

VALDOM - NLP 2 LLM

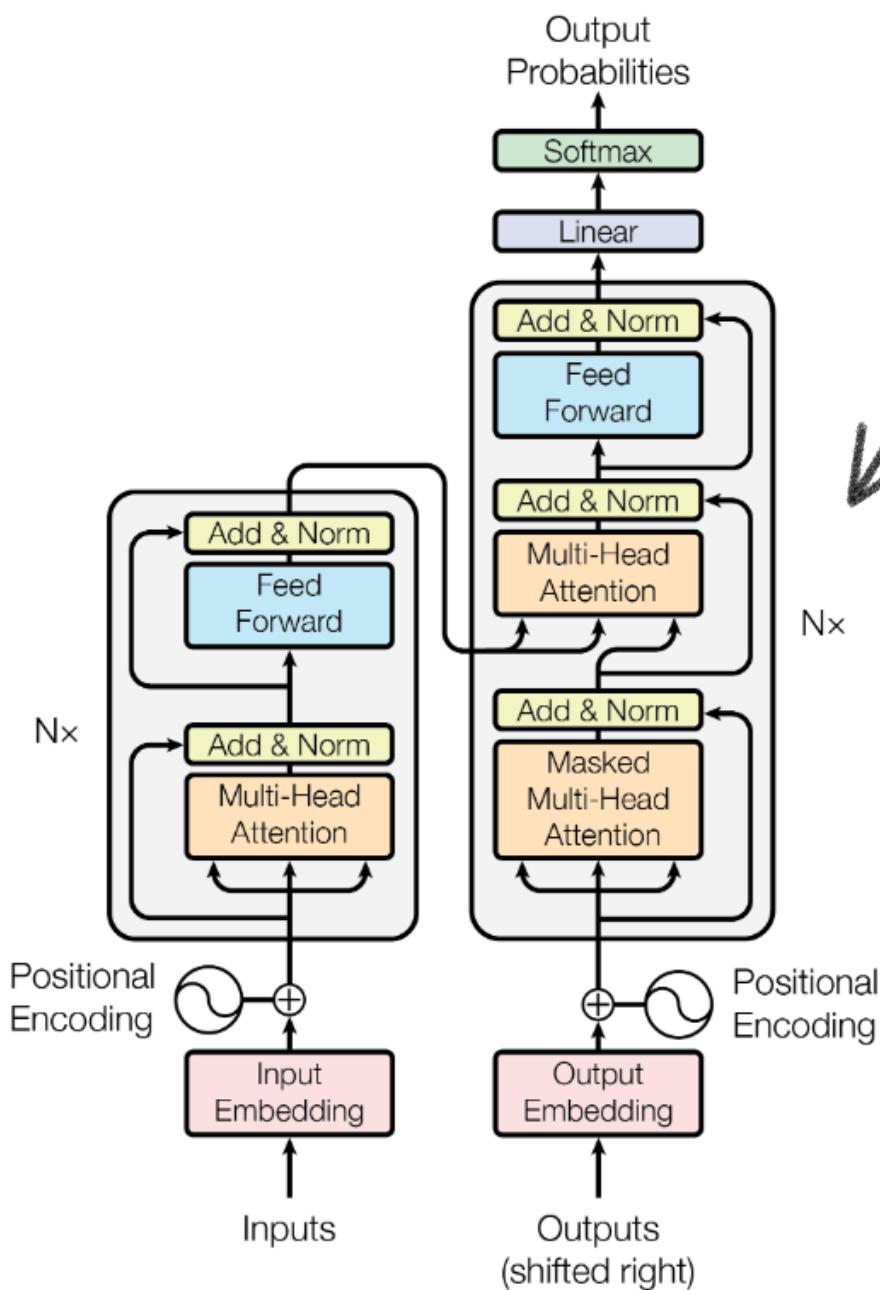
# LARGE LANGUAGE MODELS

JOSEBA DALMAU

# OBJECTIVE

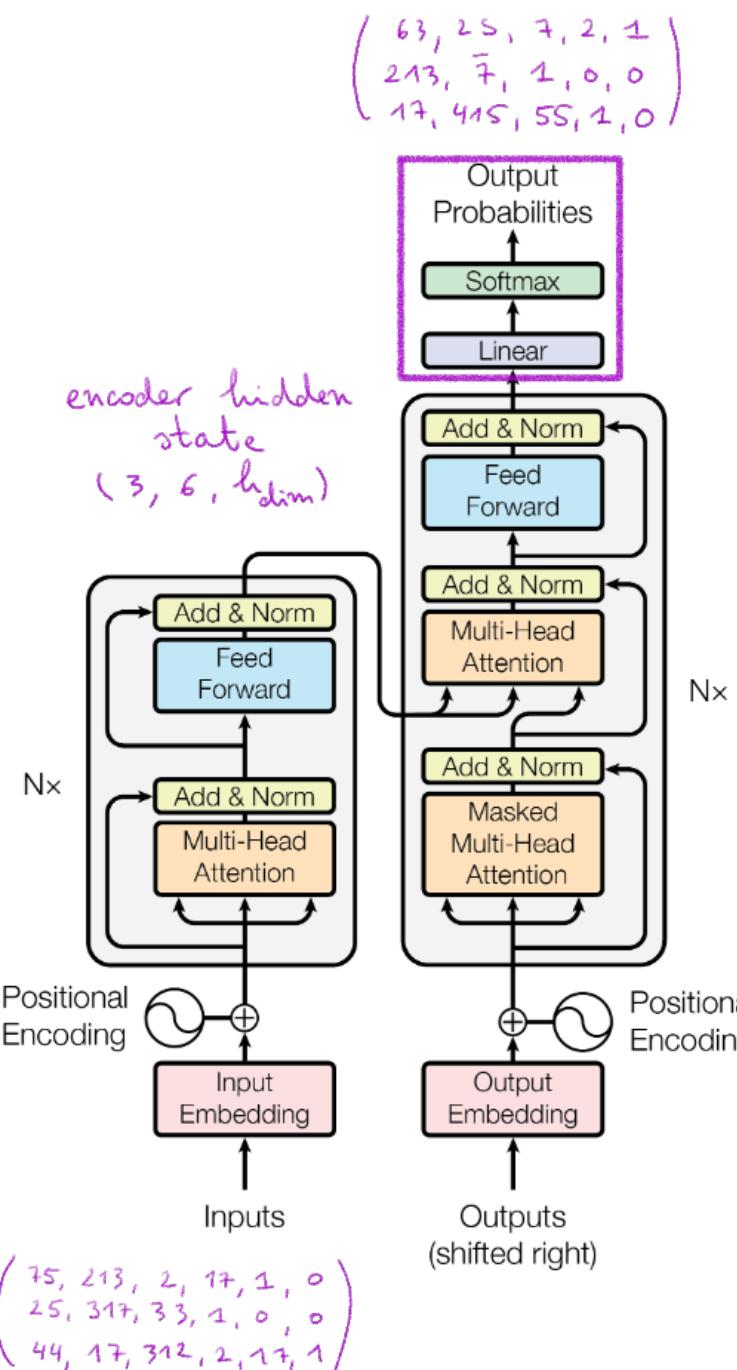
Master the Transformer  
architecture for various  
different NLP tasks

