Transformers

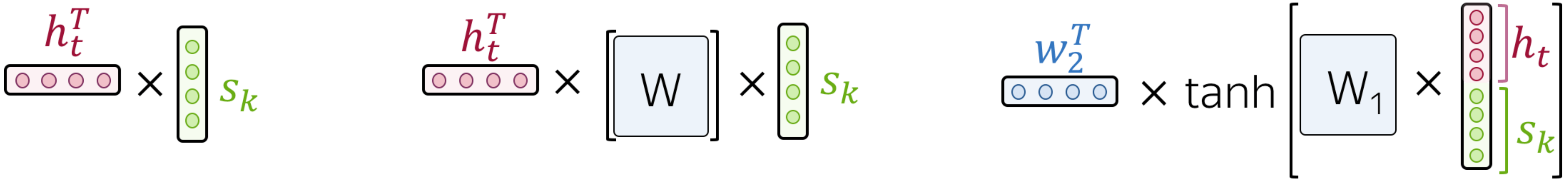
Makarand Tapaswi

CS7.505 Spring 2024

17th February 2024

Computing Attention

Dot-product | Bilinear | Multi-Layer Perceptron



$$\text{score}(h_t, s_k) = h_t^T\ s_k$$

$$\text{score}(h_t, s_k) = h_t^T\ W s_k$$

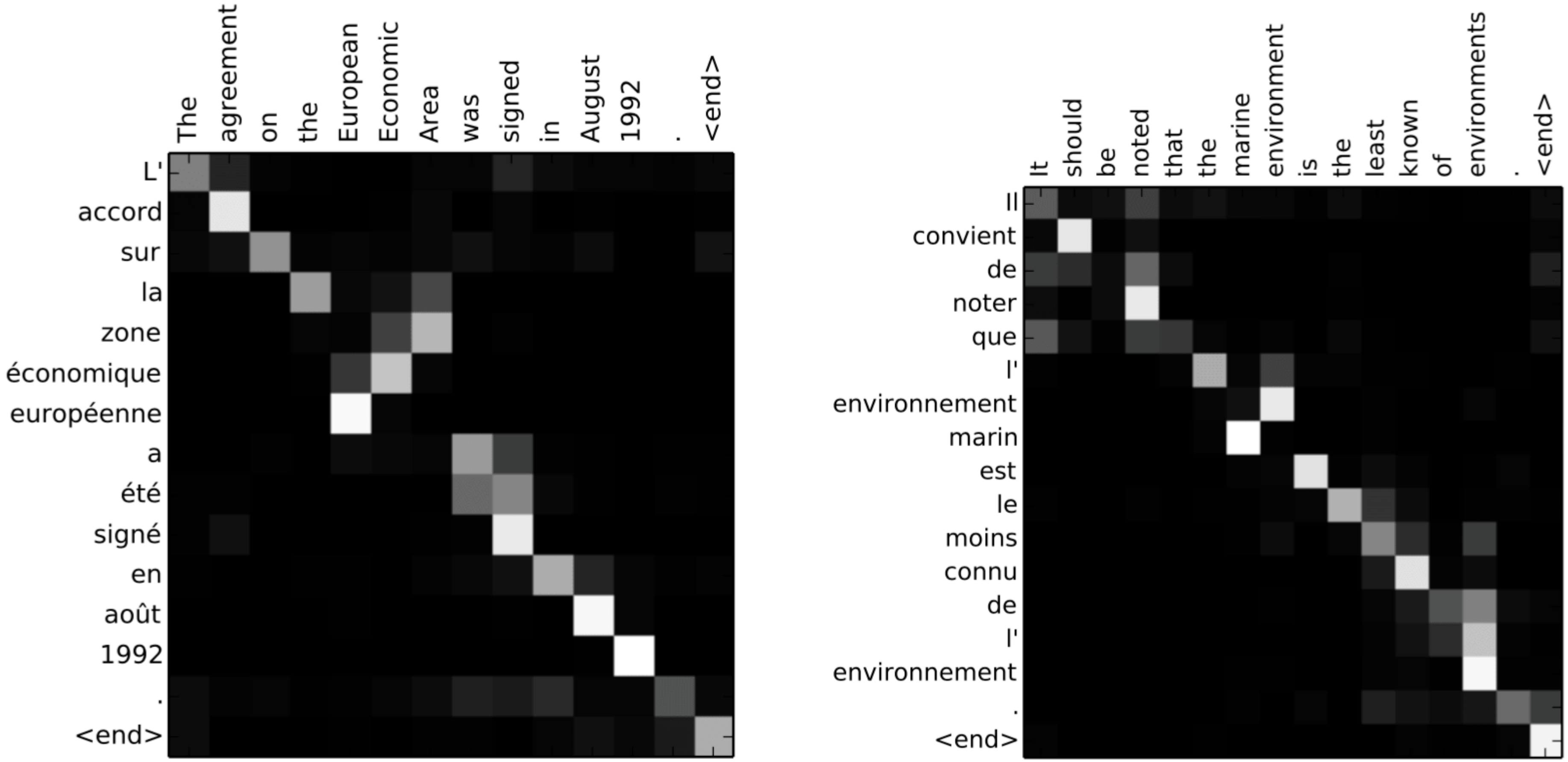
$$\text{score}(h_t, s_k) = w_2^T \cdot \tanh(W_1[h_t, s_k])$$

Softmax temperature

$$p_i = \frac{e^{z_i/T}}{\sum_j e^{z_j/T}}$$

- T == 1: normal softmax
- T >> 1: p closer to uniform distribution (flat)
- T << 1: p closer to one-hot distribution (peaky)

Attention Learns Alignment!



