E9: 309 Advanced Deep Learning
28-10-2020

Instructor: Sriram Ganapathy
sriramg@iisc.ac.in

ching 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 word abstract of the work. If modifications are needed, we will review and let you know in 2-3 days.

Housekeeping

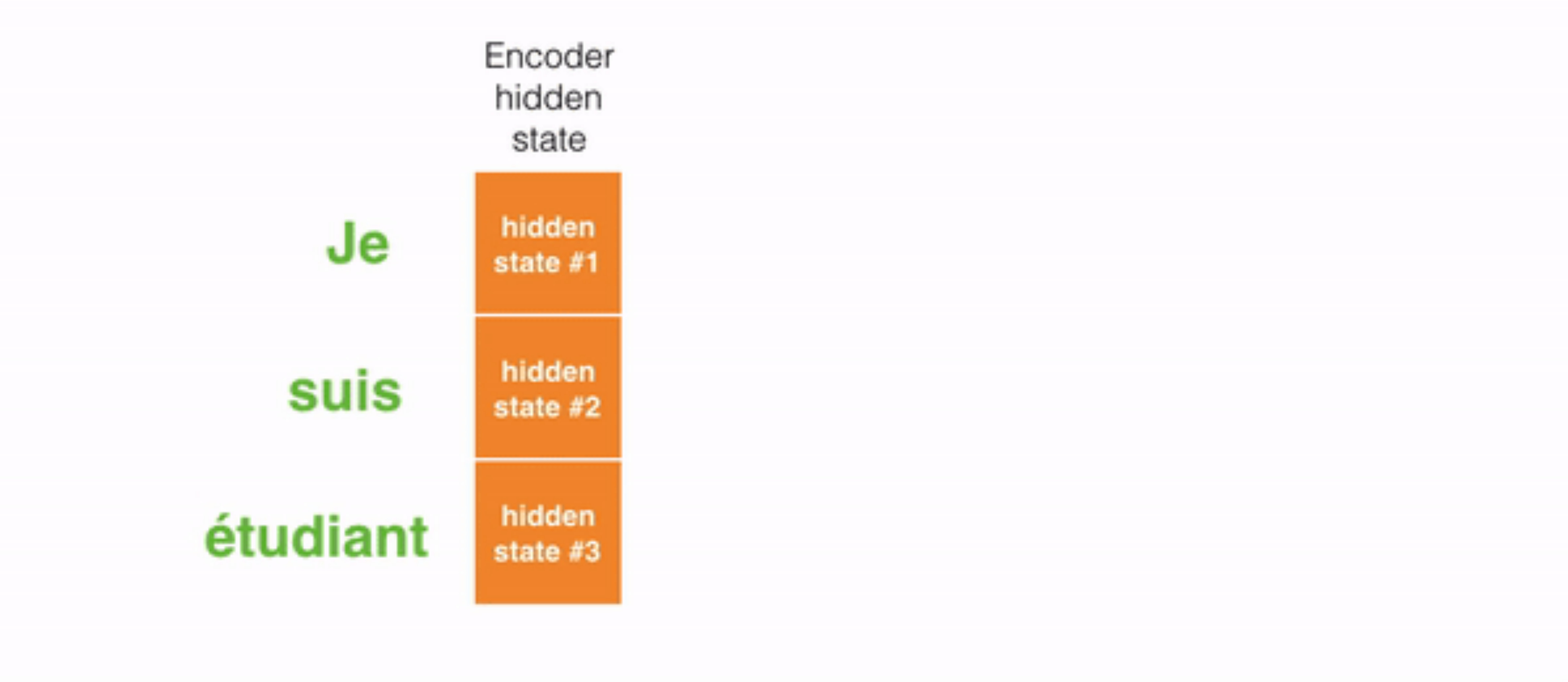
- 1st mini-project
  - ✓ Deadlines
    - ★ Report and presentation slides (Nov 19th, 10 AM).
      - ★ 1-page pdf with second page only for references and tools used (Template will be provided).
    - ★ Report - Indicate prior work, technical details and your contribution. Strictly adhere to the guidelines 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.
    - ✓ Issues with single encoder embedding for all outputs
- ➡️ Introduction to attention
    - ✓ Attention network and attention weights
- ➡️ Visualizing attention weights.
- ➡️ Self-attention and multi-head attention.,

Visualizing attention



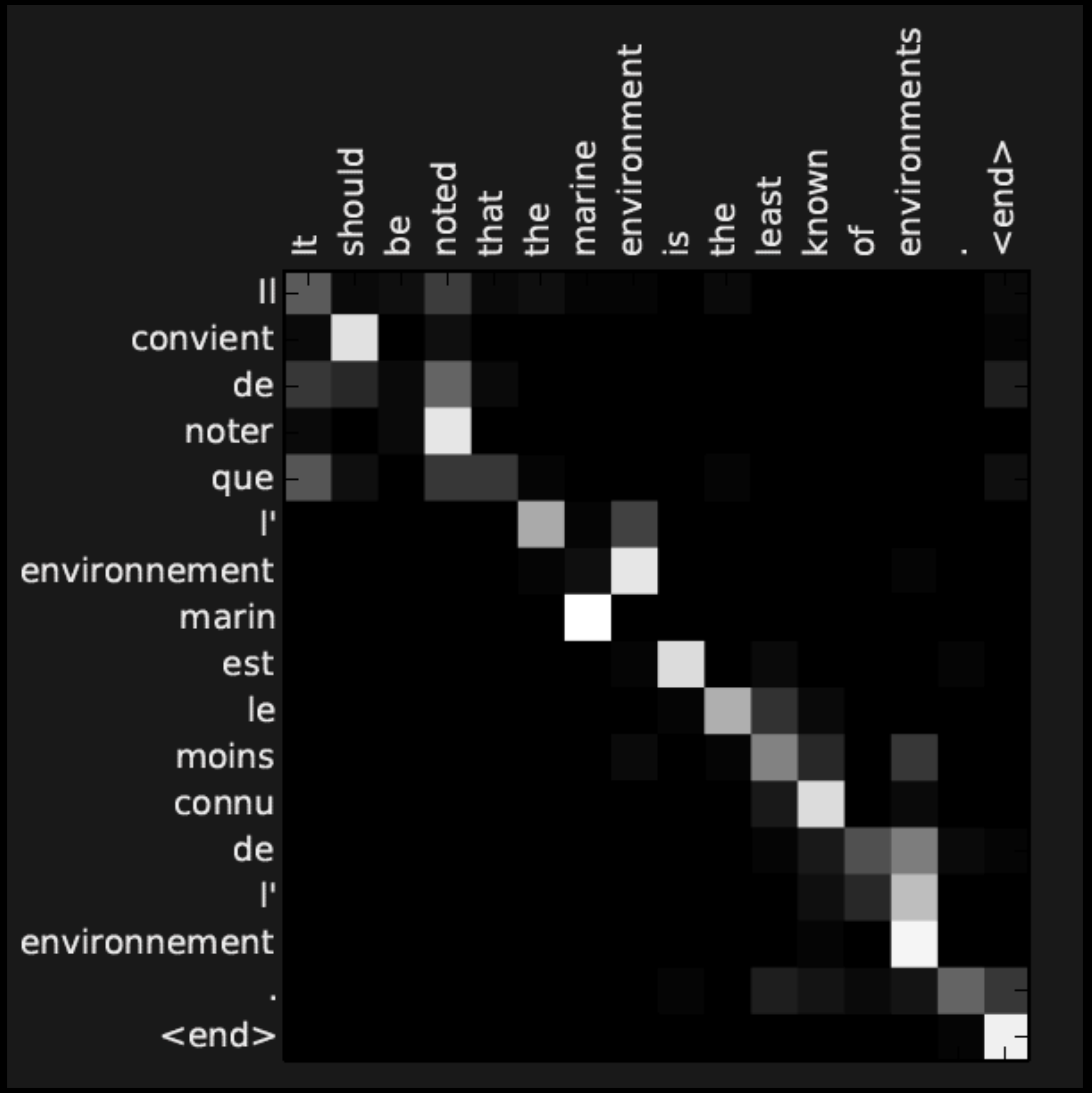
Analysis of attention networks

- Attention weights $\alpha(s, t)$

  ✓ Probability of linking (attending) to input at t for generating output at s

  ✓ Useful in analyzing the internal structure of the encoder-decoder model

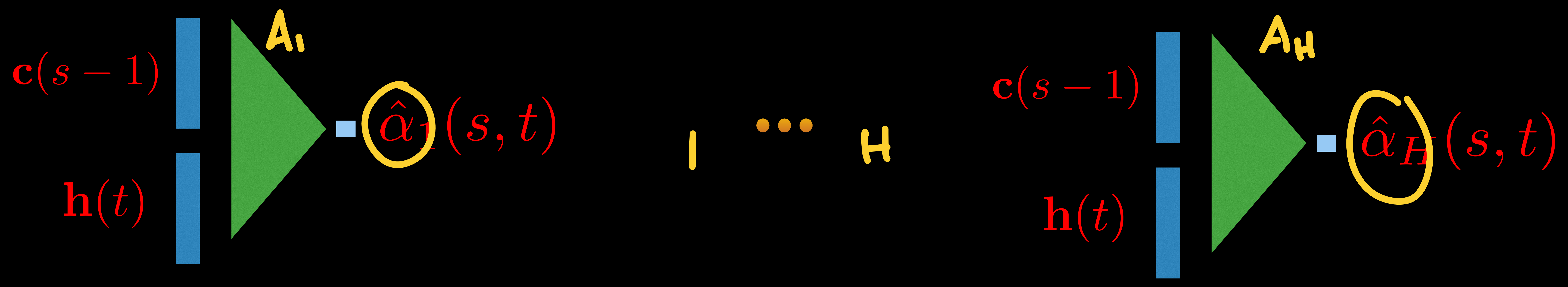
Visualizing the attention weights



Reading Assignment - "Neural Machine Translation by Jointly Learning to Align and Translate"
https://arxiv.org/pdf/1409.0473.pdf