# TRANSFORMERS



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

# LAST WEEK



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

# OBJECTIVE

Master what happens

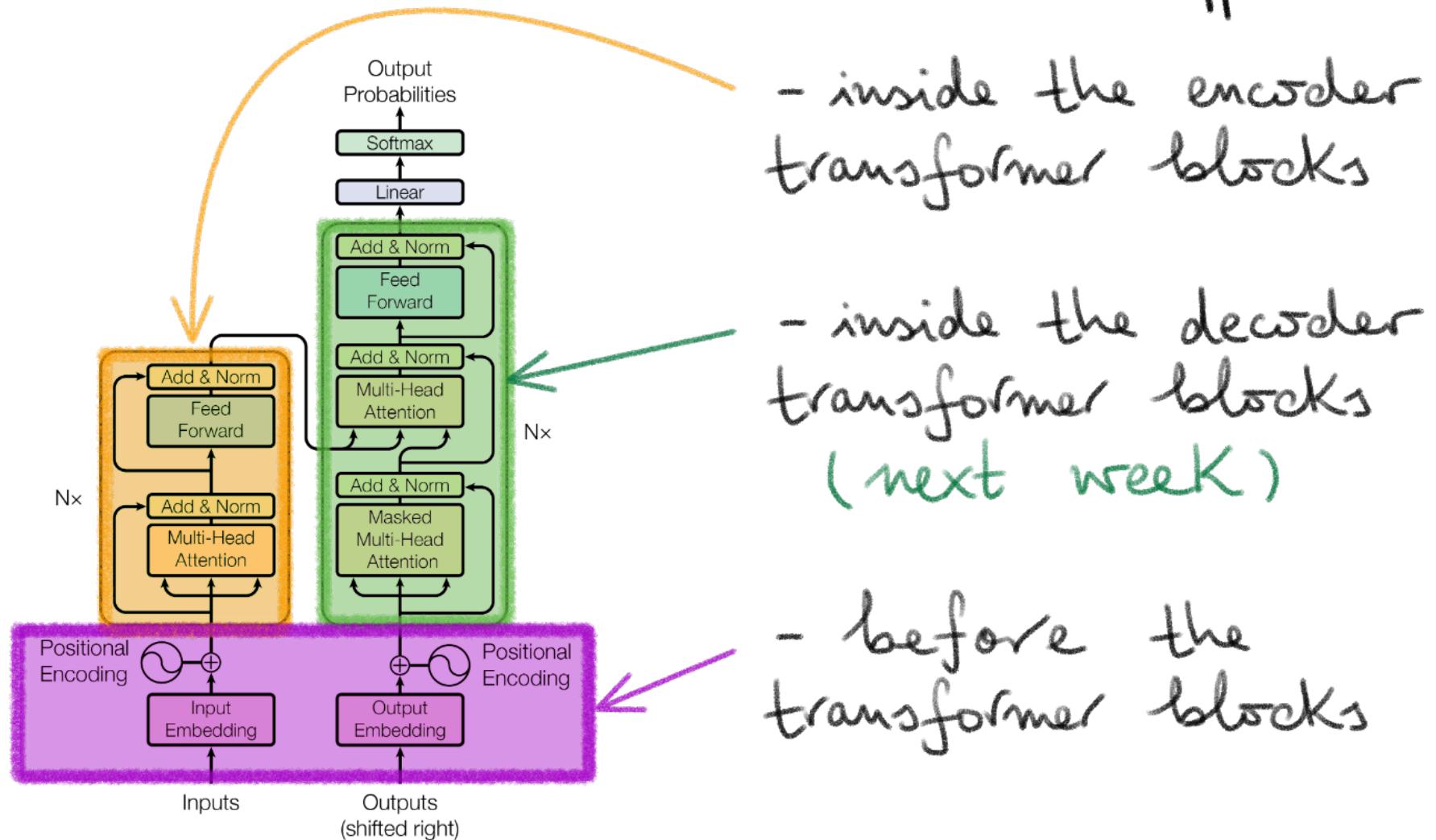
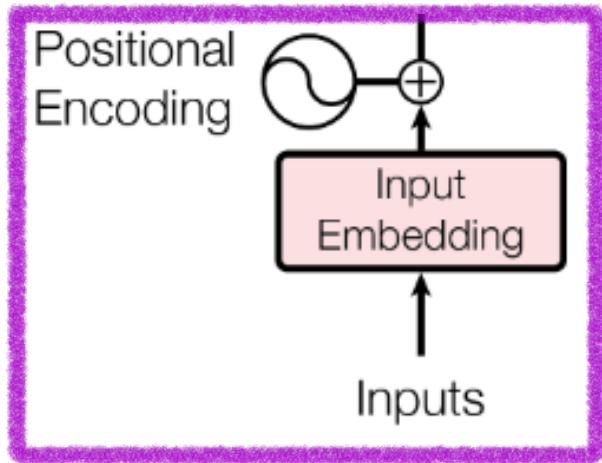


Figure 1: The Transformer - model architecture.

# INPUT EMBEDDING



set of integers of  
size  $n_{vocab}$

Purpose :

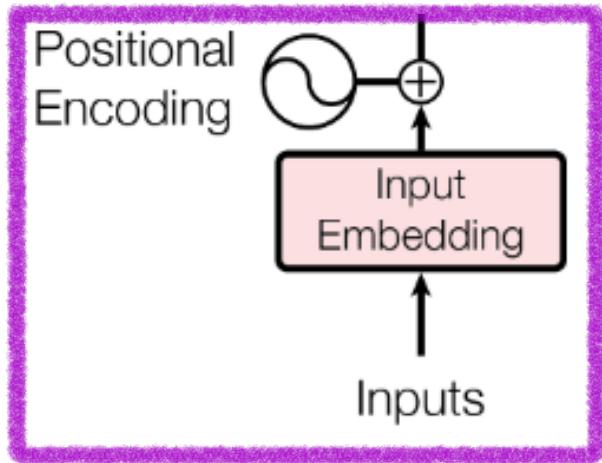
token  
 $t \in \mathcal{D}$

$\xrightarrow{\quad}$   $\mathbb{R}^{h_{dim}}$

hidden dimension

How?

# INPUT EMBEDDING



set of integers of  
size  $n_{vocab}$

Purpose :

token  
 $t \in \mathcal{D}$   $\xrightarrow{\quad} \mathbb{R}^{h_{dim}}$

hidden dimension

- Word2Vec

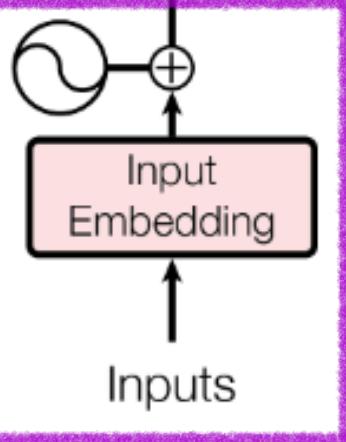
- GloVe

- ...

flow?

# POSITIONAL ENCODING

Positional Encoding



Purpose:

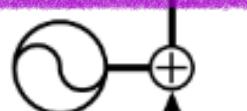
encode the information about  
the position of the tokens  
within the sentence;

you are walking very slowly

My cat seems to like you

# POSITIONAL ENCODING

Positional Encoding



Input  
Embedding

Inputs

Purpose:

