VALDOM - NLP 2 LLM

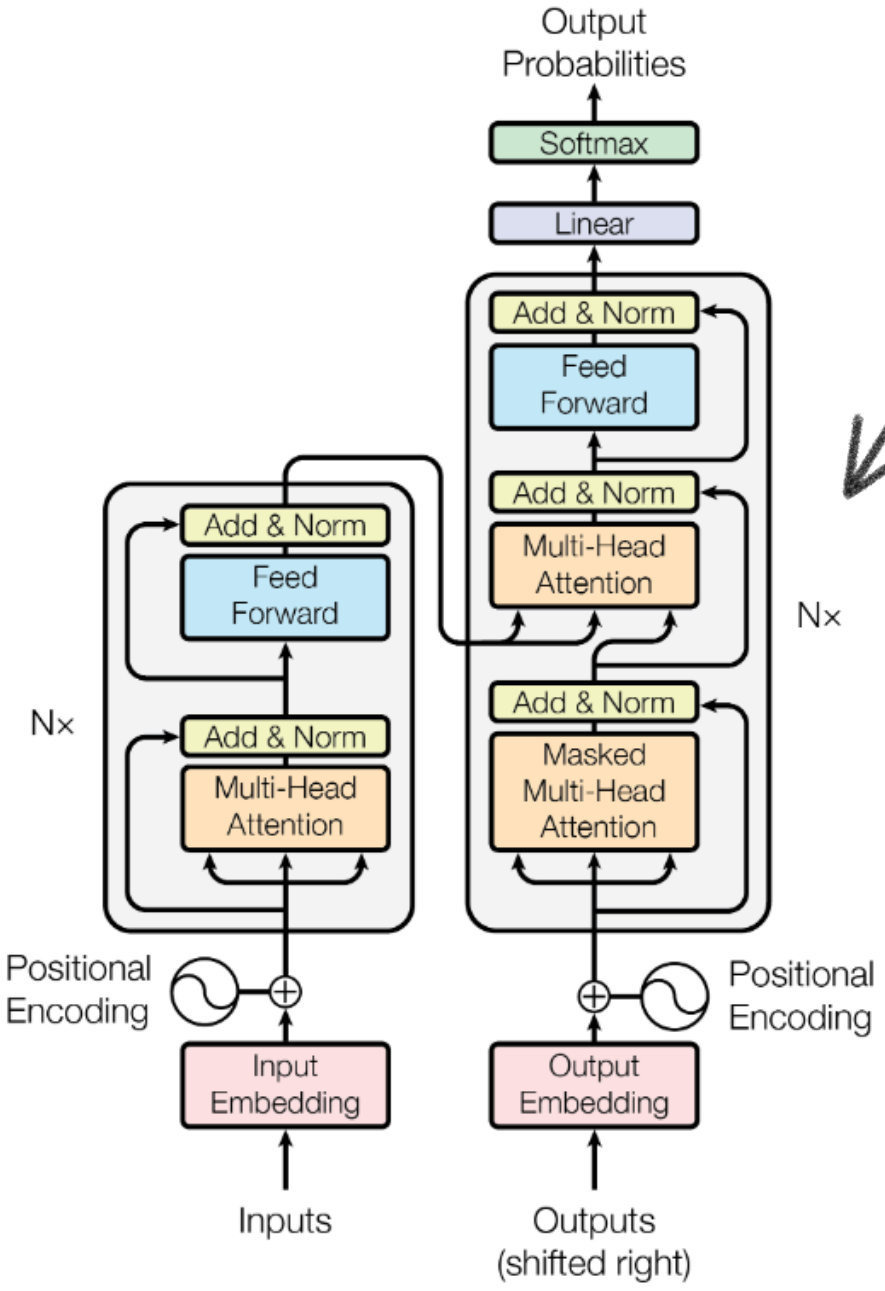
LARGE LANGUAGE MODELS

JOSEBA DALMAU

OBJECTIVE

Master the Transformer architecture for various different NLP tasks

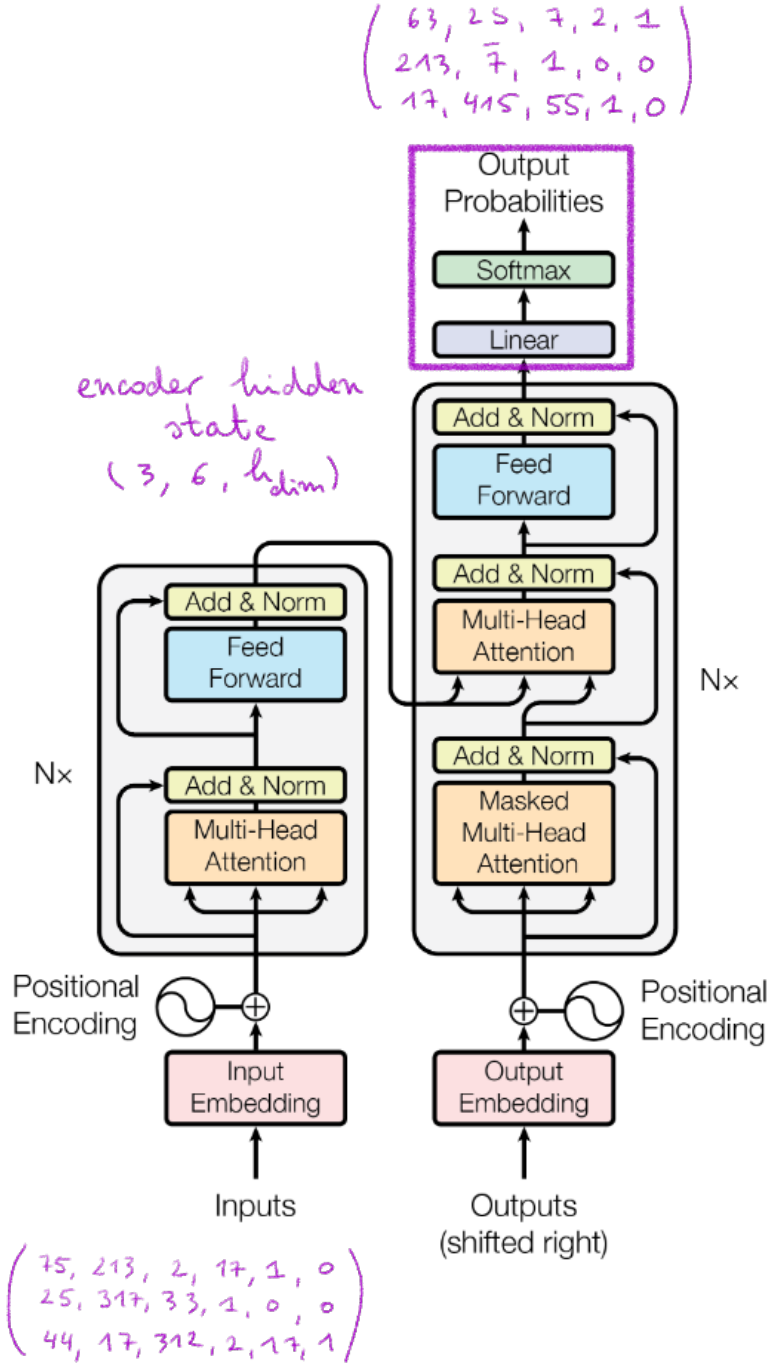
TRANSFORMERS



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

Vaswani et al. 2017

Last Lecture



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

OBJECTIVE

Master what happens

- inside the encoder transformer blocks

- inside the decoder transformer blocks (next lecture)