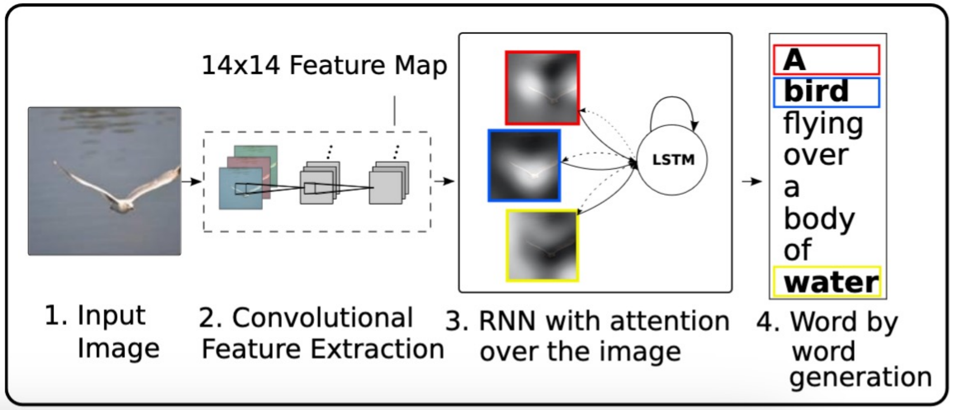
Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Kelvin Xu KELVIN.XU@UMONTREAL.CA
Jimmy Lei Ba JIMMY@PSI.UTORONTO.CA
Ryan Kiros RKIROS@CS.TORONTO.EDU
Kyunghyun Cho KYUNGHYUN.CHO@UMONTREAL.CA
Aaron Courville AARON.COURVILLE@UMONTREAL.CA
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Richard S. Zemel ZEMEL@CS.TORONTO.EDU
Yoshua Bengio FIND-ME@THE.WEB

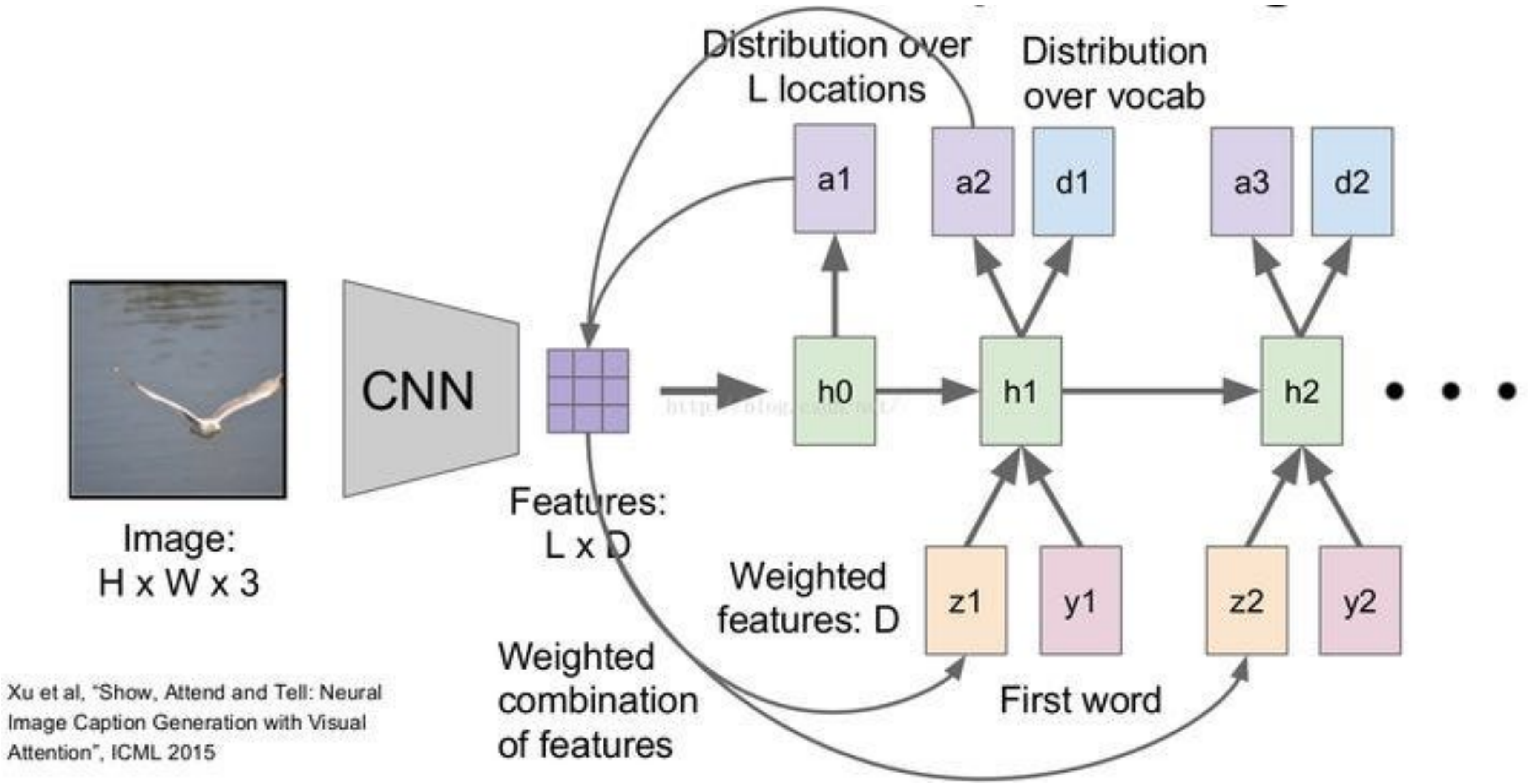
Abstract

Inspired by recent work in machine translation and object detection, we introduce an attention based model that automatically learns to describe the content of images. We describe how we can train this model in a deterministic manner using standard backpropagation techniques and stochastically by maximizing a variational lower bound. We also show through visualization how the model is able to automatically learn to fix its gaze on salient objects while generating the cor-

Figure 1. Our model learns a words/image alignment. The visualized attentional maps (3) are explained in section 3.1 & 5.4



Architecture Details

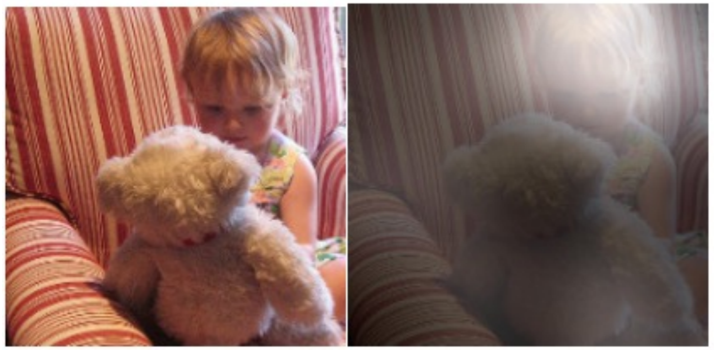


Show and Tell → Show, Attend, and Tell

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.