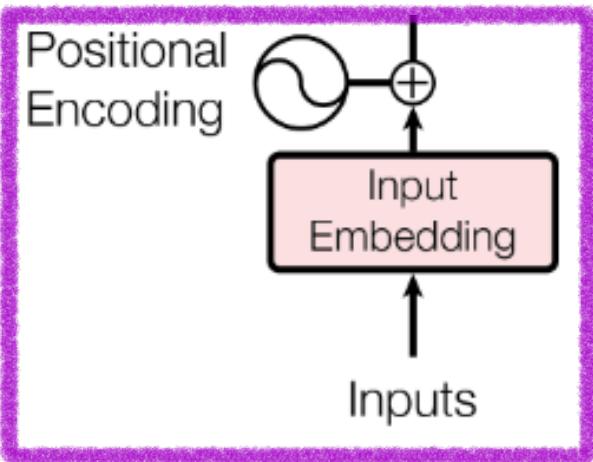
encode the information about  
the position of the tokens  
within the sentence;

you are walking very slowly

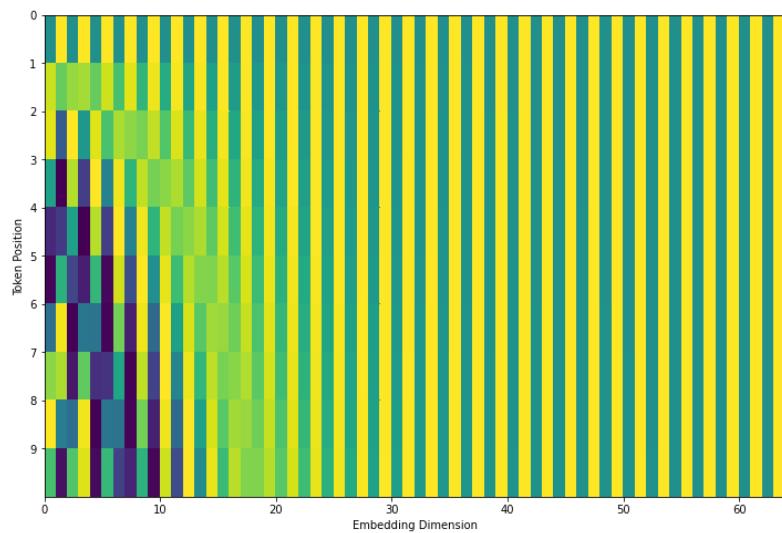
My cat seems to like you

# POSITIONAL

# ENCODING



Per sentence in the batch:



A diagram showing the addition of a positional encoding vector to an input embedding vector. A plus sign (+) is placed between two vectors. The first vector is labeled  $T_0 \begin{bmatrix} 7,23 \\ 7,1 \\ T_1 \\ \vdots \\ T_9 \end{bmatrix}$ . The second vector is a 64-dimensional vector represented as a bracketed list of values:  $\begin{bmatrix} 7,23 & -3,71 & 0,55 & \cdots & 12,08 \\ -11,2 & -2,24 & 0,09 & \cdots & -8,99 \\ 0,07 & 13,56 & 7,14 & \cdots & -6,64 \\ \vdots & \vdots & \vdots & & \vdots \\ 8,03 & 2,31 & -0,44 & \cdots & 5,27 \end{bmatrix}$ .

$$\text{hdim} = 64$$

# OBJECTIVE

Master what happens

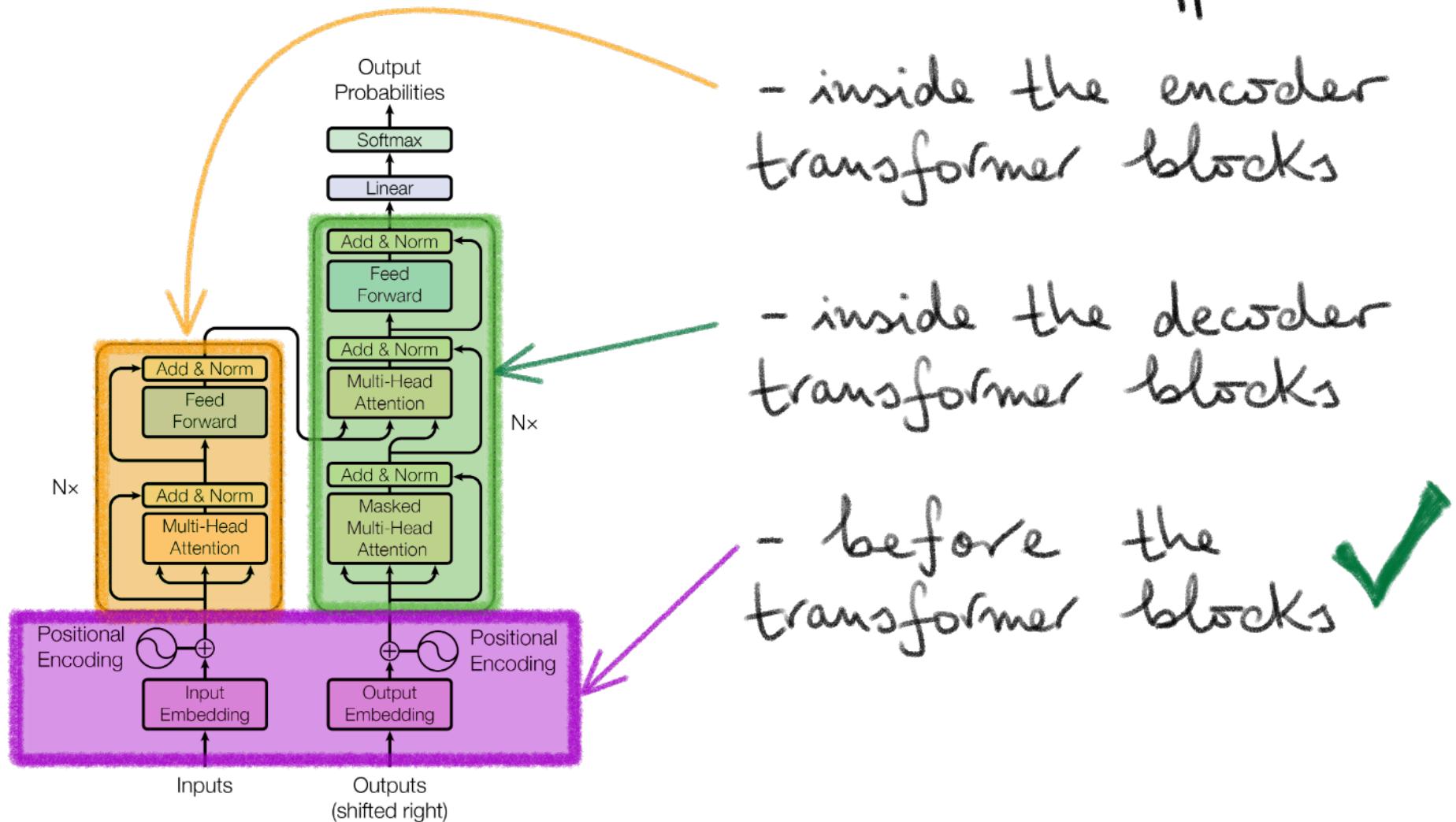
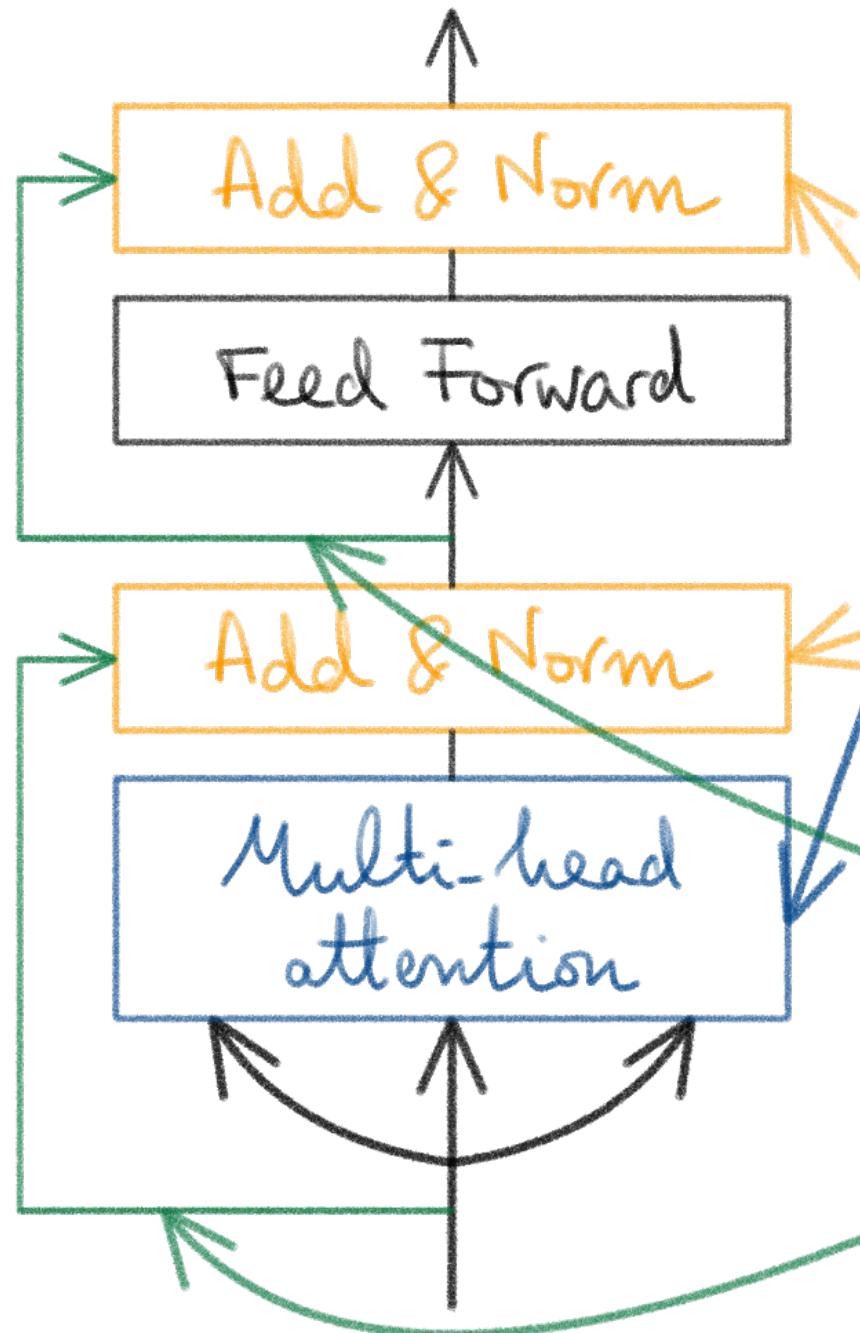


