E9: 309 Advanced Deep Learning
2-11-2020

Instructor: Sriram Ganapathy
sriramg@iisc.ac.in

Teaching Assistant : Akshara Soman, Prachi Singh, Jaswanth Reddy
aksharas@iisc.ac.in, prachis@iisc.ac.in, jaswanthk@iisc.ac.in

http://leap.ee.iisc.ac.in/sriram/teaching/ADL2020/

Housekeeping

- Attendance
  - We will use the recorded sessions for attendance
    - If you are unable to attend live sessions (due to network or other issues, please indicate by email before or after class to the instructor and copy the FAs).
- Mid-term exam
  - 1st week of Dec. (Modules I and II).

Housekeeping

- 1st mini-project
  - Deadlines
    - Abstract submission deadline (Nov 2nd, Monday)
      - Using the google form given in the webpage
    - Solo projects or 2-member projects
      - Indicate roles of each member in 2-member project
      - 200 page abstract of the work. If modifications are needed, we will review and let you know in 2-3 days.

Housekeeping

- 1st mini-project
    - Deadlines
        - Report and presentation slides (Nov 18th, 10 AM).
            - 1-page pdf with second page only for references and tools used (Template will be provided).
        - Report - Indicate prior work, technical details and your contribution. Strictly adhere to the guidelines given in the template.
        - Slides (max 4 slides) - 4 min presentation for solo project and 6 min. for two member teams. 3 mins for your presentation and 1 min for Q&A.
        - Two slots are available on 2 days (pick the suitable based on your other class schedules).

Recap of previous class

State of affairs

- Encoder-decoder models with attention.
  - ➡ self attention and multi-head attention
- Transformer models - Introduction

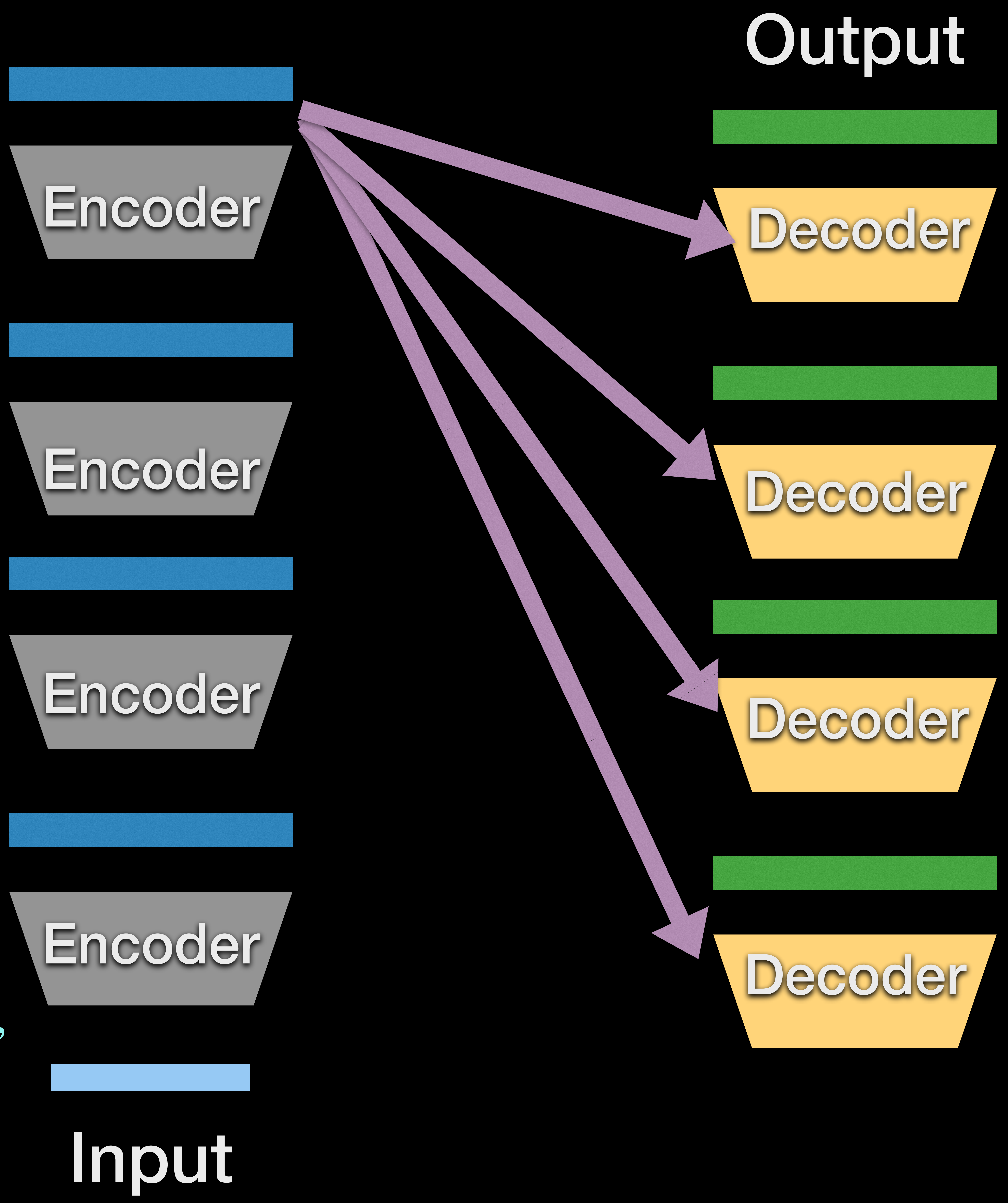
Transformers

* Encoder Decoder architecture based models.

* Uses only feed forward architectures with self-attention.

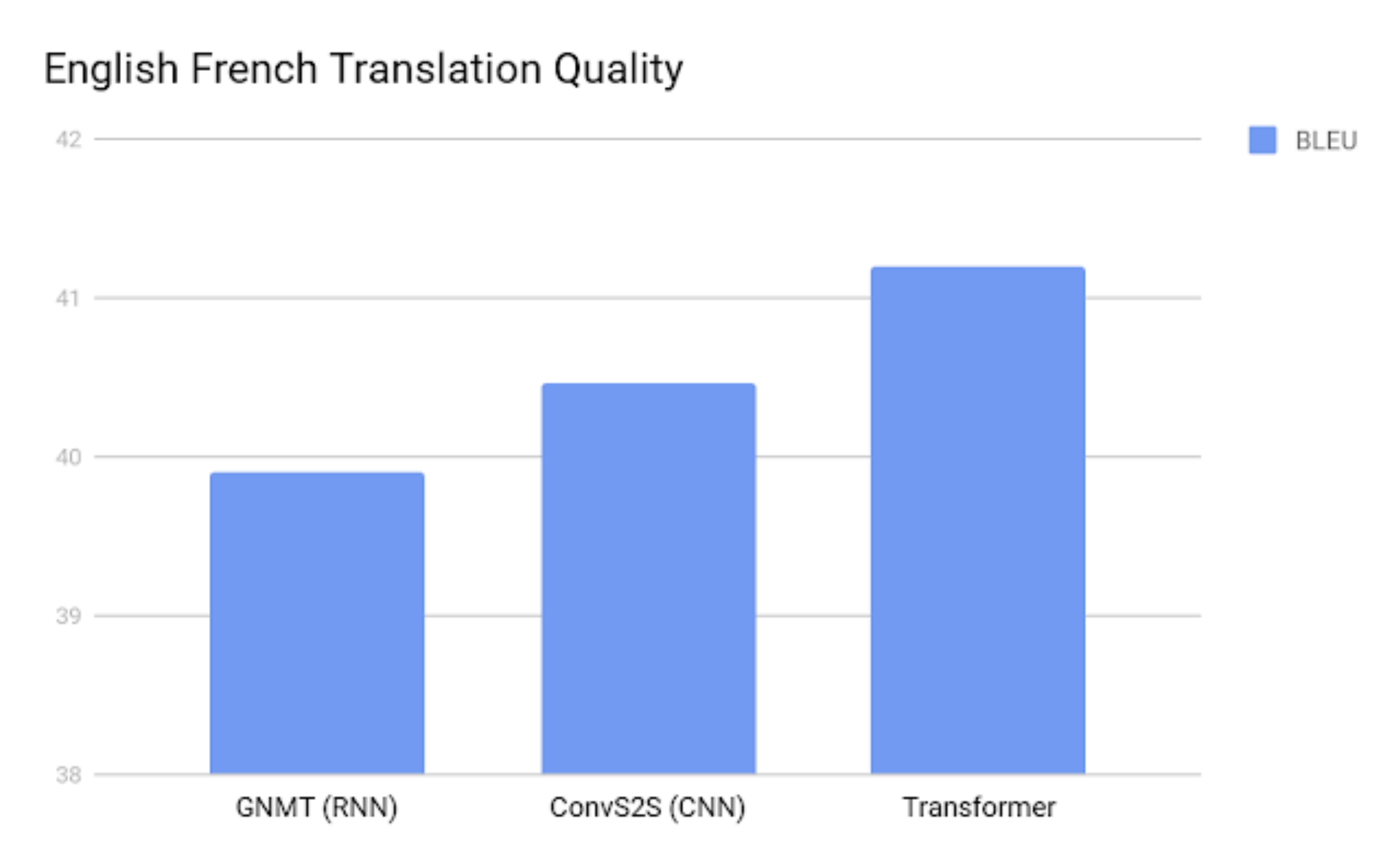
  ➡ Multi-head self attention.

* All the encoder layers and the decoder layers have the same set of operations.



