

Seq2seq and Attention

OCTOBER 3, 2019

Lena Voita

 lena-voita@yandex-team.ru

 <https://lena-voita.github.io>

 @lena_voita

 lena-voita

Plan

- Machine translation task
- Encoder-decoder framework
- Decoding
- Bahdanau attention model
- Attention is all you need: Transformer
- What do encoder self-attention heads learn?
- Attention: other use cases
- Hack of the day!

Machine translation task (seq2seq)

$\mathbf{x} = (x_1, x_2, \dots, x_{T_x})$ - source sentence

$\mathbf{y} = (y_1, y_2, \dots, y_{T_y})$ - target sentence

Any type of machine translation system can be defined as a function

$$\hat{\mathbf{y}} = \text{mt}(\mathbf{x})$$

Translation is equivalent to finding a target sentence that maximizes the conditional probability of \mathbf{y} given a source sentence \mathbf{x} , i.e.

$$\arg \max_y p(y|x)$$

Machine translation task (seq2seq)

Machine translation systems create a probabilistic model for the probability of y given x ,

$$p(y|x, \theta),$$

and find the target sentence that maximizes this probability:

$$\hat{y} = \arg \max_y p(y|x, \theta).$$

(θ – the parameters of the model specifying the probability distribution)

Machine translation task (seq2seq)

- Modeling

What the model $p(y|x, \theta)$ will look like?

- Learning

We need a method to learn appropriate values for parameters θ from training data.

- Search

We need to solve the problem of finding the most probable sentence (solving “argmax”)

Conditional Language Model

Seq2seq directly models probability $P(y|x)$:

$$P(y|x) = P(y_1|x) P(y_2|y_1, x) P(y_3|y_1, y_2, x) \dots, P(y_T|y_1, \dots, y_{T-1}, x)$$

Conditional Language Model

Seq2seq directly models probability $P(y|x)$:

$$P(y|x) = P(y_1|x) P(y_2|y_1, x) P(y_3|y_1, y_2, x) \dots, P(y_T|y_1, \dots, y_{T-1}, x)$$



Probability of next target words so
far and source sentence x

Conditional Language Model

Seq2seq directly models probability $P(y|x)$:

$$P(y|x) = P(y_1|x) P(y_2|y_1, x) P(y_3|y_1, y_2, x) \dots, P(y_T|y_1, \dots, y_{T-1}, x)$$

Why conditional language model?



Probability of next target words so far and source sentence x

Conditional Language Model

Seq2seq directly models probability $P(y|x)$:

$$P(y|x) = P(y_1|x) \underbrace{P(y_2|y_1, x)}_{\text{Probability of next target words so far and source sentence } x} P(y_3|y_1, y_2, x) \dots, P(y_T|y_1, \dots, y_{T-1}, x)$$

Why conditional language model?

- LM – predicts next word
- Conditional – predictions conditioned on source sentence