Figure 1: The Transformer - model architecture.

# TRANSFORMER BUILDING BLOCKS

- Scaled dot product attention
- Residual connections
- Layer normalization
- Feed-forward networks

# TRANSFORMER BLOCKS



Scaled dot  
product attention

Layer  
Normalization

Residual  
Connections

# 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



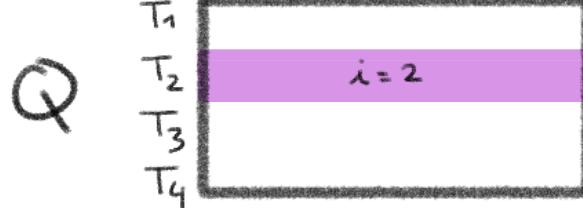
V



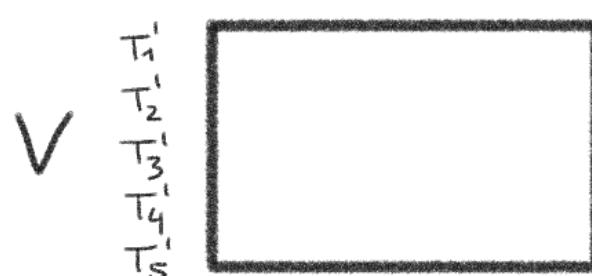
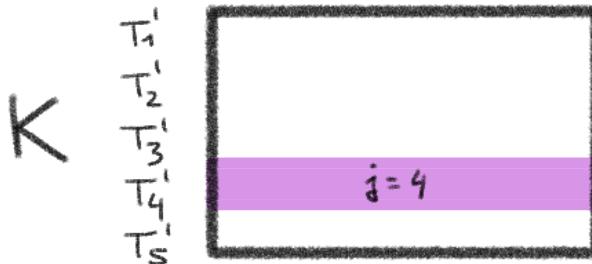
d<sub>k</sub>

# SCALED DOT-PRODUCT ATTENTION

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



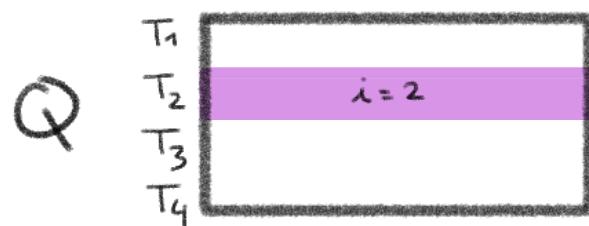
$$(QK^T)_{i,j} = \sum_{h=1}^{d_K} Q_{i,h} \cdot K_{j,h}$$



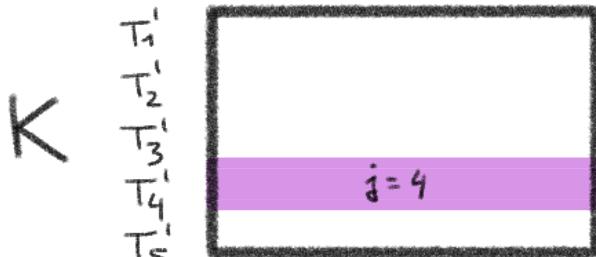
$d_K$

# SCALED DOT-PRODUCT ATTENTION

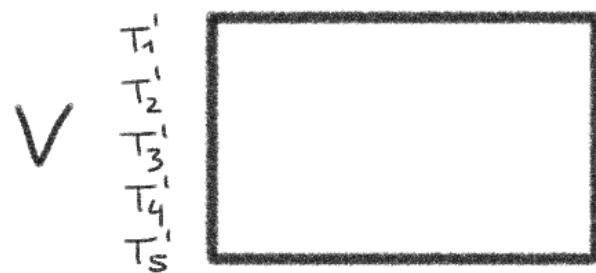
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V$$



$$(QK^T)_{ij} = \sum_{h=1}^d Q_{ih} \cdot K_{j,h}$$



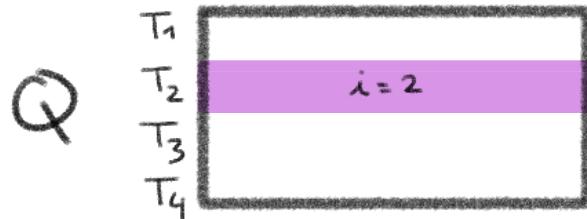
$$\text{softmax} \left( \frac{\begin{pmatrix} QK^T \\ j^1 \end{pmatrix}}{\sqrt{d_k}}, \dots, \frac{\begin{pmatrix} QK^T \\ j^5 \end{pmatrix}}{\sqrt{d_k}} \right)$$



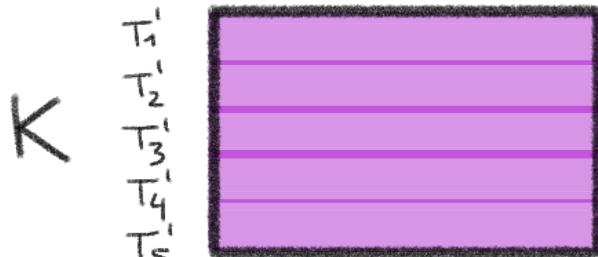
attention scores for the  
 $i$ -th query