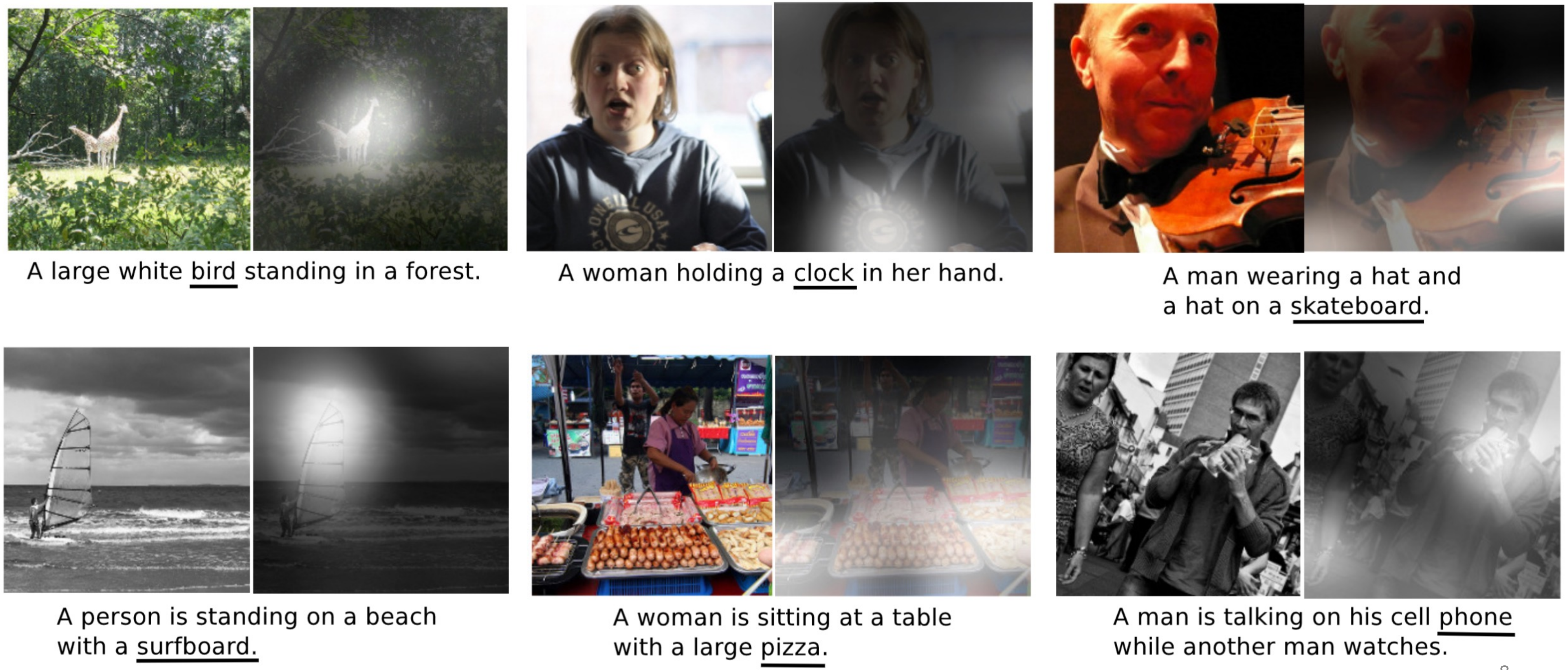
A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Understanding when things go wrong

Figure 5. Examples of mistakes where we can use attention to gain intuition into what the model saw.



Source

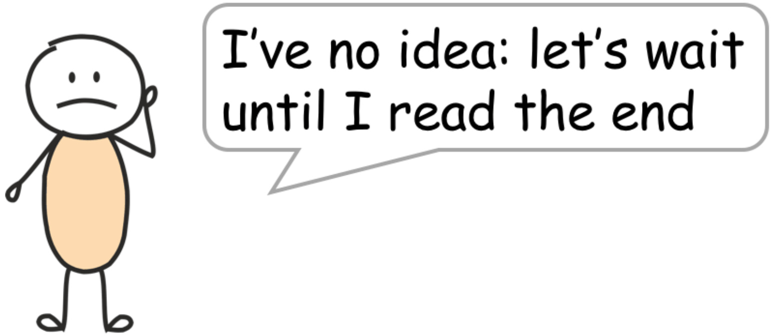
- https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html#transformer_intro

Continuing this very nice blog from last time! Please do read

RNNs → Transformers

I arrived at the bank after crossing the … …street? …river?

What does **bank** mean in this sentence?



RNNs

O(N) steps to process a sentence with length N

Transformer

Constant number of steps to process any sentence

Attention is All You Need!