Encoder-Decoder Framework

RECAP: RNN Language Models

Output distribution

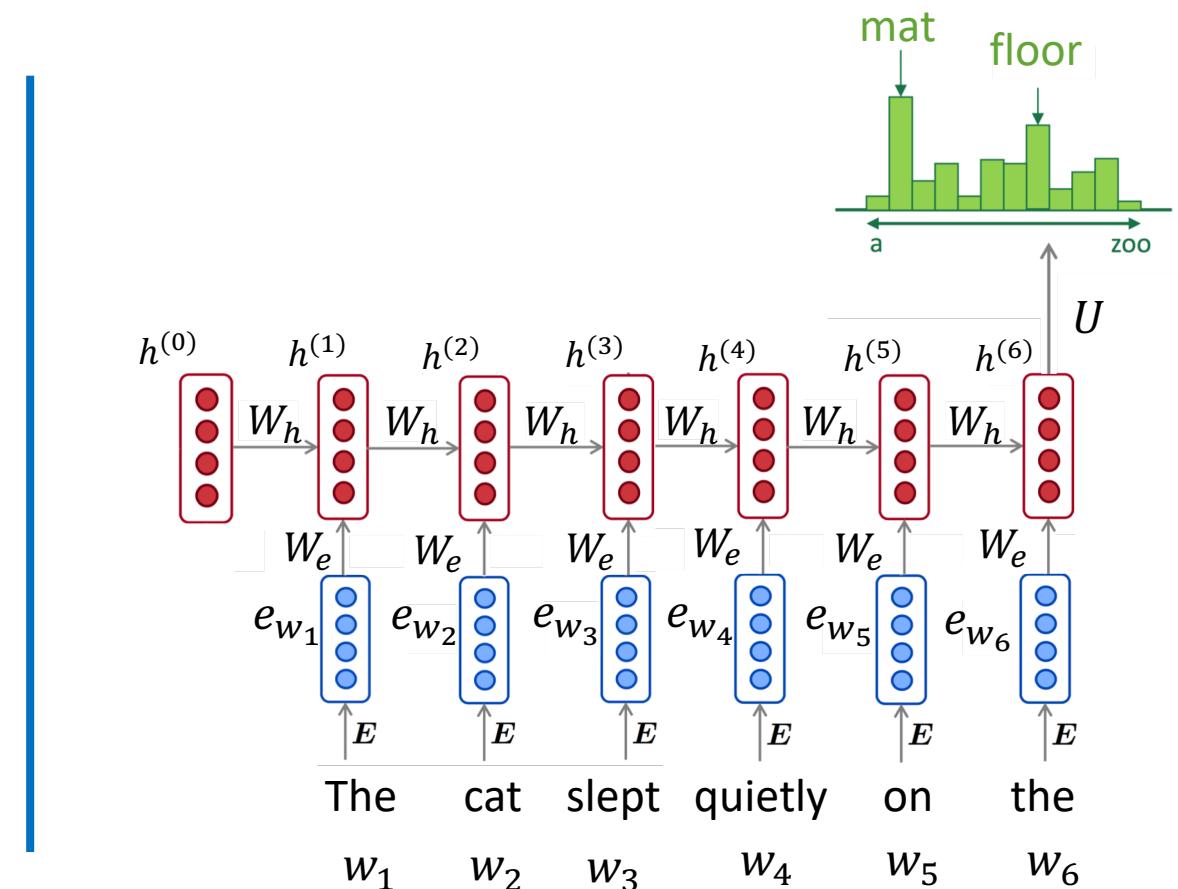
$$\hat{y}_t = \text{softmax}(Uh^{(t)} + b_2) \in \mathbb{R}^{|V|}$$

Hidden states

$$h^{(t)} = \sigma(W_h h^{(t-1)} + W_e e_{w_{t-1}} + b_1)$$

Word embeddings

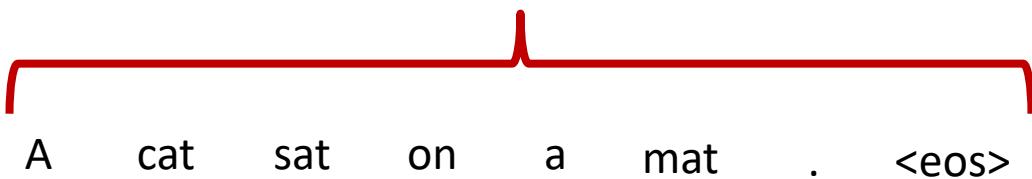
Words/tokens



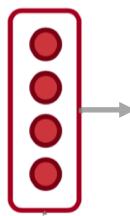
RNN Encoder-Decoder (Vanilla)