Multi-head attention

- Having more than one attention heads



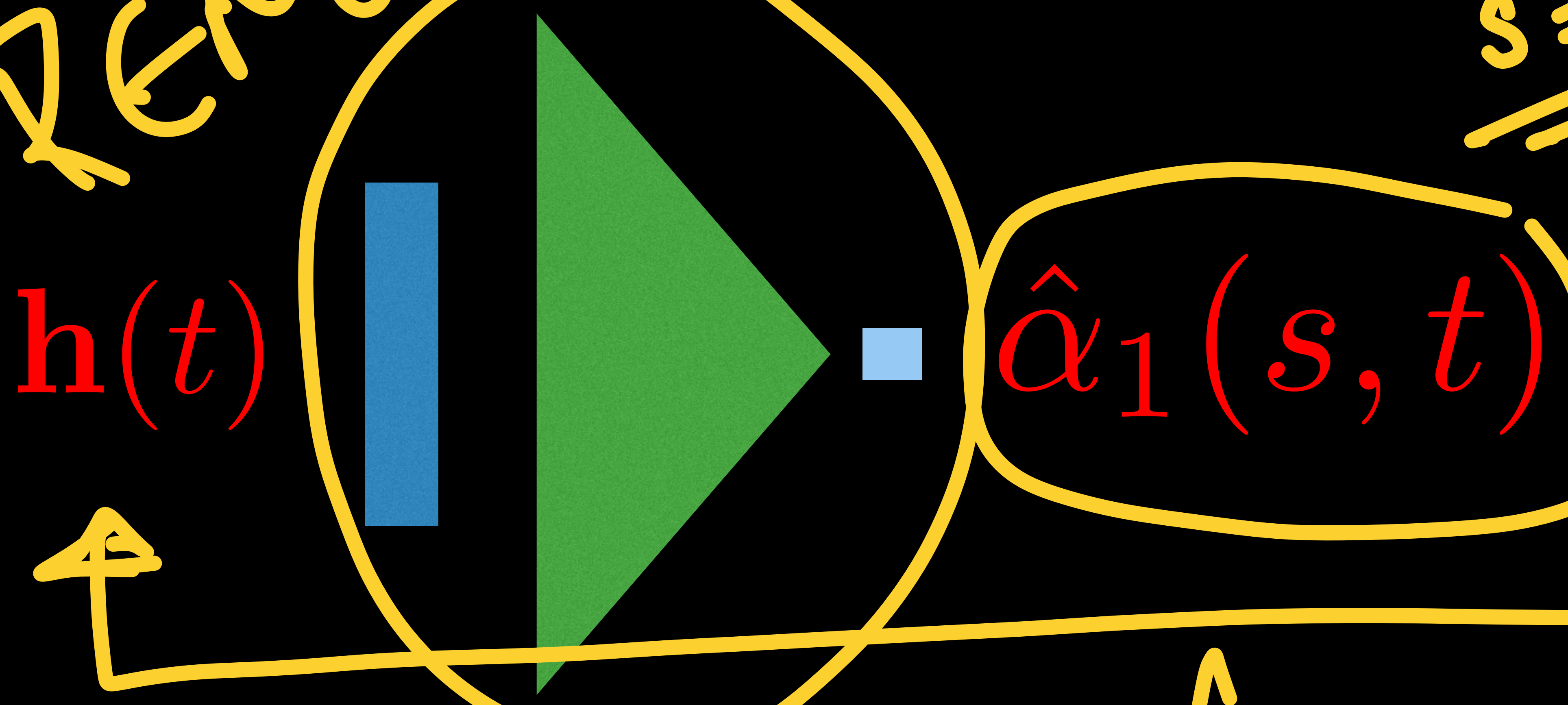
$$\hat{\alpha}_1(s,t) = \mathbf{A}_1[\mathbf{c}(s-1); \mathbf{h}(t)] \qquad \cdots \qquad \hat{\alpha}_H(s,t) = \mathbf{A}_H[\mathbf{c}(s-1); \mathbf{h}(t)]$$

$$\mathbf{e}_1(s) = \sum_t \alpha_1(s,t)\mathbf{h}(t) \qquad \cdots \qquad \mathbf{e}_H(s) = \sum_t \alpha_H(s,t)\mathbf{h}(t)$$

$$\mathbf{e}(s) = [\mathbf{e}_1^T(s); \ldots; \mathbf{e}_H^T(s)]^T$$

Self-attention

- Using attention layers without feedback from decoder.



- Without feedback the attention performs,
  - temporal relevance weighting of the input time-series (hidden layer representations).

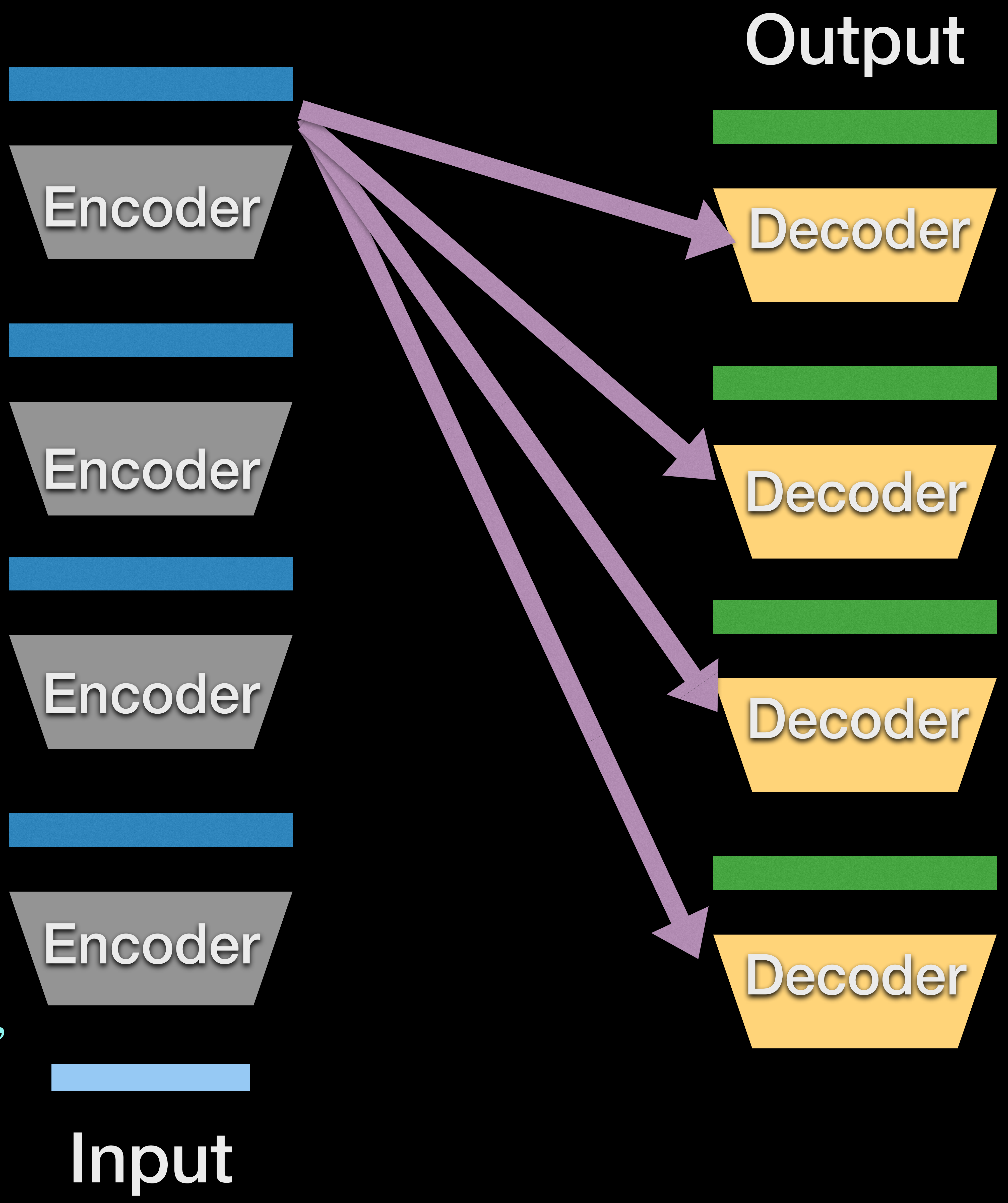
Issues in RNNs/LSTMs

- Issues of long-term dependency
  - LSTMs have partial solutions
- Back propagation through time
  - Does not allow parallelism in forward pass or backward pass.
  - Significant increase in training time as well as in forward propagation.
- Question - can we use attention mechanism itself to build temporal dependencies without recurrence.