- ~~Almost there … but, before that~~

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions

Information Flow

https://blog.research.google/2017/08/transformer-novel-neural-network.html

Auto-regressive decoding, Seq2Seq

Encoder

Who is doing:

- all source tokens

What they are doing:

- look at each other
- update representations

repeat
N times

Decoder

Who is doing:

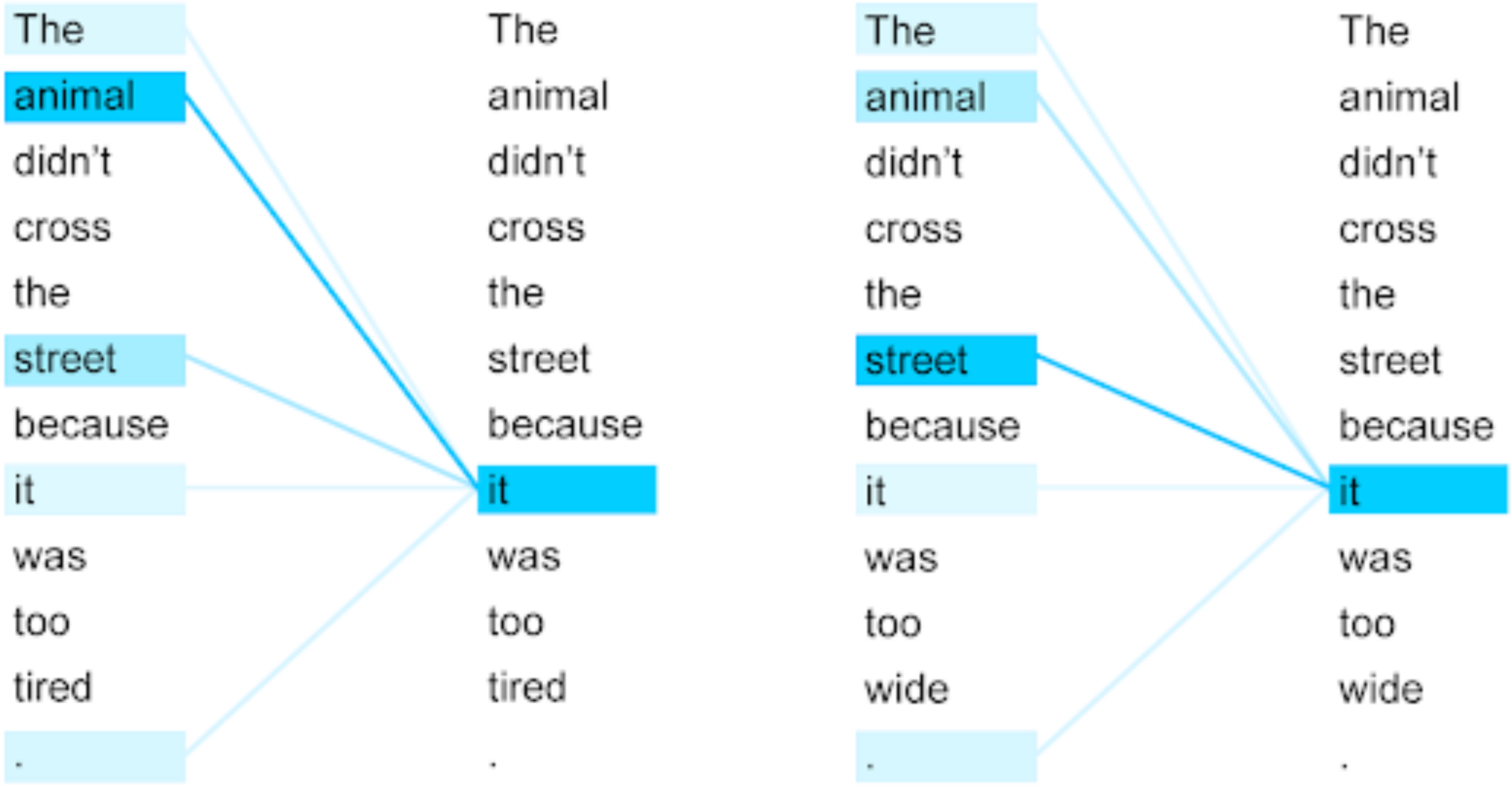
- target token at the current step

What they are doing:

- looks at previous target tokens
- looks at source representations
- update representation