Source sentence

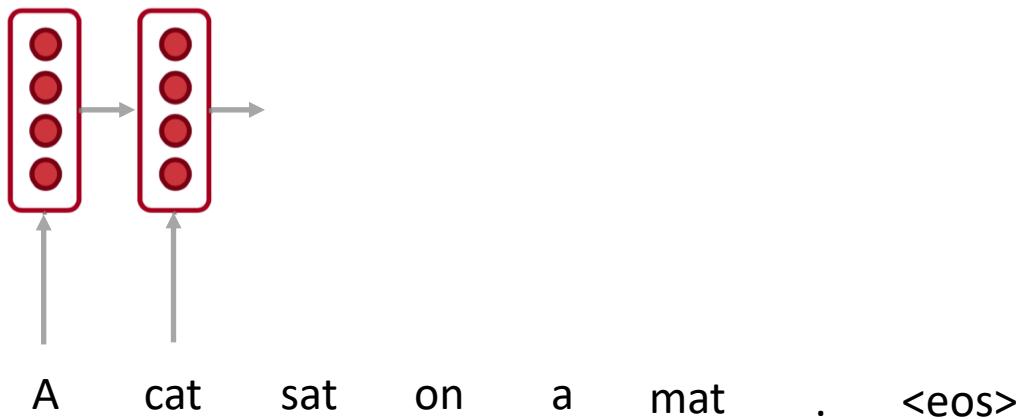


RNN Encoder-Decoder (Vanilla)

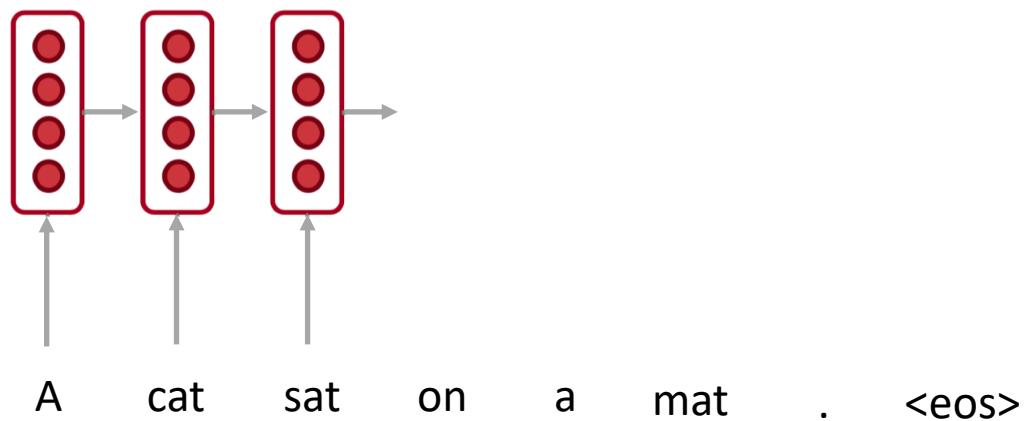


A cat sat on a mat . <eos>

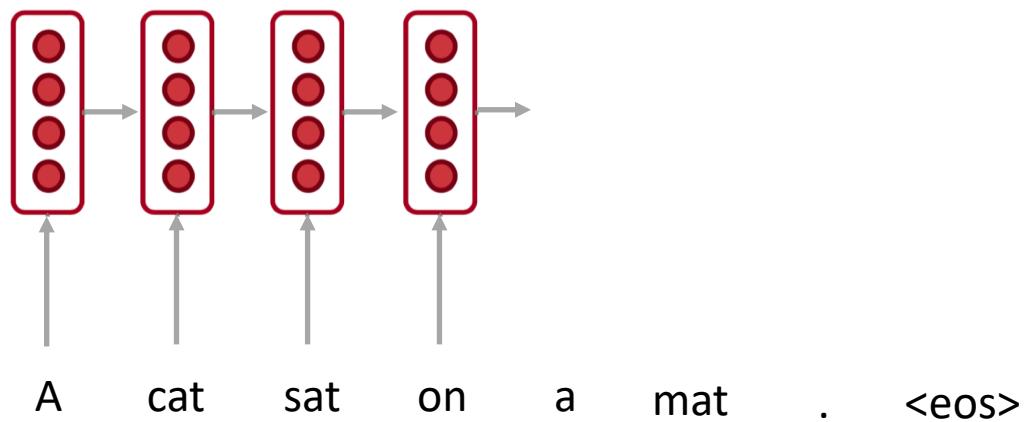
RNN Encoder-Decoder (Vanilla)