Reading Assignment - "Attention is All You Need"
https://arxiv.org/pdf/1706.03762.pdf

Transformers - the state of art in NMT



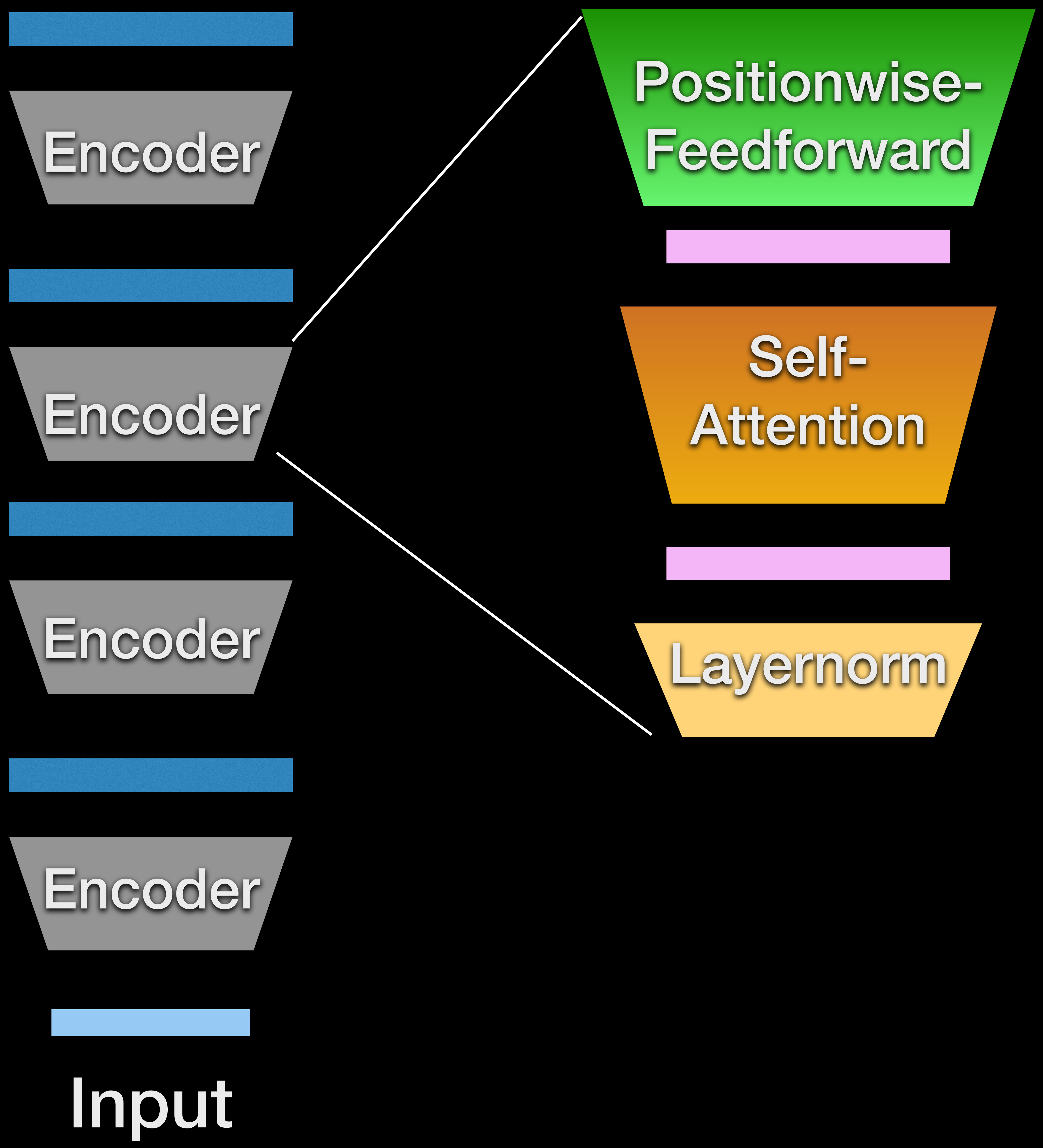
https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Transformers - the state of art in NMT

https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Transformers

- Encoder layers
  - Consist of layer norm
  - Self attention (multi-head)
  - Positionwise feedforward
    - May also consist of skip connections.



Transformers - encoder

- Let $\mathbf{x}(1)...\mathbf{x}(T)$ denote the input and let $\mathbf{e}^l(1)...\mathbf{e}^l(T)$ denote encoder outputs at layer l.

$$\overline{\mathbf{E}}^{l-1} = Layernorm([\mathbf{e}^{l-1}(1)...\mathbf{e}^{l-1}(T)]^T) \in \mathcal{R}^{T \times D}$$

- Definition of layer norm

$$Layernorm(\mathbf{e}^l(t)) = \frac{\boldsymbol{\alpha}^l}{\boldsymbol{\sigma}_{\mathbf{e}^l(t)}} \odot (\mathbf{e}^l(t) - \boldsymbol{\mu}_{\mathbf{e}^l(t)}) + \boldsymbol{\beta}^l$$

Transformers - encoder

- Querry, Key and Value

$$\mathbf{Q}_h^l = \overline{\mathbf{E}}^{l-1}\mathbf{W}_h^{l,Q} + \mathbf{1}(\mathbf{b}_h^{l,Q})^T \in \mathcal{R}^{T\times d}$$

$$\mathbf{K}_h^l = \overline{\mathbf{E}}^{l-1}\mathbf{W}_h^{l,K} + \mathbf{1}(\mathbf{b}_h^{l,K})^T \in \mathcal{R}^{T\times d}$$

$$\mathbf{V}_h^l = \overline{\mathbf{E}}^{l-1}\mathbf{W}_h^{l,V} + \mathbf{1}(\mathbf{b}_h^{l,V})^T \in \mathcal{R}^{T\times d}$$

- $\mathbf{W}_h^{l,Q}, \mathbf{W}_h^{l,K}, \mathbf{W}_h^{l,V} \in \mathcal{R}^{D\times d} \quad \mathbf{b}_h^{l,Q}, \mathbf{b}_h^{l,K}, \mathbf{b}_h^{l,V} \in \mathcal{R}^{d\times 1}$

$h = \{1..H\}$ heads $\qquad d = \dfrac{D}{H} \qquad \mathbf{1} \in \mathcal{R}^{T\times 1}$ all ones

Transformers - encoder

- Multi-head attention

$$\hat{\mathbf{A}}_h^l = \mathbf{Q}_h^l(\mathbf{K}_h^l)^T \in \mathcal{R}^{T\times T}$$

$$\hat{\mathbf{A}}_h^l = softmax(\frac{\hat{\mathbf{A}}_h^l}{\sqrt{d}})$$

$$\mathbf{C}_h^l = \mathbf{A}_h^l \mathbf{V}_h^l \in \mathcal{R}^{T\times D}$$

- Context vector from self-attention

$$\mathbf{C}^l = [\mathbf{C}_1^1 ... \mathbf{C}_H^l] \in \mathcal{R}^{T\times D}$$