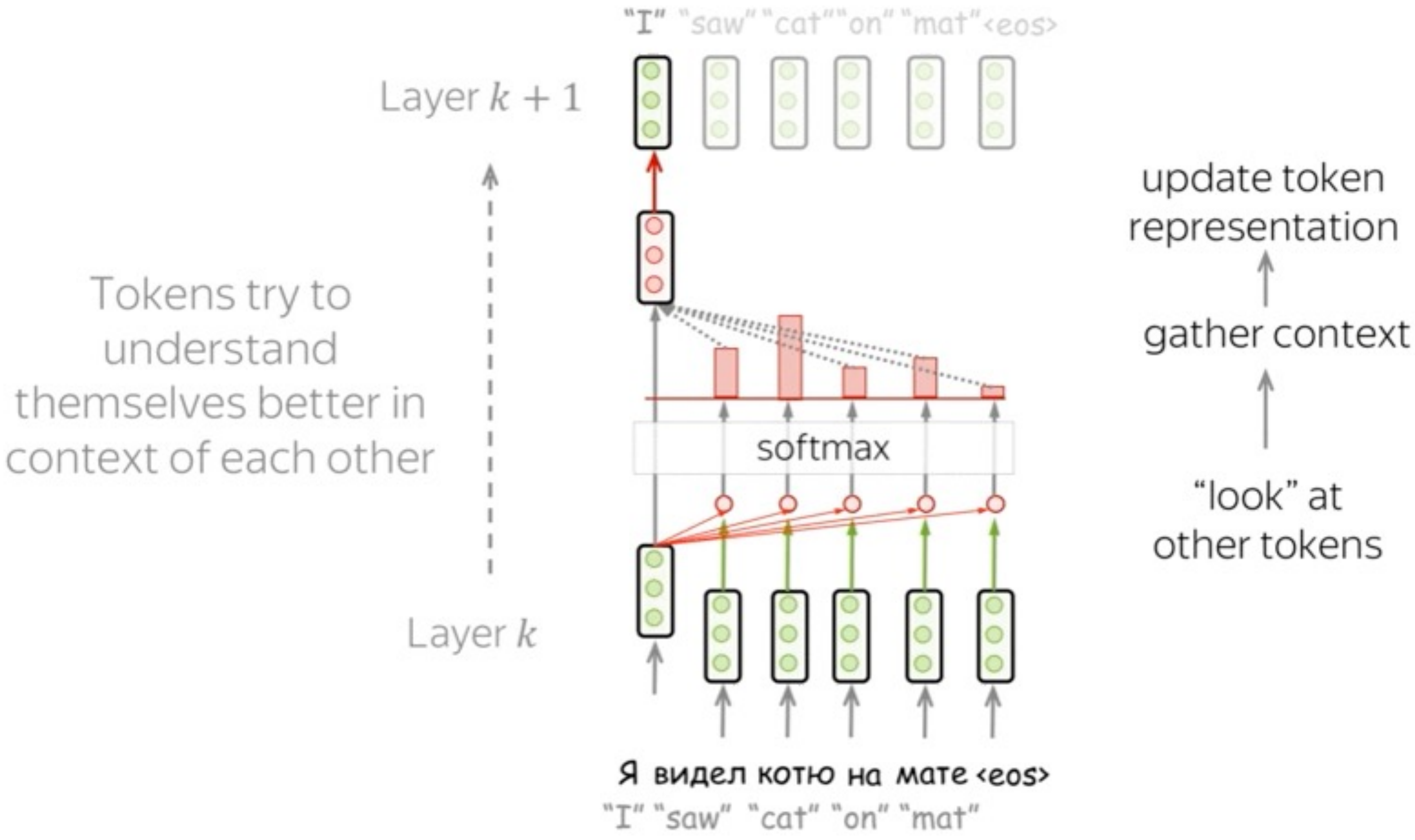
repeat
N times

Self-attention



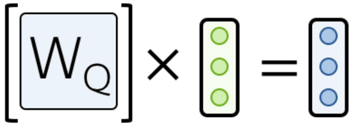
https://blog.research.google/2017/08/transformer-novel-neural-network.html

Encoder Self-attention



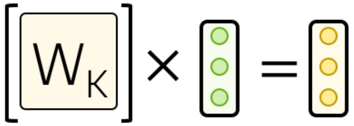
Self-Attention

Each vector receives three representations ("roles")



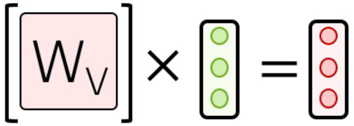
Query: vector **from** which the attention is looking

"Hey there, do you have this information?"



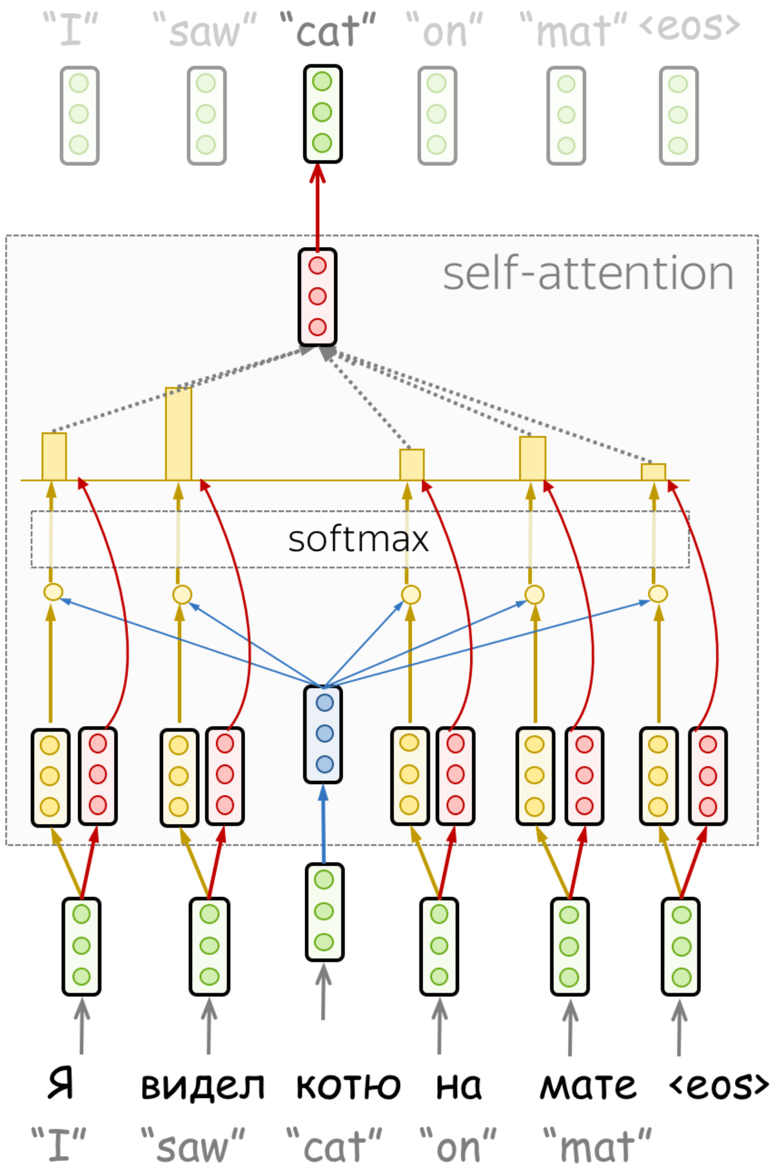
Key: vector **at** which the query looks to compute weights

"Hi, I have this information – give me a large weight!"



Value: their weighted sum is attention output

"Here's the information I have!"