- before the transformer blocks

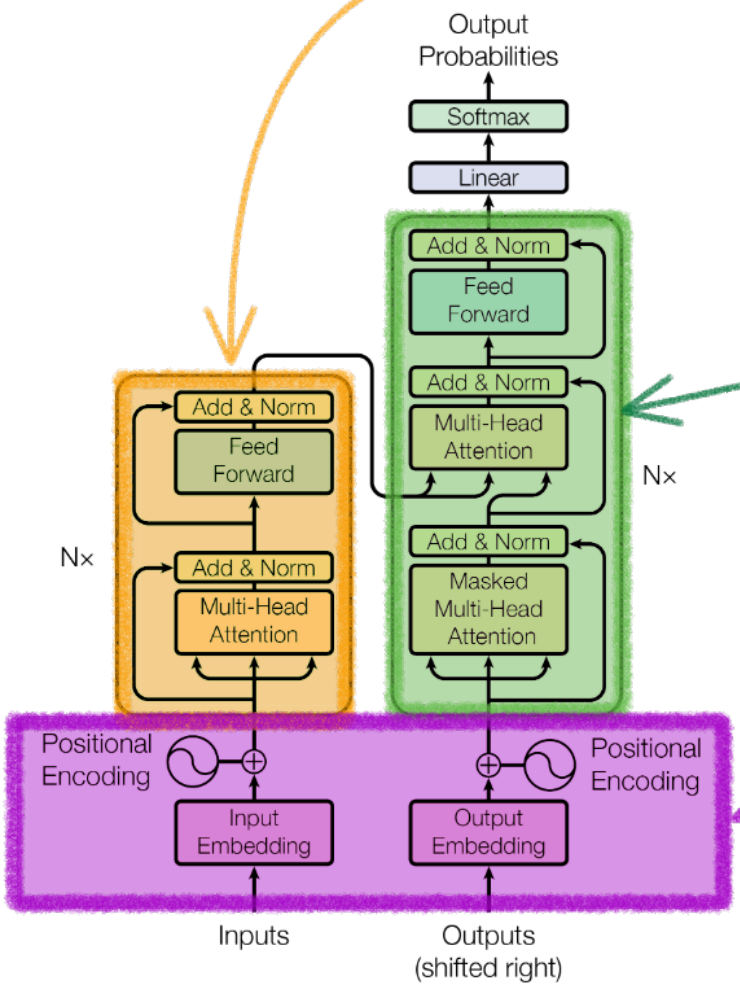
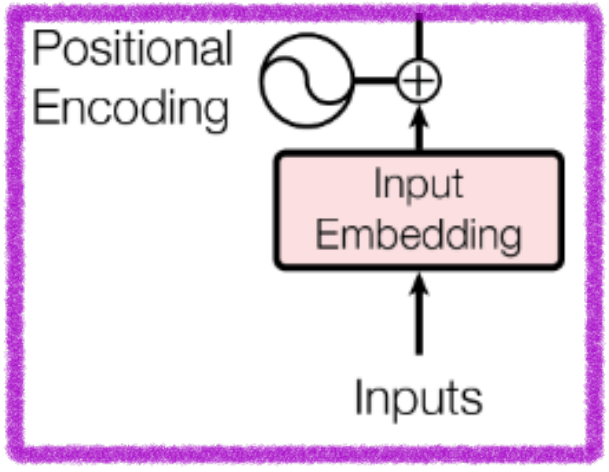


Figure 1: The Transformer - model architecture.

Input Embedding



Purpose :

token $\longmapsto$ $\mathbb{R}^{h_{dim}}$

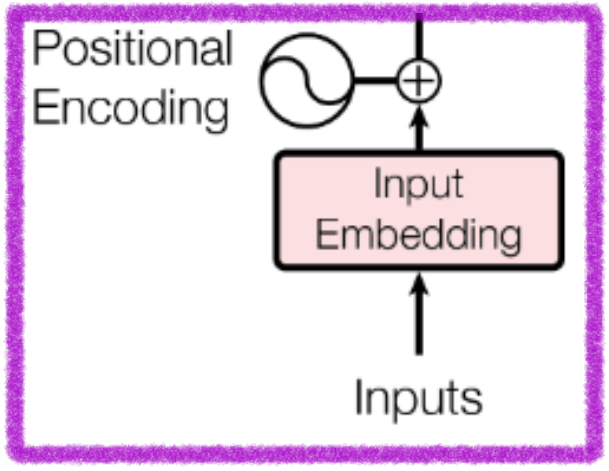
$t \in \mathcal{D}$

set of integers of size n_{vocab}

hidden dimension

How?

Input Embedding



Purpose :

token $\longmapsto$ $\mathbb{R}^{h_{dim}}$

$t \in \mathcal{D}$

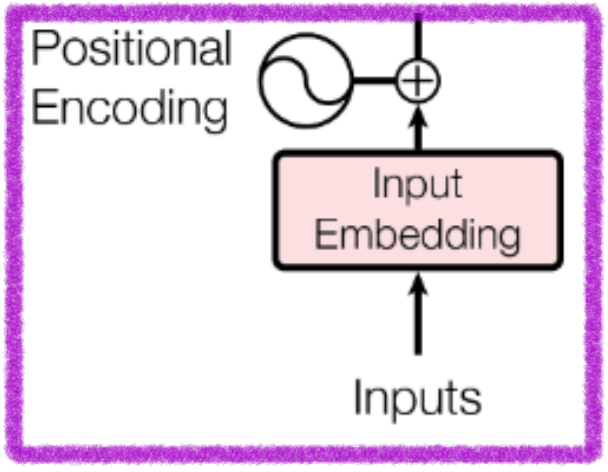
set of integers of size n_{vocab}

hidden dimension

How ?

- Word2Vec
- GloVe
- ...

Positional Encoding



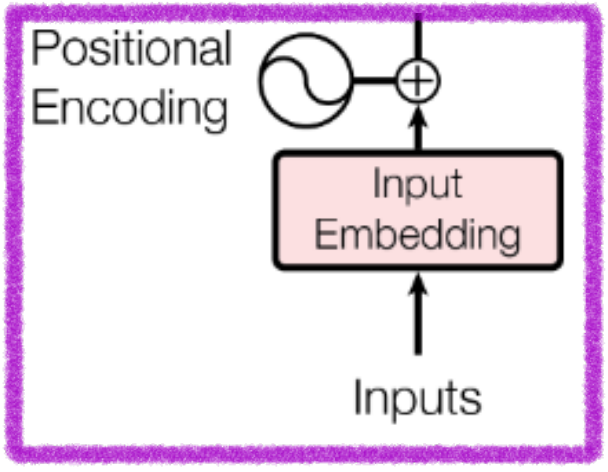
Purpose:

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



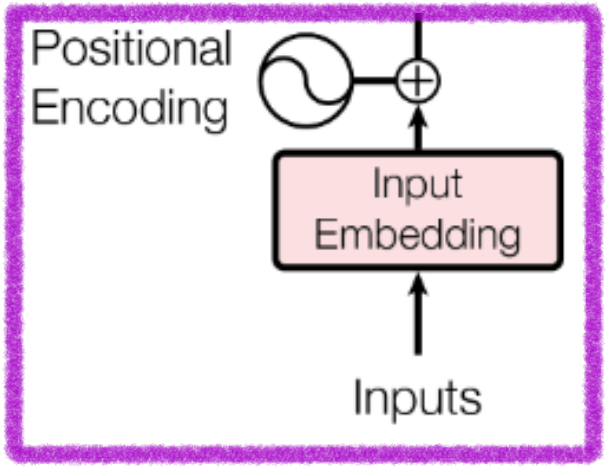
Purpose:

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

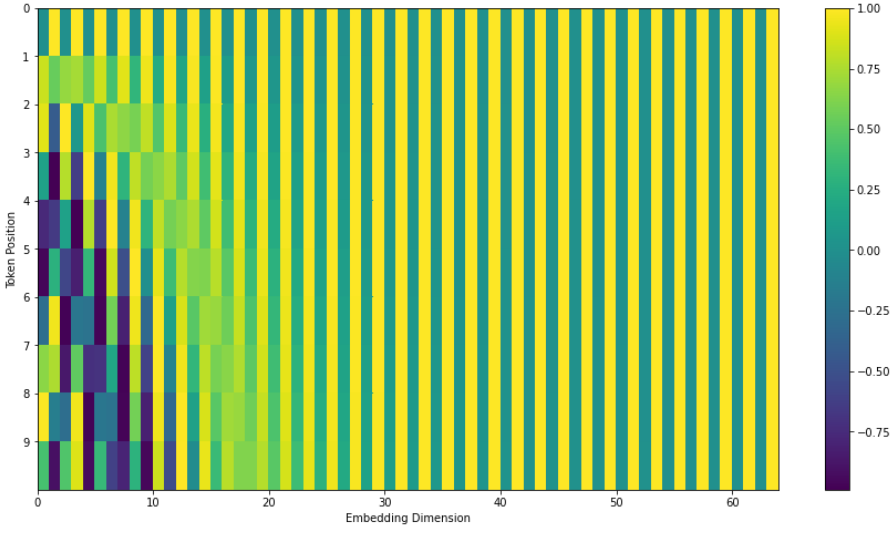
you are walking very slowly

My cat seems to like you

POSITIONAL ENCODING



Per sentence in the batch:



$$+ \begin{matrix} T_0 \\ T_1 \\ T_2 \\ \vdots \\ T_9 \end{matrix} \begin{bmatrix} 7{,}23 & -3{,}71 & 0{,}55 & \cdots & 12{,}08 \\ -11{,}2 & -2{,}24 & 0{,}09 & \cdots & -8{,}99 \\ 0{,}07 & 13{,}56 & 7{,}14 & \cdots & -6{,}64 \\ \vdots & \vdots & \vdots & & \vdots \\ 8{,}03 & 2{,}31 & -0{,}44 & \cdots & 5{,}27 \end{bmatrix}$$