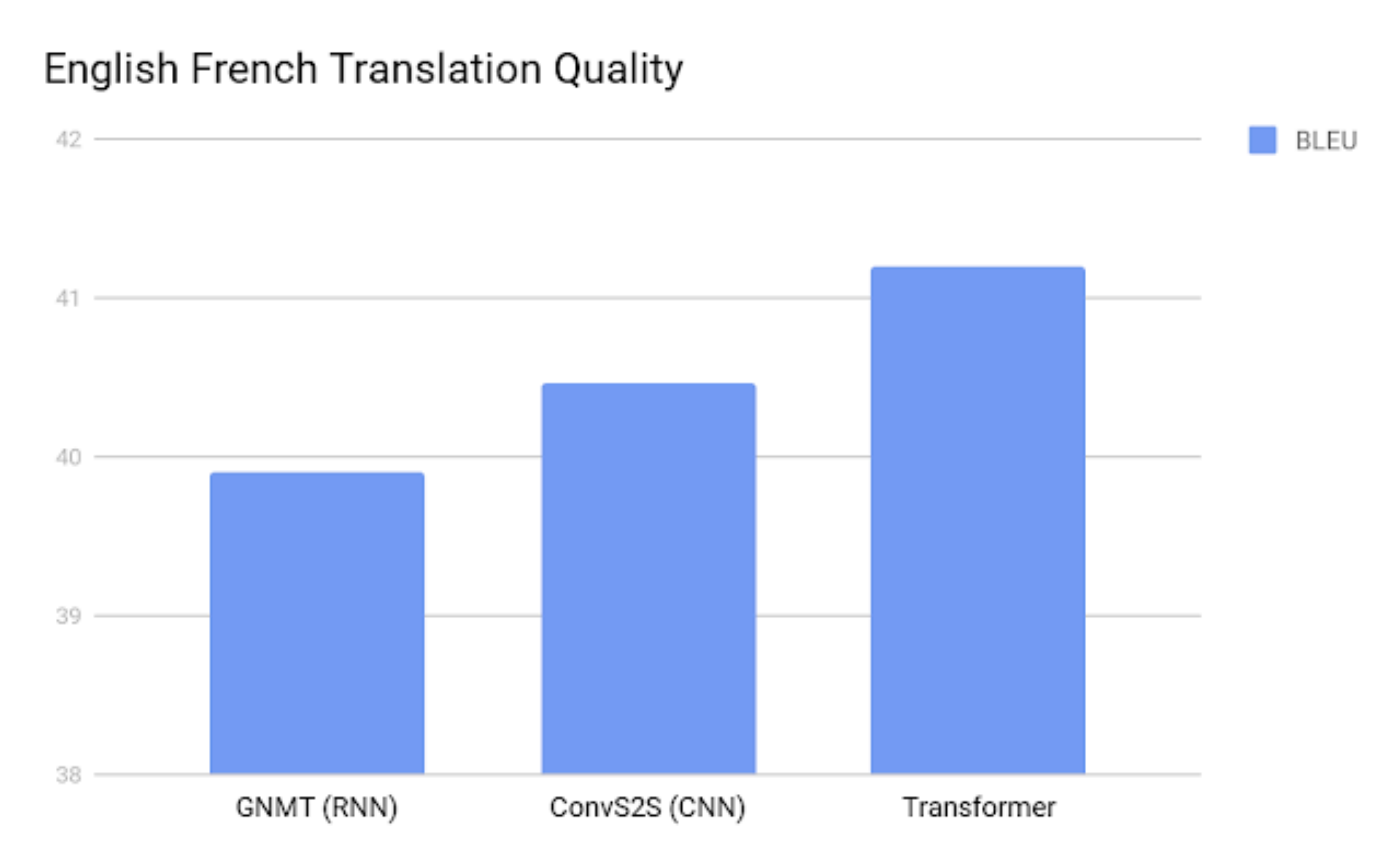
Transformers

- 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



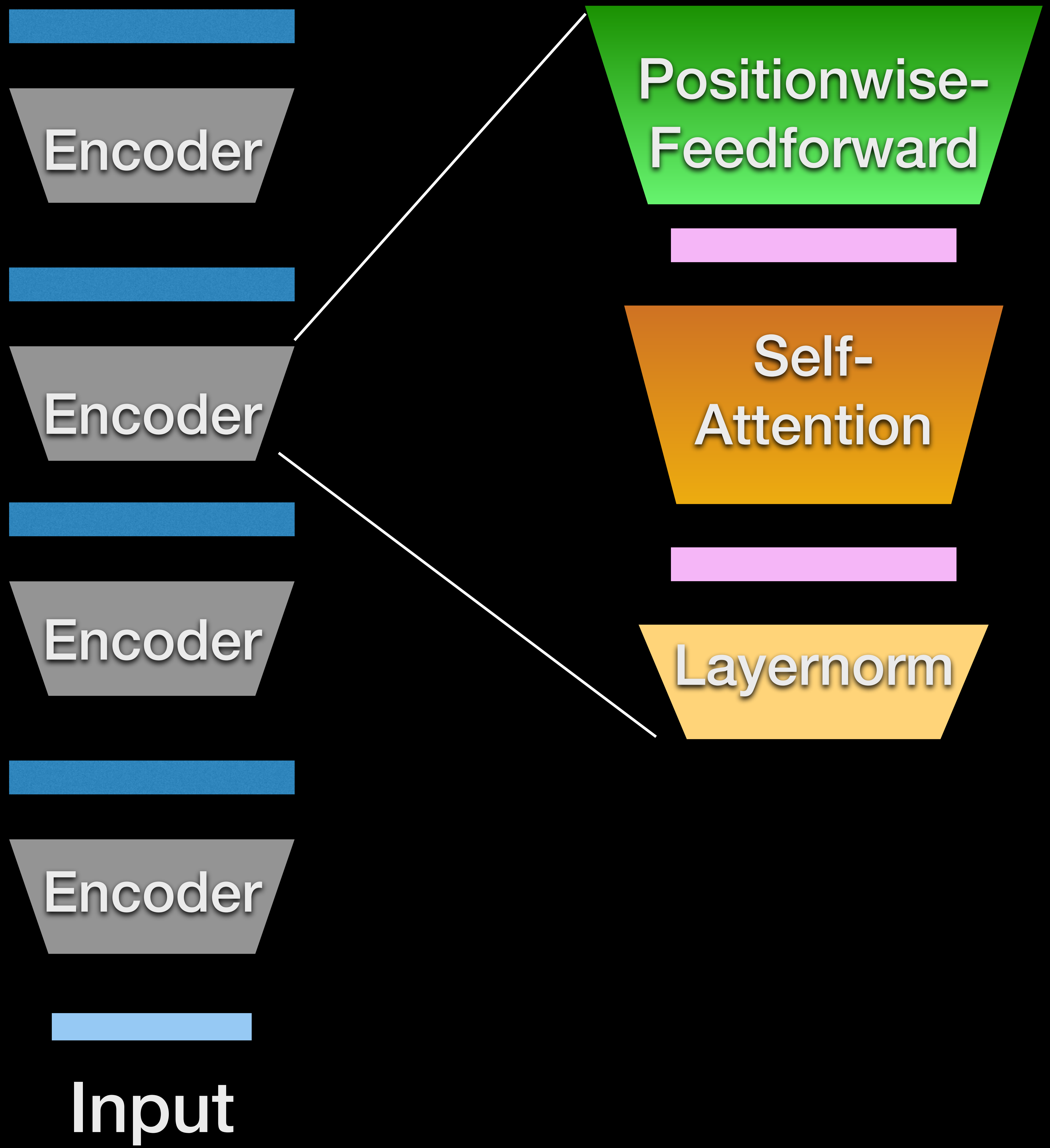
https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

