Machine Learning to Language Model

Topic 03 - Self-Attention

Jaihua Yen
https://jaihuayen.github.io/

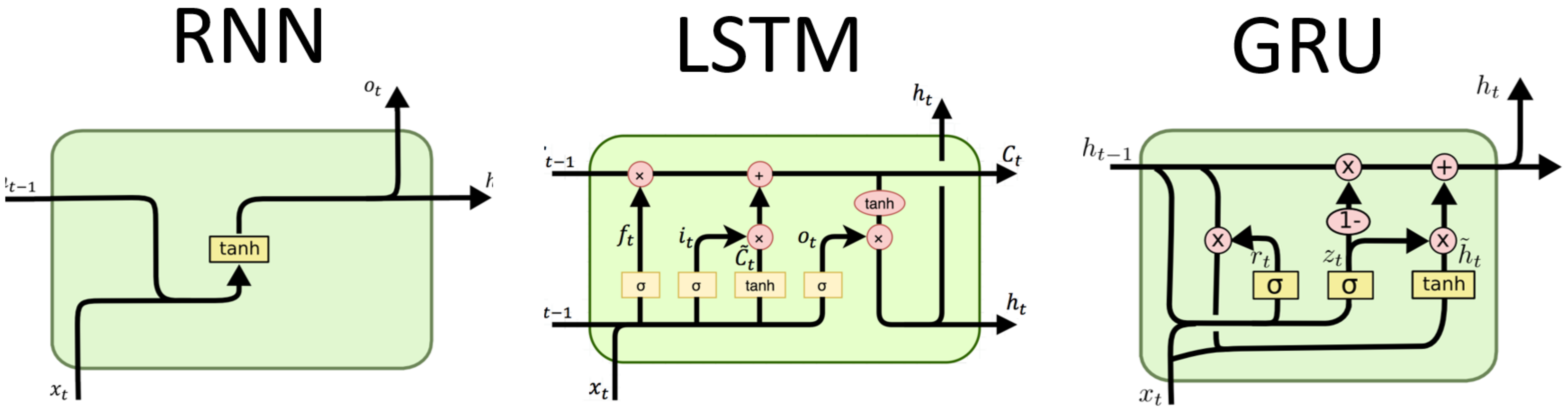
Contents

- Transformer Overview
- Self-Attention
- Wrap Up

Why Transformer?