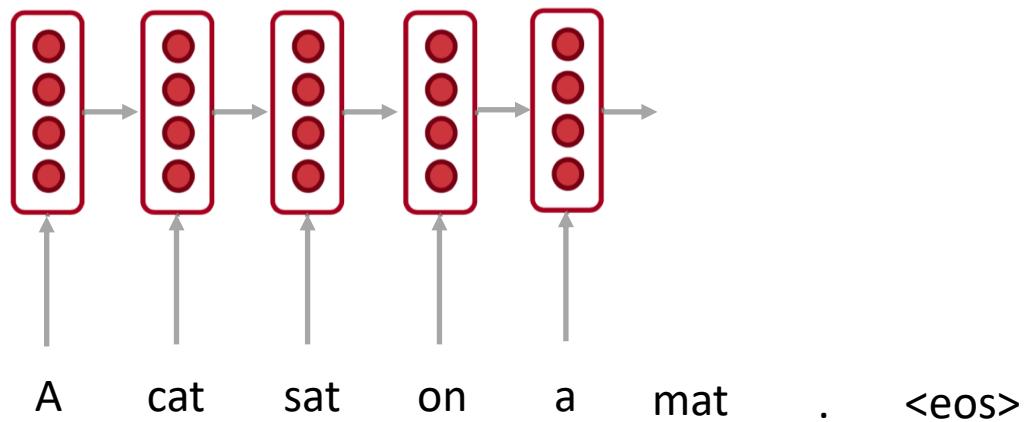
RNN Encoder-Decoder (Vanilla)



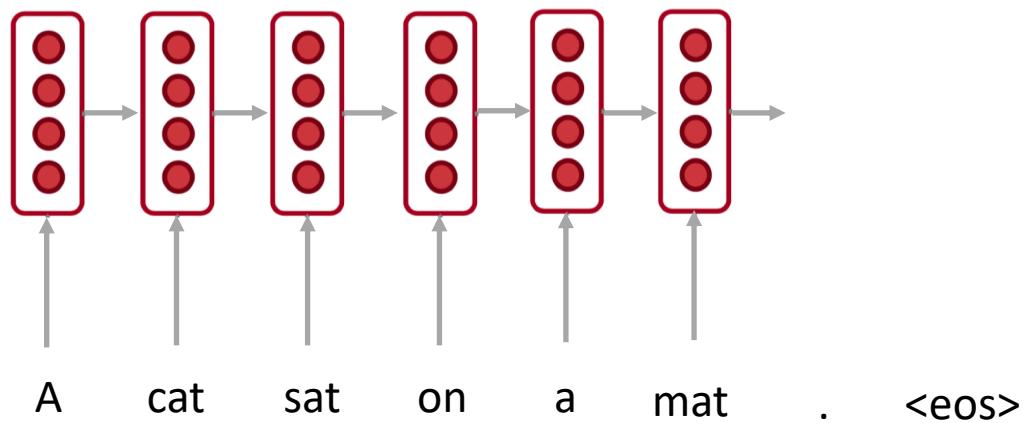
RNN Encoder-Decoder (Vanilla)



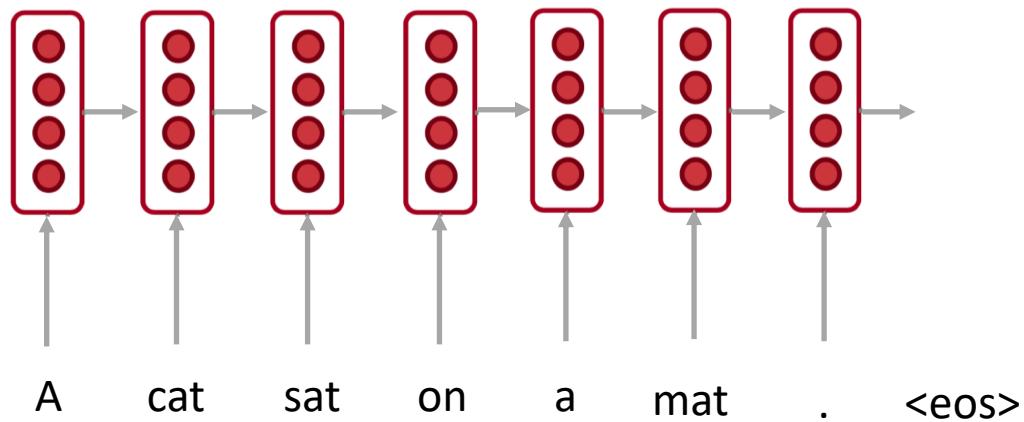
RNN Encoder-Decoder (Vanilla)