$h_{dim} = 64$

Objective

Master what happens

- inside the encoder transformer blocks

- inside the decoder transformer blocks

- before the transformer blocks ✓

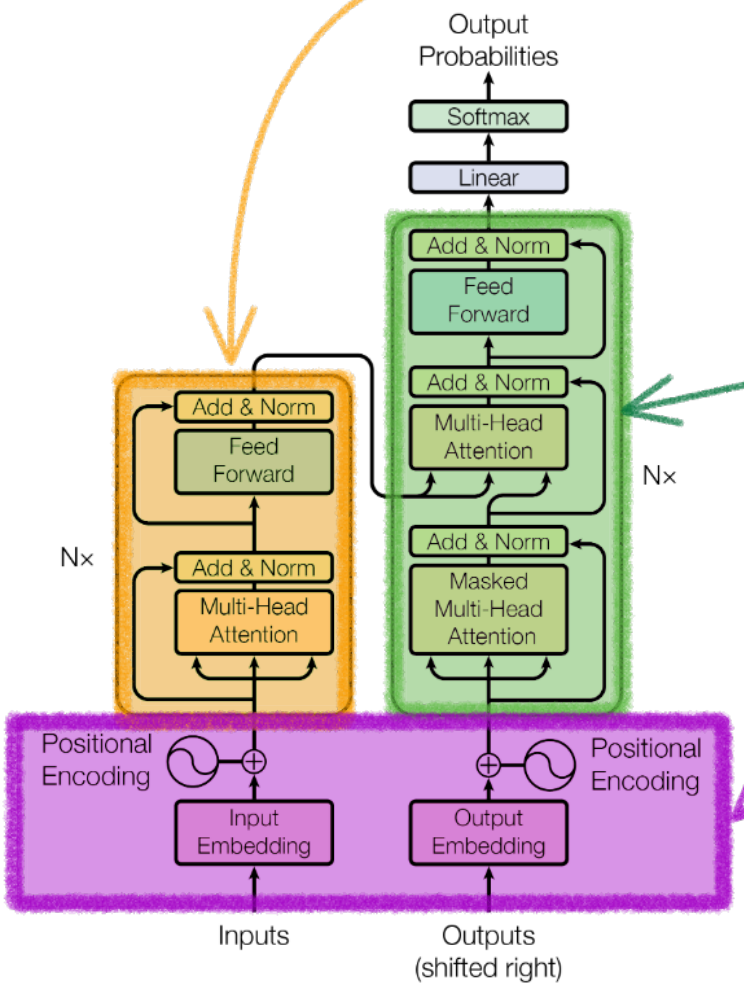
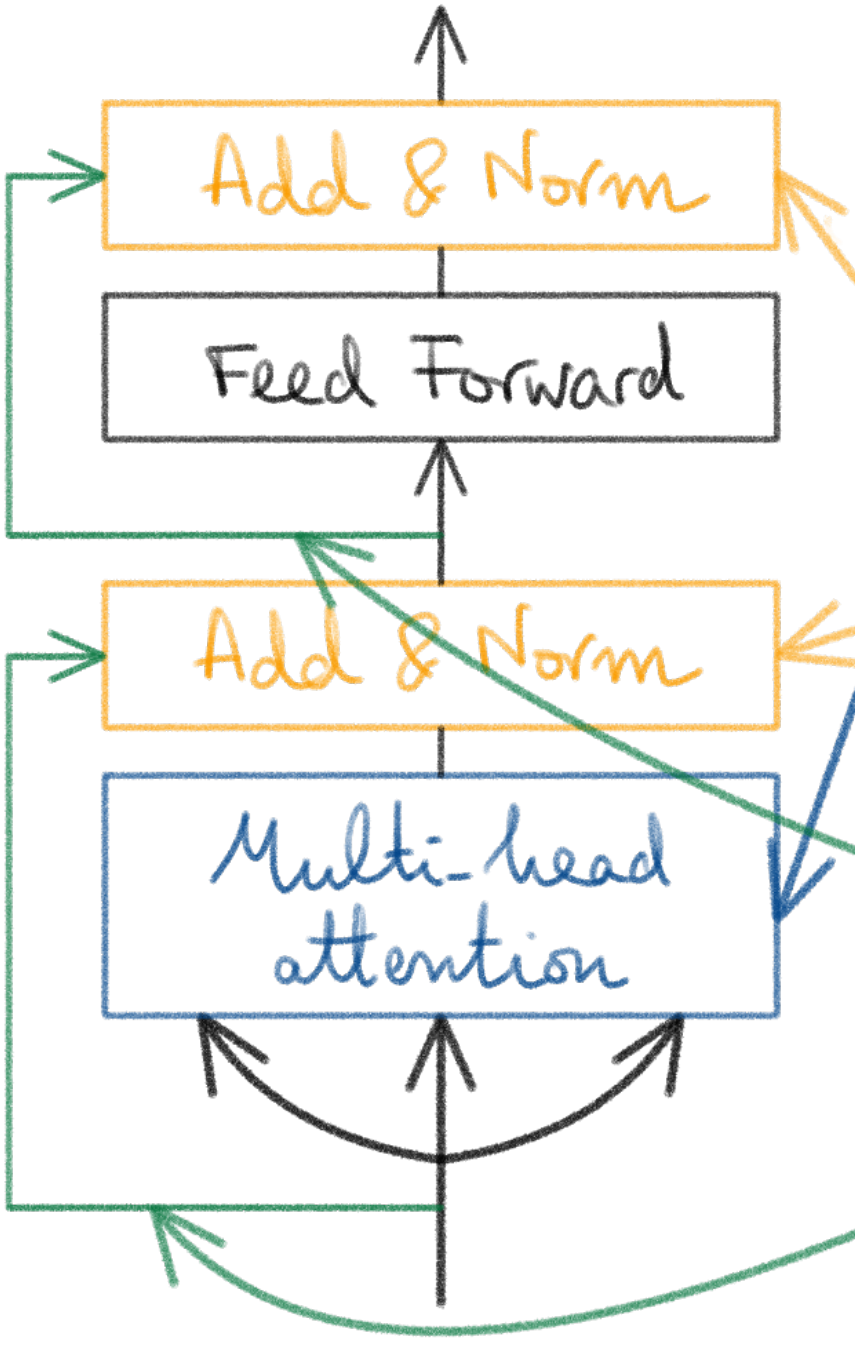


Figure 1: The Transformer - model architecture.