Low Efficiency of Recurrent Neural Network

Step-by-Step is time-consuming

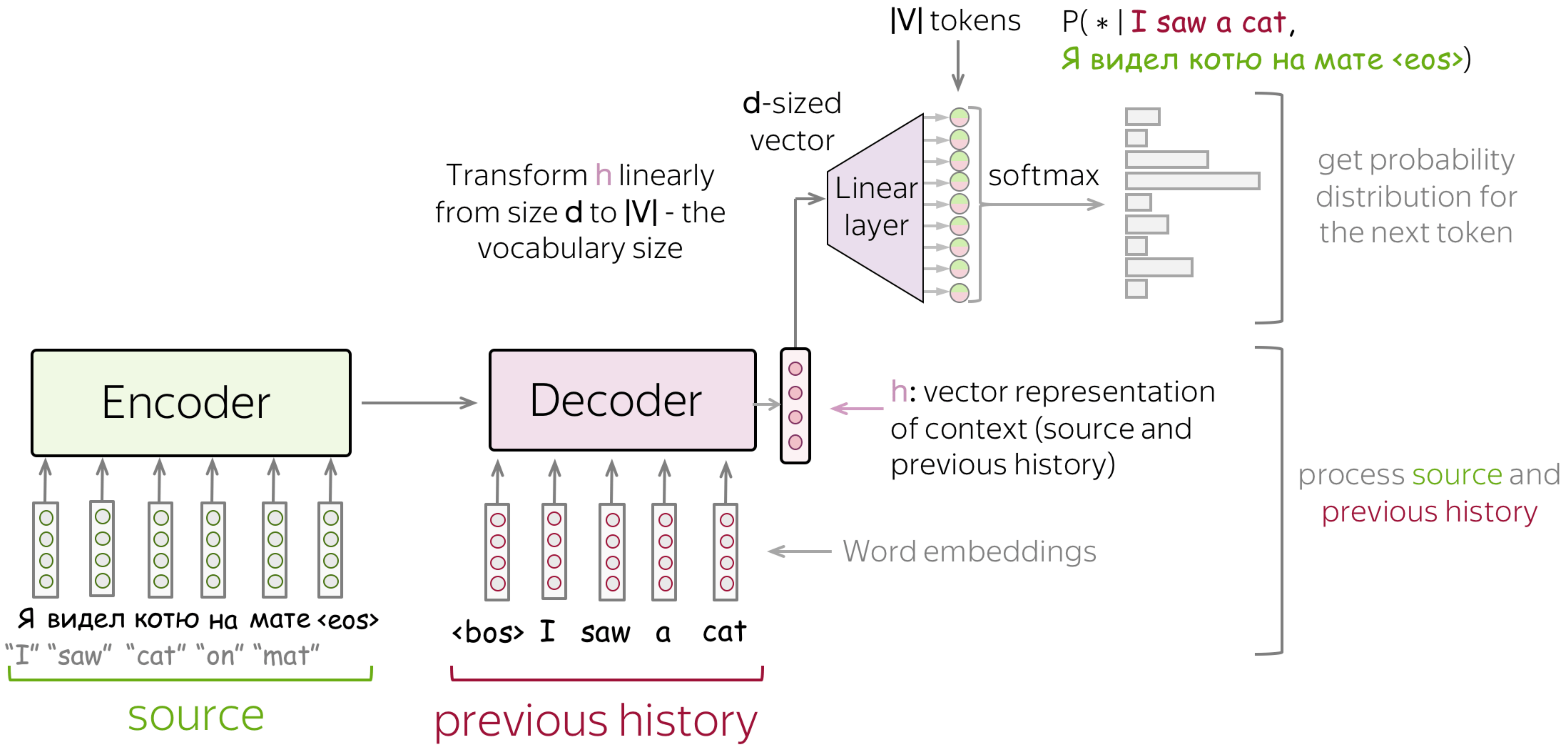


http://dprogrammer.org/rnn-lstm-gru

Lost Information in Recurrent Neural Network

The information in the beginning will degrade in the future steps

Is there a way to see the whole picture of the sentence at one time?

