL MASK

The window is closed $\langle \text{eos} \rangle$



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

CAUSAL MASK

In code :

Tensor of shape (b, H, ℓ, ℓ)

same mask for all segs
same mask for all heads

CAUSAL MASK

In code :

Tensor of shape (b, H, l, l)

↑
same mask for all segs
same mask for all heads

\Rightarrow input mask of shape (l, l)

CAUSAL MASK

In code :

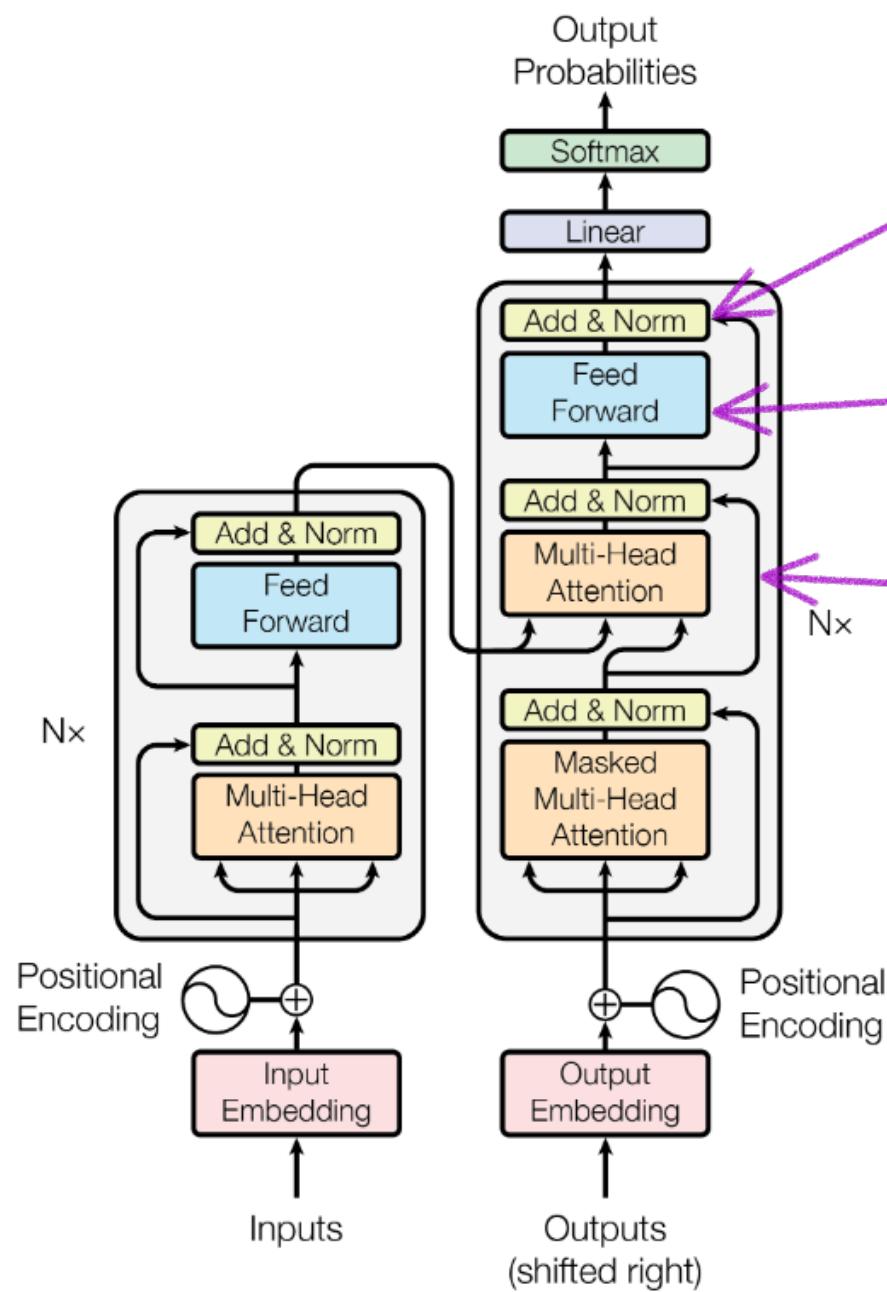
Tensor of shape (b, H, ℓ, ℓ)