TRANSFORMER BUILDING BLOCKS

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

Transformer Blocks



Scaled dot product attention

Layer Normalization

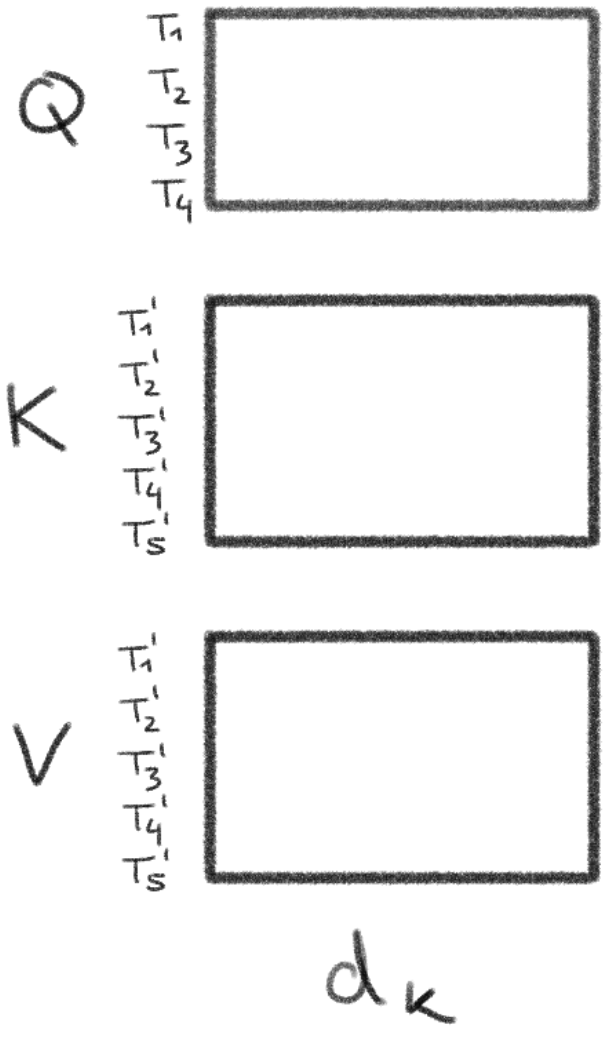
Residual Connections

Scaled Dot-Product Attention

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

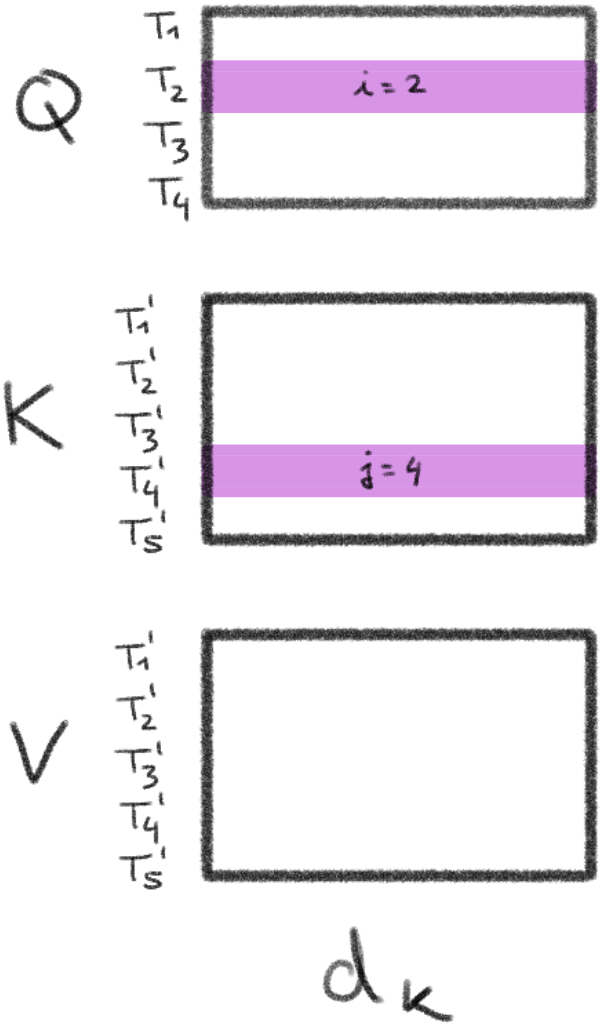
Scaled Dot-Product Attention

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



Scaled Dot-Product Attention

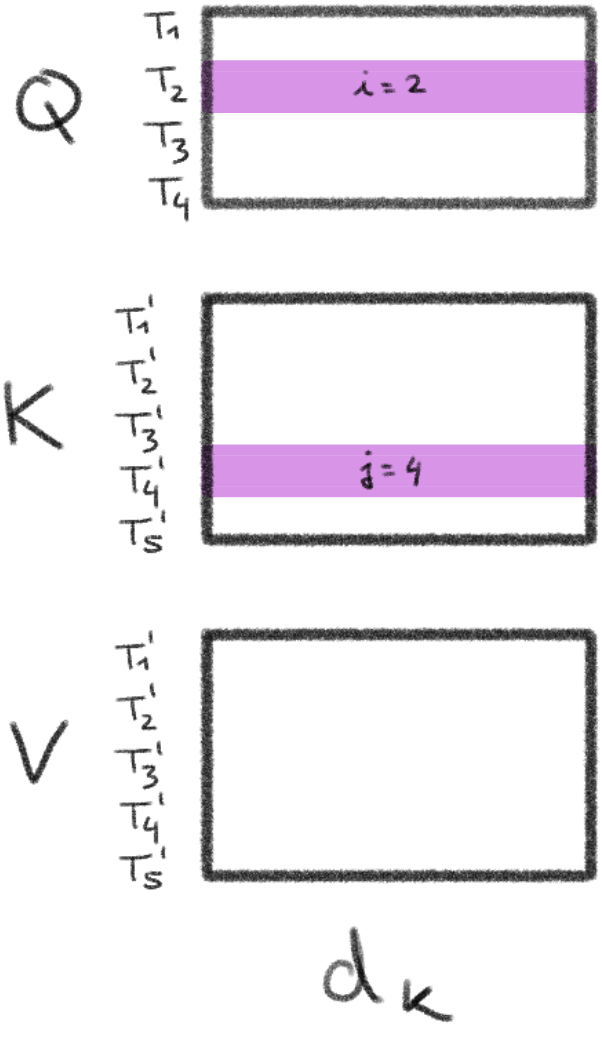
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



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

Scaled Dot-Product Attention

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



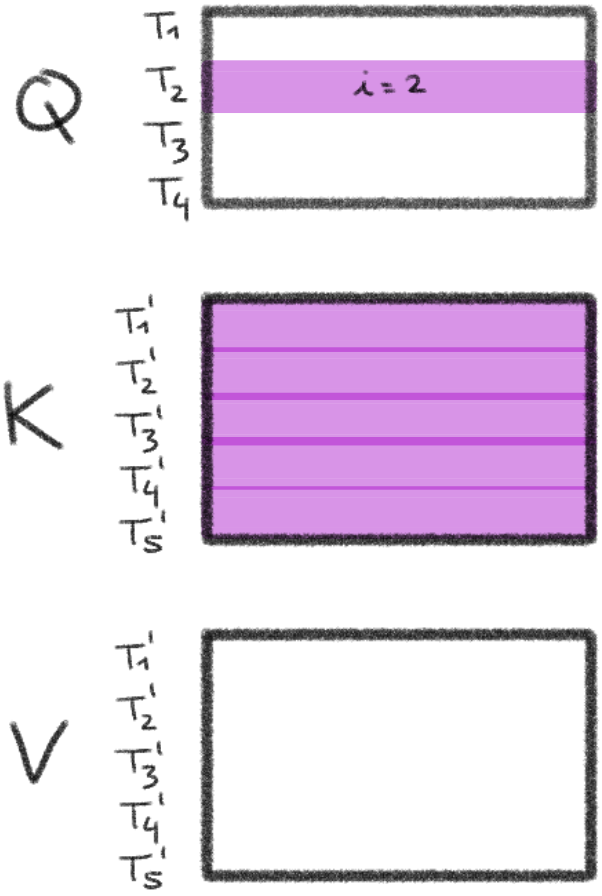
$$(QK^T)_{i,j} = \sum_{h=1}^{d_k} Q_{i,h} \cdot K_{j,h}$$