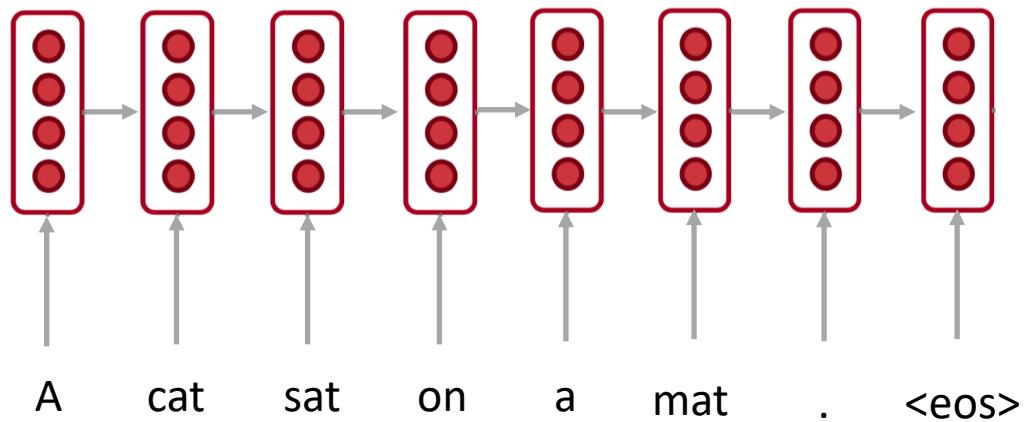
RNN Encoder-Decoder (Vanilla)



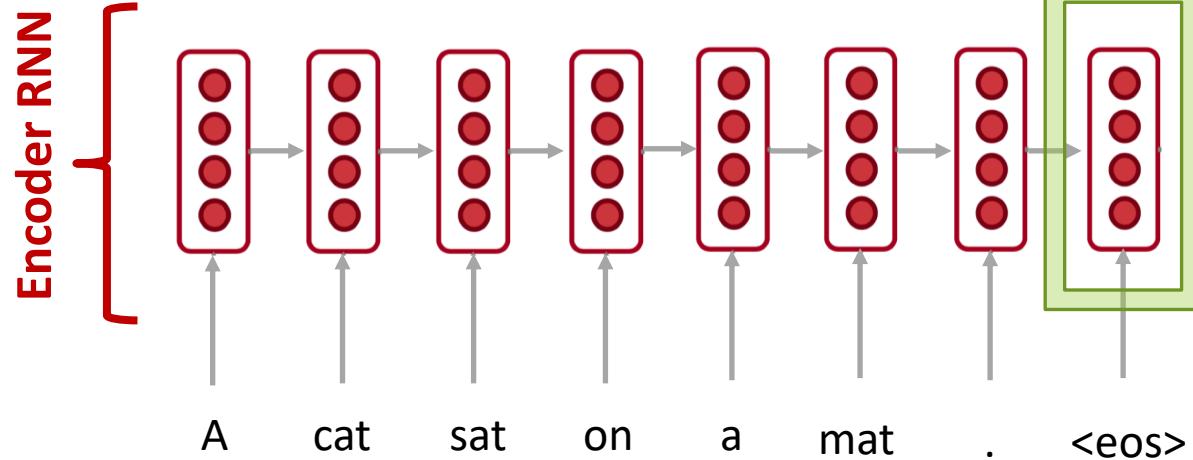
RNN Encoder-Decoder (Vanilla)



RNN Encoder-Decoder (Vanilla)