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

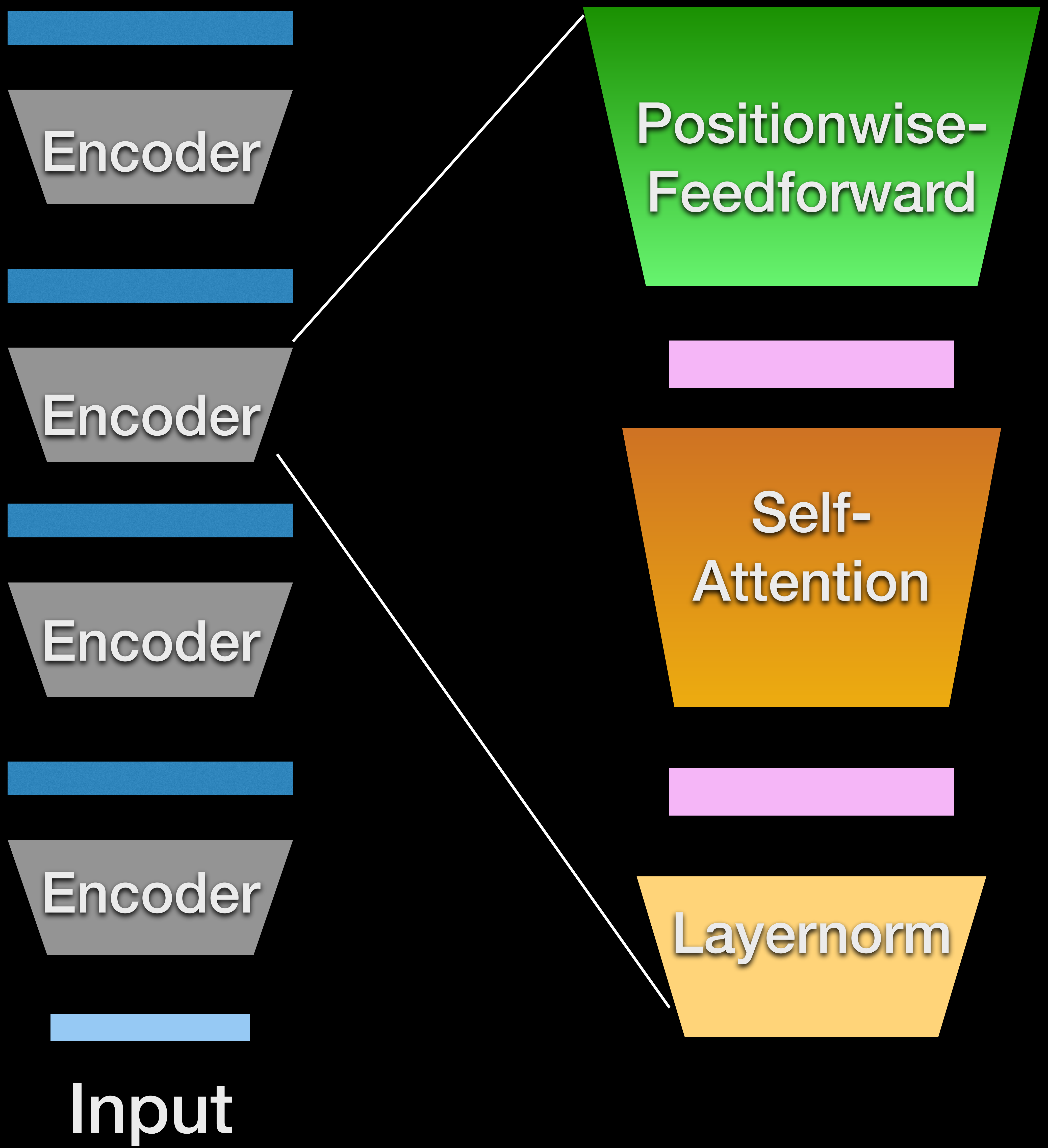
- Position wise feedforward layer

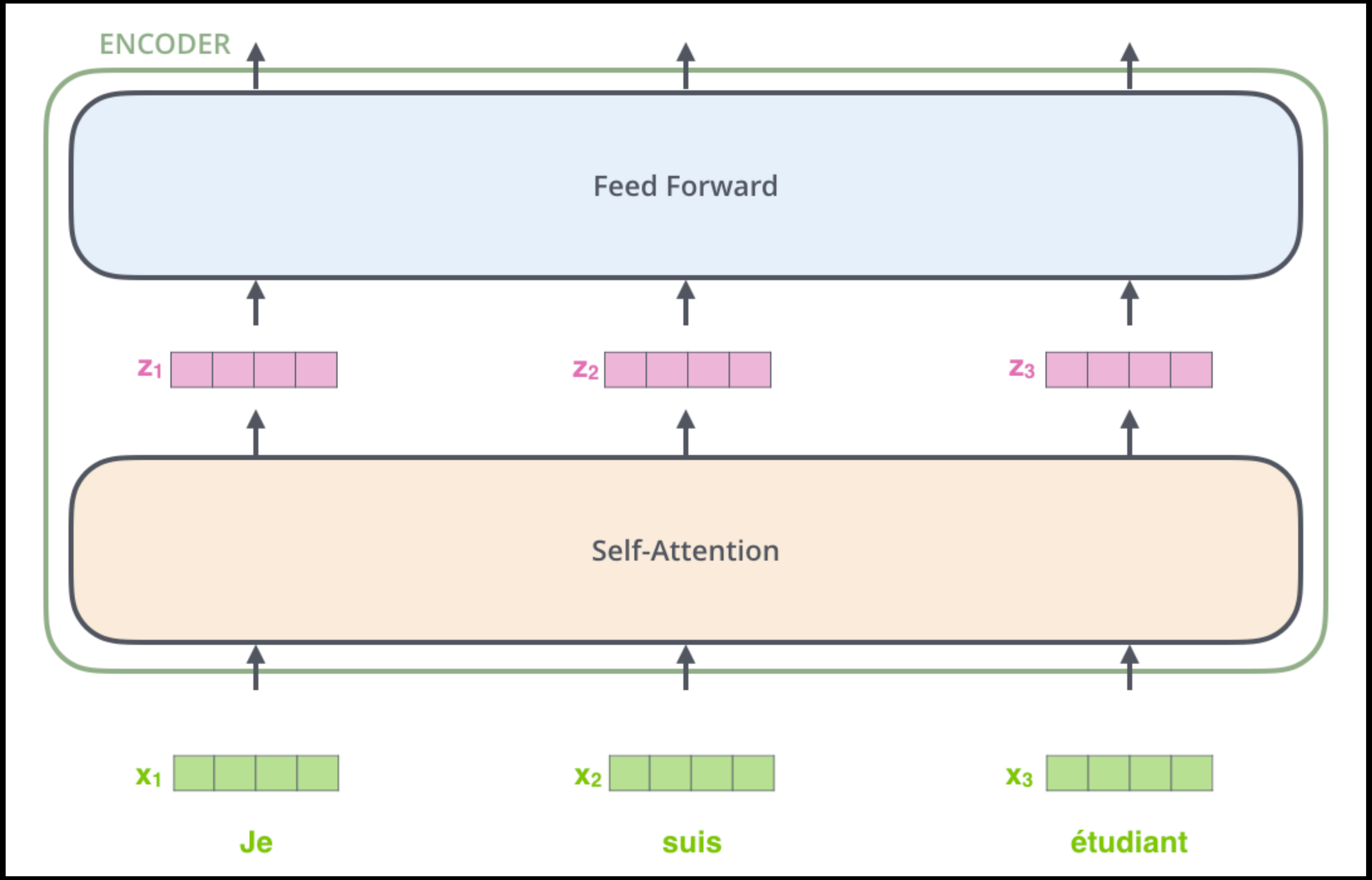
$$\mathbf{E}^l_{ff} = ReLU(\mathbf{C}^l \mathbf{W}^l_{ff} + \mathbf{1}\mathbf{b}^T_{ff}) \in \mathcal{R}^{T \times d_{ff}}$$

- Encoder layer output

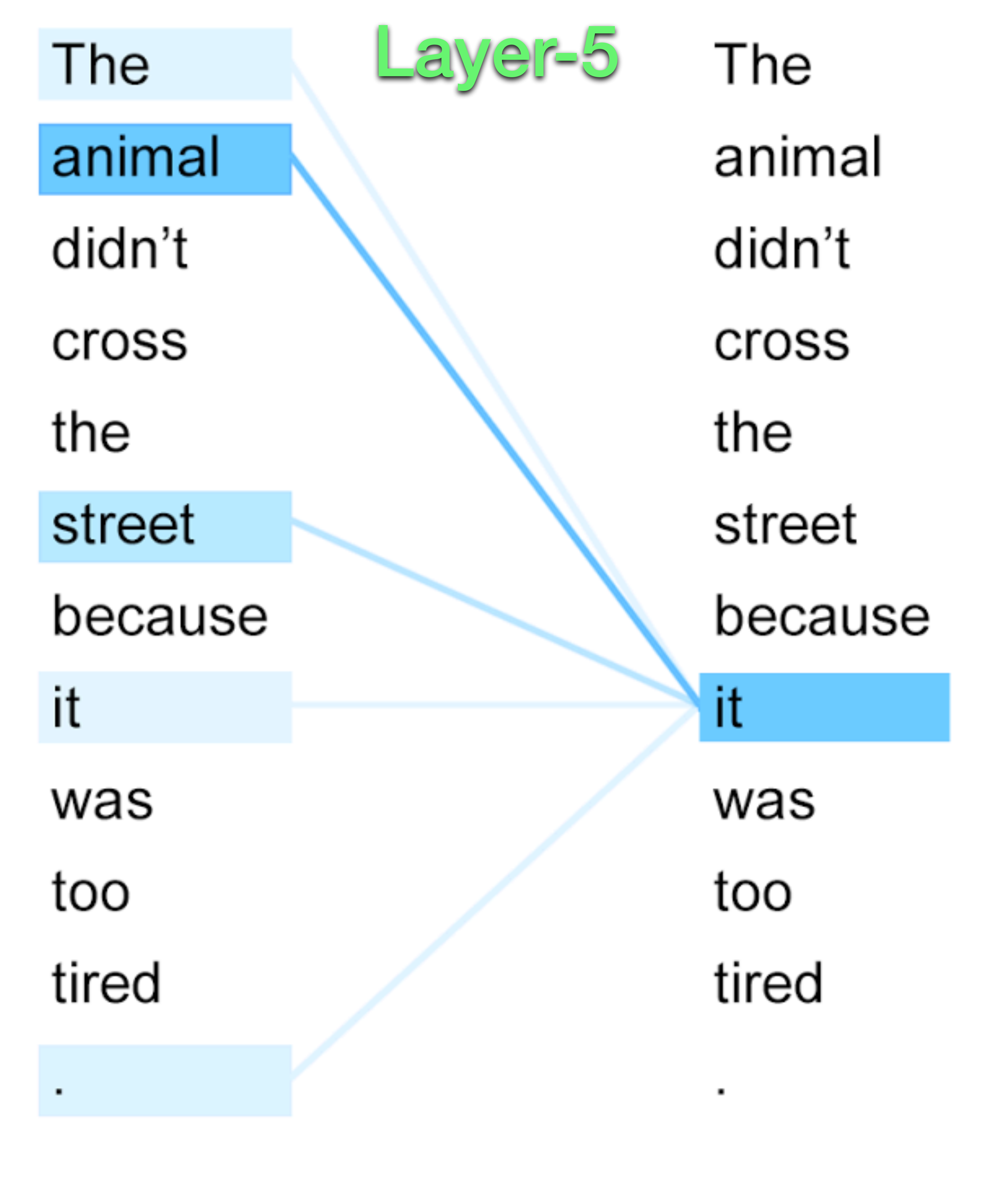
$$[\mathbf{e}^l(1) ... \mathbf{e}^l(T)] = \mathbf{E}^l_{ff} \mathbf{W}^l_{of} + \mathbf{1}(\mathbf{b}^l_{of})^T \in \mathcal{R}^{T \times D}$$