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

attention scores for the i-th query

Scaled Dot-Product Attention

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



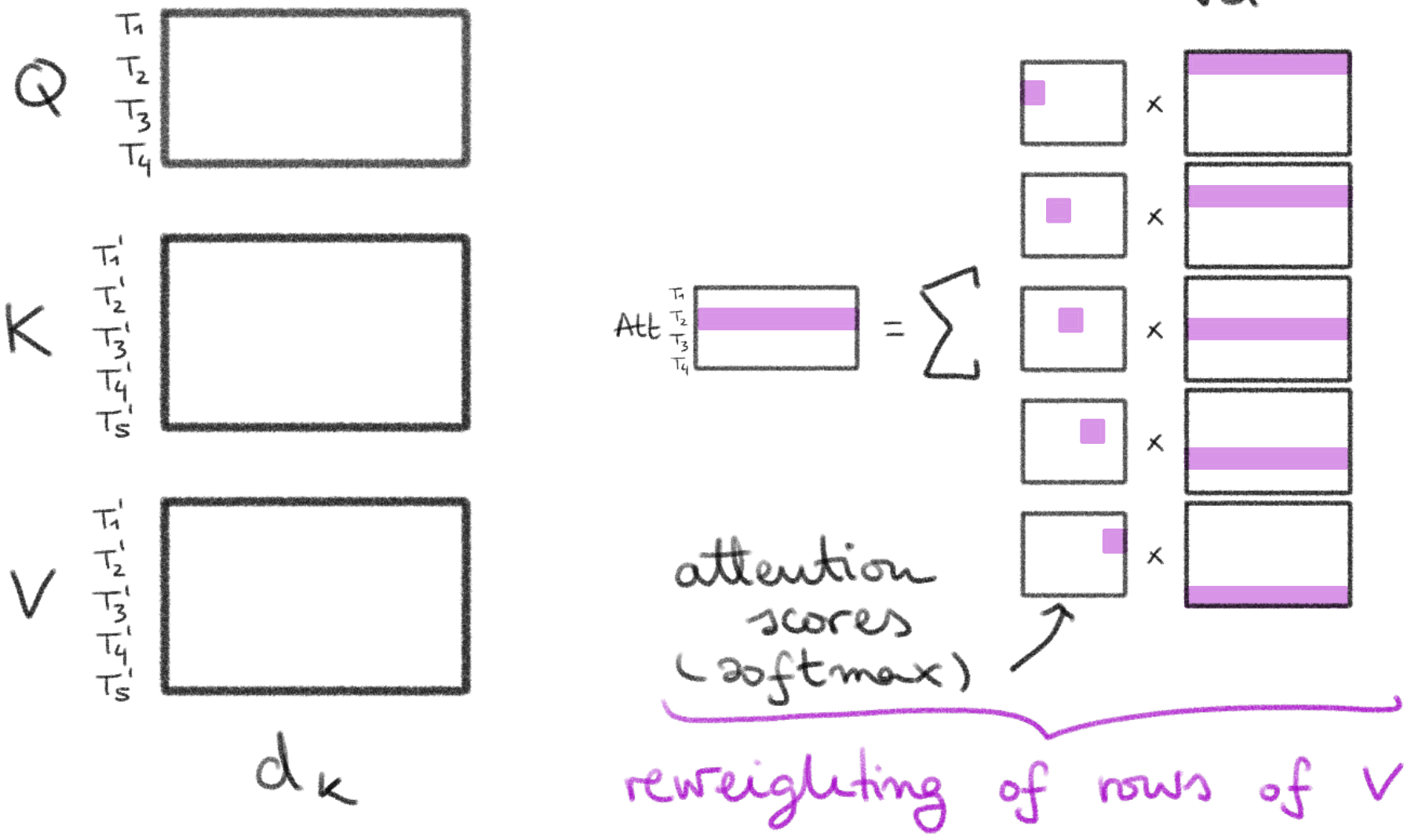
$$(QK^T)_{i,j} = \sum_{h=1}^{d_k} Q_{i,h} \cdot K_{j,h}$$

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

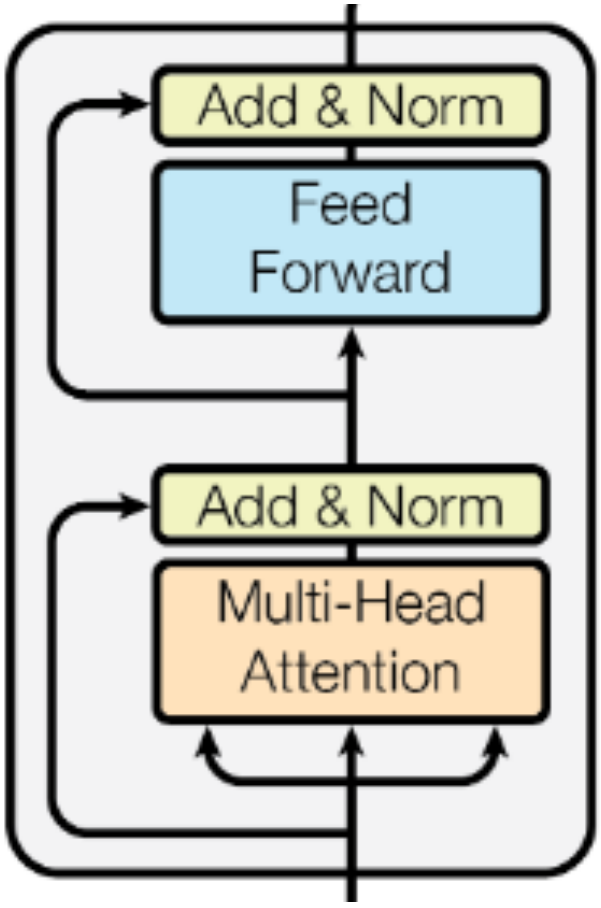
attention scores for the i-th query

Scaled Dot-Product Attention

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

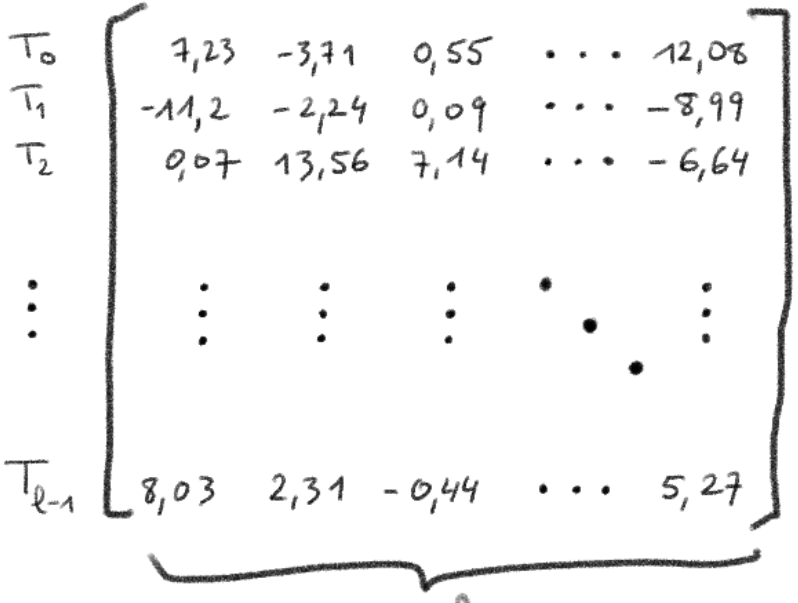


Multi-Head Attention



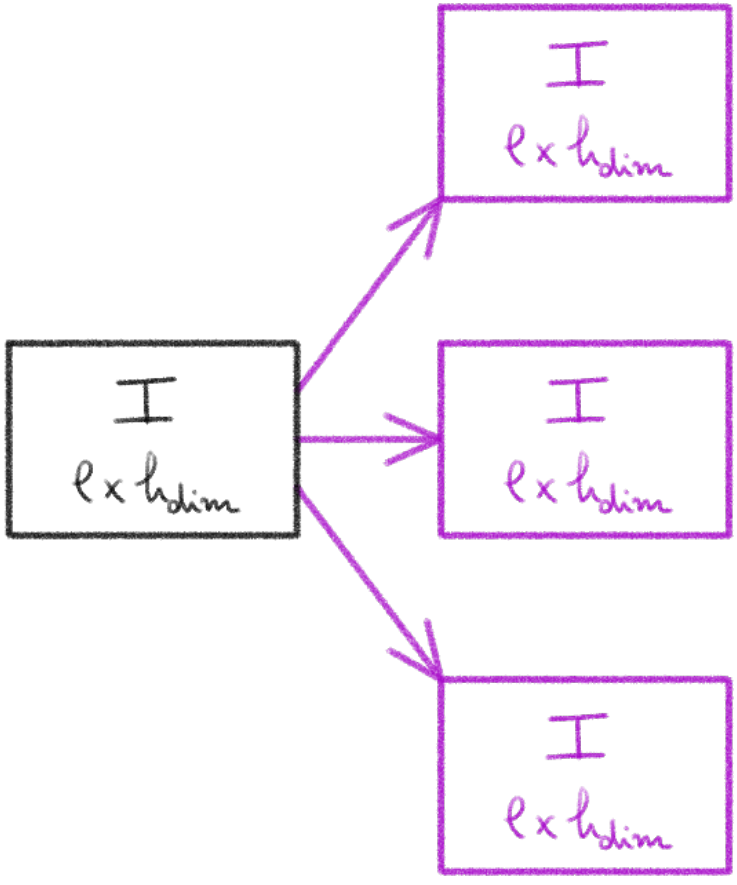
$$
\begin{matrix} T_0 \\ T_1 \\ T_2 \\ \vdots \\ T_{l-1} \end{matrix}
\begin{bmatrix}
7{,}23 & -3{,}71 & 0{,}55 & \cdots & 12{,}08 \\
-11{,}2 & -2{,}24 & 0{,}09 & \cdots & -8{,}99 \\
0{,}07 & 13{,}56 & 7{,}14 & \cdots & -6{,}64 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
8{,}03 & 2{,}31 & -0{,}44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