Transformer - encoder

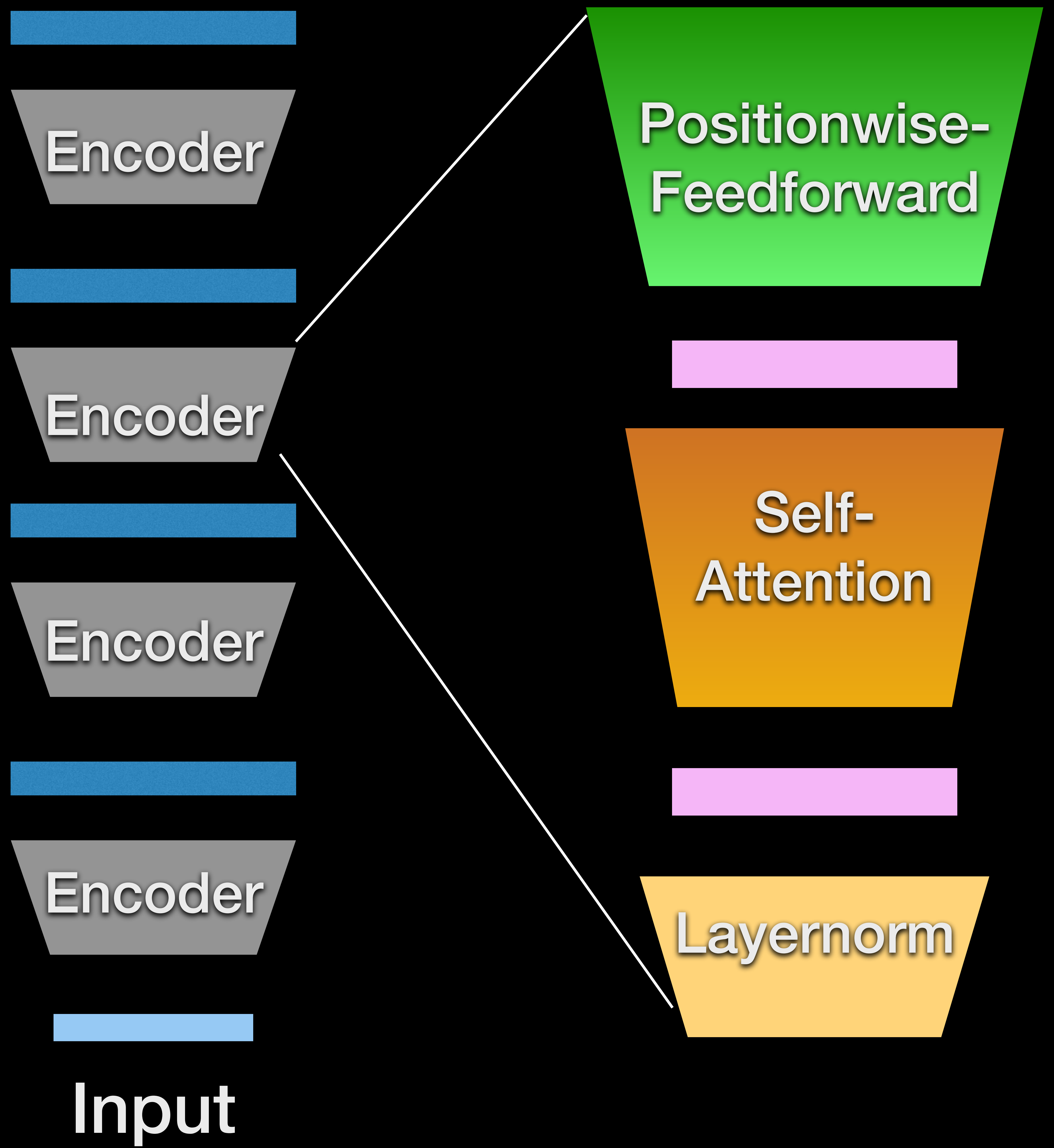
- Position wise feedforward layer

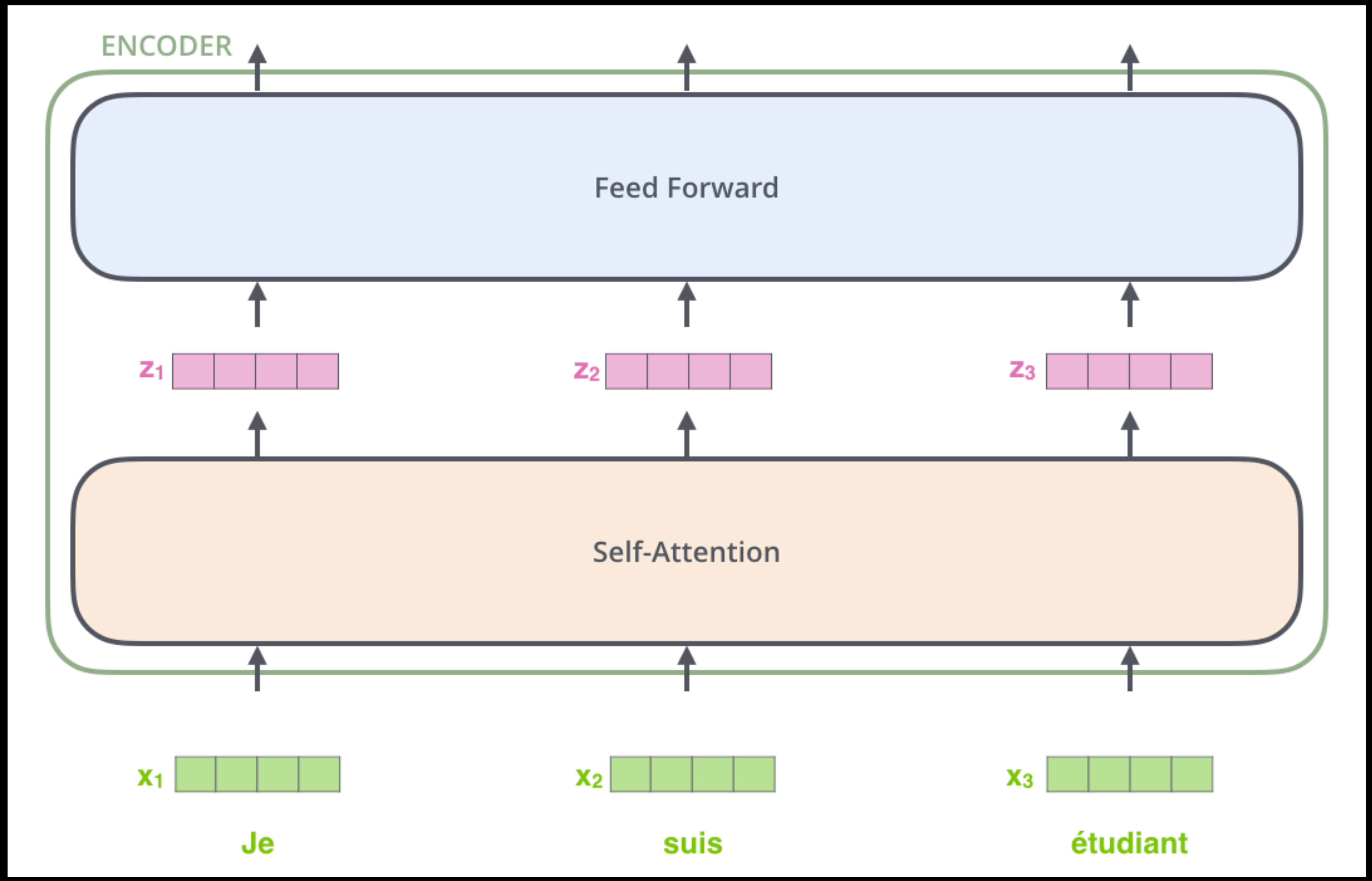
$$\mathbf{E}^l_{ff} = ReLU(\mathbf{C}^l\mathbf{W}^l_{ff} + \mathbf{1}\mathbf{b}^T_{ff}) \in \mathcal{R}^{T\times d_{ff}}$$

- Encoder layer output

$$[\mathbf{e}^l(1)...\mathbf{e}^l(T)] = \mathbf{E}^l_{ff}\mathbf{W}^l_{of} + \mathbf{1}(\mathbf{b}^l_{of})^T \in \mathcal{R}^{T\times D}$$

Transformers - encoder



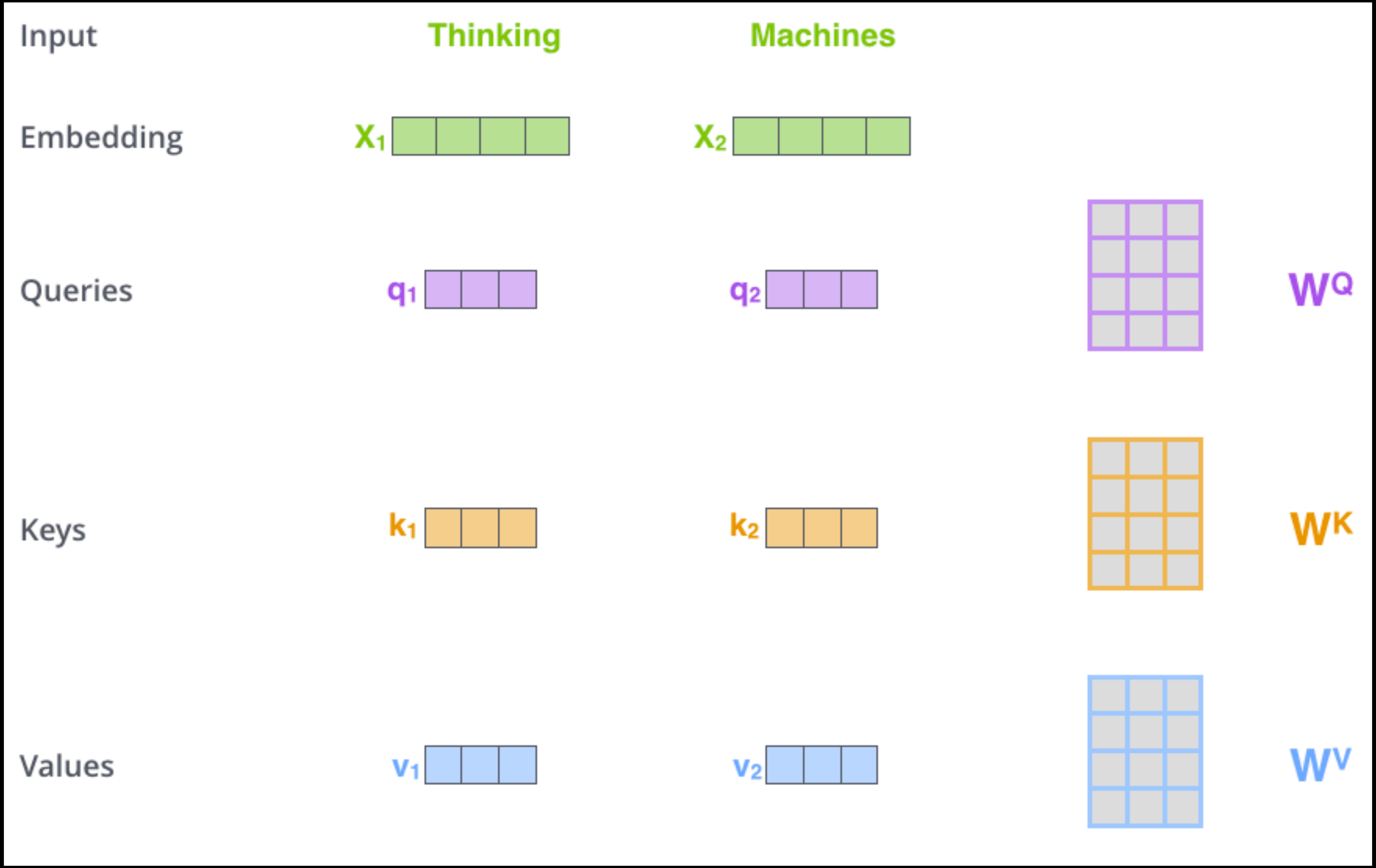


Self Attention - recap

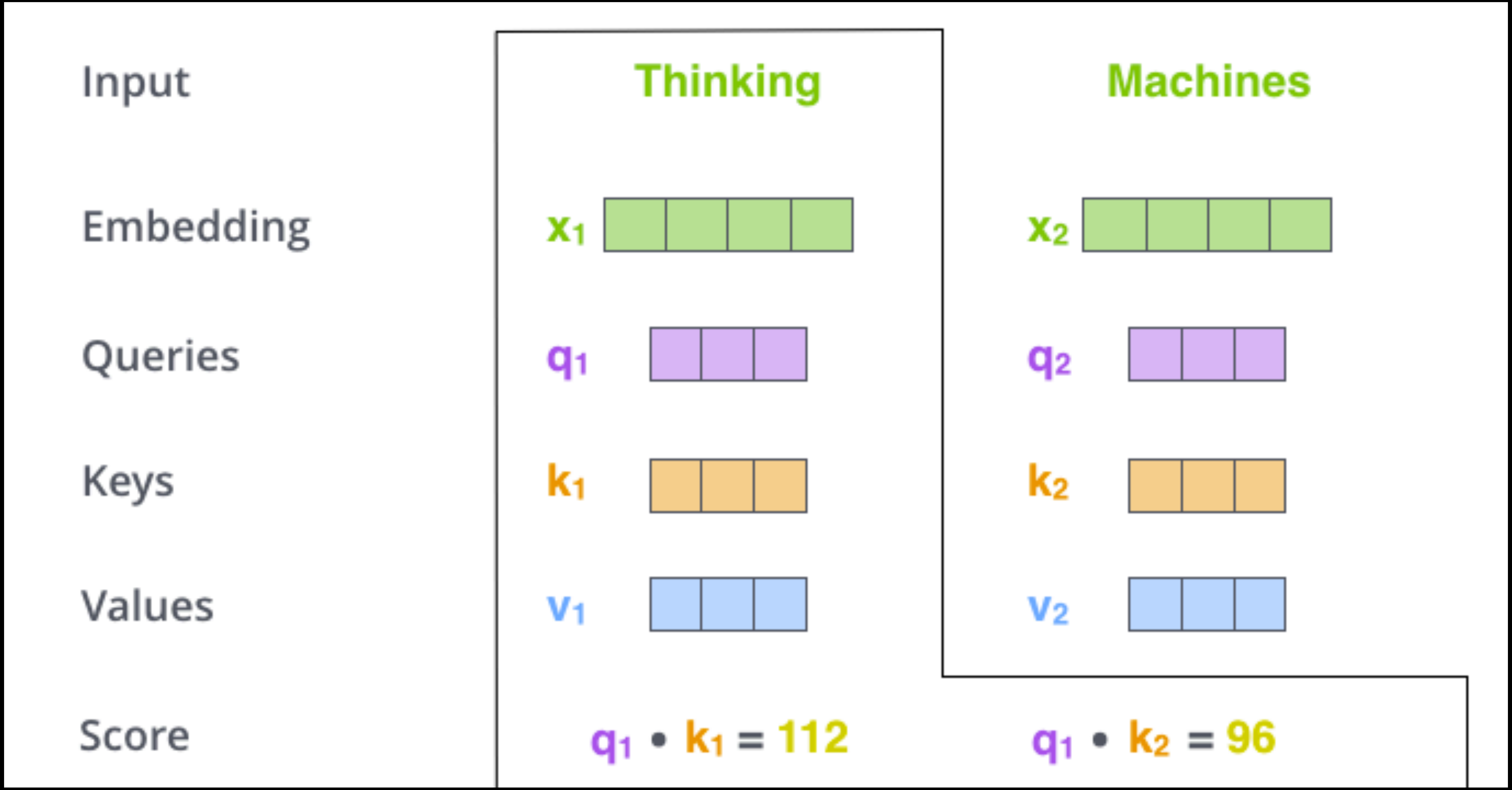
* Illustrative example - The quick brown fox (English) —> Der shnelle brane fuchs (German)