Attention computation

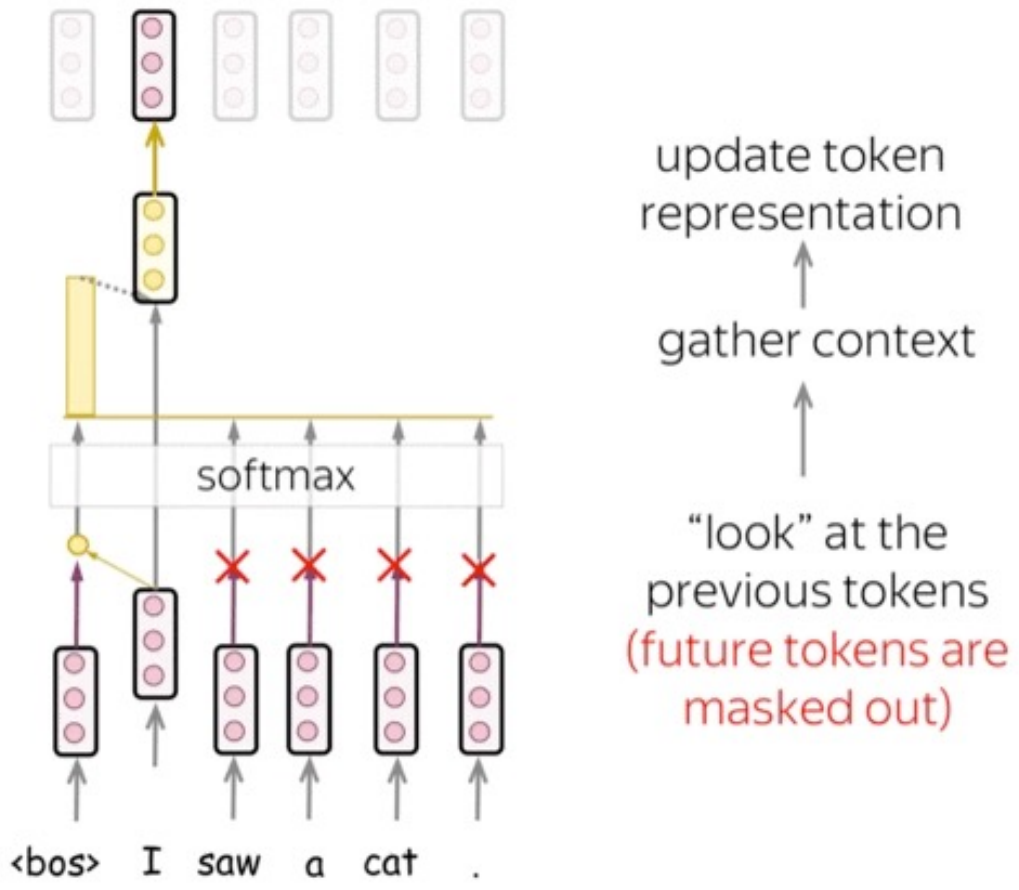
$$Attention(\underbrace{q}_{\text{from}}, \underbrace{k, v}_{\text{to}}) = \overbrace{softmax\left(\frac{qk^T}{\sqrt{d_k}}\right)}^{\text{Attention weights}} v$$