↑
same mask for all segs
same mask for all heads

\Rightarrow input mask of shape (ℓ, ℓ)

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

DECODER BLOCK



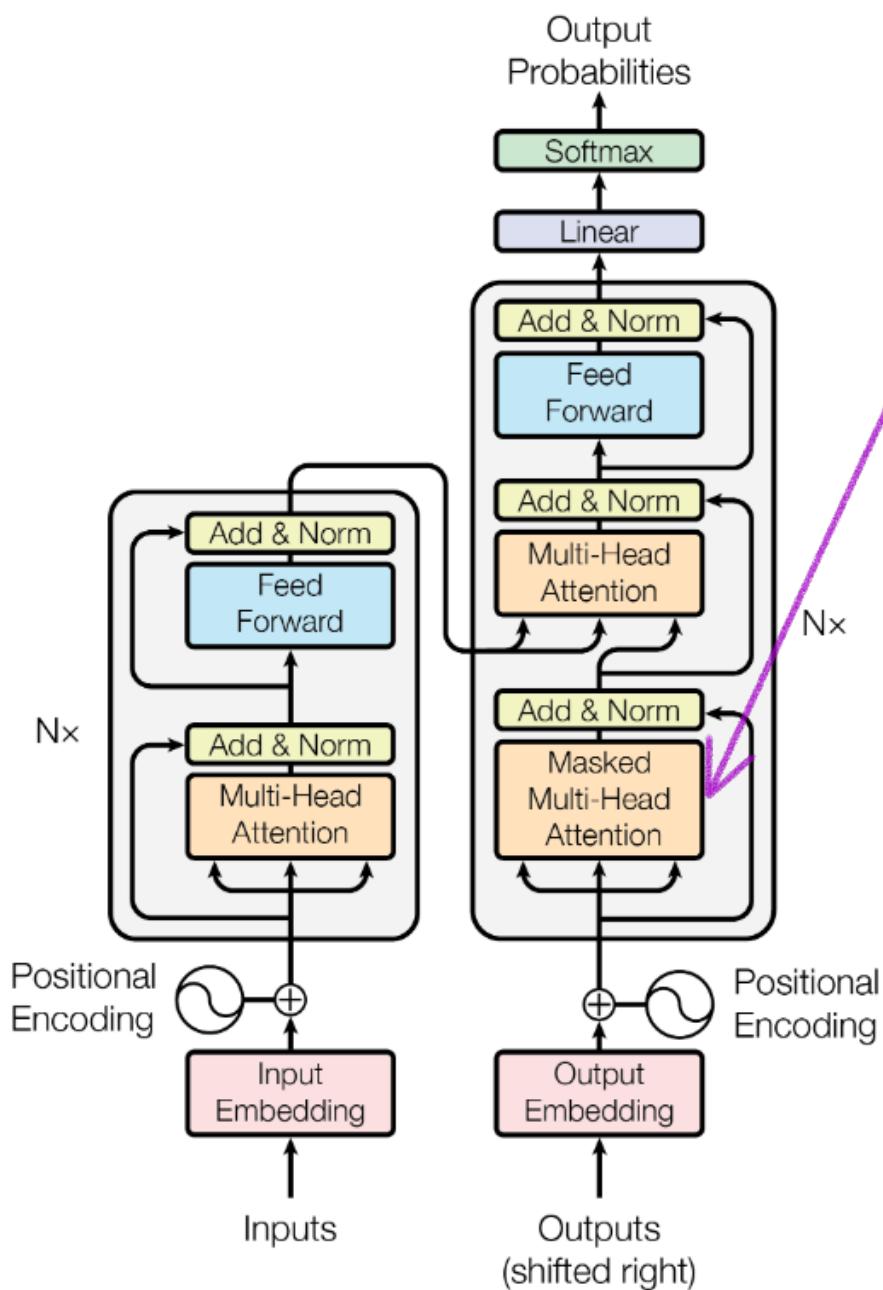
Layer Norm

Feed-forward module

Residual
Connections

Exactly the same
as for the decoder!