Transformers - encoder

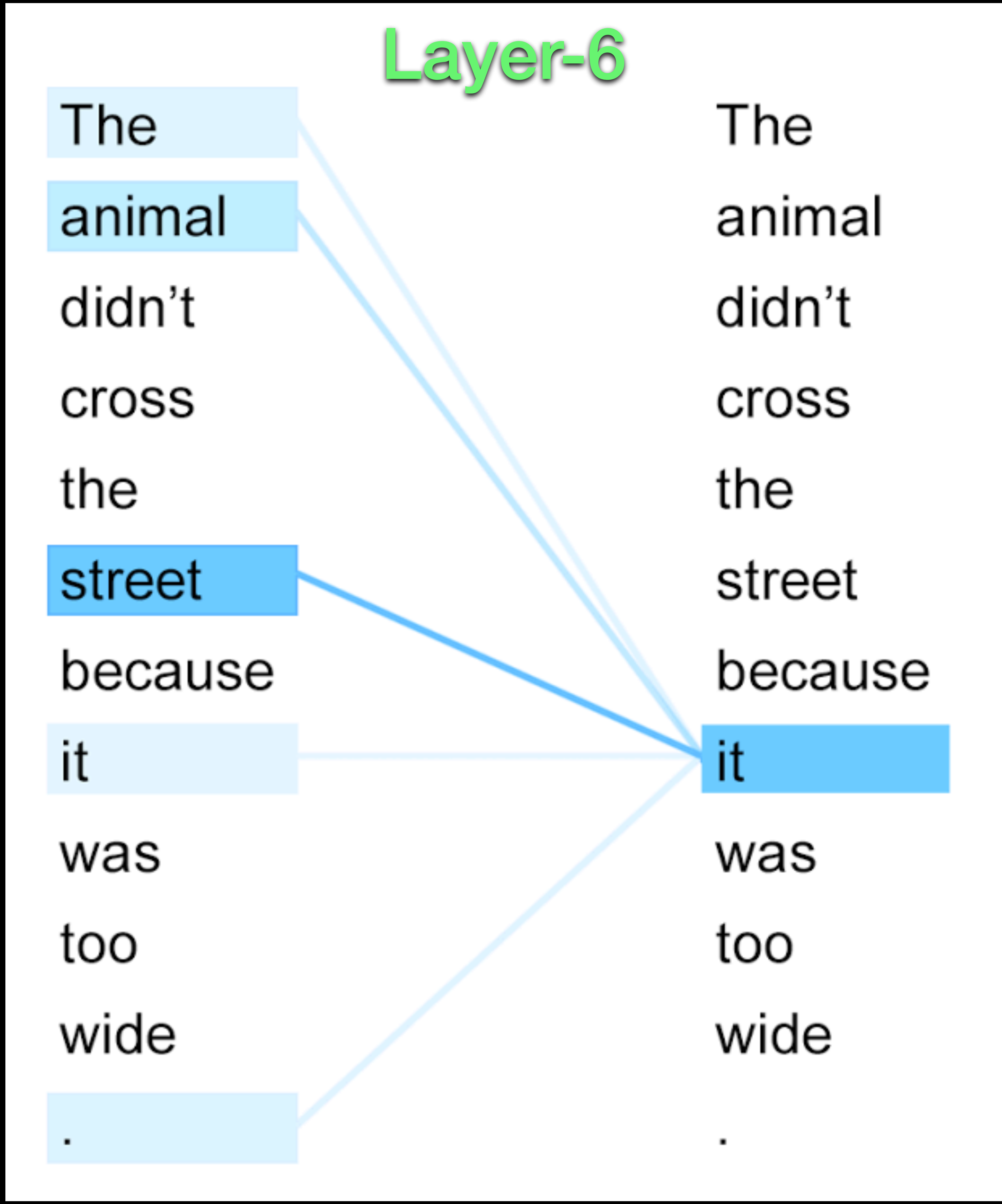


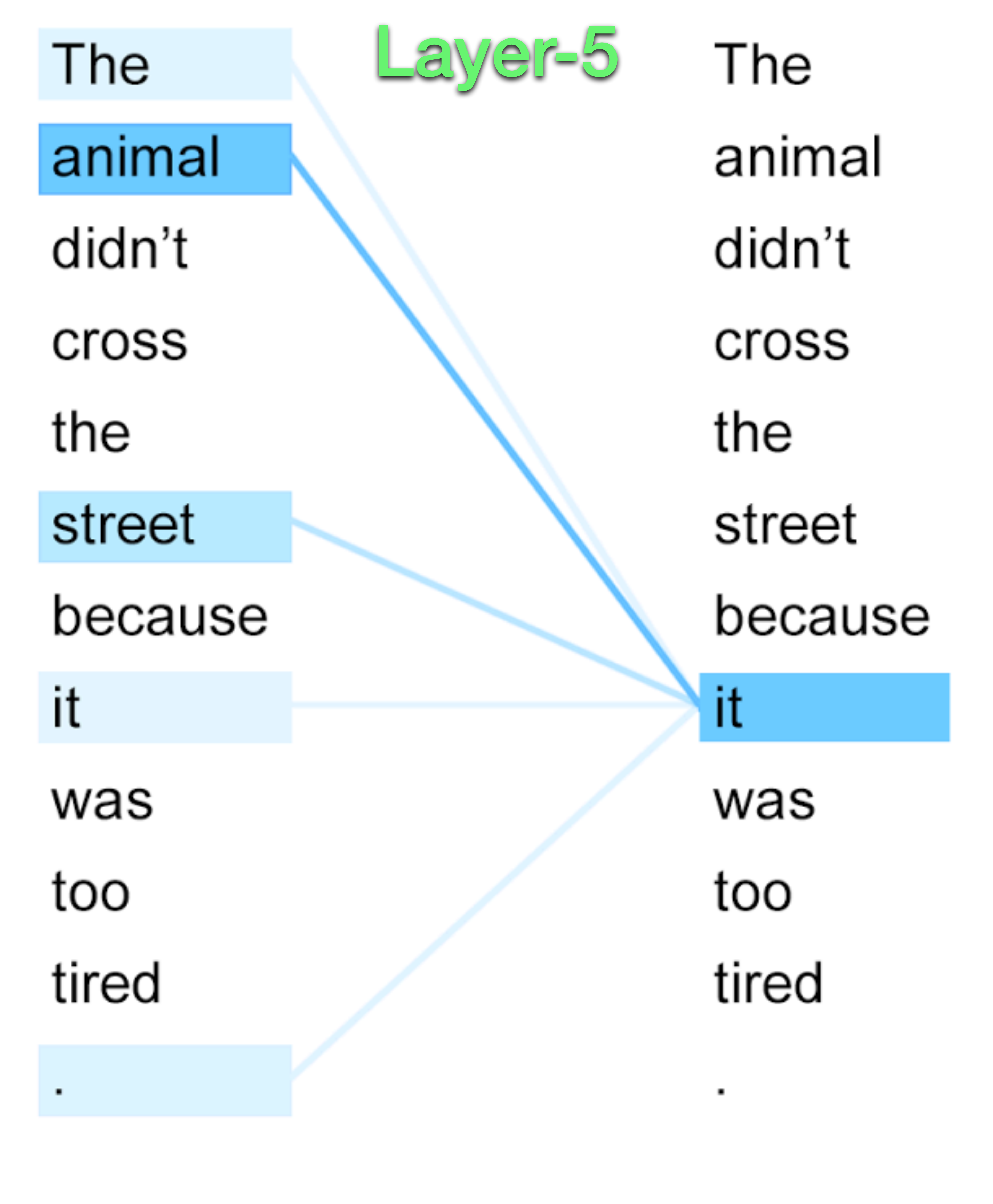


Self-attention revisited

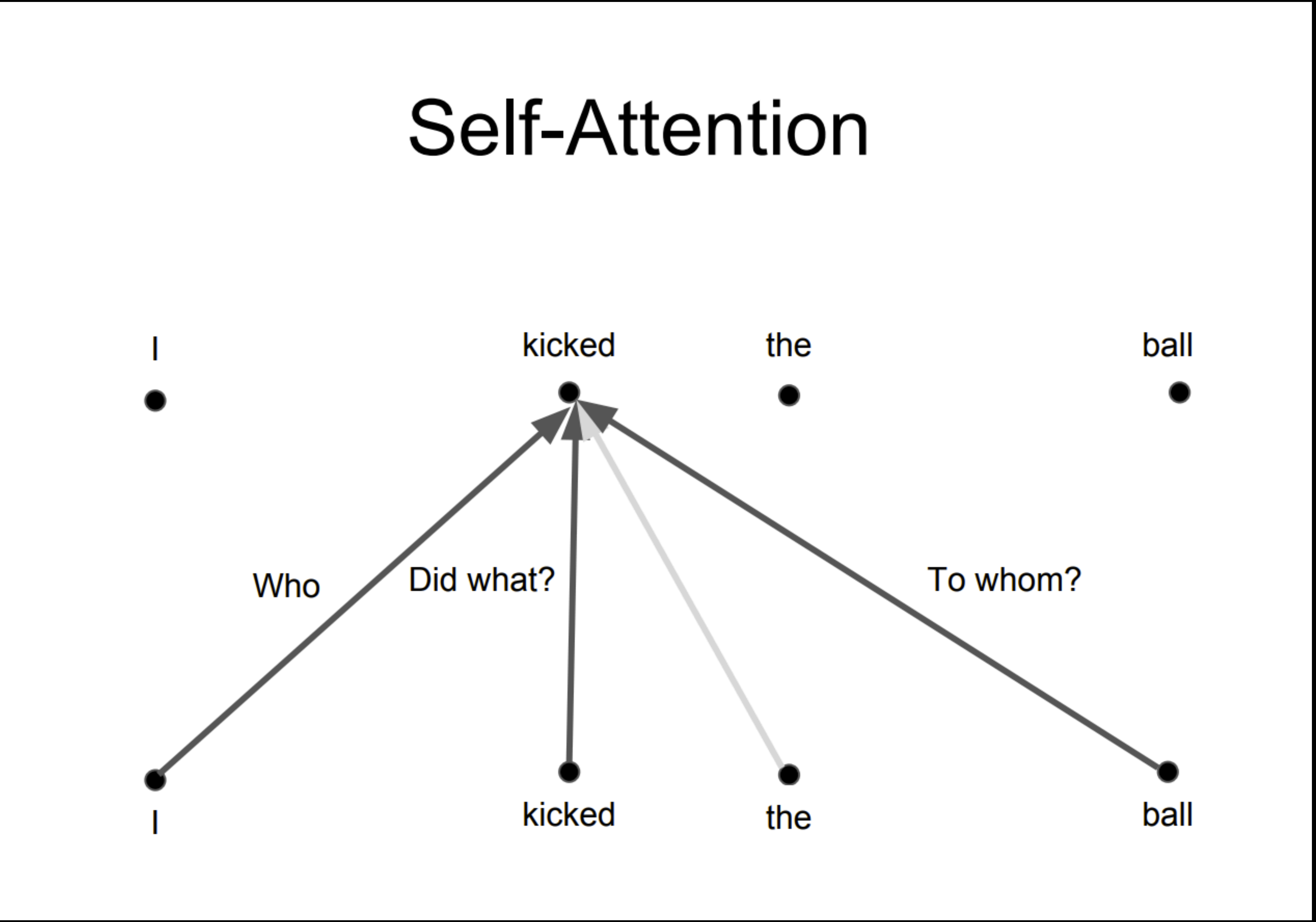