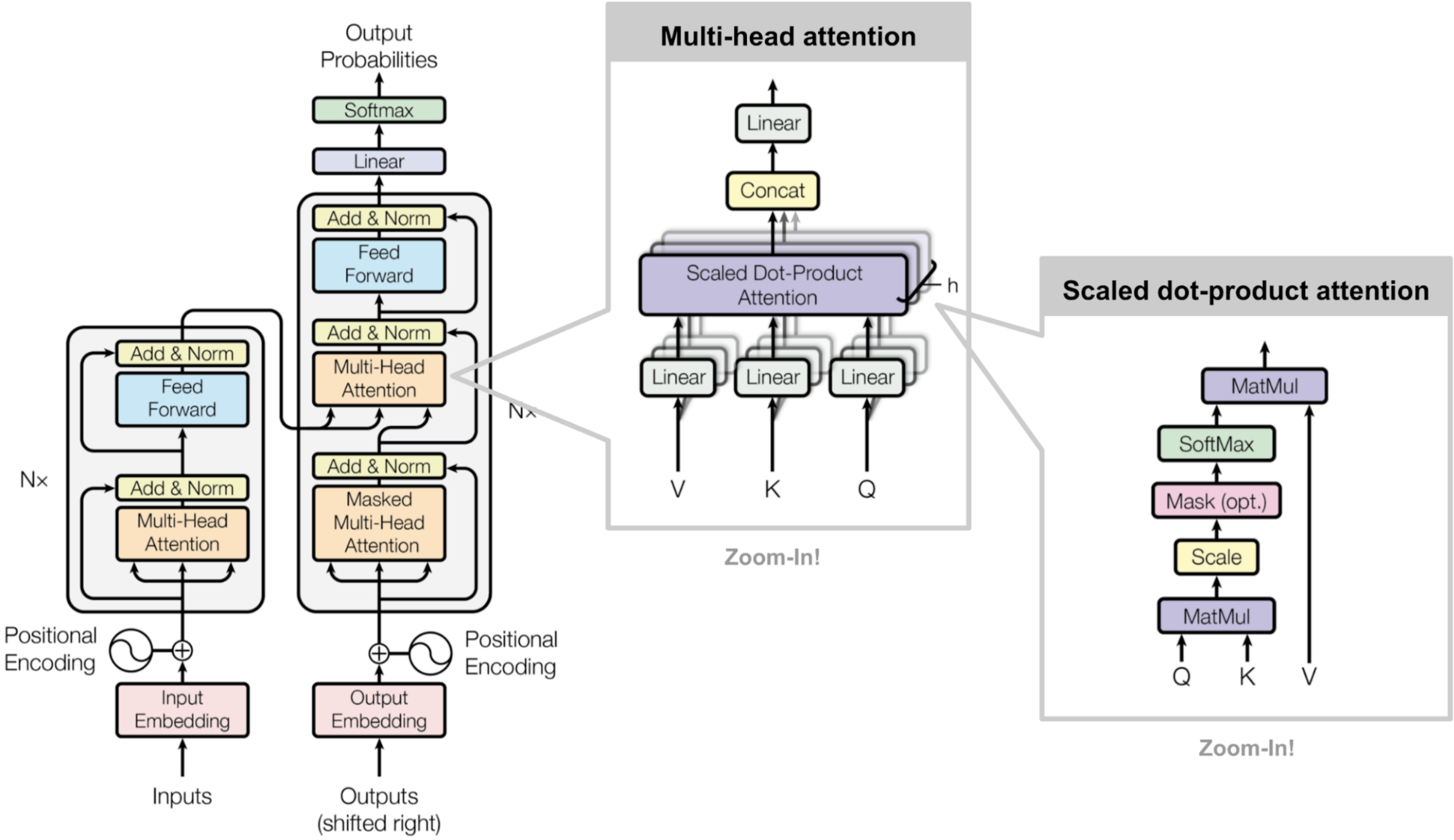


|V| tokens
P(∗ | I saw a cat,
Я видел котю на мате <eos>)
d-sized vector
Transform h linearly from size d to |V| - the vocabulary size
Linear layer
softmax
get probability distribution for the next token
Encoder
Decoder
h: vector representation of context (source and previous history)
process source and previous history
Word embeddings
Я видел котю на мате <eos>
"I" "saw" "cat" "on" "mat"
source
<bos> I saw a cat
previous history