	THE	QUICK	BROWN	FOX
THE	0.9	0.1	0	0
QUICK	0.1	0.75	0	0.15
BROWN	0	0	0.7	0.3
FOX	0	0.2	0.35	0.55

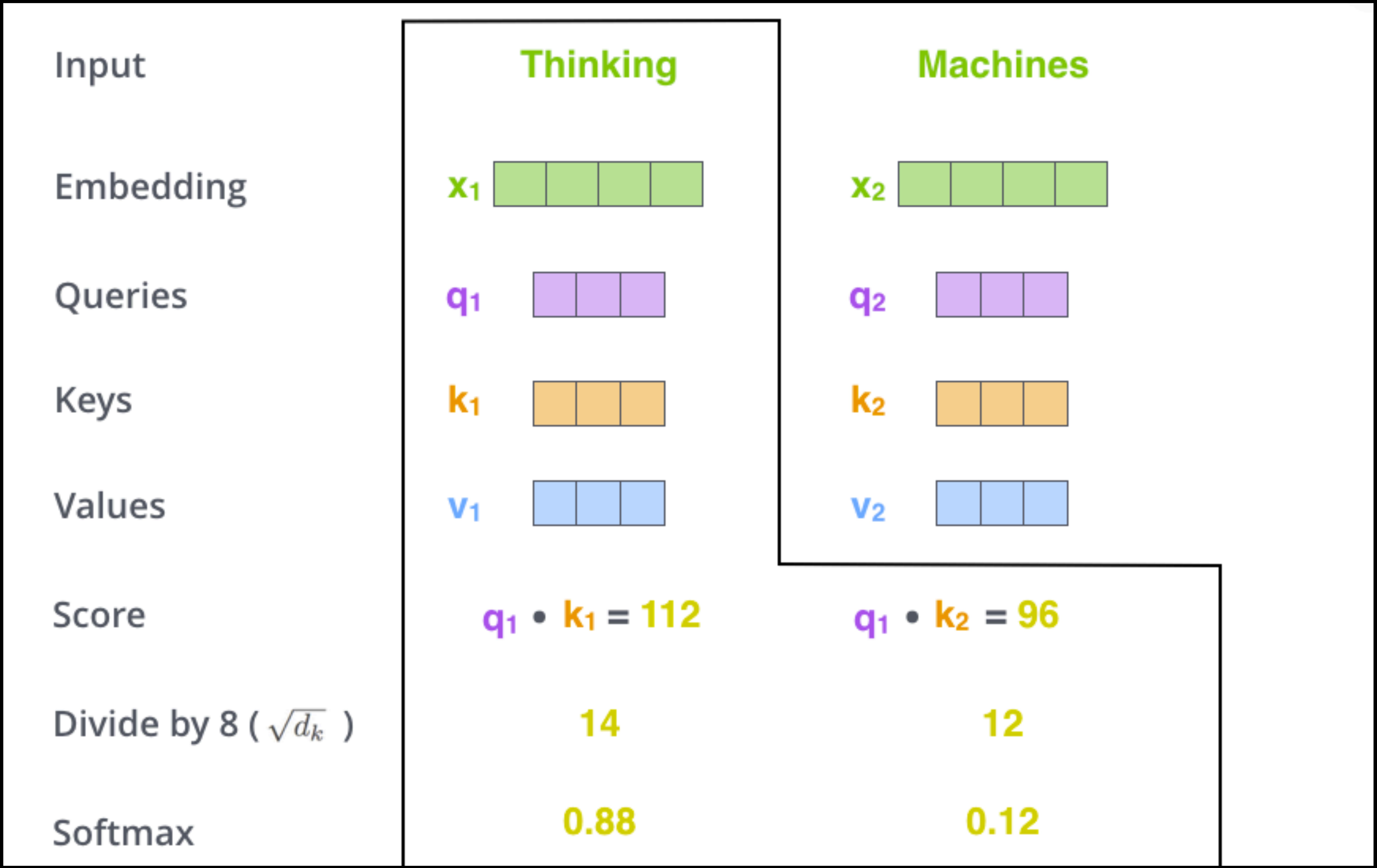
Self-attention revisited



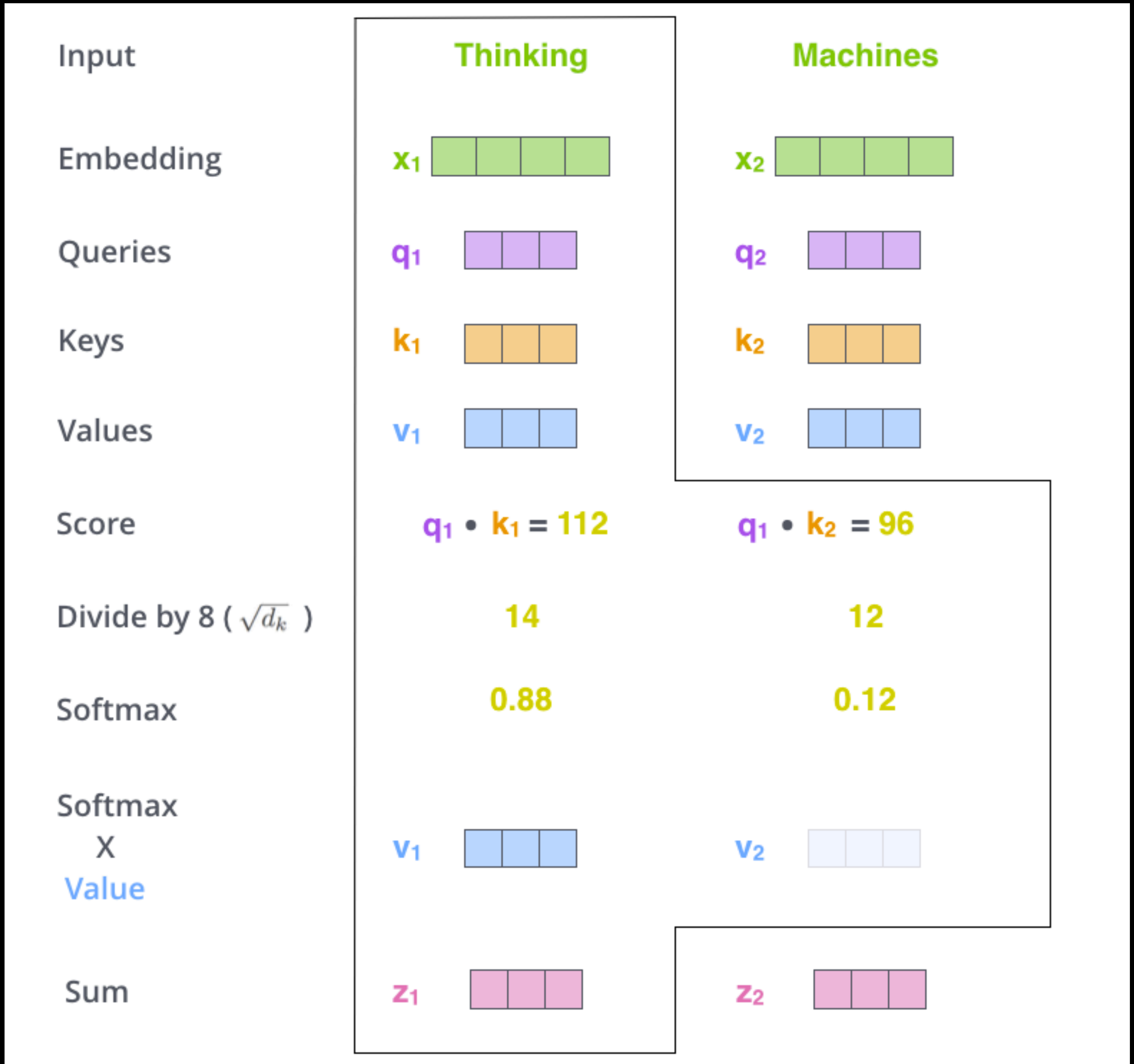