Self-attention revisited

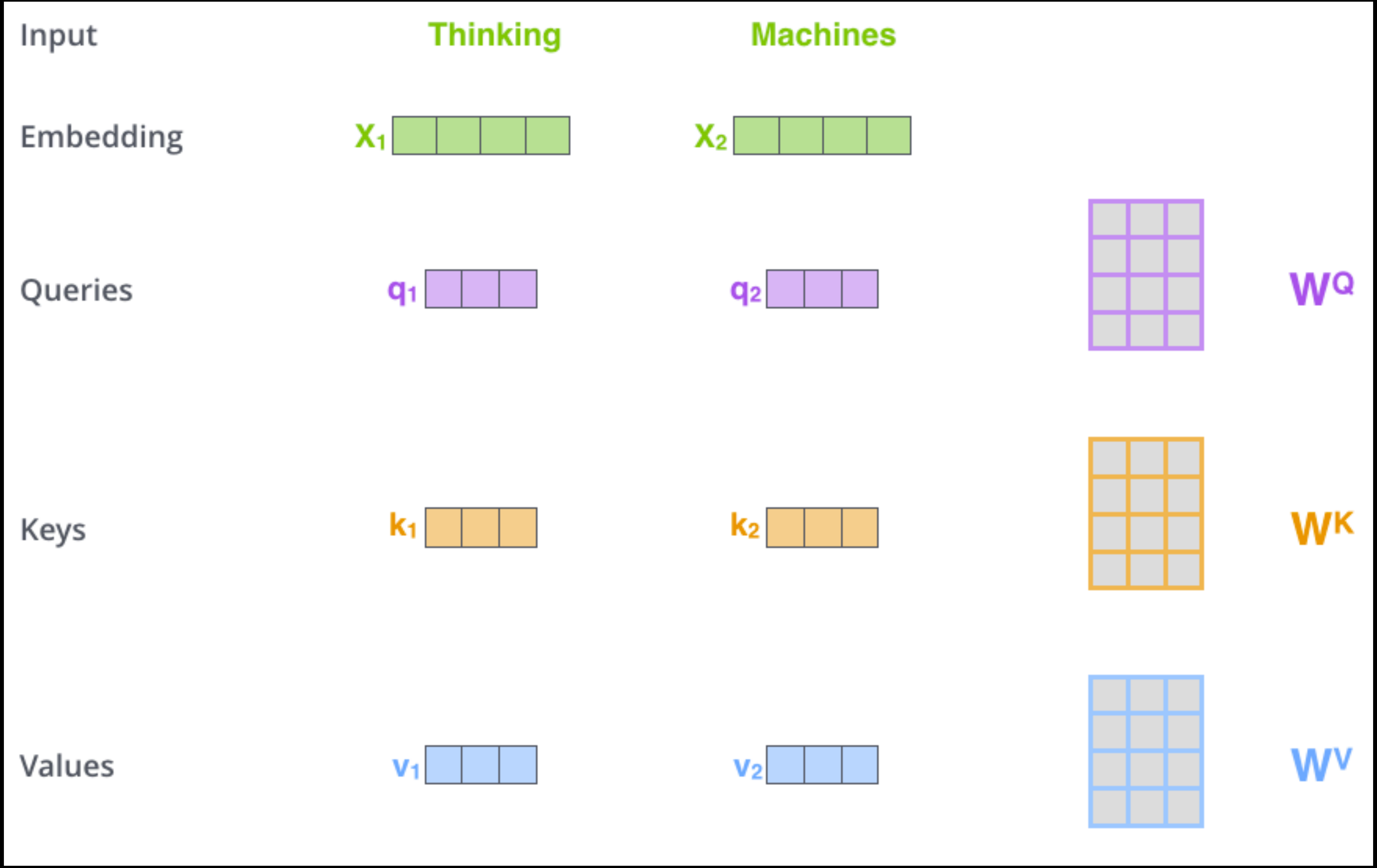




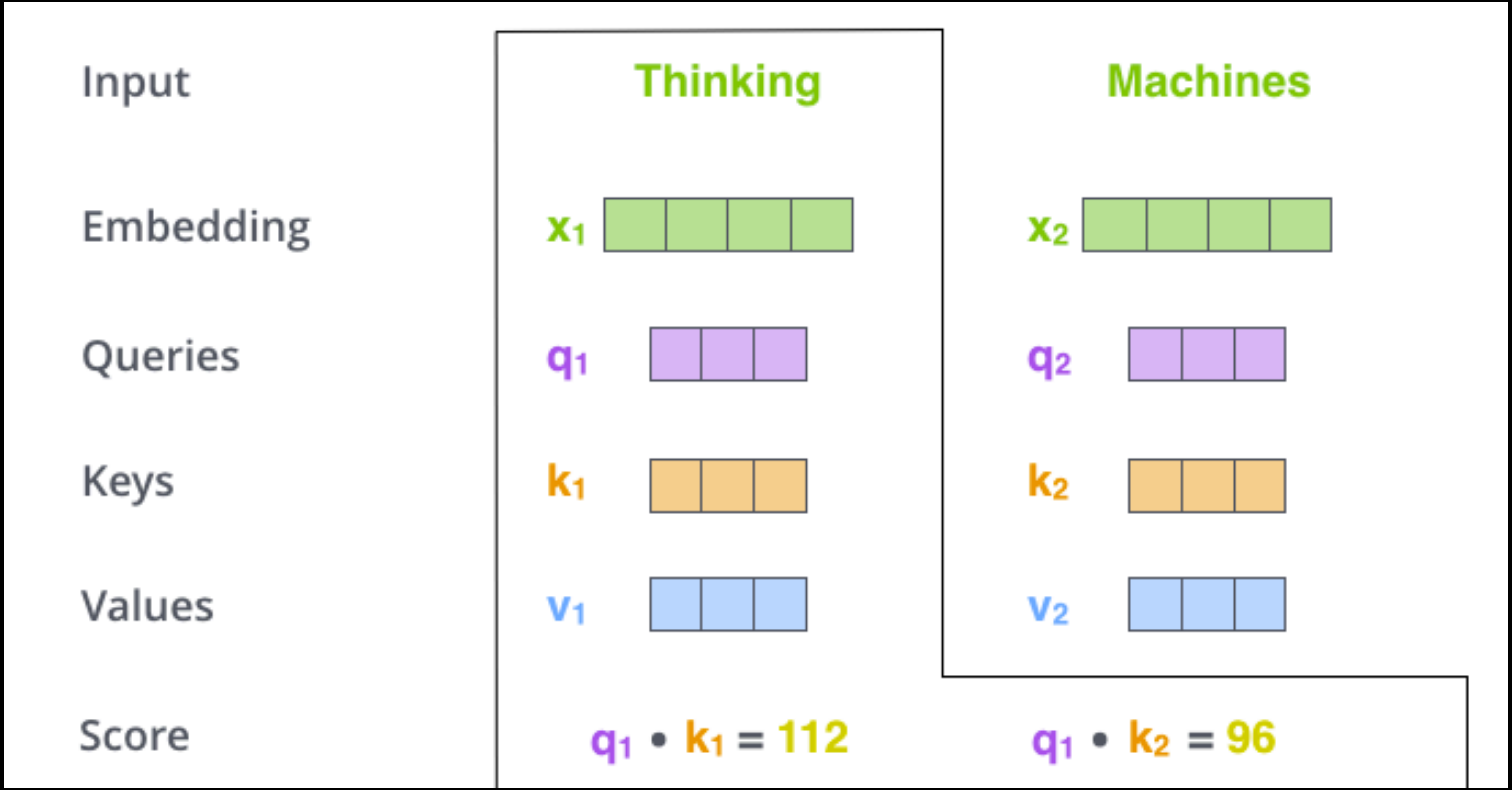
Self-attention revisited



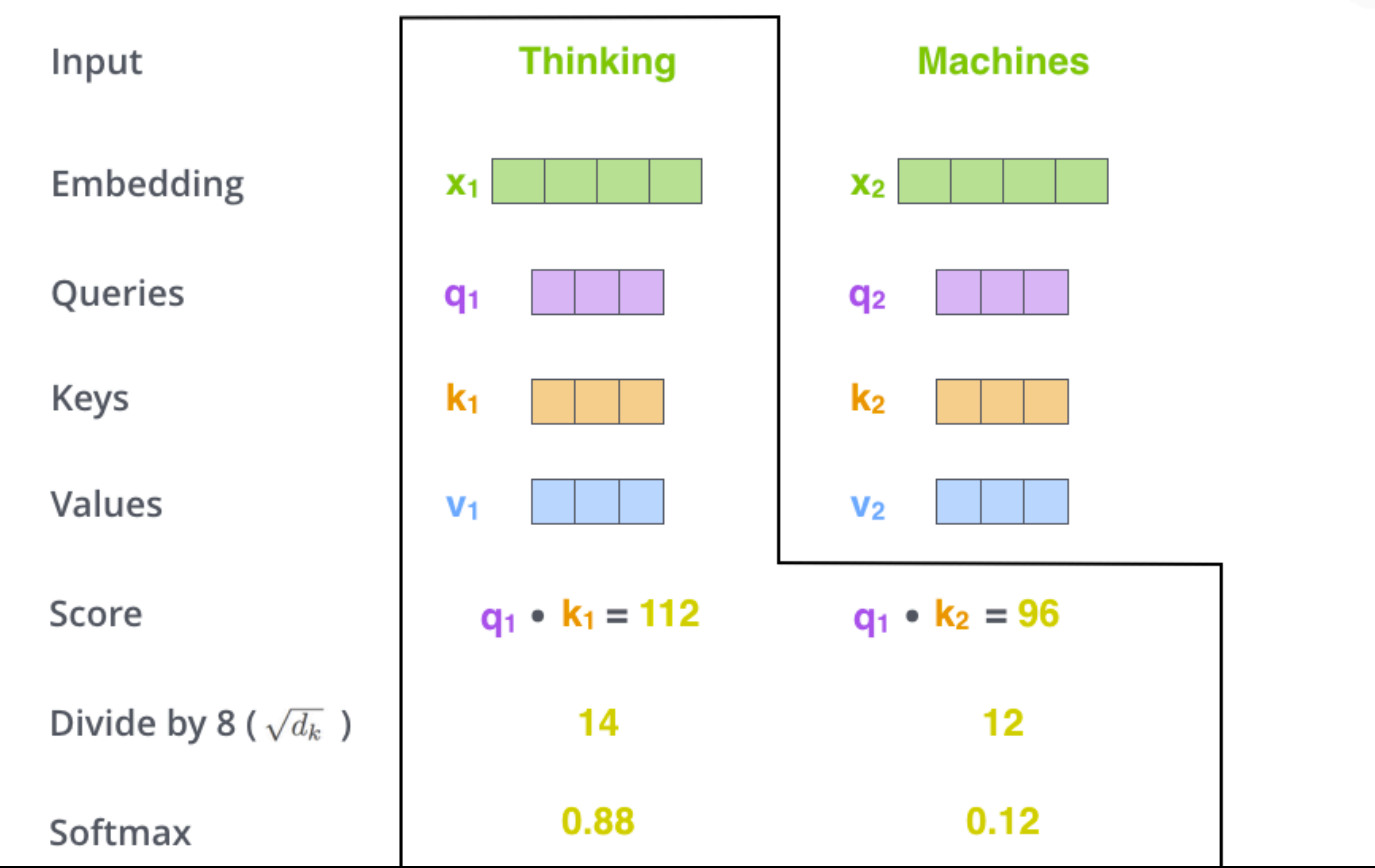