https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

Transformer

The Key to the AI Era

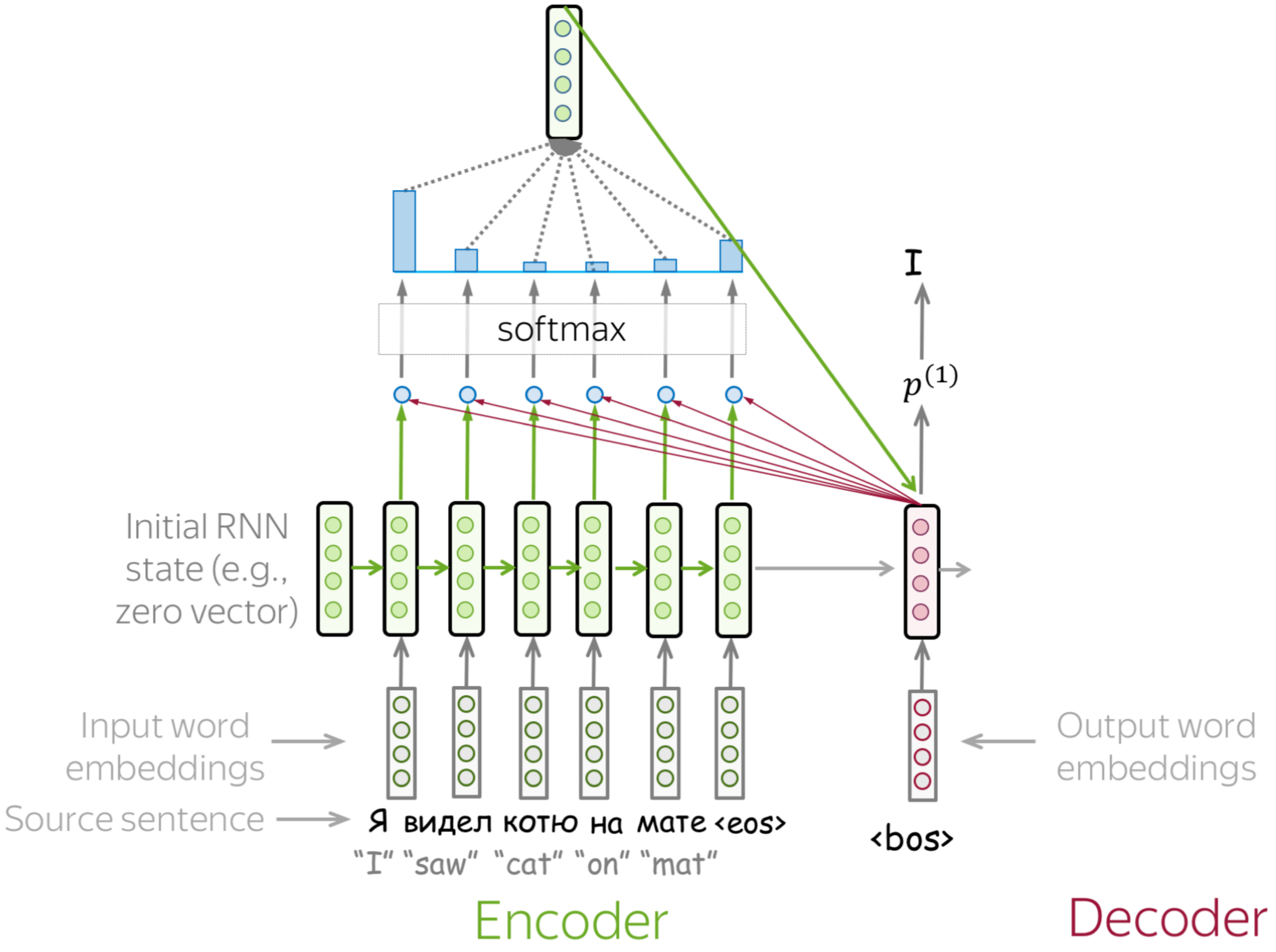


https://lilianweng.github.io/posts/2018-06-24-attention/

Self-Attention

Attention Mechanism

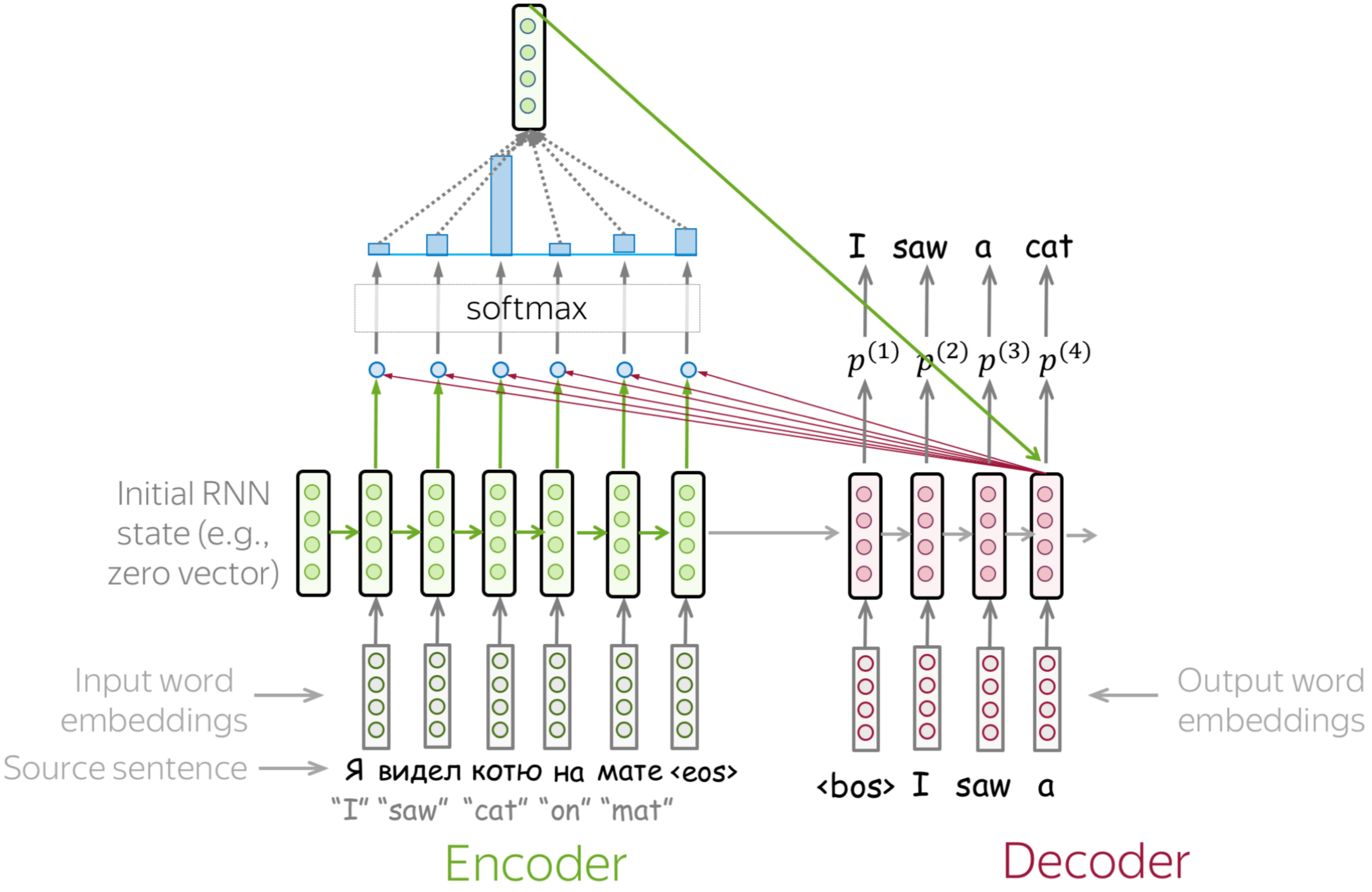
Remain the information of all words in all steps