from

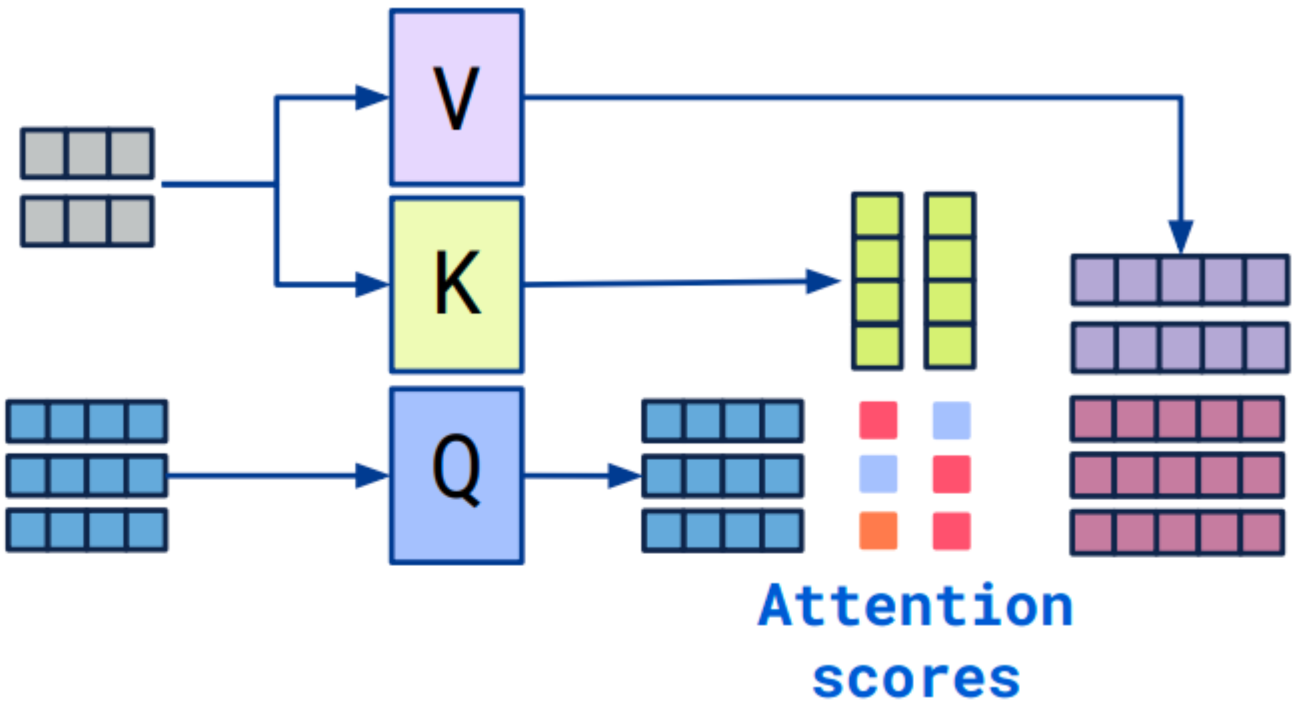
to

vector dimensionality of K, V

Masked self-attention

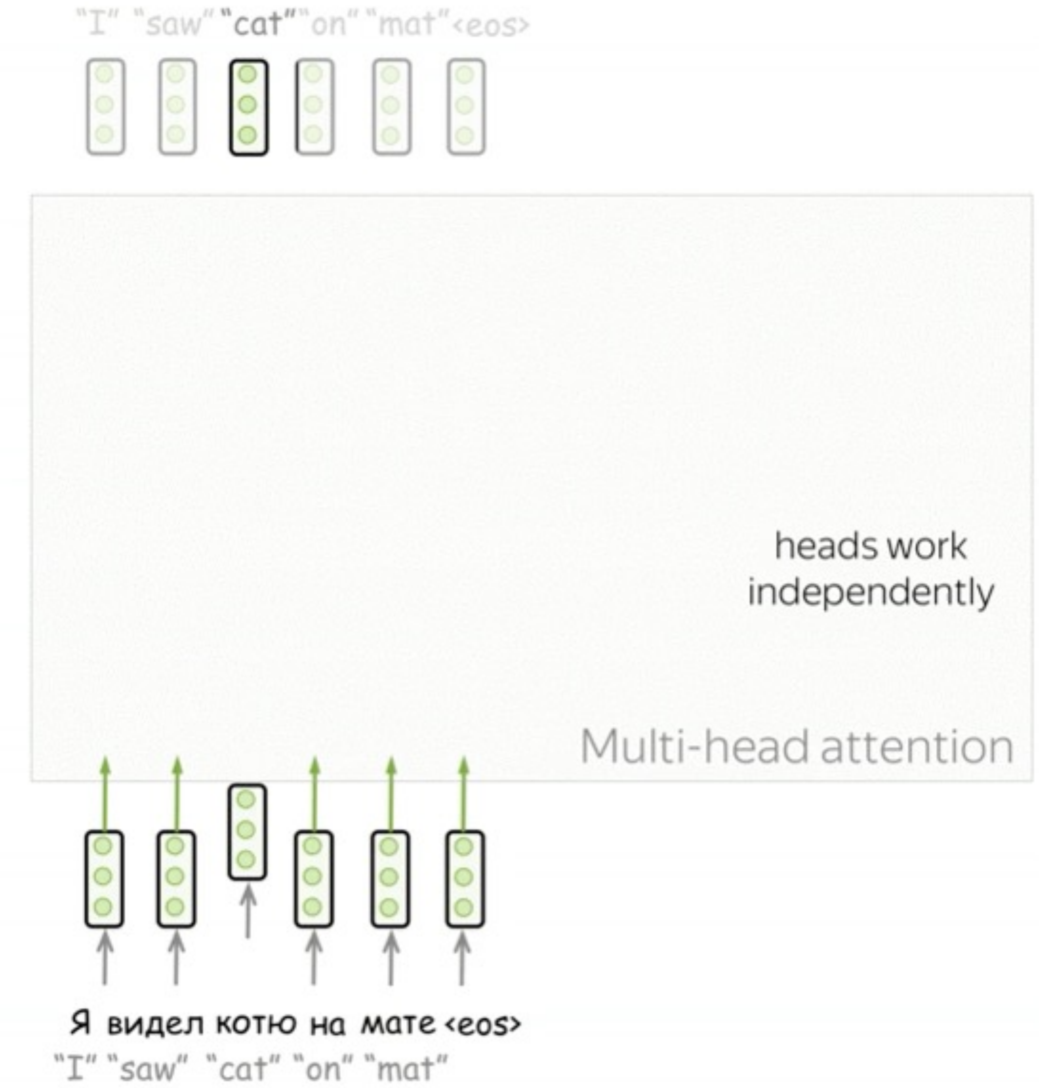


Cross-attention



https://vaclavkosar.com/ml/cross-attention-in-transformer-architecture

Multi-head attention



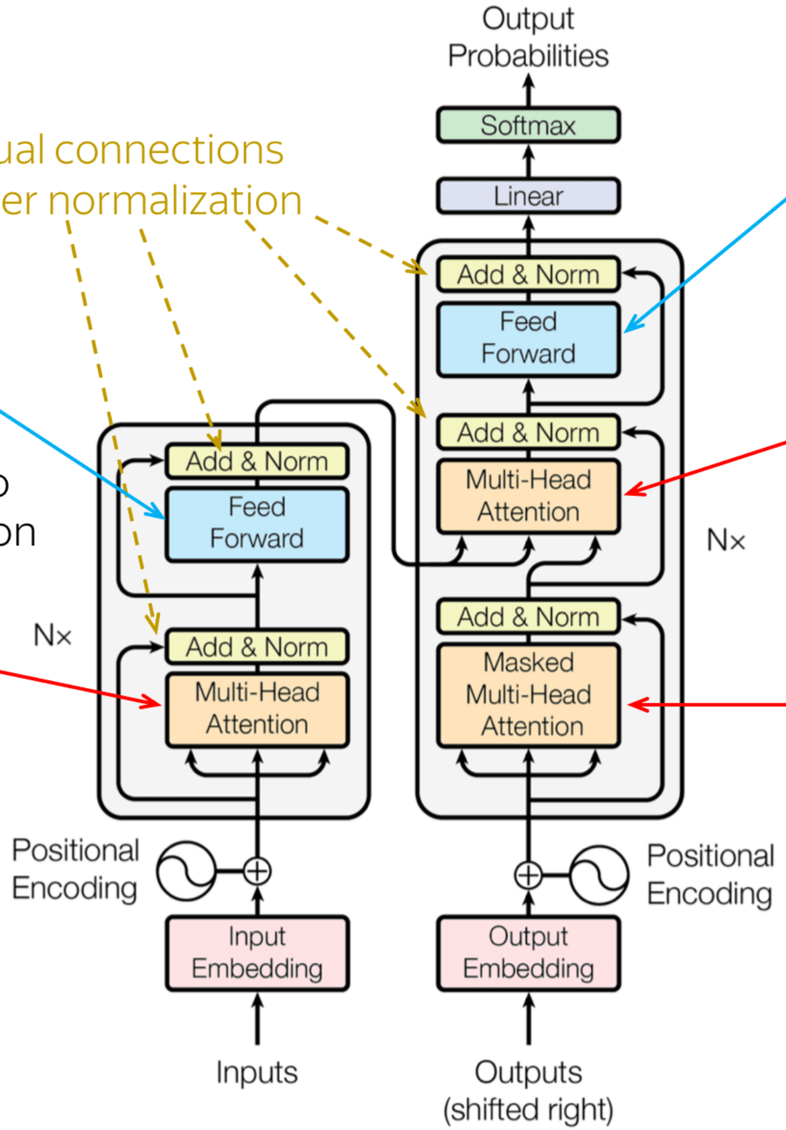
Overall architecture

Residual connections
and layer normalization

Feed-forward network:
after taking information from
other tokens, take a moment to
think and process this information

Encoder self-attention:
tokens look at each other

queries, keys, values
are computed from
encoder states



Feed-forward network:
after taking information from