Self-attention revisited



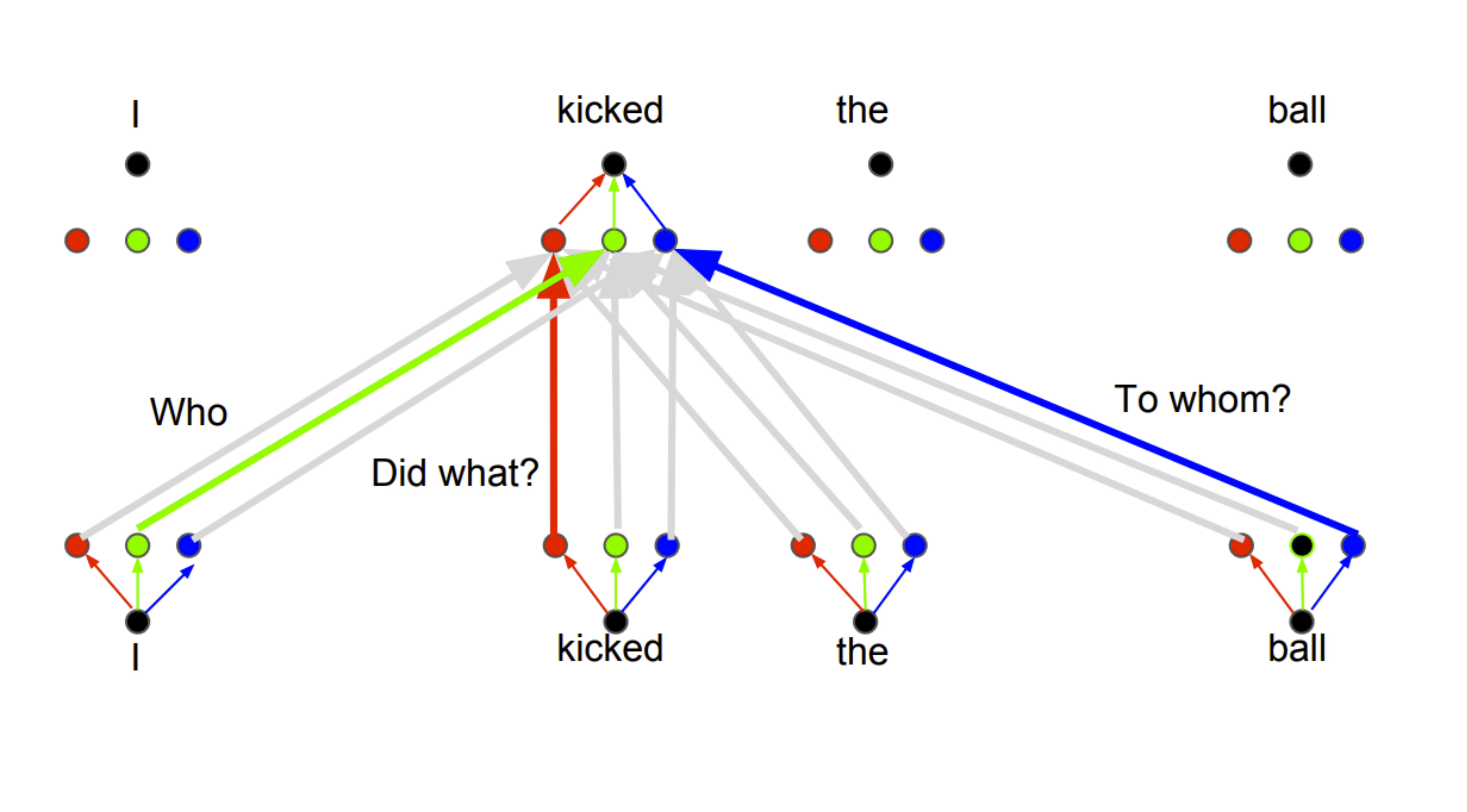
Self-attention revisited



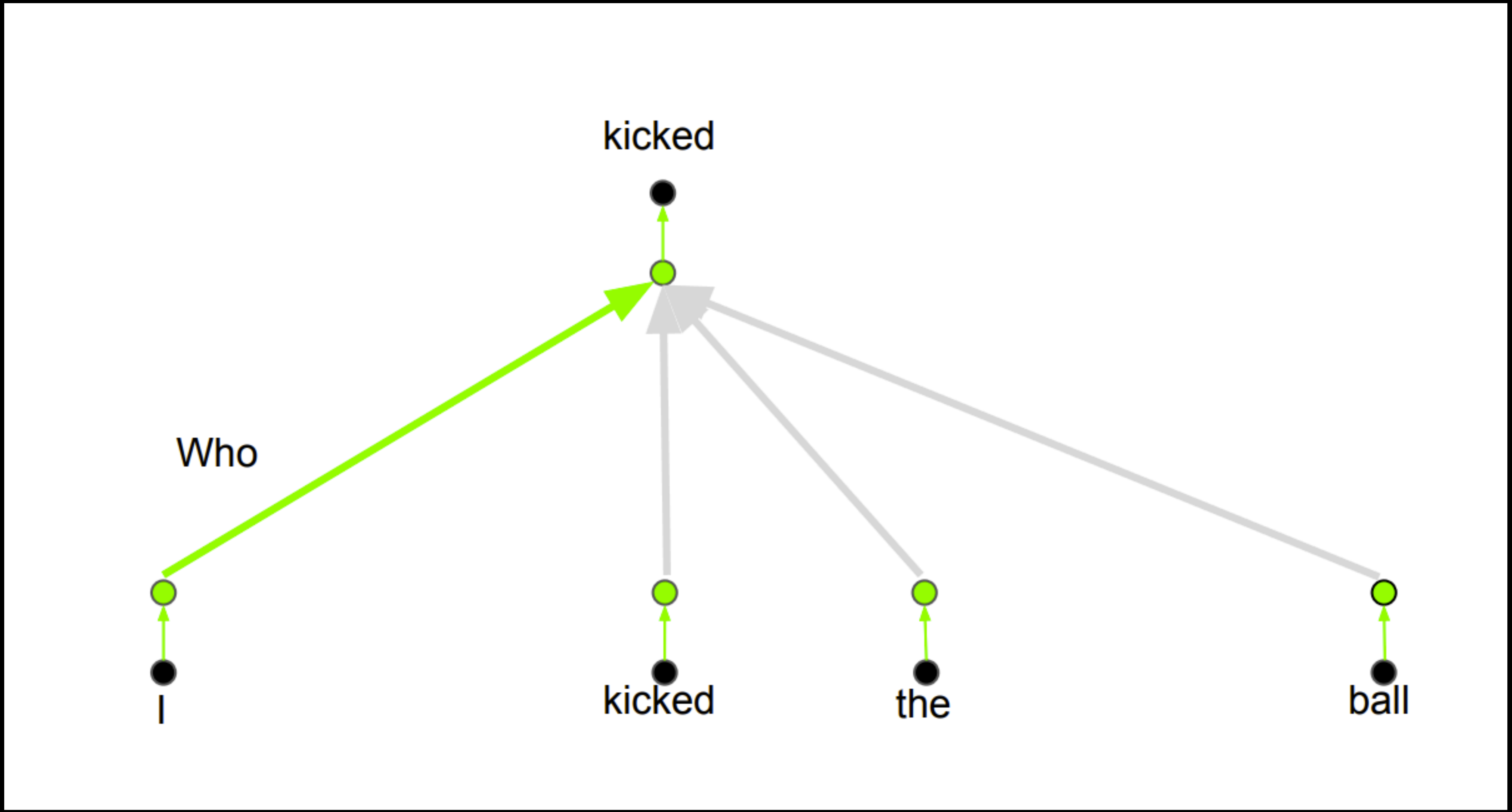
Self-attention revisited



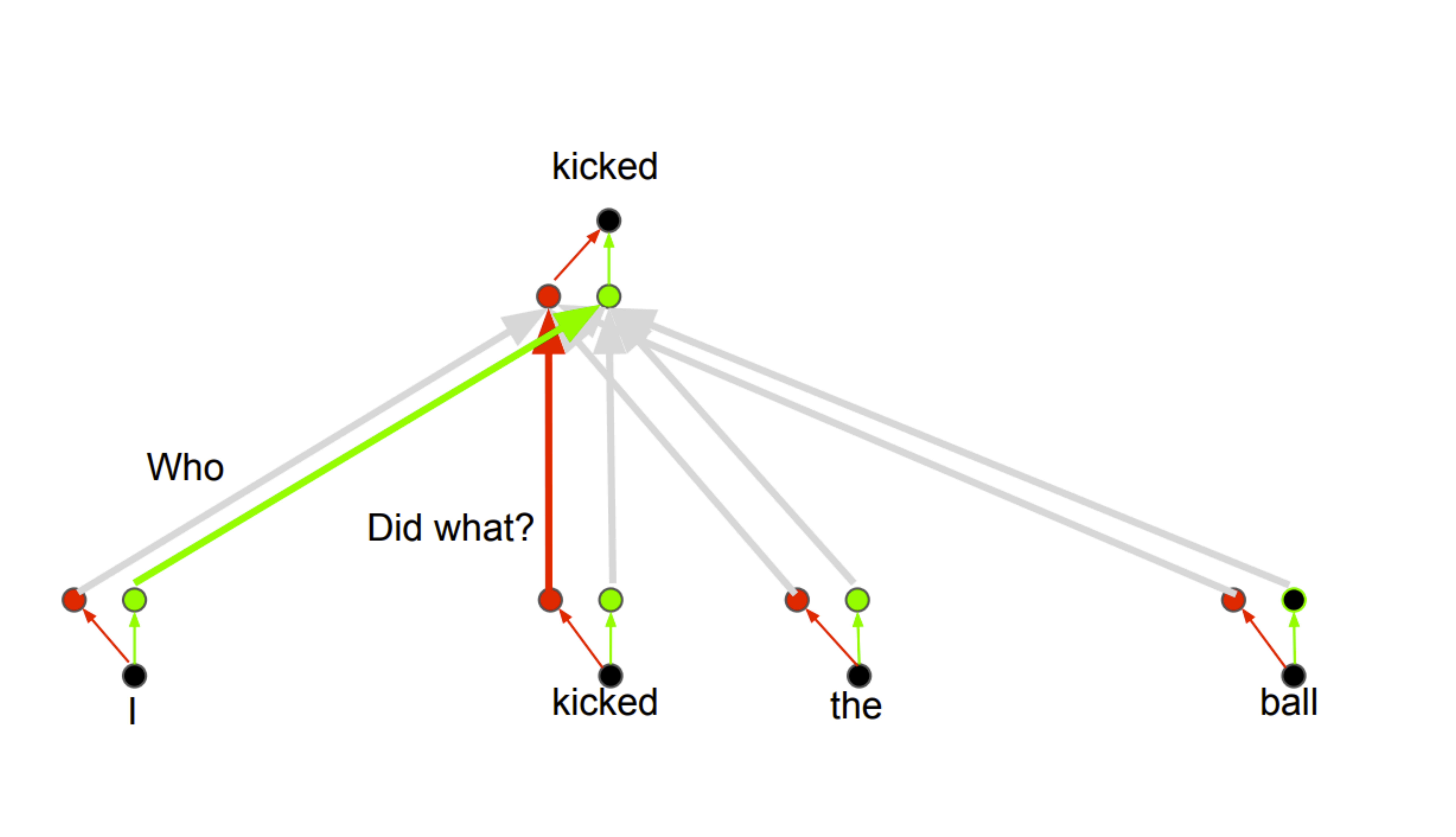