https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

Attention Mechanism

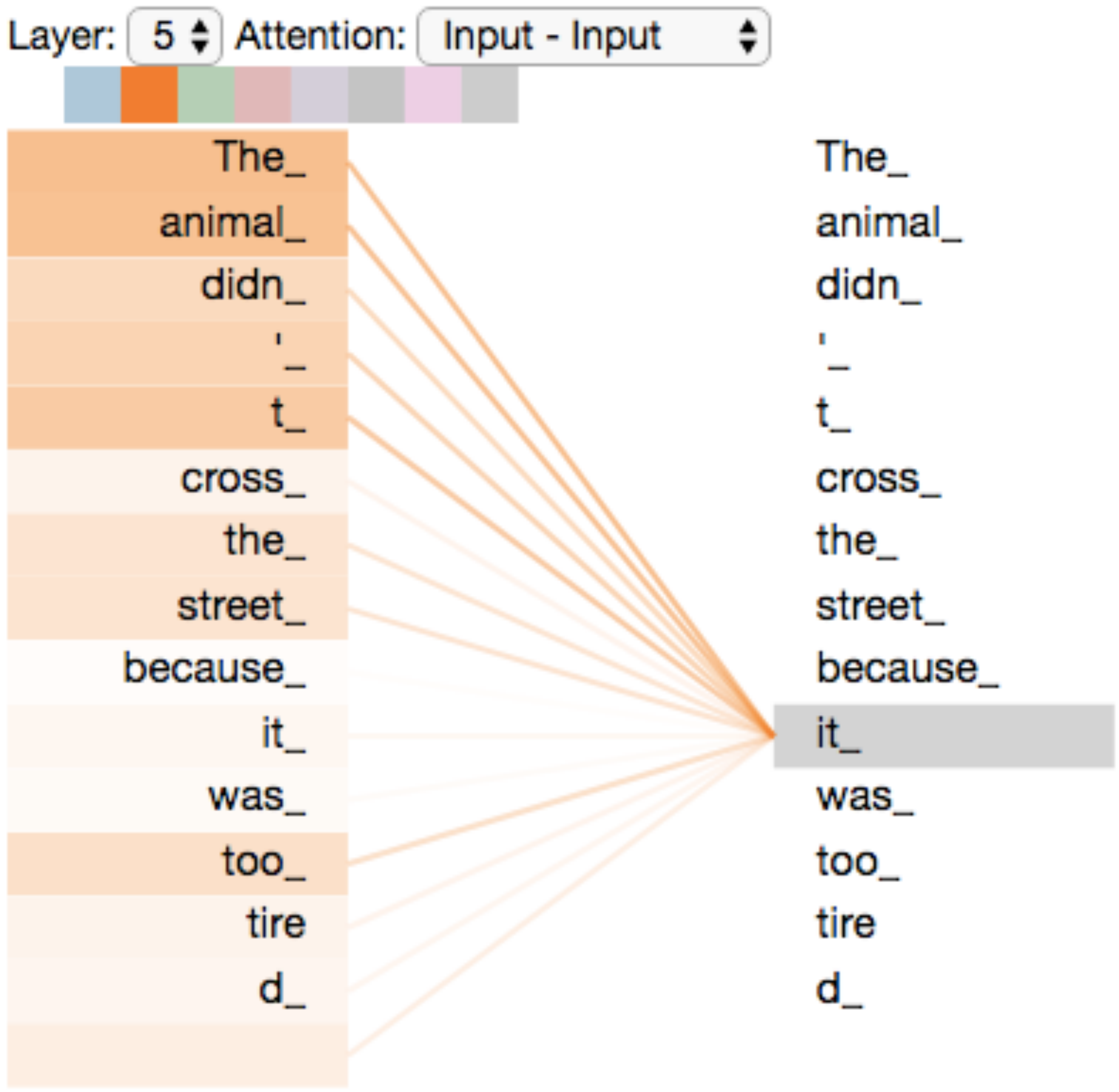
Remain the information of all words in all steps


I saw a cat
softmax
$p^{(1)}$ $p^{(2)}$ $p^{(3)}$ $p^{(4)}$
Initial RNN state (e.g., zero vector)
Input word embeddings
Output word embeddings
Source sentence
Я видел котю на мате <eos>
"I" "saw" "cat" "on" "mat"
<bos> I saw a
Encoder
Decoder