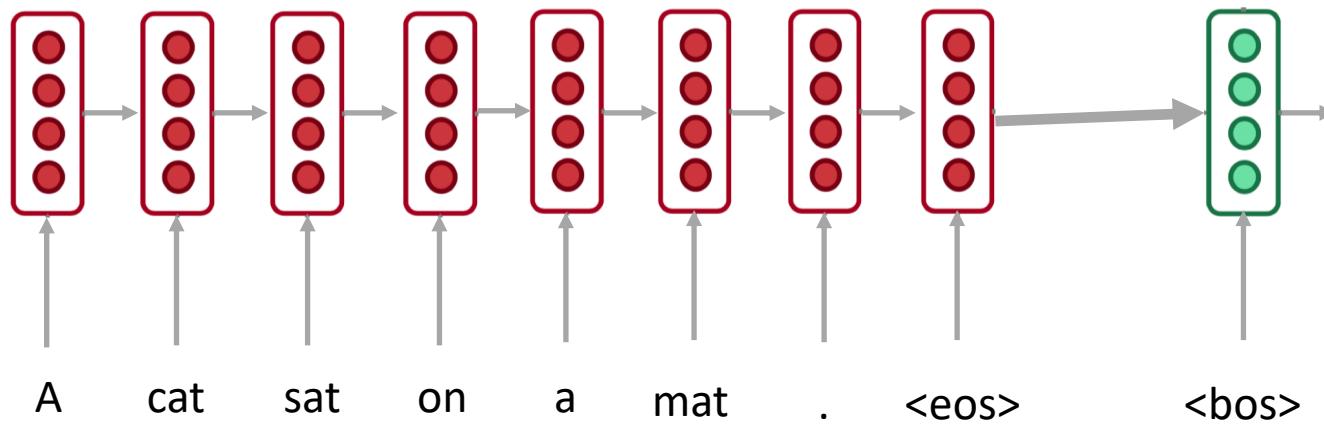


RNN Encoder-Decoder (Vanilla)

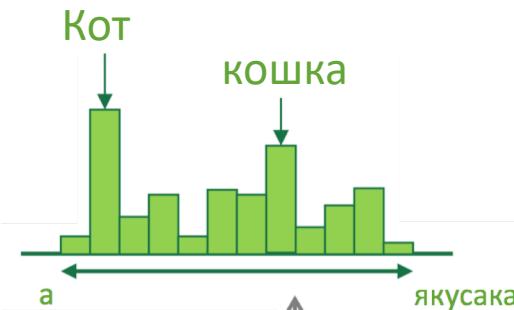
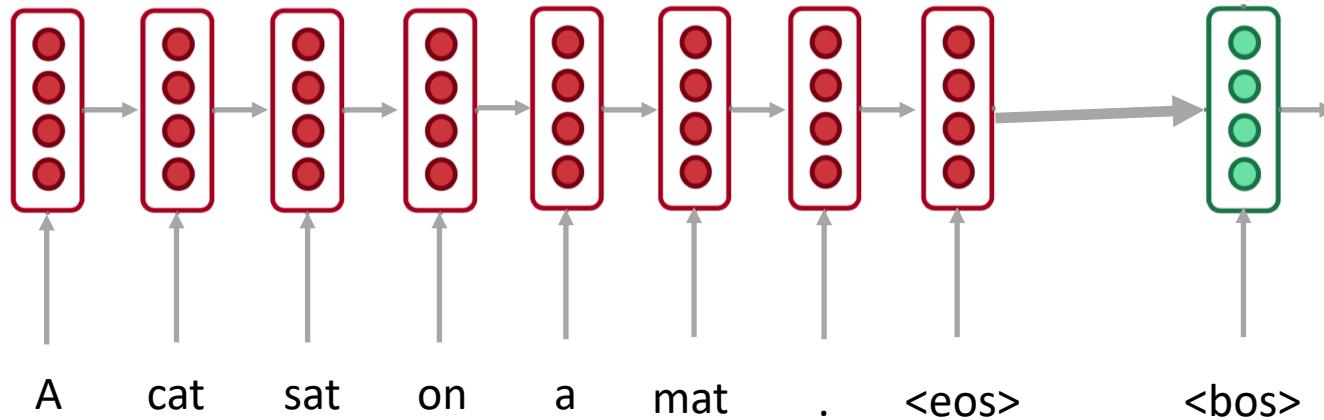