Self-attention multi-head



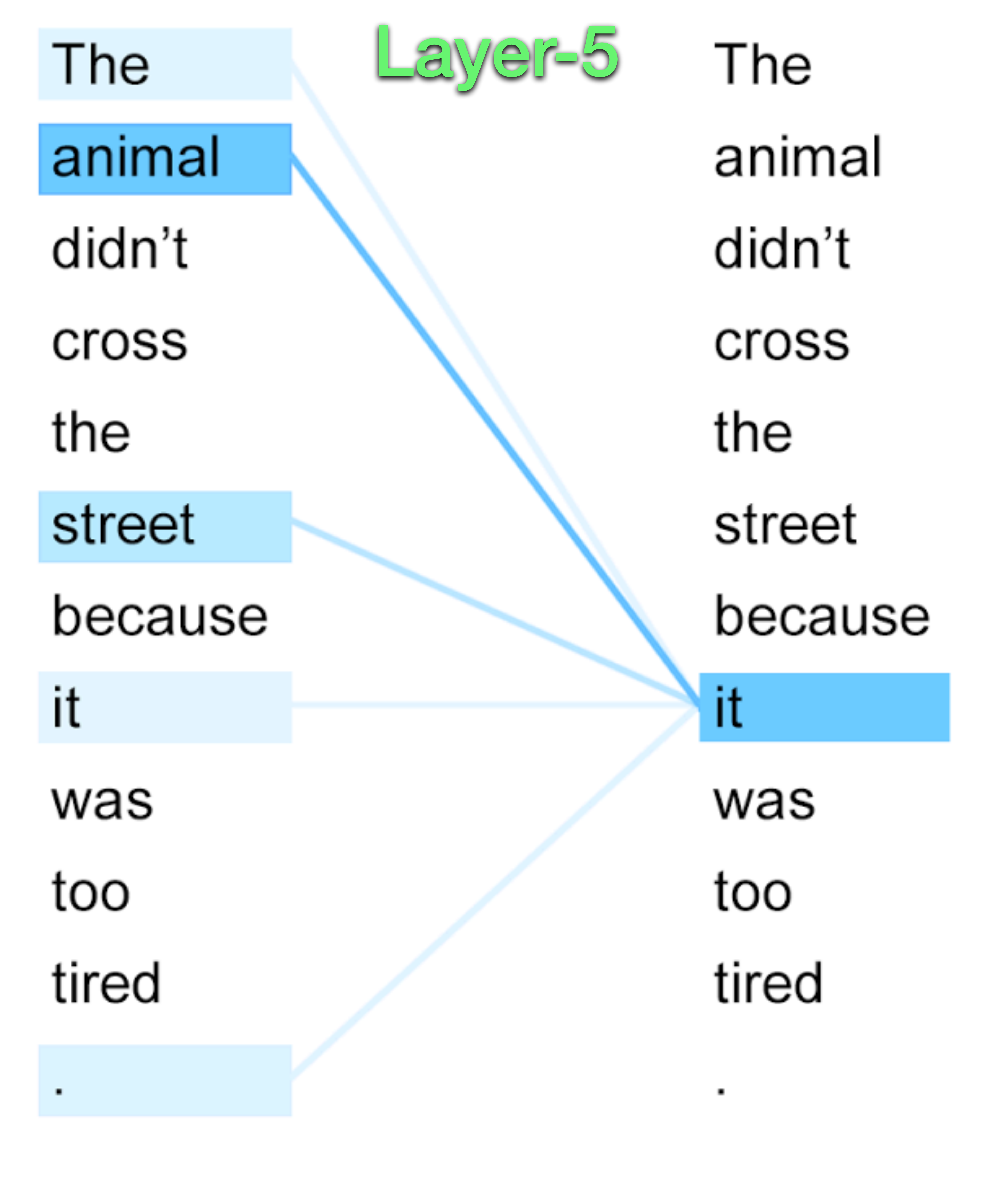
Self-attention multi-head - role of attention heads



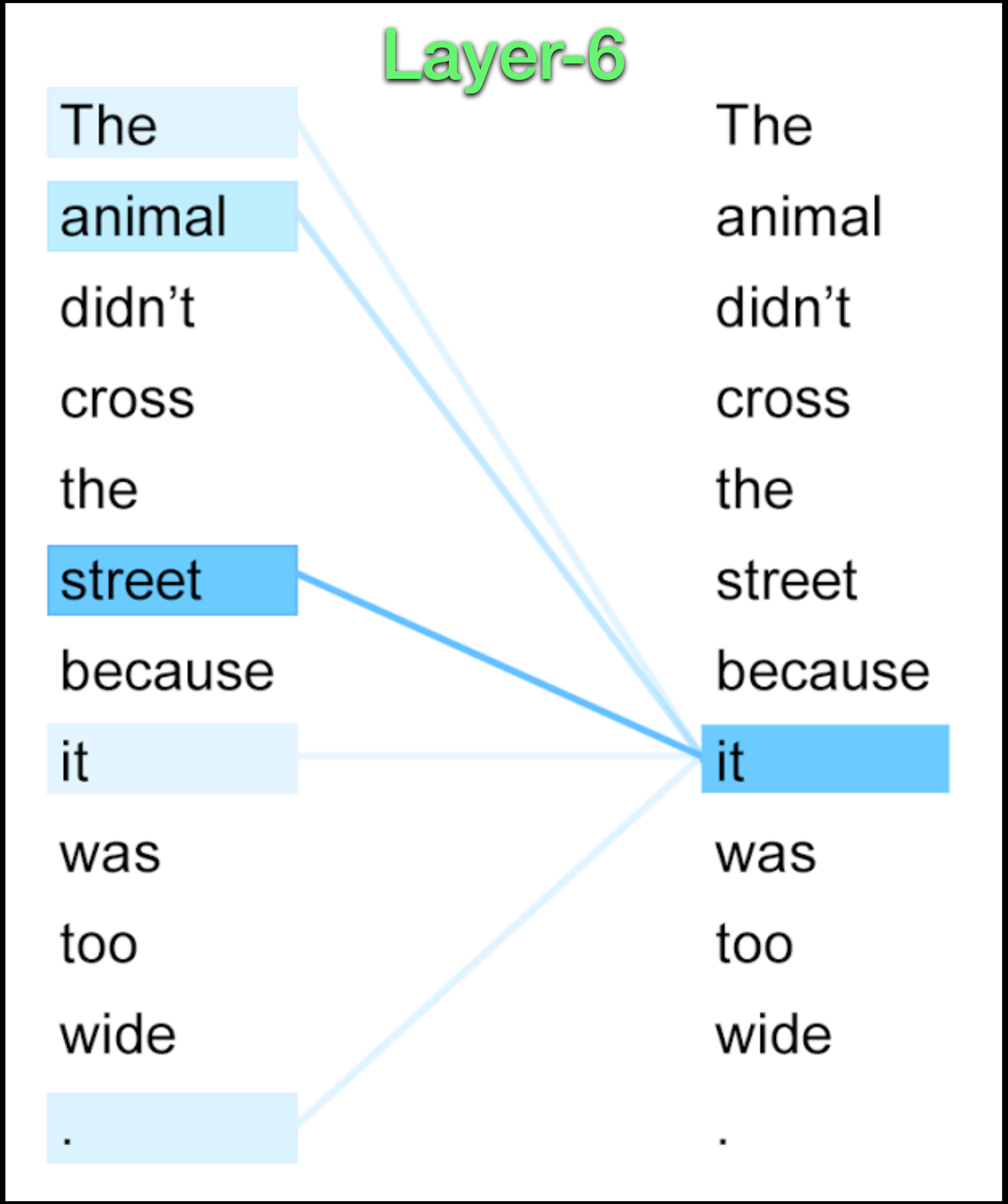
Self-attention multi-head - role of attention heads