# SCALED DOT-PRODUCT ATTENTION

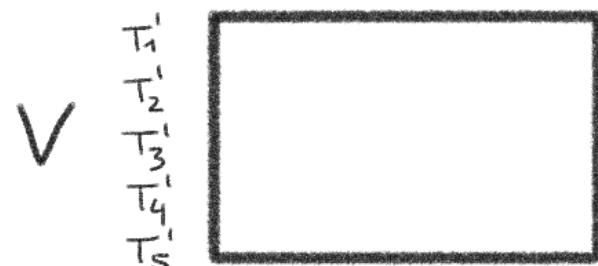
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



$$(QK^T)_{i,j} = \sum_{h=1}^{d_k} Q_{i,h} \cdot K_{j,h}$$



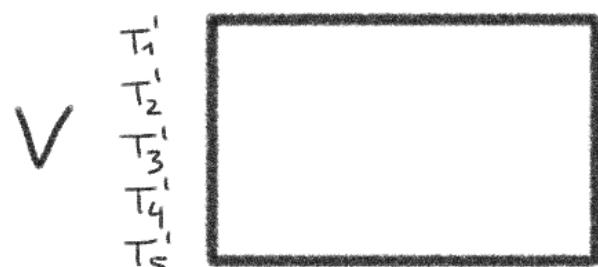
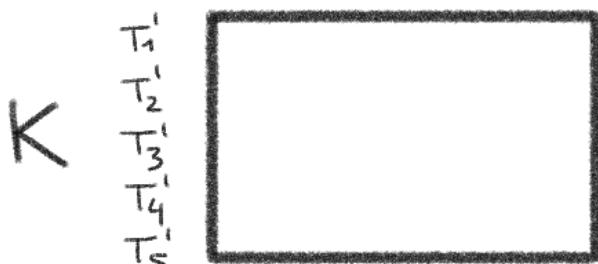
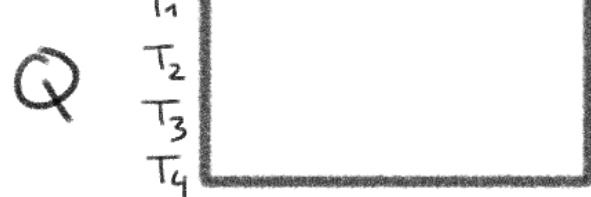
$$\text{softmax}\left(\frac{(QK^T)_{i,1}}{\sqrt{d_k}}, \dots, \frac{(QK^T)_{i,5}}{\sqrt{d_k}}\right)$$



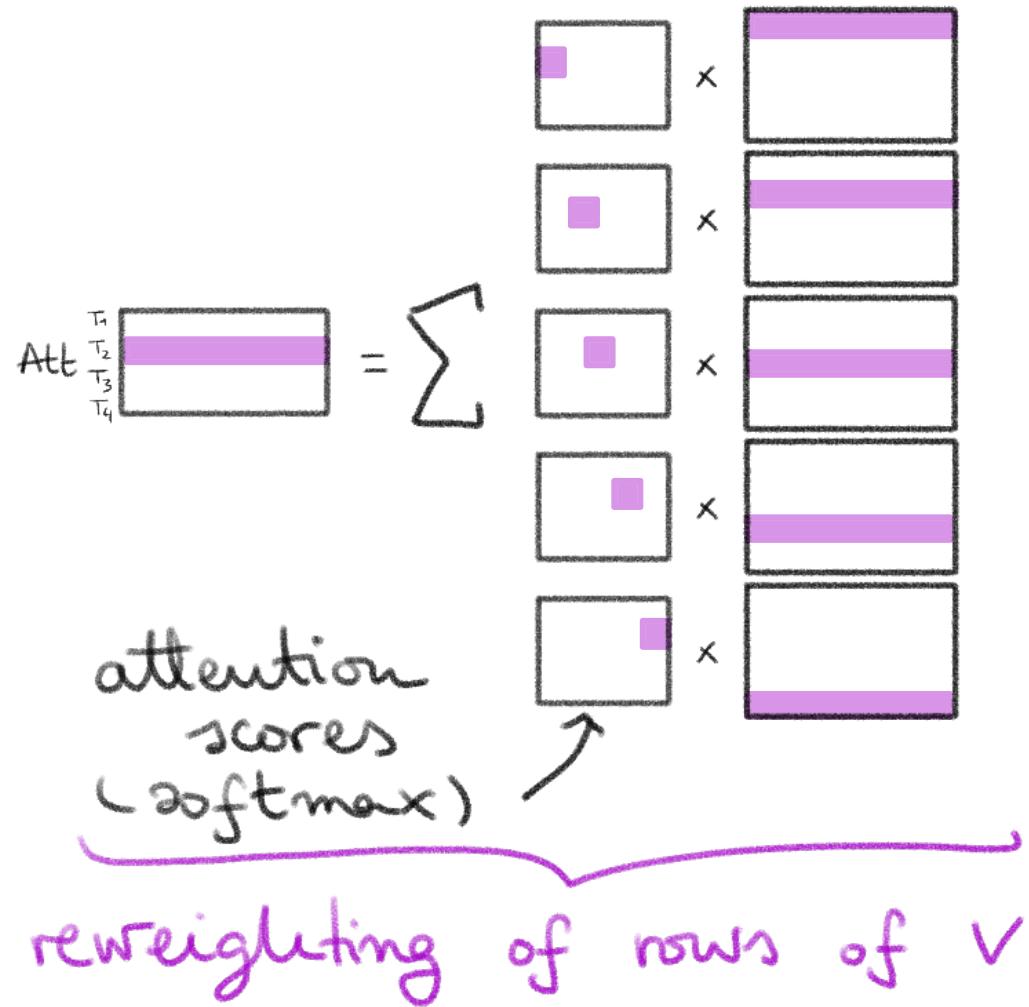
attention scores for the  
 $i$ -th query

# SCALED DOT-PRODUCT ATTENTION

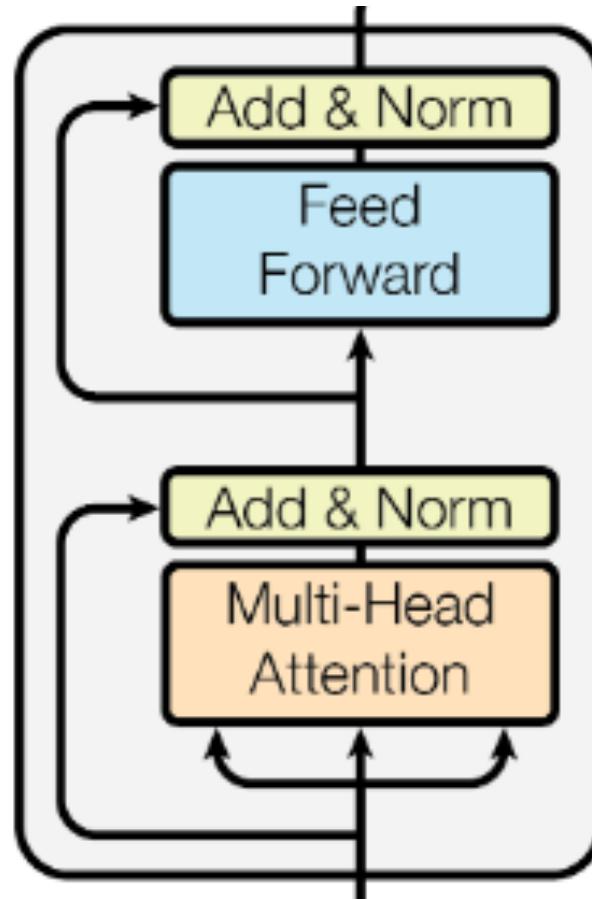
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V$$



$d_K$



# MULTI-HEAD ATTENTION



$$\begin{matrix} T_0 & \begin{matrix} 7,23 & -3,71 & 0,55 & \dots & 12,08 \end{matrix} \\ T_1 & \begin{matrix} -11,2 & -2,24 & 0,09 & \dots & -8,99 \end{matrix} \\ T_2 & \begin{matrix} 0,07 & 13,56 & 7,14 & \dots & -6,64 \end{matrix} \\ \vdots & \vdots \quad \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \\ T_{l-1} & \begin{matrix} 8,03 & 2,31 & -0,44 & \dots & 5,27 \end{matrix} \end{matrix}$$

$\underbrace{\quad\quad\quad}_{l \times h_{\text{dim}}}$

This diagram shows a sequence of inputs  $T_0, T_1, T_2, \dots, T_{l-1}$  represented as vectors. The dimension of each vector is  $l \times h_{\text{dim}}$ , where  $l$  is the sequence length and  $h_{\text{dim}}$  is the dimension of each head. The vectors are stacked horizontally to form the input matrix.