https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

Self-Attention

Attention to the same sentence

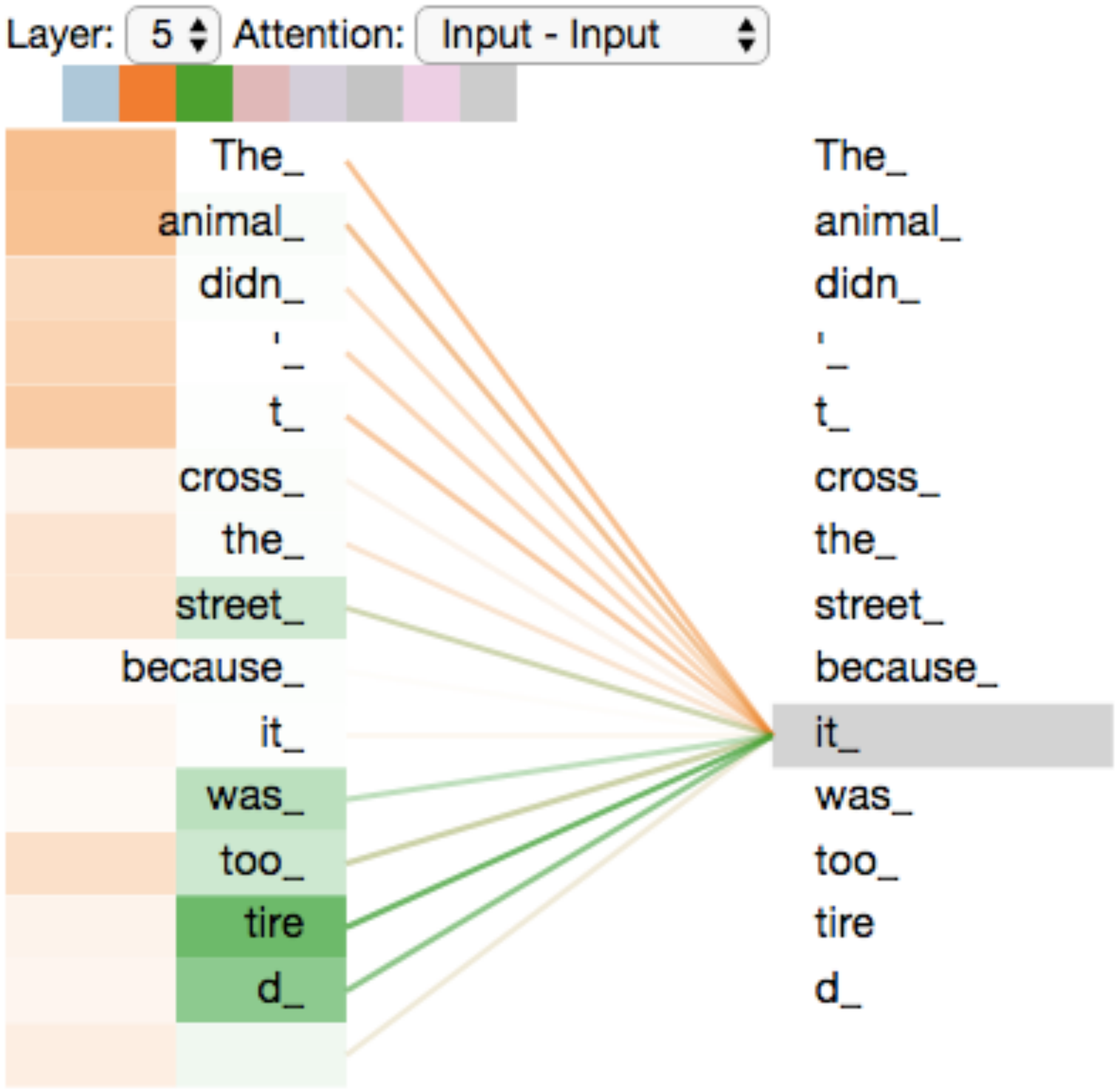


Layer: 5 Attention: Input - Input

https://jalammar.github.io/illustrated-transformer/

Multi-Head Self-Attention

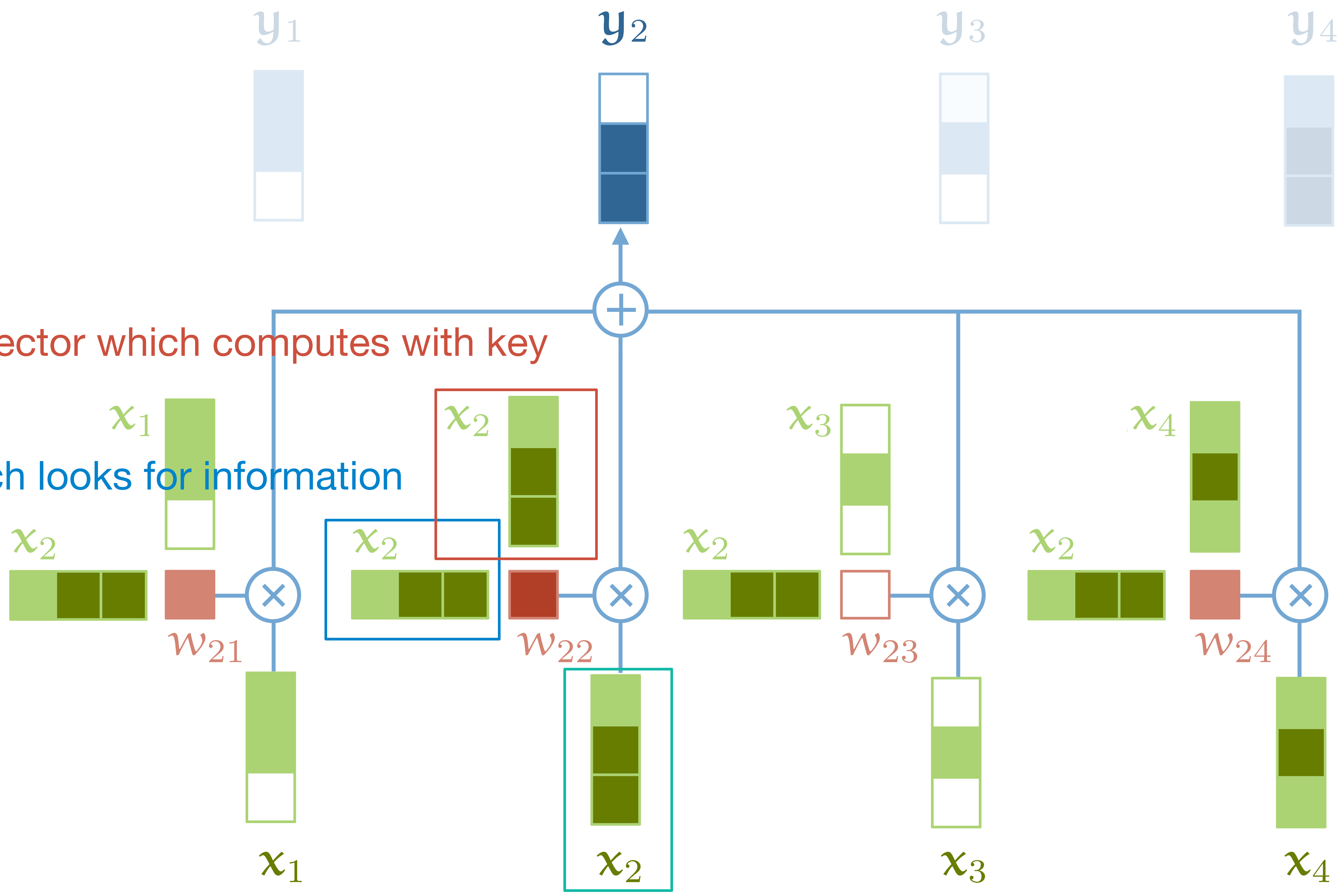
Attention to the same sentence



Layer: 5 Attention: Input - Input