other tokens, take a moment to
think and process this information

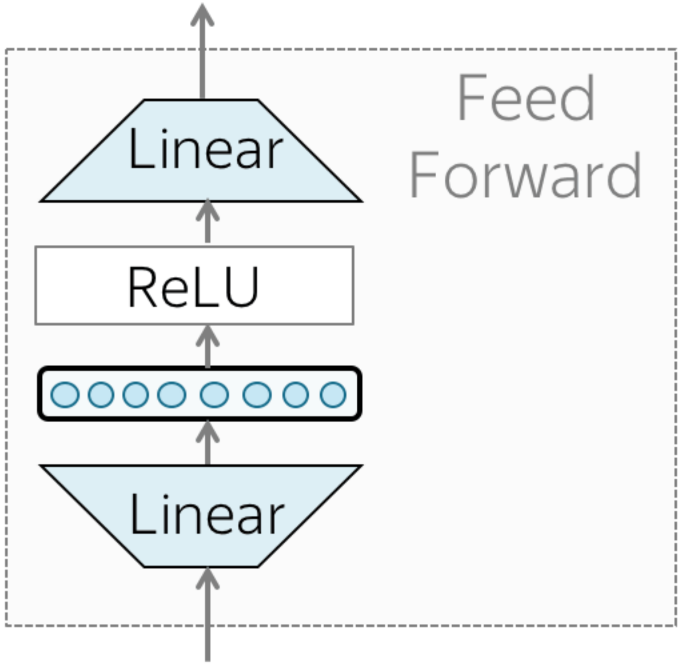
Decoder-encoder attention:
target token looks at the source

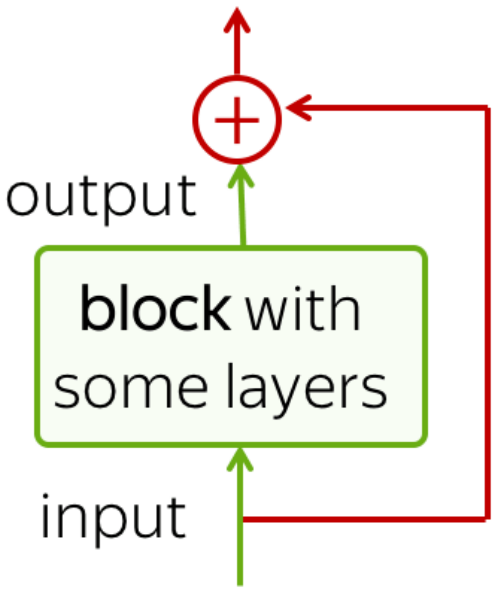
queries – from decoder states; keys
and values from encoder states

Decoder self-attention (masked):
tokens look at the previous tokens

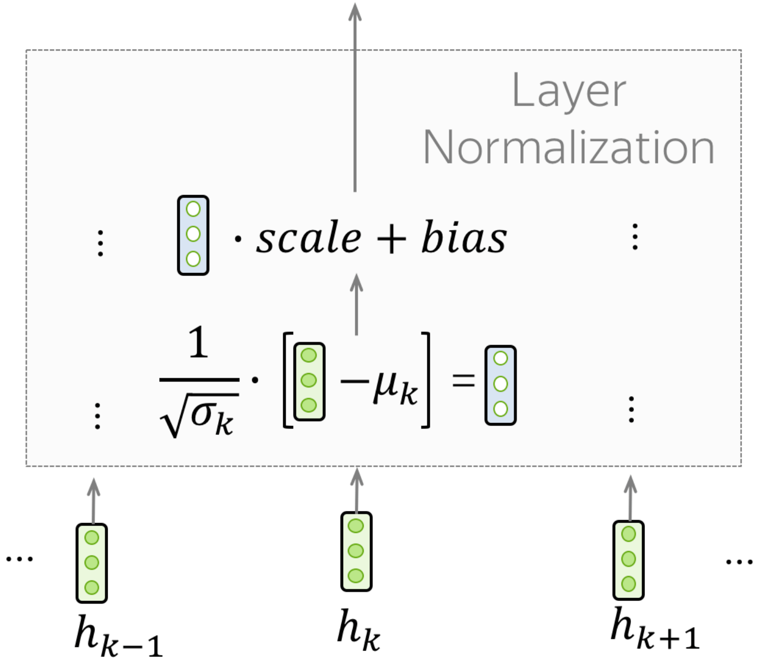
queries, keys, values are computed
from decoder states

Blocks





Residual connection: add a block's input to its output



Permutations

- (encoder)
- Shuffling order of input tokens shuffles representations!
- Attention is permutation “invariant” to order

Position encoding / embedding