Multi-Head Attention

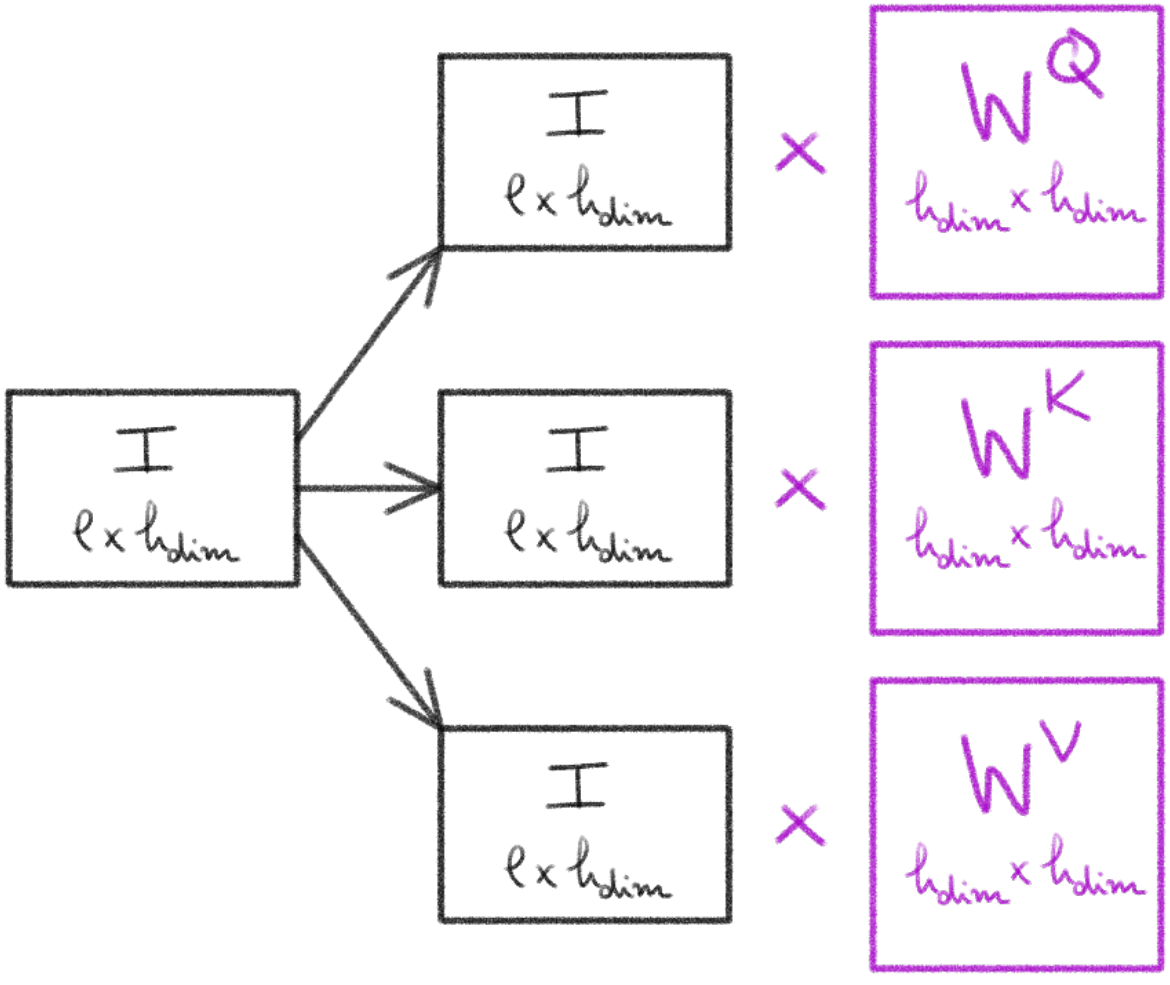


Multi-Head Attention

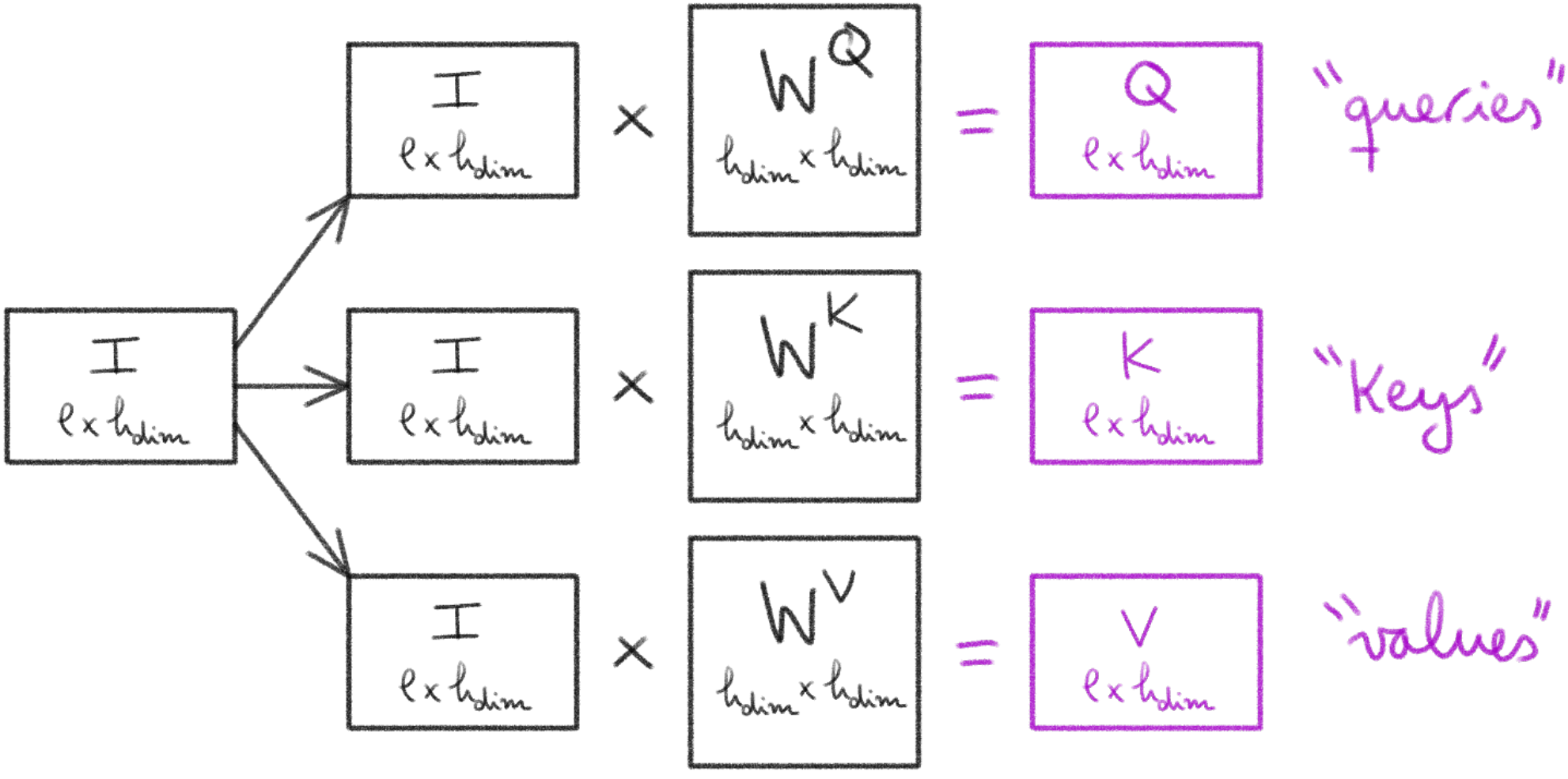
Multi-Head Attention



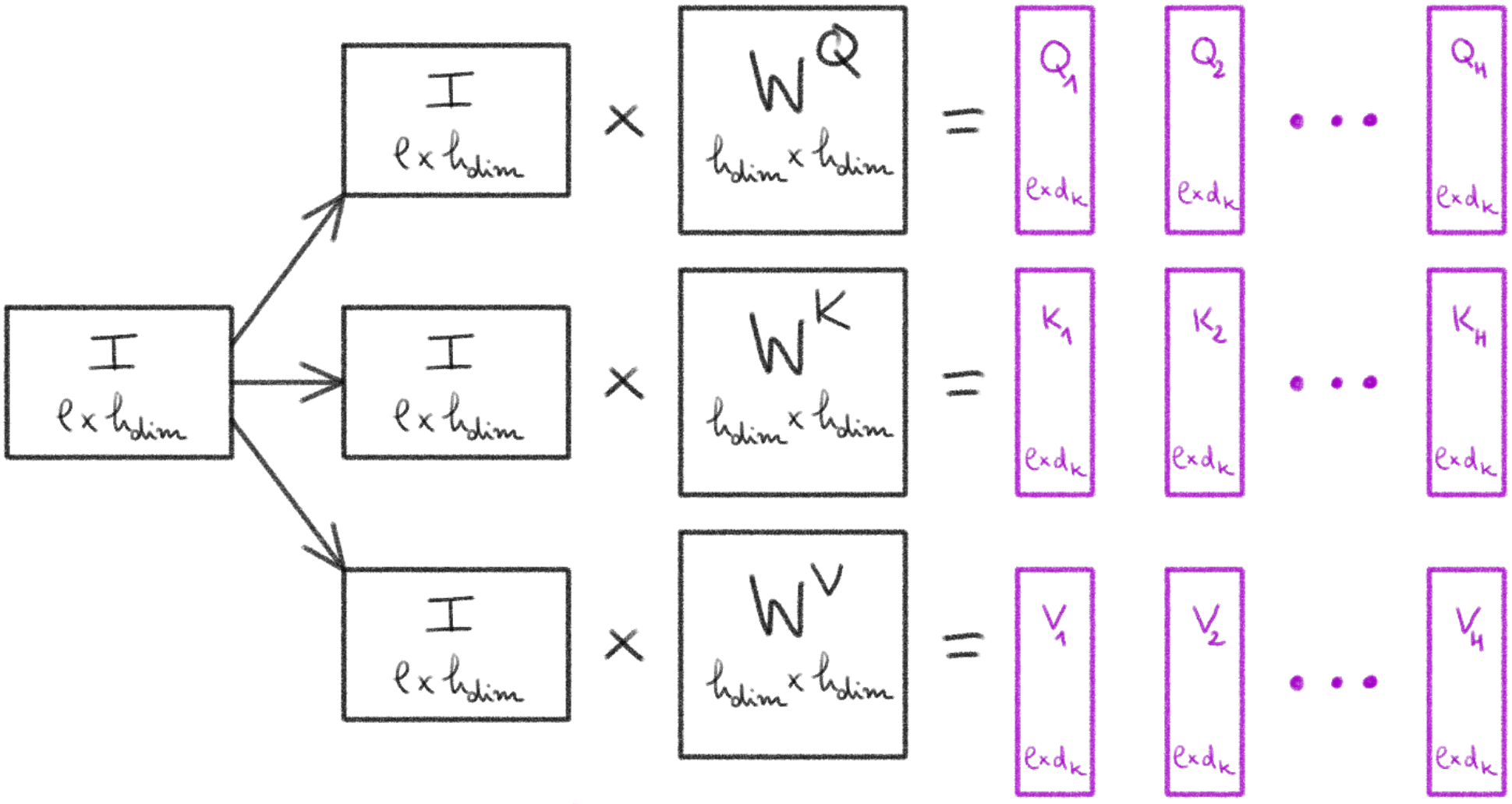
Multi-Head Attention



Multi-Head Attention



Multi-Head Attention

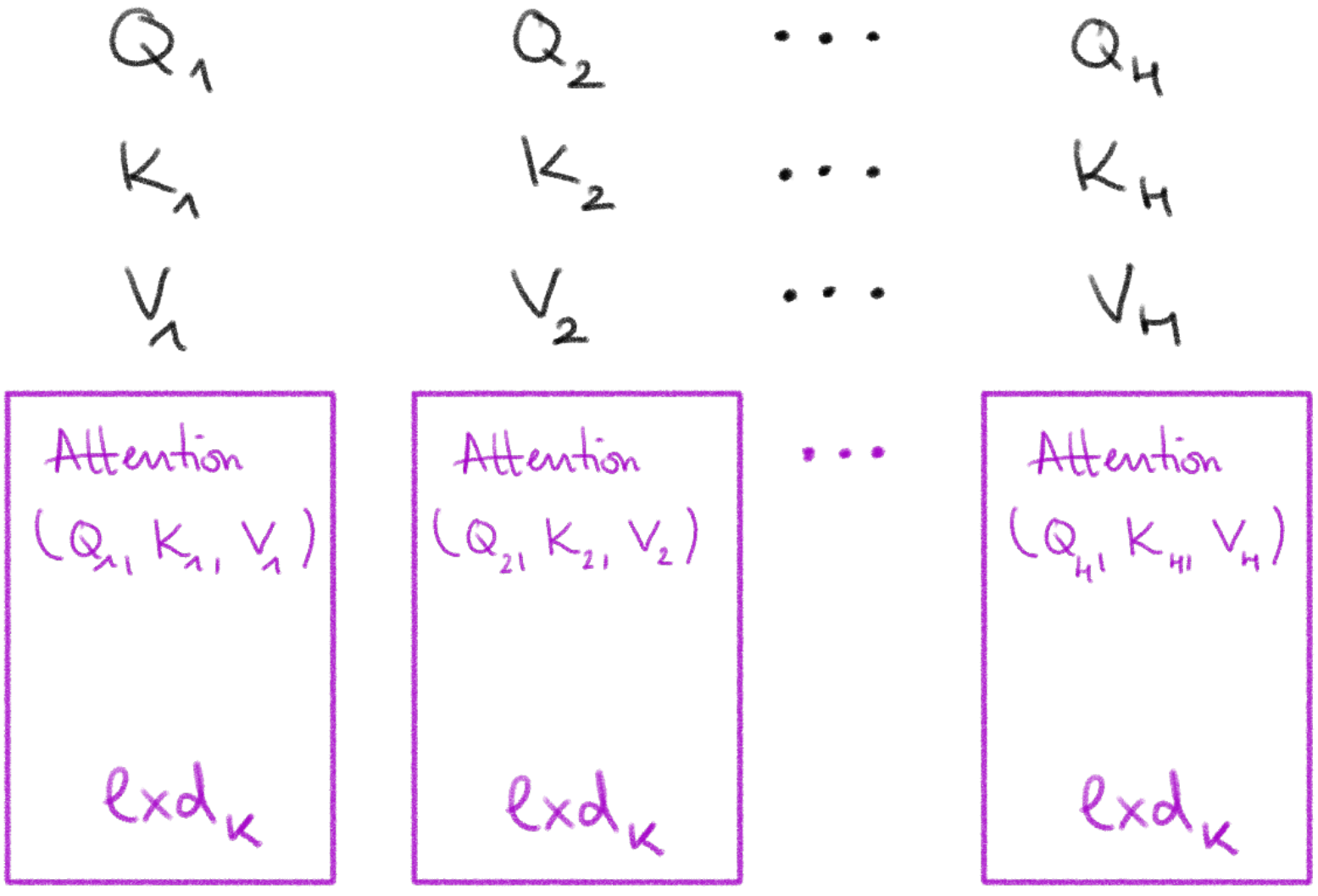


Relationship : $h_{dim} = d_k \times H$

d_k ↑ nb. of dims per head

H ↑ nb. of heads

Multi-Head Attention



Multi-Head Attention

Q_1 Q_2 ... Q_H

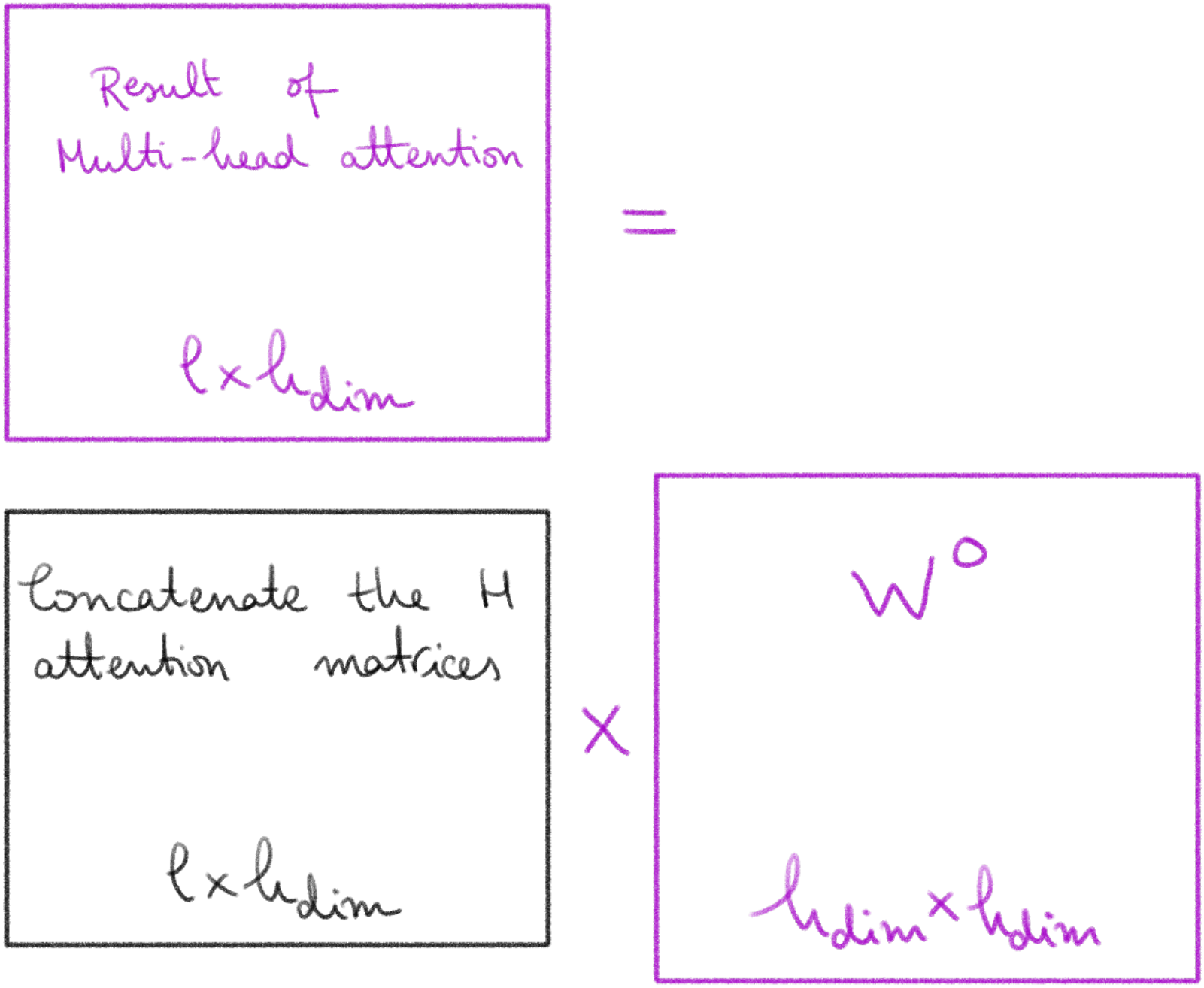
K_1 K_2 ... K_H

V_1 V_2 ... V_H

Concatenate the H attention matrices

$l \times h_{dim}$

Multi-Head Attention



Multi-Head Attention

1) This is our input sentence*

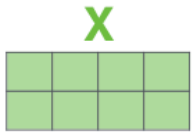
2) We embed each word*