# MULTI-HEAD ATTENTION

$T_0$	7,23	-3,71	0,55	...	12,08
$T_1$	-11,2	-2,24	0,09	...	-8,99
$T_2$	9,07	13,56	7,14	...	-6,64
:	:	:	:	.	:
$T_{l-1}$	8,03	2,31	-0,44	...	5,27

$l \times h_{\text{dim}}$

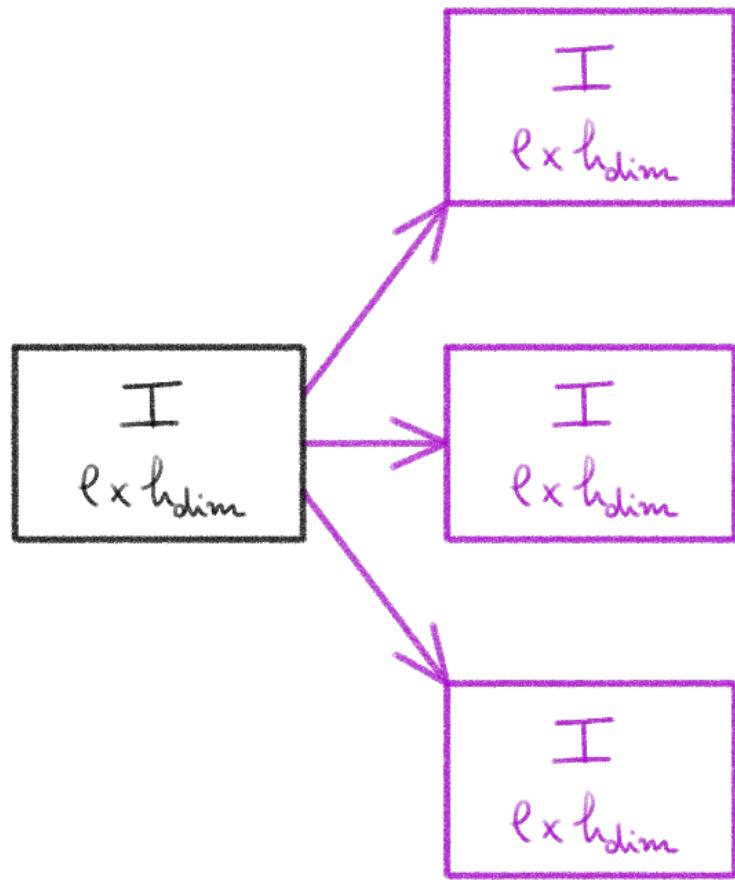


nb. of tokens  
in the sentence

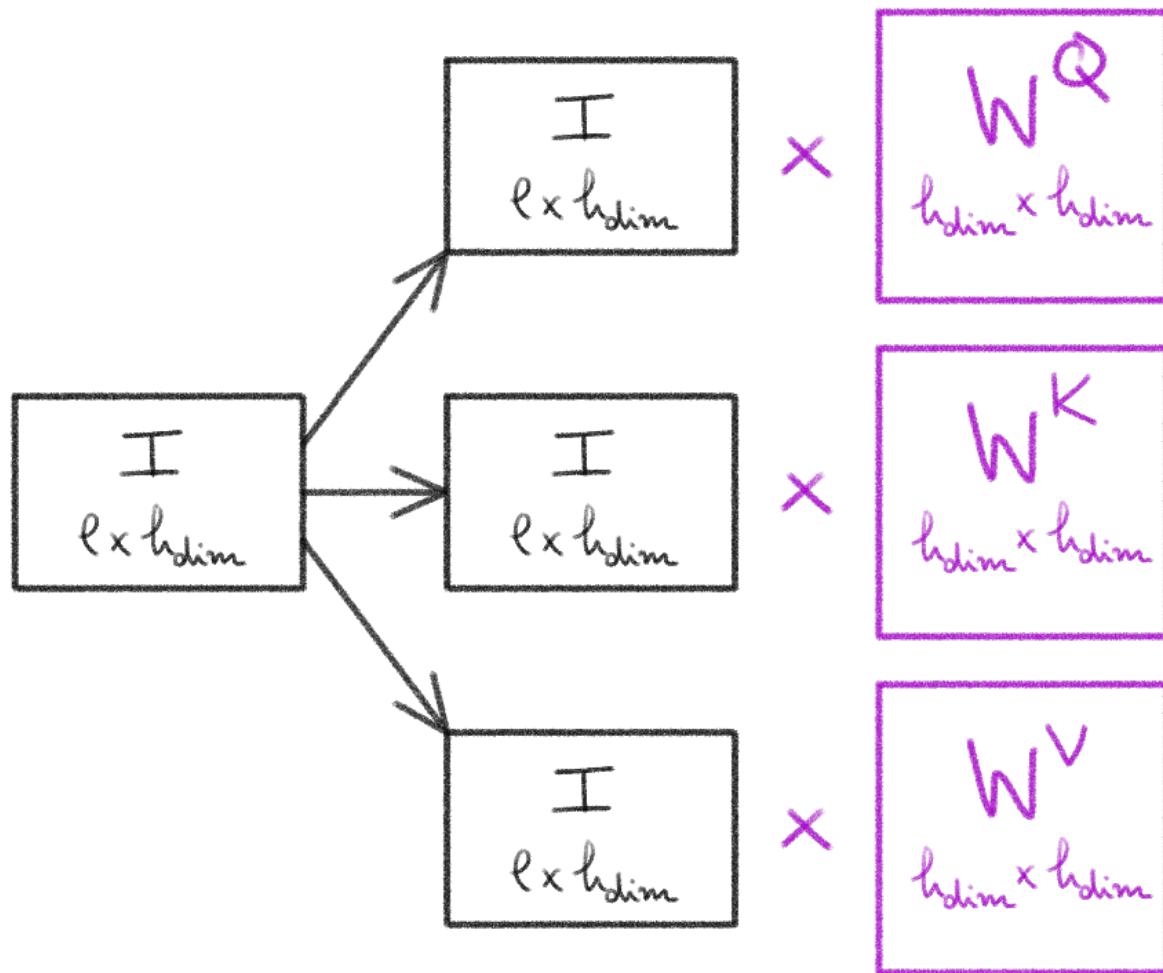
# MULTI-HEAD ATTENTION

I  
Ex hdim

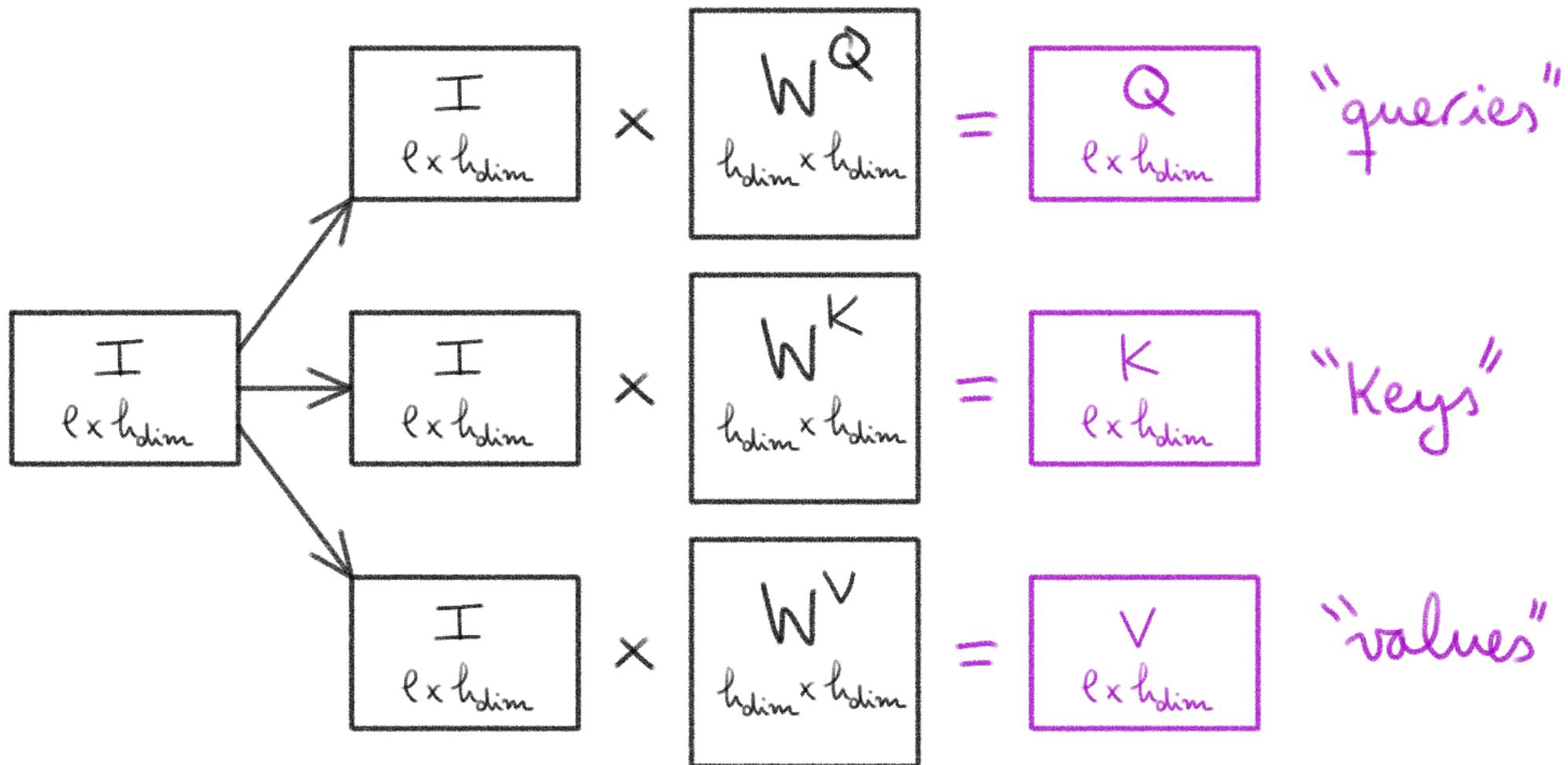