Encoding of a source sentence.
Provides an initial state for decoder RNN

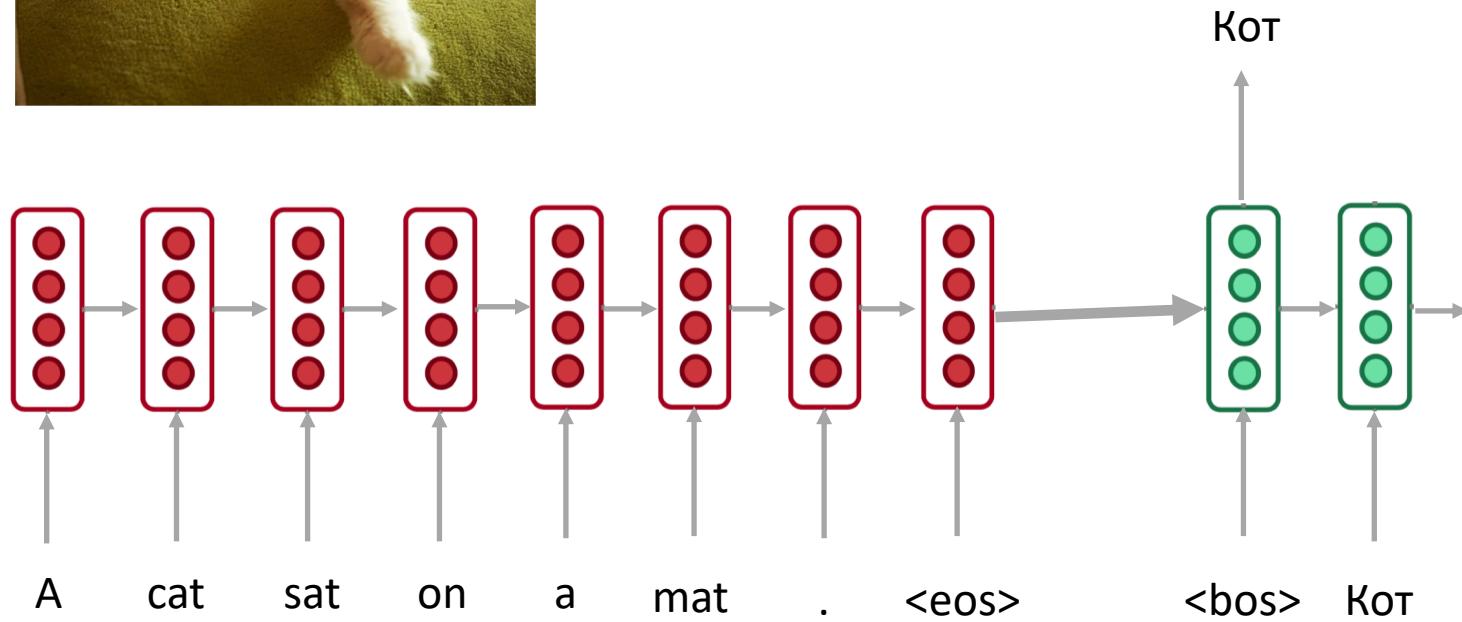
RNN Encoder-Decoder (Vanilla)



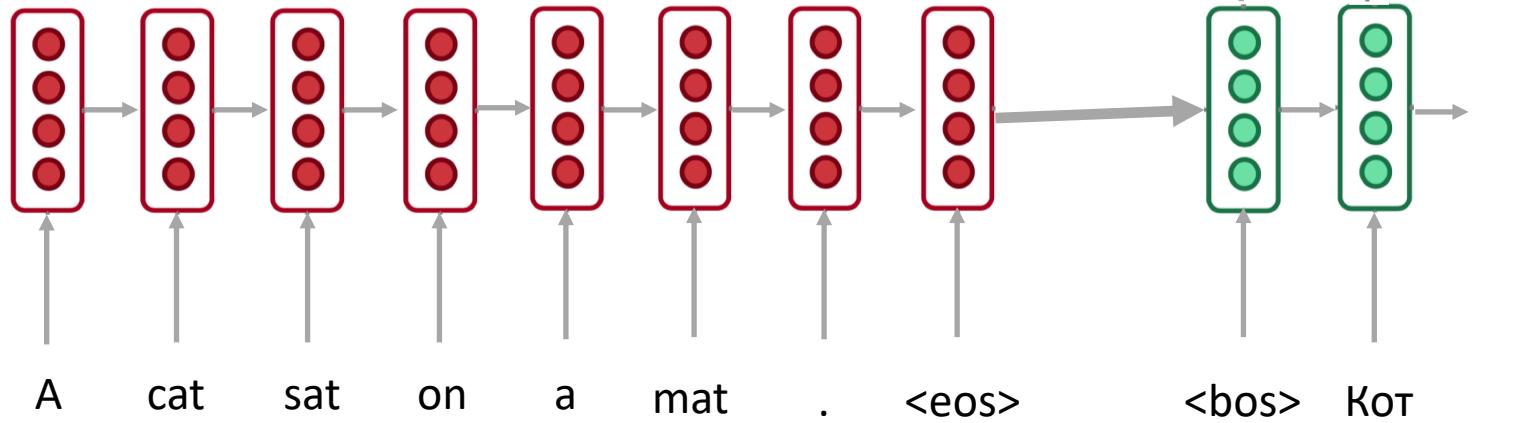
RNN Encoder-Decoder (Vanilla)