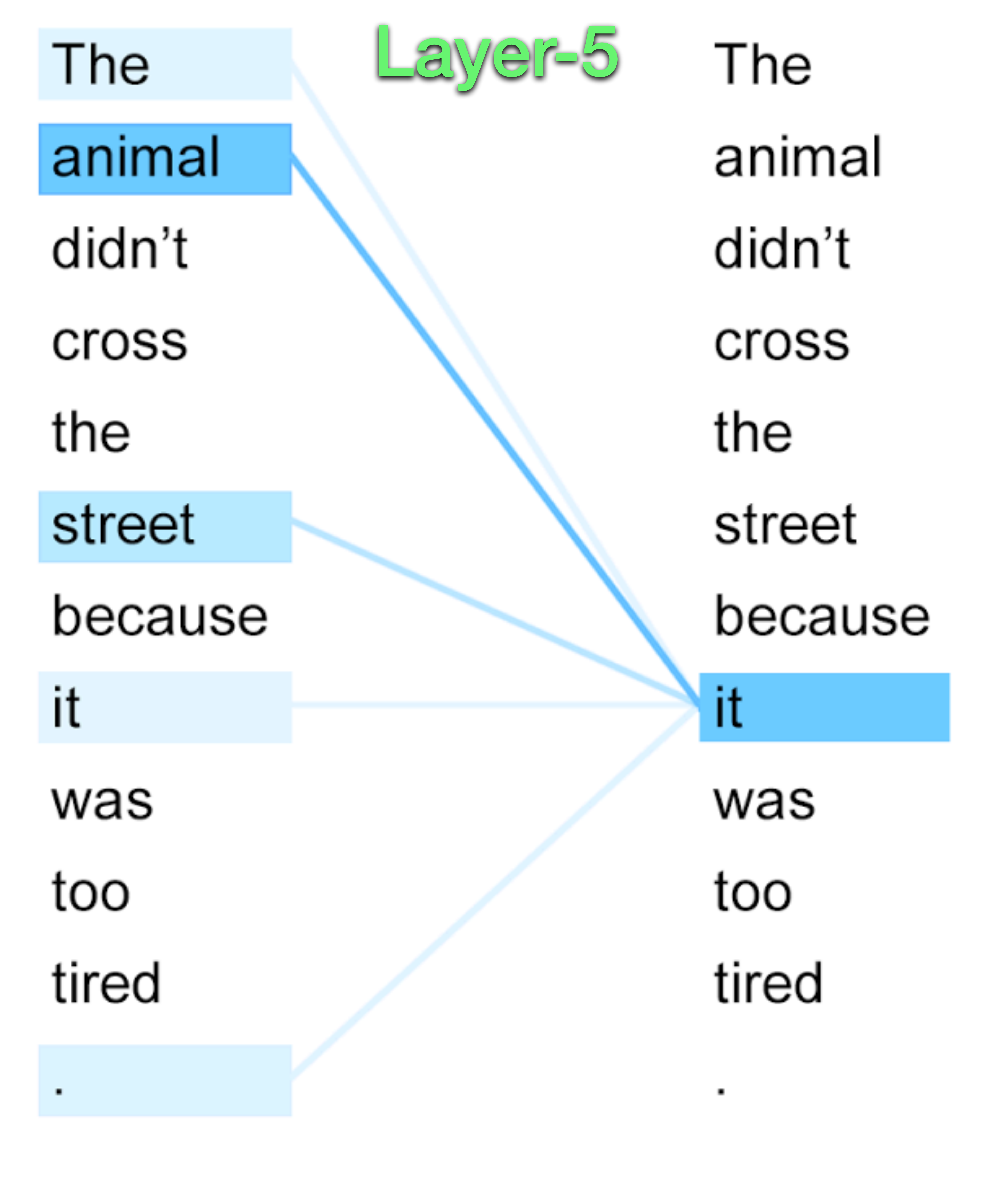


Self-attention - need for depth

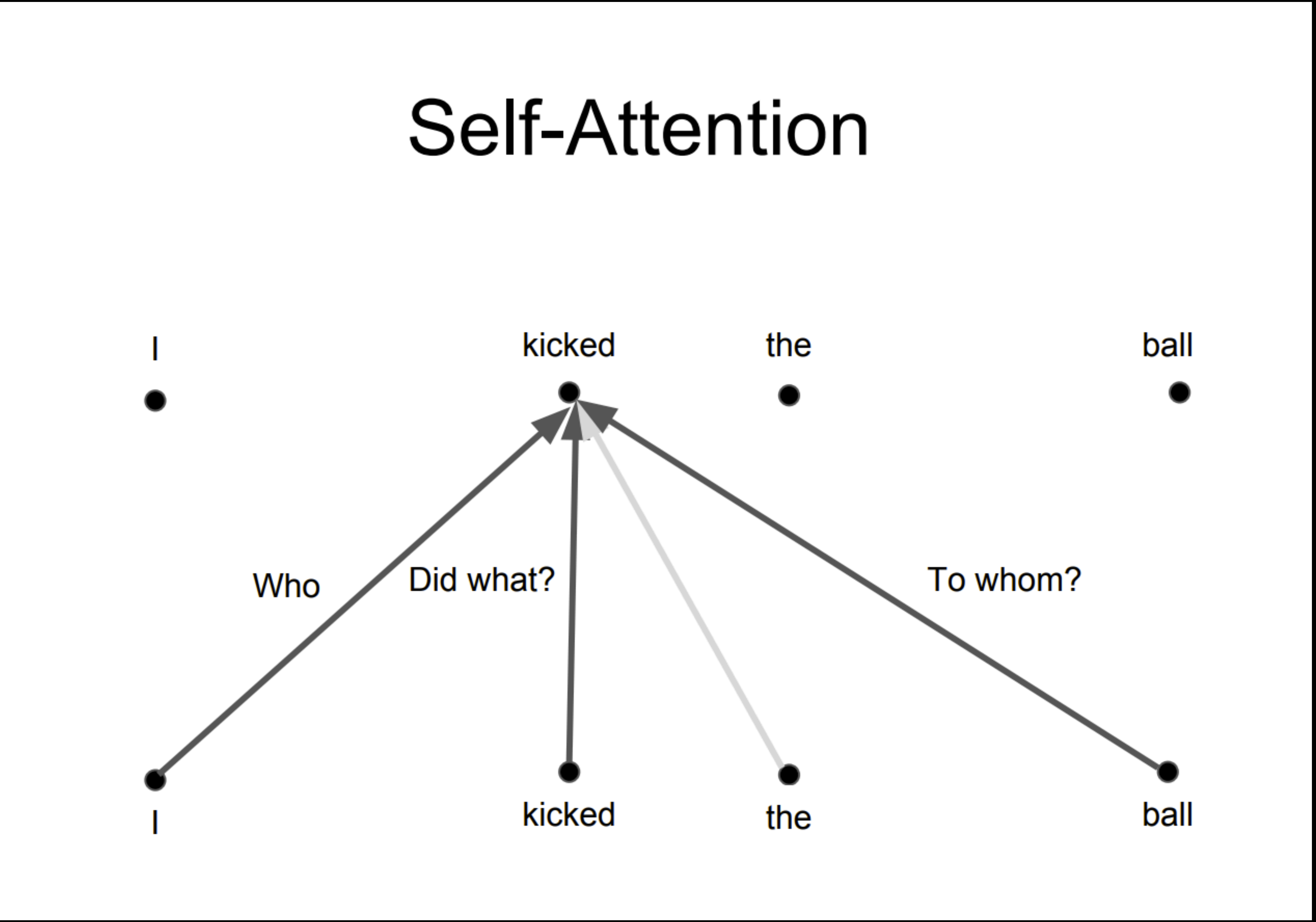


Self-attention - need for depth



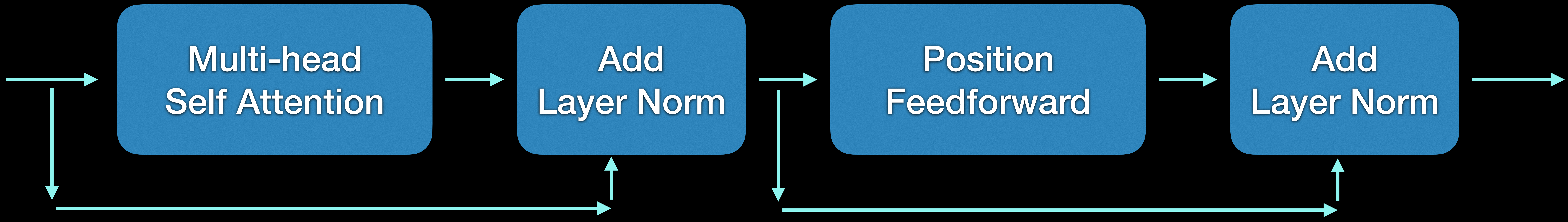


Need for multi-head attention



Single layer of encoder (typical implementation)

- Single encoder layer has typically self-attention skip connection, layer norm and feedforward layer



Positional encoding

✳ No recurrence or position awareness yet in the model

Binary format -
position can encode the rate of change of bits across time

In floating format - one can use sines and cosines

0 :	0	0	0	0	8 :	1	0	0	0
1 :	0	0	0	1	9 :	1	0	0	1
2 :	0	0	1	0	10 :	1	0	1	0
3 :	0	0	1	1	11 :	1	0	1	1
4 :	0	1	0	0	12 :	1	1	0	0
5 :	0	1	0	1	13 :	1	1	0	1
6 :	0	1	1	0	14 :	1	1	1	0
7 :	0	1	1	1	15 :	1	1	1	1

Positional encoding

- An example used in the first paper

$$\mathbf{p}(t) \in \mathcal{R}^D$$

$$p_i(t) = \begin{cases} sin(\omega_k t), & \text{if } i = 2k \\ cos(\omega_k t), & \text{if } i = 2k+1 \end{cases}$$

$$\boldsymbol{k} \in \{1 ... \frac{D}{2}\}$$

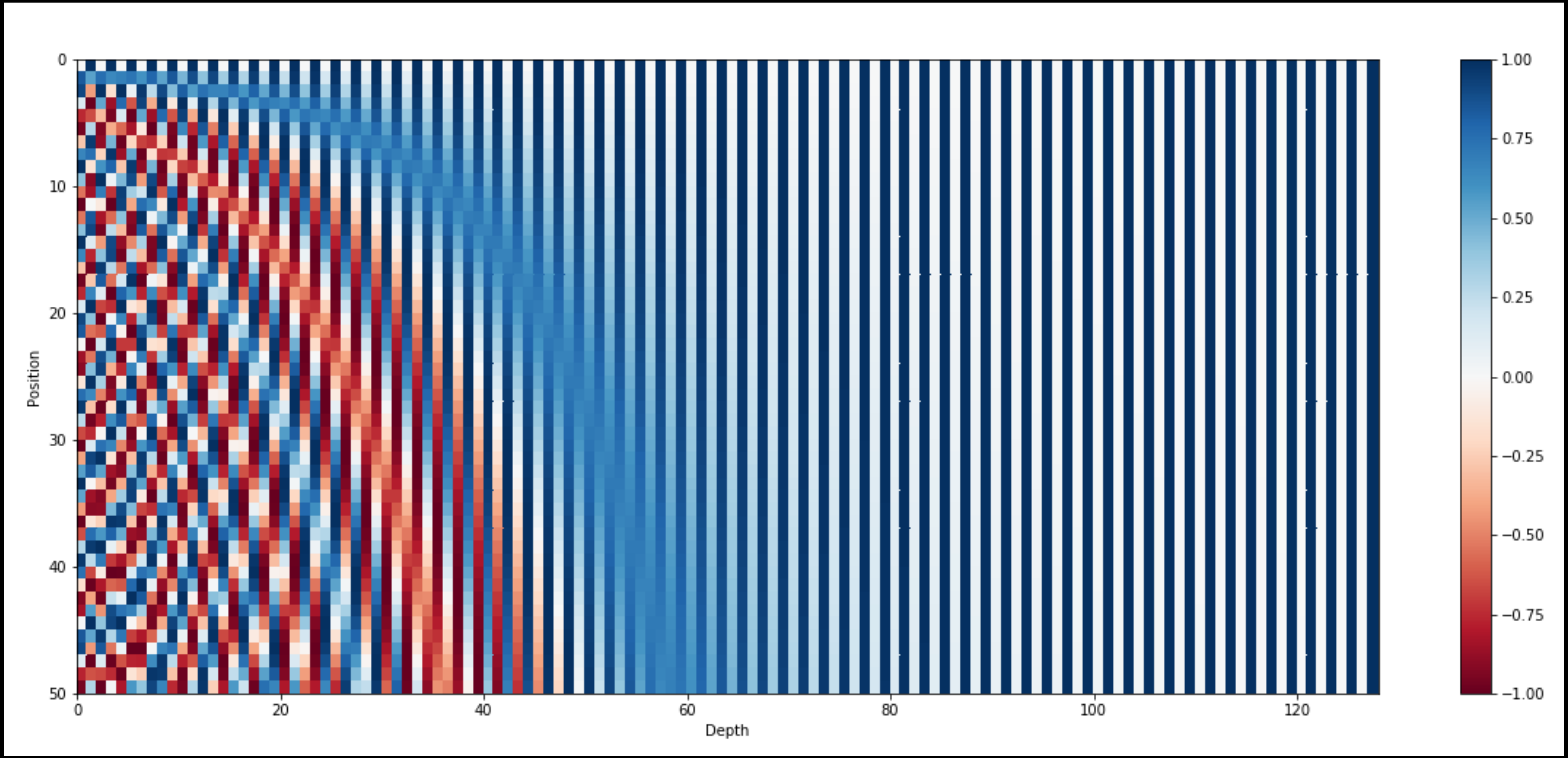
$$\omega_k = \frac{1}{10000^{\frac{2k}{D}}}$$

$$\mathbf{x}(t) = \mathbf{x}(t) + \mathbf{p}(t)$$

Positional encoding

* An example used in the first paper $\mathbf{p}(t) \in \mathcal{R}^D$ [T=50, D=128]