# MULTI-HEAD ATTENTION



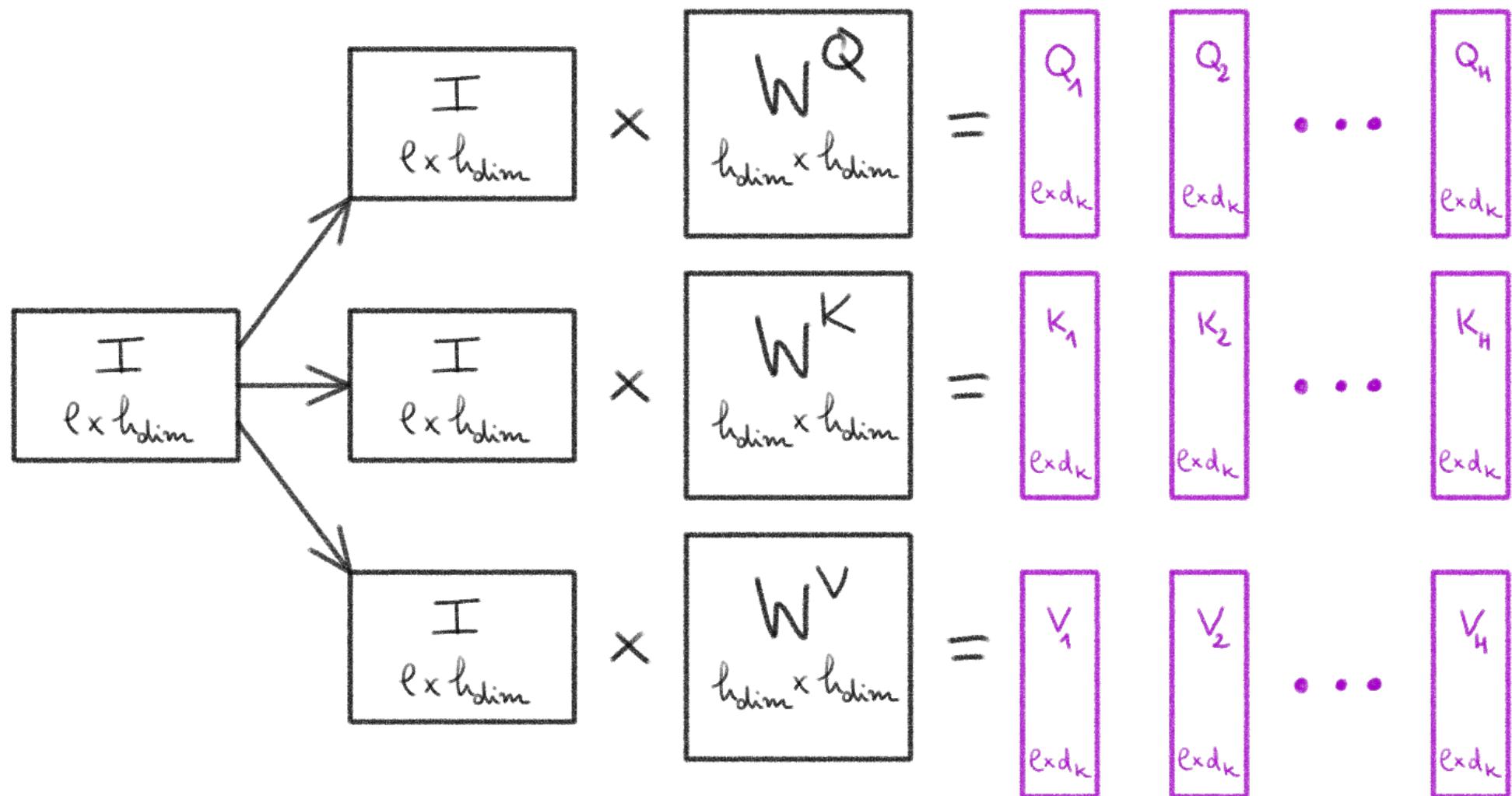
# MULTI-HEAD ATTENTION



# MULTI-HEAD ATTENTION



# MULTI-HEAD ATTENTION



Relationship :  $h_{\text{dim}} = d_k \times H$

nb. of dims per head ↑      ↑ nb. of heads

# MULTI-HEAD ATTENTION

$Q_1$

$Q_2$

...

$Q_H$

$K_1$

$K_2$

...

$K_H$

$V_1$

$V_2$

...

$V_H$

Attention  
 $(Q_1, K_1, V_1)$

$l \times d_K$

Attention  
 $(Q_2, K_2, V_2)$

$l \times d_K$

...