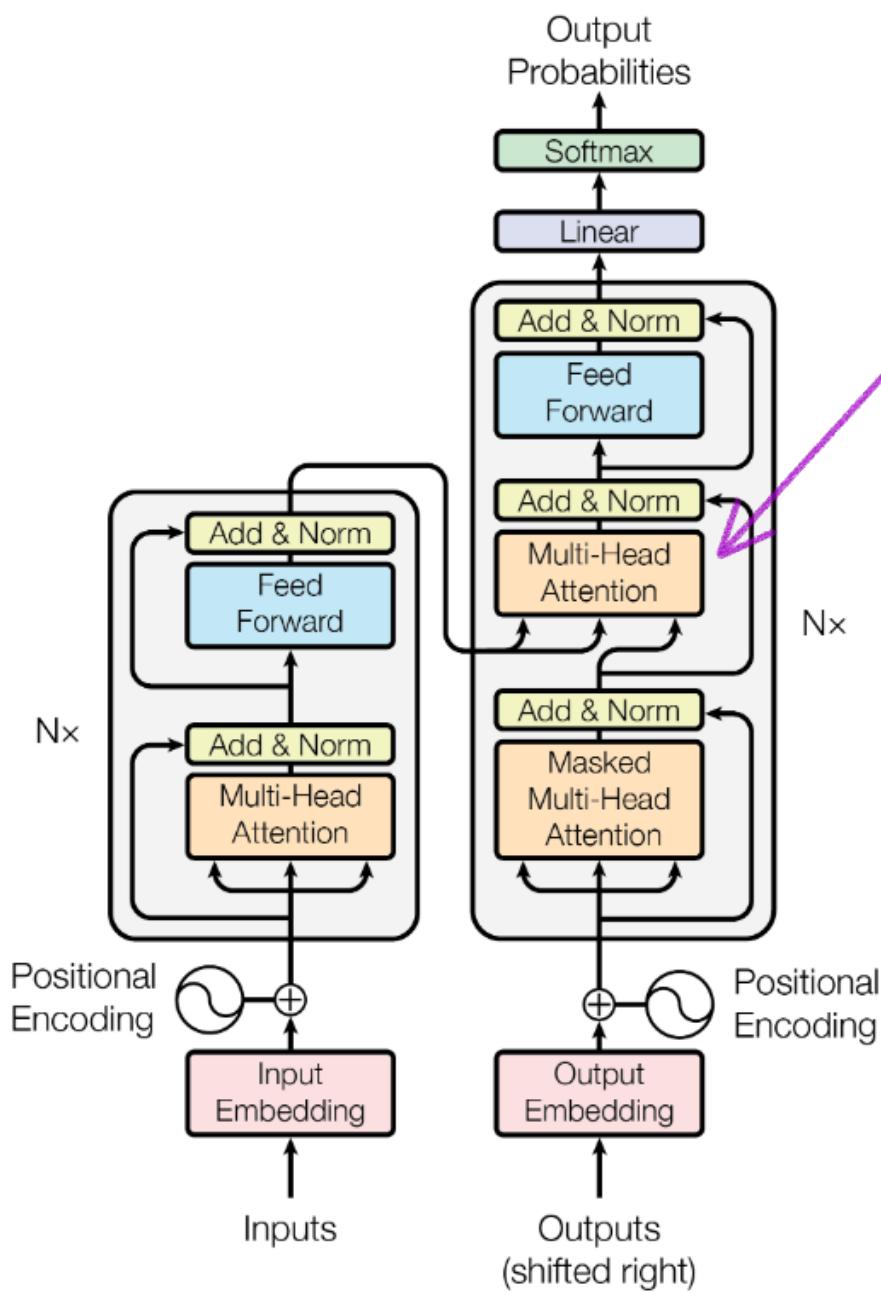
DECODER BLOCK



Masked Multi-head
(self) attention:

Self attention
with a
causal mask

DECODER BLOCK



Multi-head
(cross) attention:

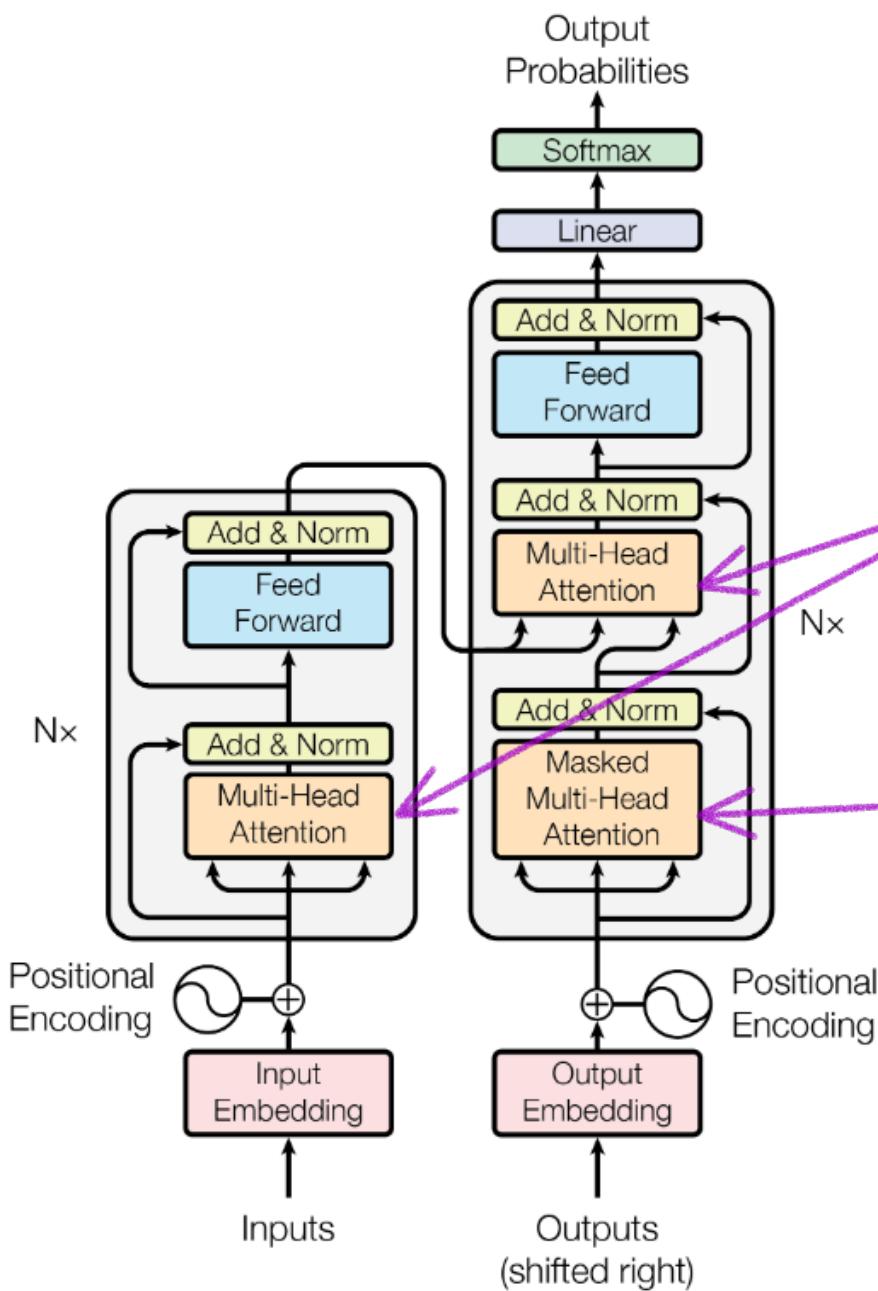
Queries

→ from decoder
input

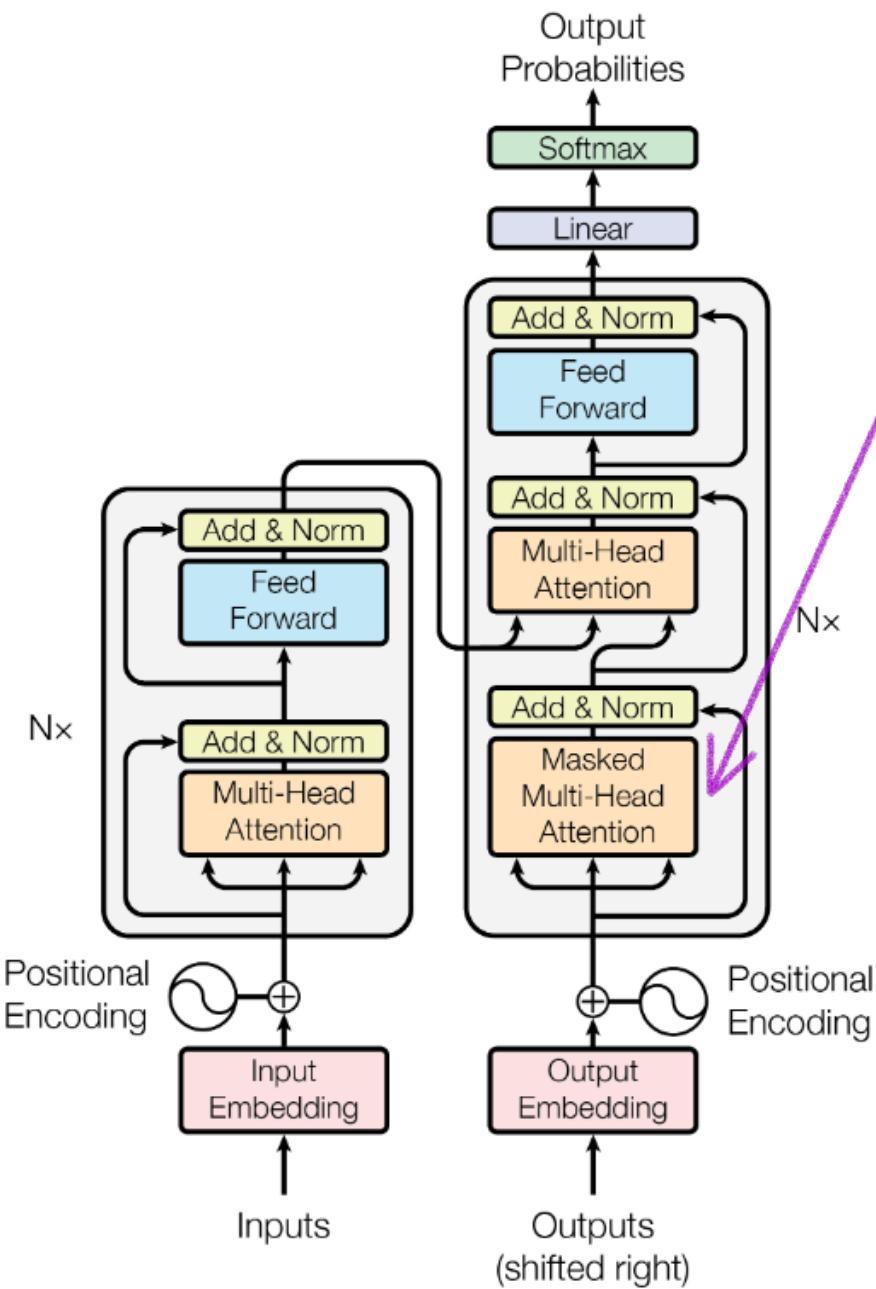
Keys, Values

→ from encoder
hidden state

PADDING MASKS EVERYWHERE



COMBINING MASKS