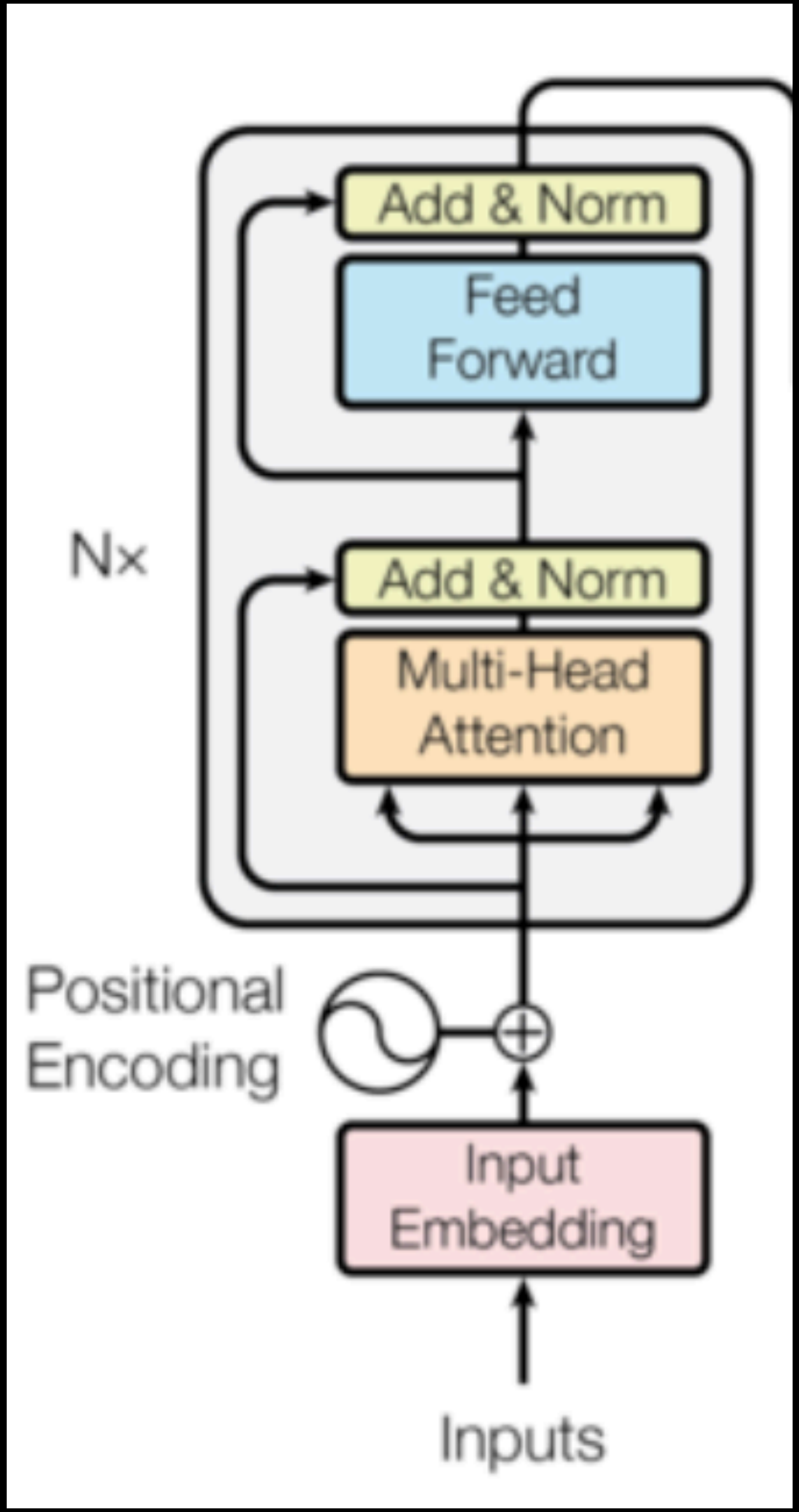
T



D

Transformer encoder - overview

Reading Assignment - “Attention is All You Need”
https://arxiv.org/pdf/1706.03762.pdf



Transformer decoder

- Masked self-attention layer -
  - Mask makes the output dependecies causal
    - Only the past is used to encode the attention.

	<s>	der	schnelle	braune	fuchs	</s>
<s>						
der						
schnelle			100	20	80	
braune				100	80	
fuchs						
</s>						

$QxK^T / \sqrt{d}$

+

0	-inf	-inf	-inf	-inf	-inf
0	0	-inf	-inf	-inf	-inf
0	0	0	-inf	-inf	-inf
0	0	0	0	-inf	-inf
0	0	0	0	0	-inf
0	0	0	0	0	0

Mask Shape = TxT = 6x6

=

	<s>	der	schnelle	braune	fuchs	</s>
<s>		-inf	-inf	-inf	-inf	-inf
der			inf	-inf	-inf	-inf
schnelle			100	-inf	-inf	-inf
braune				100	-inf	-inf
fuchs						-inf
</s>						

$QxK^T / \sqrt{d}$

Source - https://lionbridge.ai/articles/what-are-transformer-models-in-machine-learning/

Transformer decoder

- Masked self-attention layer -
  - ✓ Mask makes the output dependecies causal
    - ★ Only the past is used to encode the attention.

$$Softmax\left\{\frac{\mathbf{QK}^T}{\sqrt{d}}\right\}\mathbf{V} \longrightarrow Softmax\left\{Mask + \frac{\mathbf{QK}^T}{\sqrt{d}}\right\}\mathbf{V}$$

    - ★ Make the attention matrix to be lower triangular

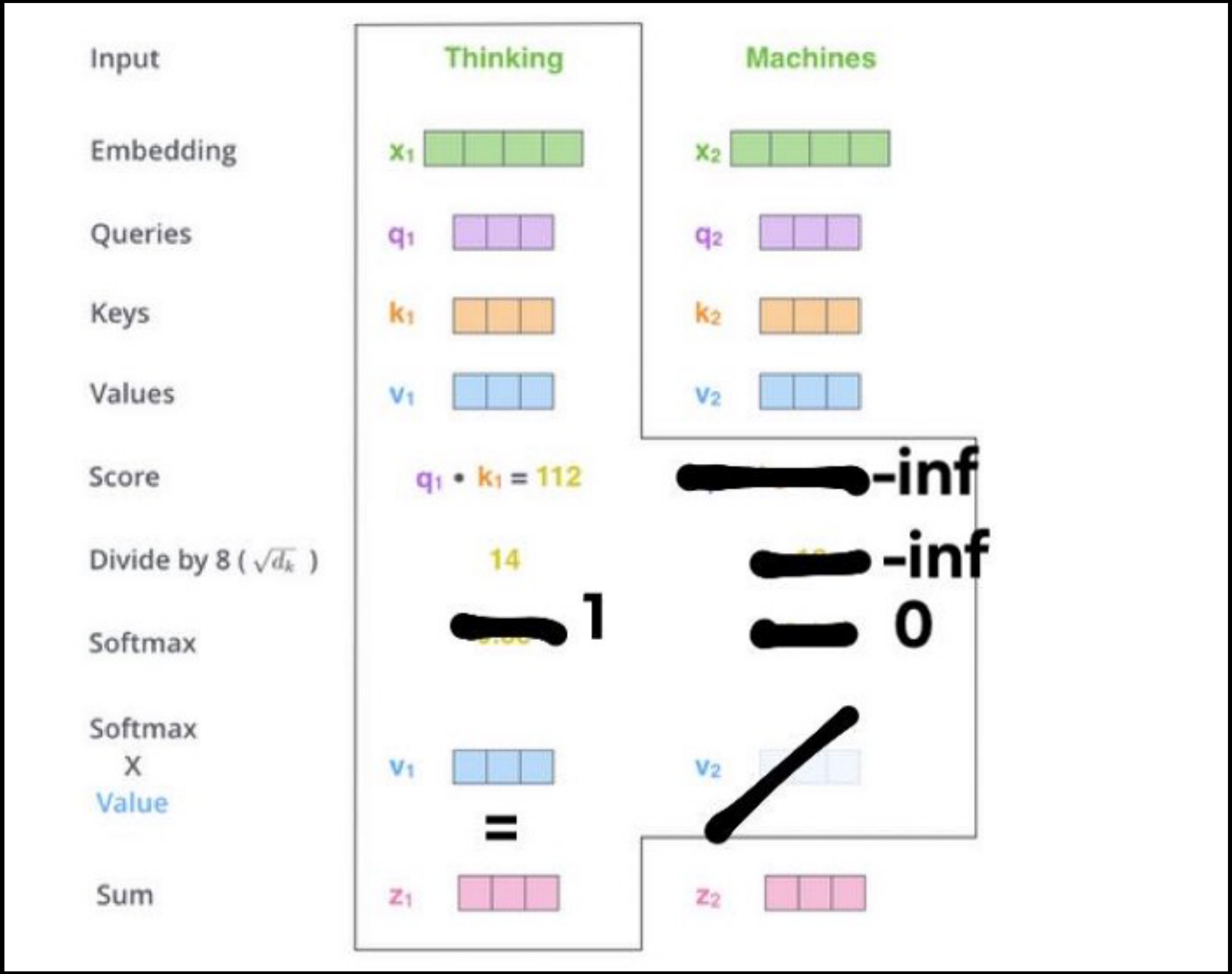
Transformer decoder

- Masked self-attention layer -
  - ✓ Mask makes the output dependecies causal
    - ★ Only the past is used to encode the attention.

	<s>	der	schnelle	braune	fuchs	</s>
<s>		0	0	0	0	0
der			0	0	0	0
schnelle			.75	0	0	0
braune				.85	0	0
fuchs						0
</s>						

$$\textbf{softmax}(Q \times K^T / \sqrt{d})$$

Transformer decoder



Encoder-decoder attention

✳ Use the Key and Value matrices from the last layer of the encoder

$$\mathbf{Q}_h^p = \overline{\mathbf{D}}^{p-1}\mathbf{W}_h^{p,Q} + \mathbf{1}(\mathbf{b}_h^{p,Q})^T \in \mathcal{R}^{S\times d}$$

$$\mathbf{K}_h^p = \mathbf{E}^L\mathbf{W}_h^{p,K} + \mathbf{1}(\mathbf{b}_h^{p,K})^T \in \mathcal{R}^{T\times d}$$

$$\mathbf{V}_h^p = \mathbf{E}^L\mathbf{W}_h^{p,V} + \mathbf{1}(\mathbf{b}_h^{p,V})^T \in \mathcal{R}^{T\times d}$$

$$\mathbf{D}_h^p = softmax\left(\frac{\mathbf{Q}_h^p(\mathbf{K}_h^p)^T}{\sqrt{d}}\right)\mathbf{V}_h^p \in \mathcal{R}^{S\times d}$$

$h = \{1..H\}$ heads $\quad d = \dfrac{D}{H}$

Transformer - decoder

- Decoder Layer Output

- $$[\mathbf{d}^p(1)...\mathbf{d}^p(S)] = ReLU\left(\mathbf{D}^p_{ff}\mathbf{W}^p_{of} + \mathbf{1}(\mathbf{b}^p_{of})^T\right) \in \mathcal{R}^{S\times D}$$

Transformer - full pipeline