Attention  
 $(Q_H, K_H, V_H)$

$l \times d_K$

# MULTI-HEAD ATTENTION

$$\begin{matrix} Q_1 & Q_2 & \dots & Q_H \\ K_1 & K_2 & \dots & K_H \\ V_1 & V_2 & \dots & V_H \end{matrix}$$

Concatenate the H  
attention matrices

$l \times h \text{dim}$

# MULTI-HEAD ATTENTION

Result of  
Multi-head attention

$l \times h \text{dim}$

=

Concatenate the  $H$   
attention matrices

$l \times h \text{dim}$

$\times$

$w^o$

$h \text{dim} \times h \text{dim}$

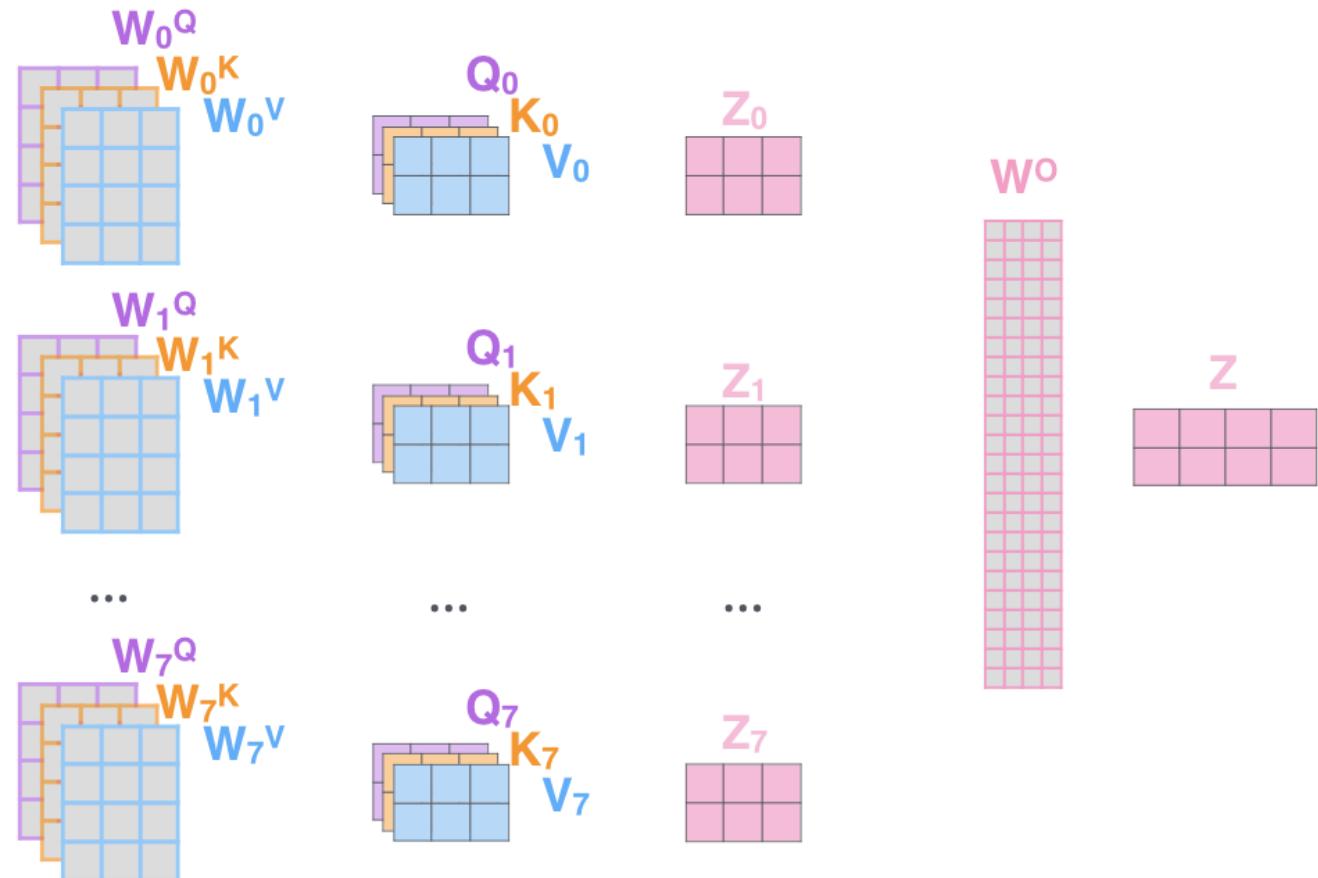
# MULTI-HEAD ATTENTION

- 1) This is our input sentence\*  $X$
- 2) We embed each word\*
- 3) Split into 8 heads. We multiply  $X$  or  $R$  with weight matrices
- 4) Calculate attention using the resulting  $Q/K/V$  matrices
- 5) Concatenate the resulting  $Z$  matrices, then multiply with weight matrix  $W^o$  to produce the output of the layer

Thinking  
Machines

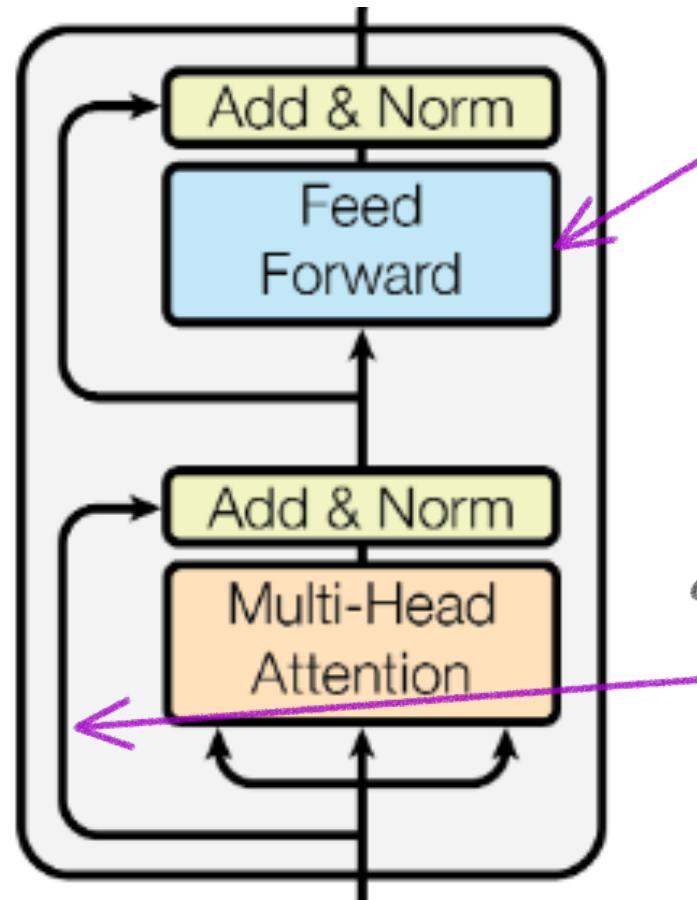


\* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one



credit: Jay Alammar

# TRANSFORMER ENCODER BLOCK



Example of FF block:

- Linear layer
- Activation (ReLU)
- Linear layer

Skip / Residual Connections:

$x$  = input of MHA

$y$  = output of MHA

$\Rightarrow x + y = \text{input of Layer Norm}$

# LAYER NORMALIZATION

Idea: normalize across features

$$x = (x_1, \dots, x_N)$$

$$\Rightarrow \mu := \frac{1}{N} \sum_{i=1}^N x_i \quad \sigma^2 := \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$$

normalize:  $\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}$

output:  $y_i = \gamma_i \hat{x}_i + \beta_i$  learnable

↑ typically  $10^{-5}$

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 the input embedding? And of the positional encoding?
- What are the main building blocks of a Transformer Encoder Block?
- Write down the formula for the "Scaled dot-product attention".

## 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