Self-attention revisited



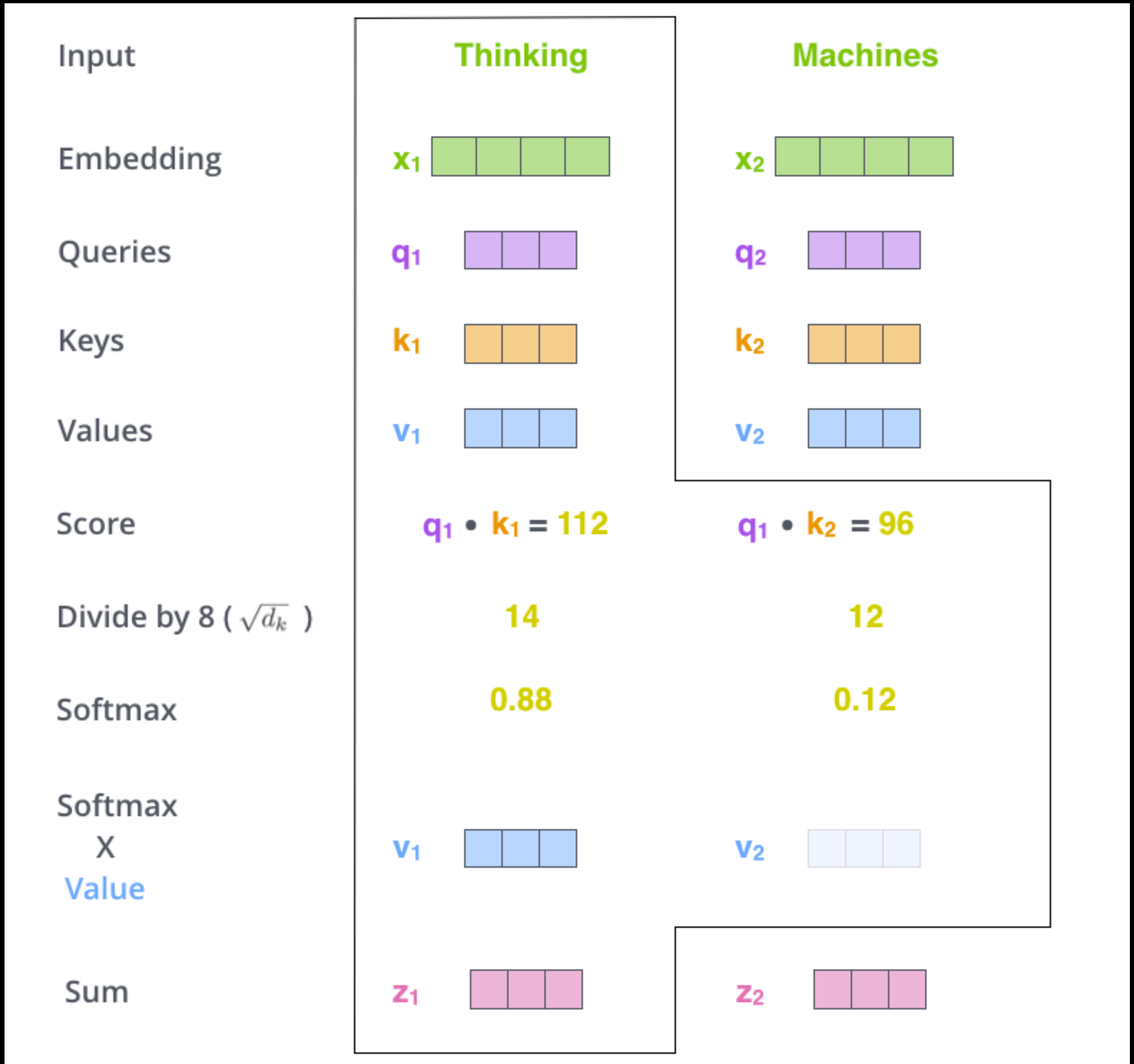
Self-attention revisited



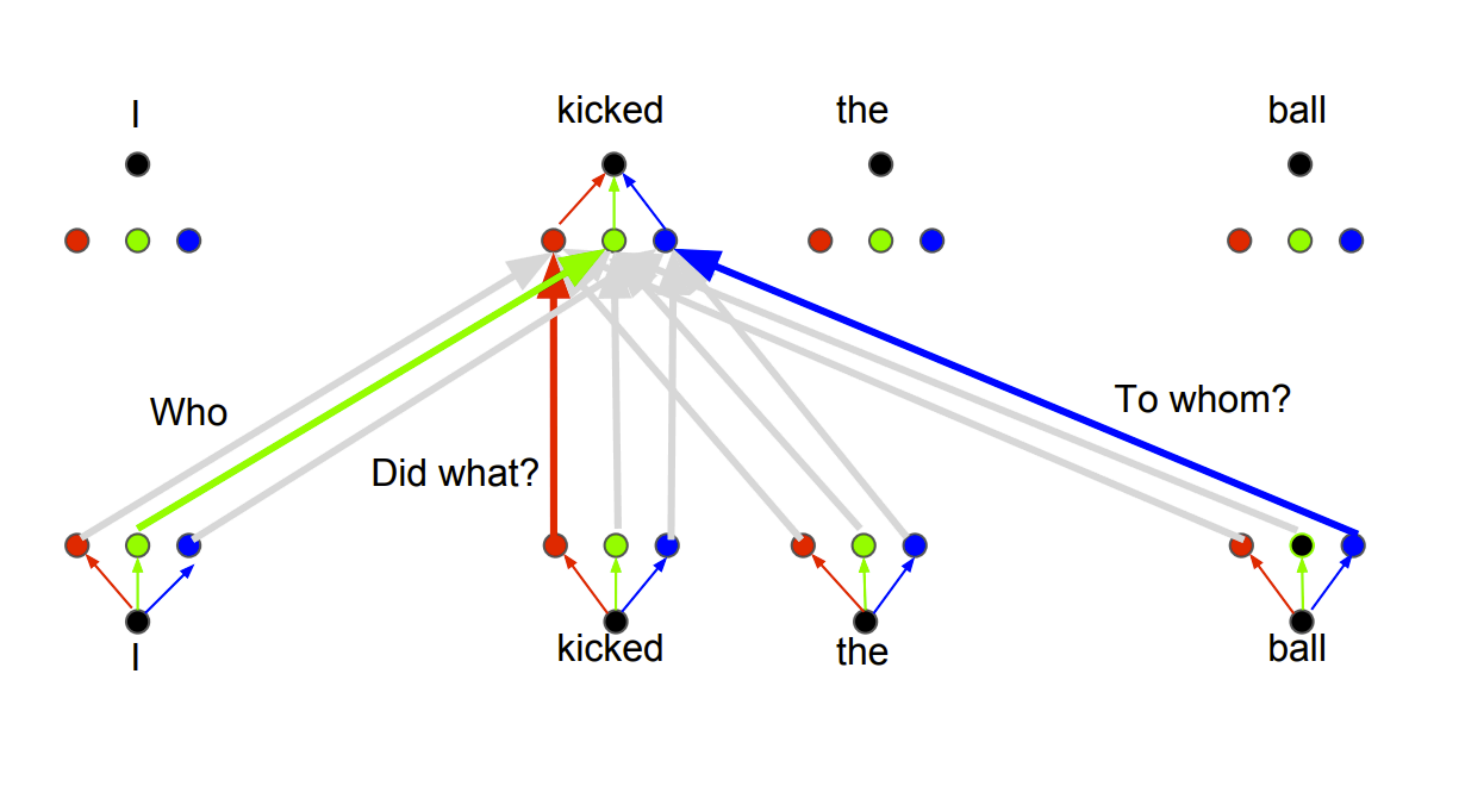
Self-attention revisited



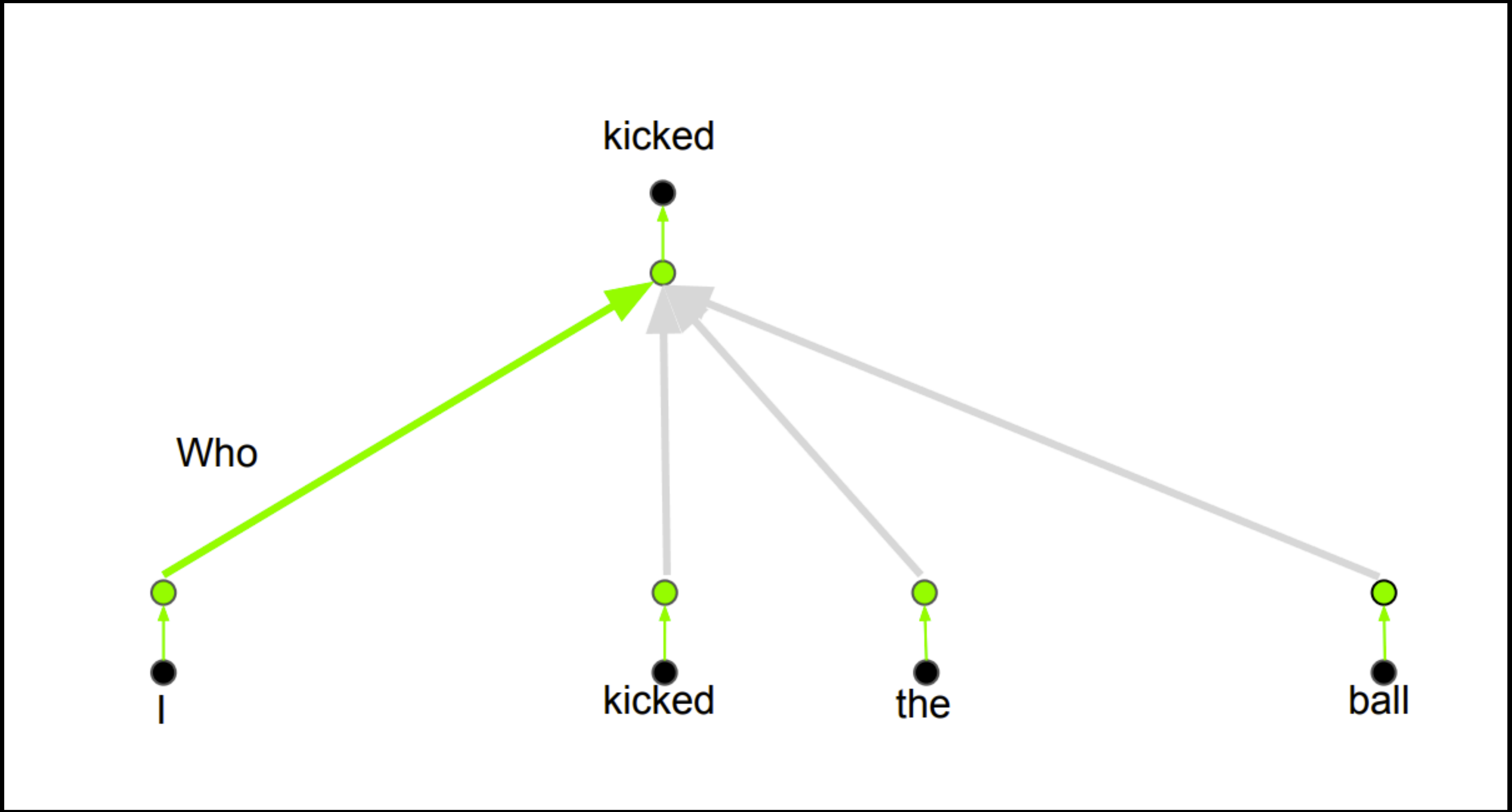
Self-attention revisited



Self-attention multi-head



Self-attention multi-head - role of attention heads



Self-attention multi-head - role of attention heads