https://jalammar.github.io/illustrated-transformer/

Query Key Value in Self-Attention



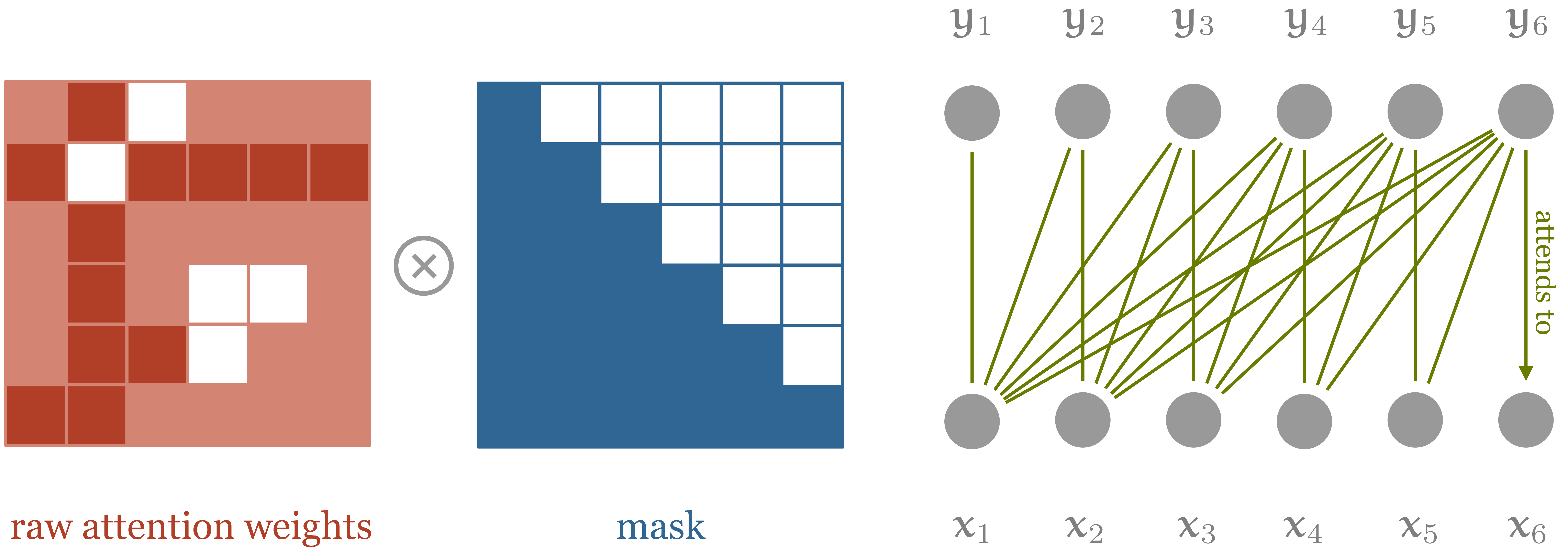
Key: The vector which computes with key

Query: The vector which looks for information

Value: The vector which contains information

https://peterbloem.nl/blog/transformers

Masked Self-Attention



https://peterbloem.nl/blog/transformers

Let’s do this in Colab!