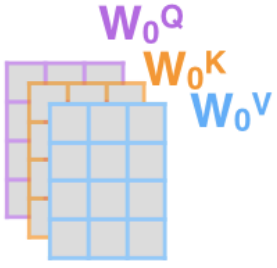
3) Split into 8 heads. We multiply X or R with weight matrices

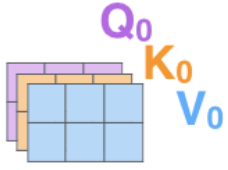
4) Calculate attention using the resulting Q/K/V matrices

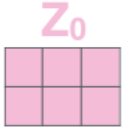
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

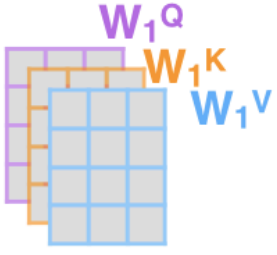
Thinking Machines

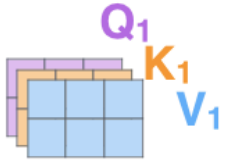


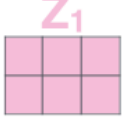


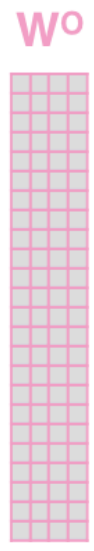


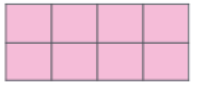




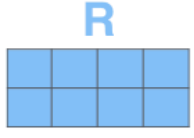


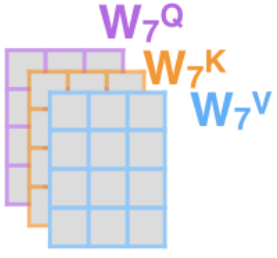


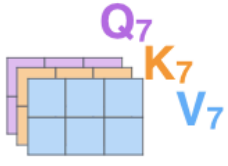


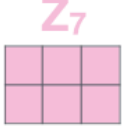


* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one



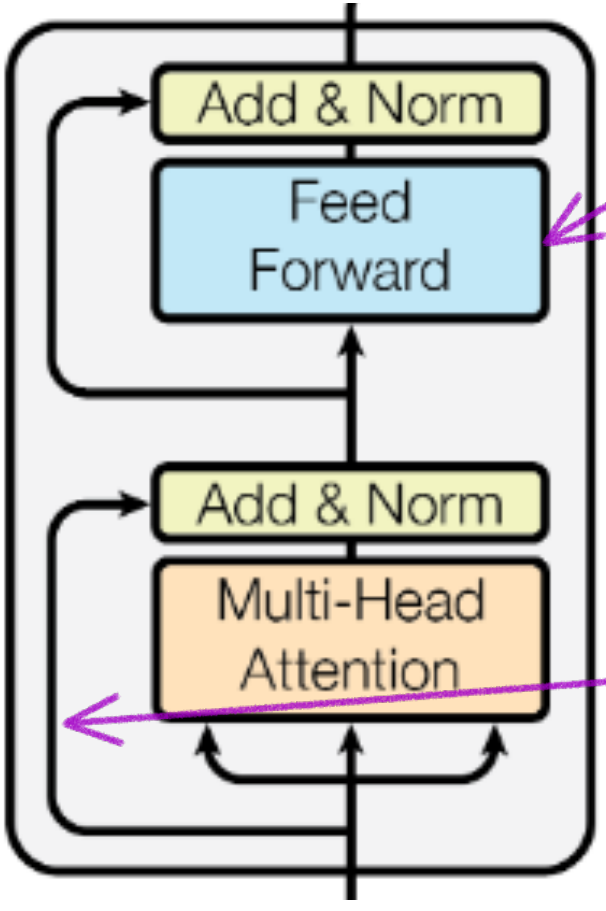






credit: Jay Alammar

Transformer Encoder Block



Example of FF block:

- Linear layer
- Activation (ReLU)
- Linear layer

Skip / Residual Connections:

x = input of MHA

y = output of MHA

$\Rightarrow x + y$ = input of LayerNorm

Layer Normalization

Idea: normalize across features

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

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

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

↑ typically 10^{-5}

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

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. 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