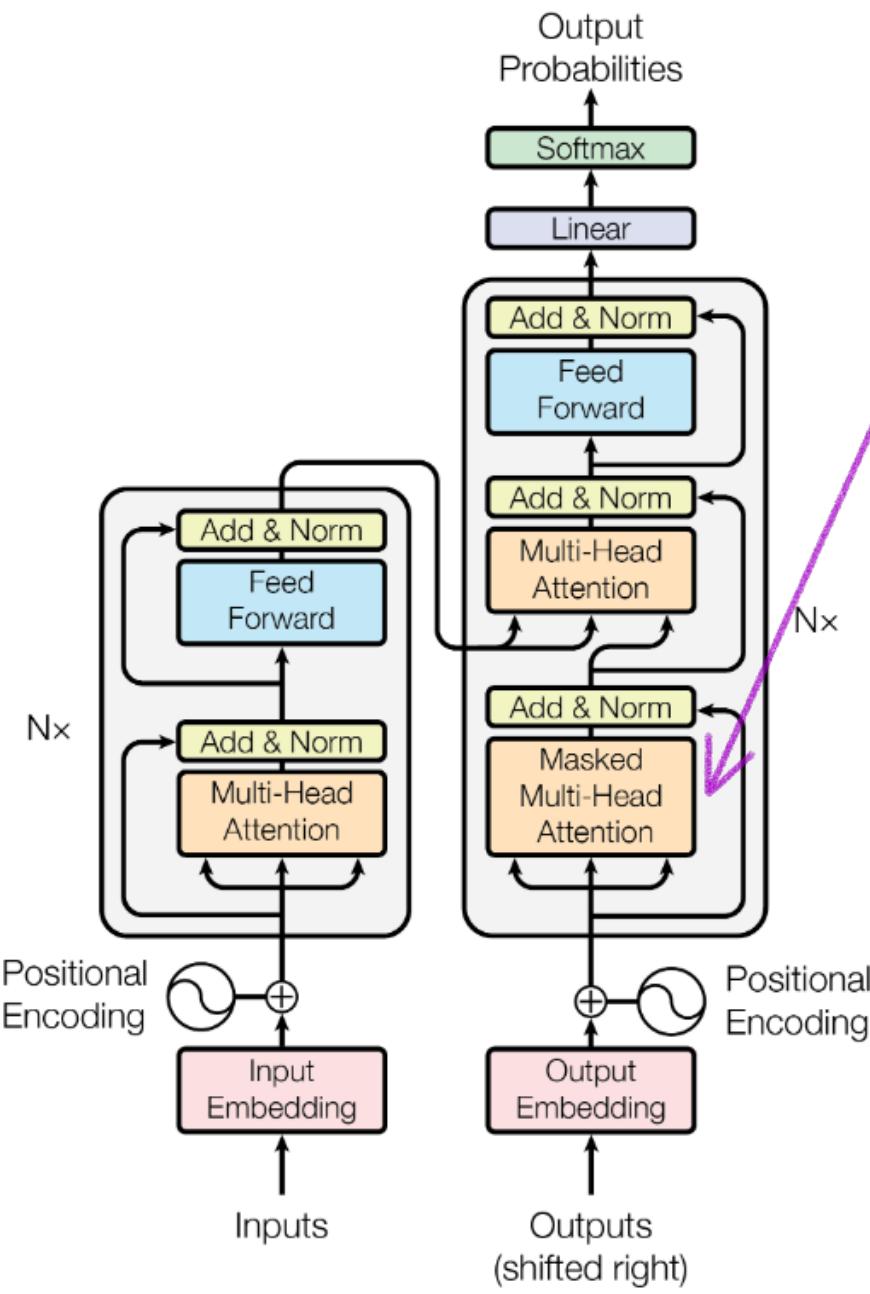
Padding mask
+
Causal mask

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

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

$M_1 \& M_2 =$
attend if both masks = 1

COMBINING MASKS



Padding mask + Causal mask

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

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

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

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. Almanar
- "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