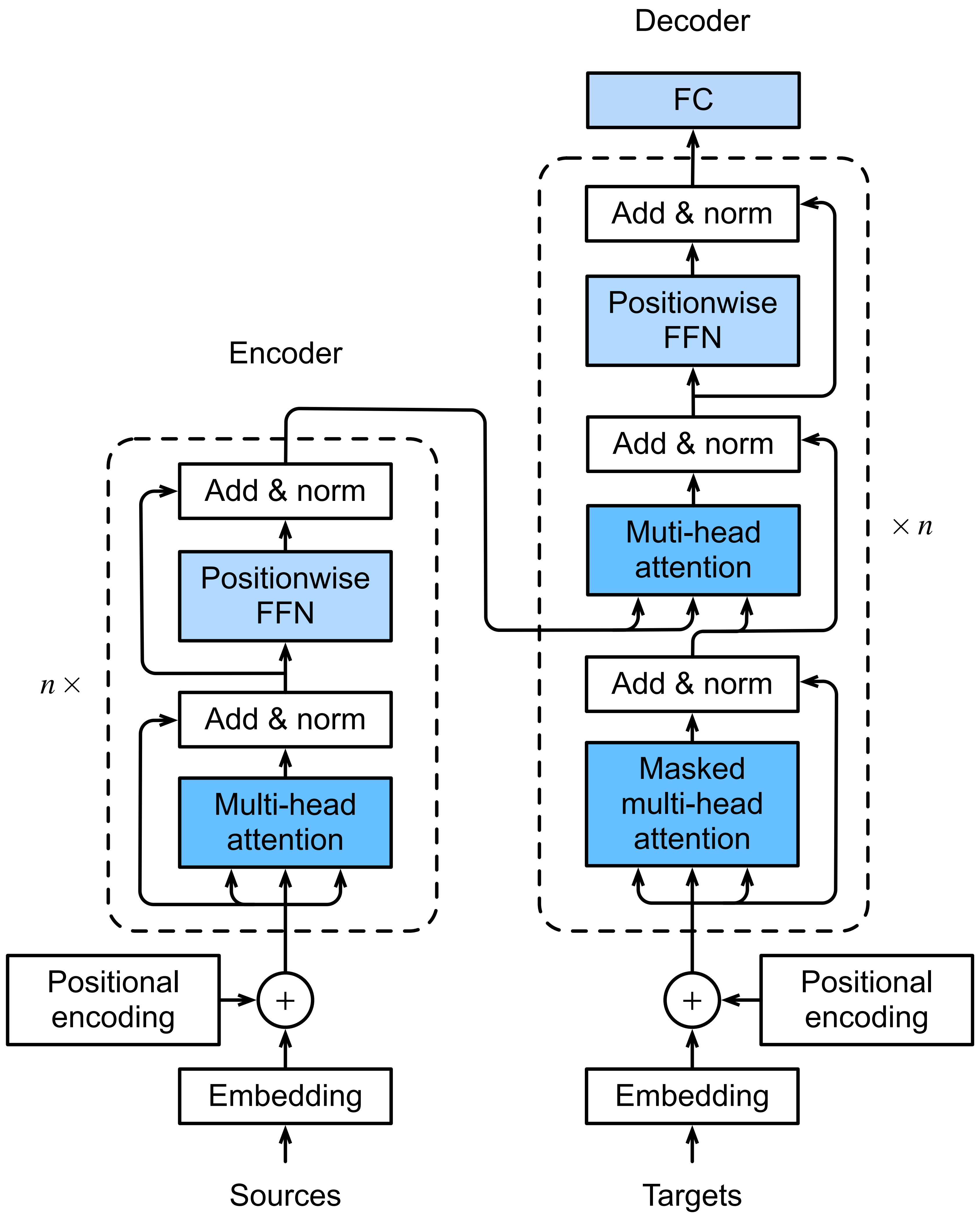
Wrap Up

What We Have Gone Through

- Overview of Transformer
- Self-Attention Mechanism

What’s Next

- Transformer Encoder
- Transformer Decoder
- Positional Encoding
- Wrap Up Transformer



Q & A