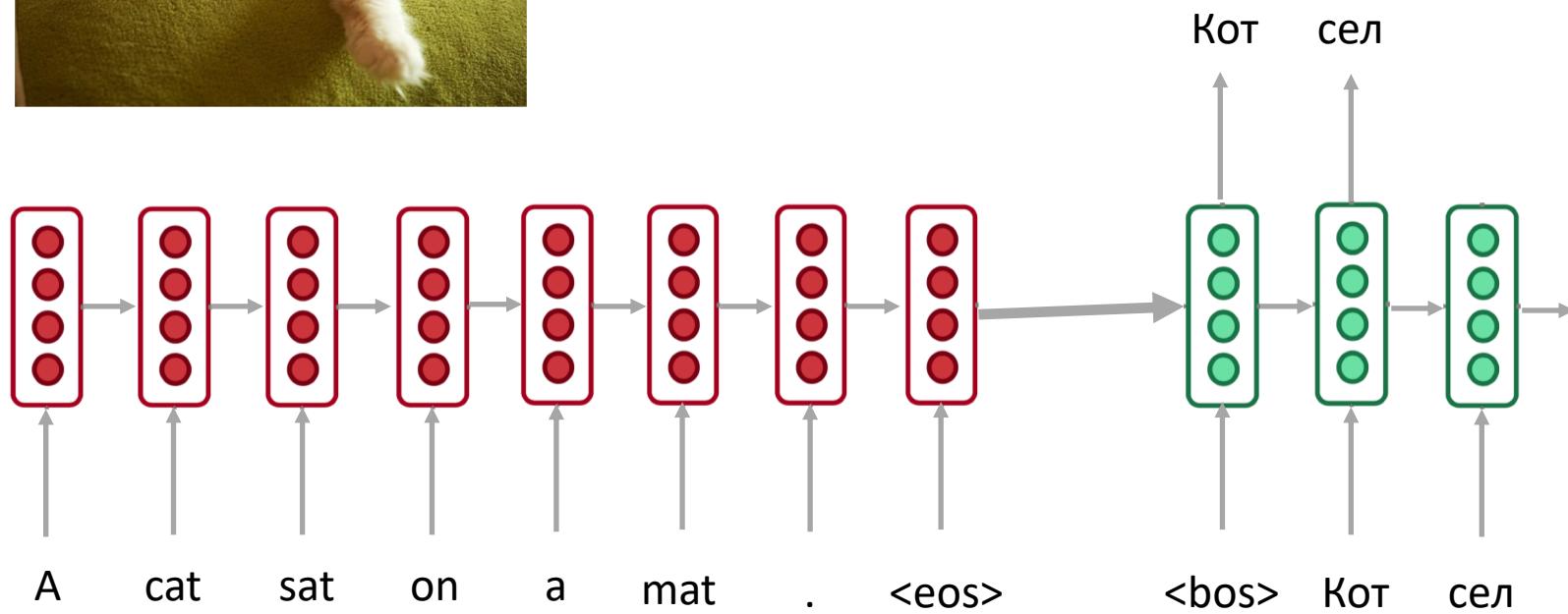
RNN Encoder-Decoder (Vanilla)



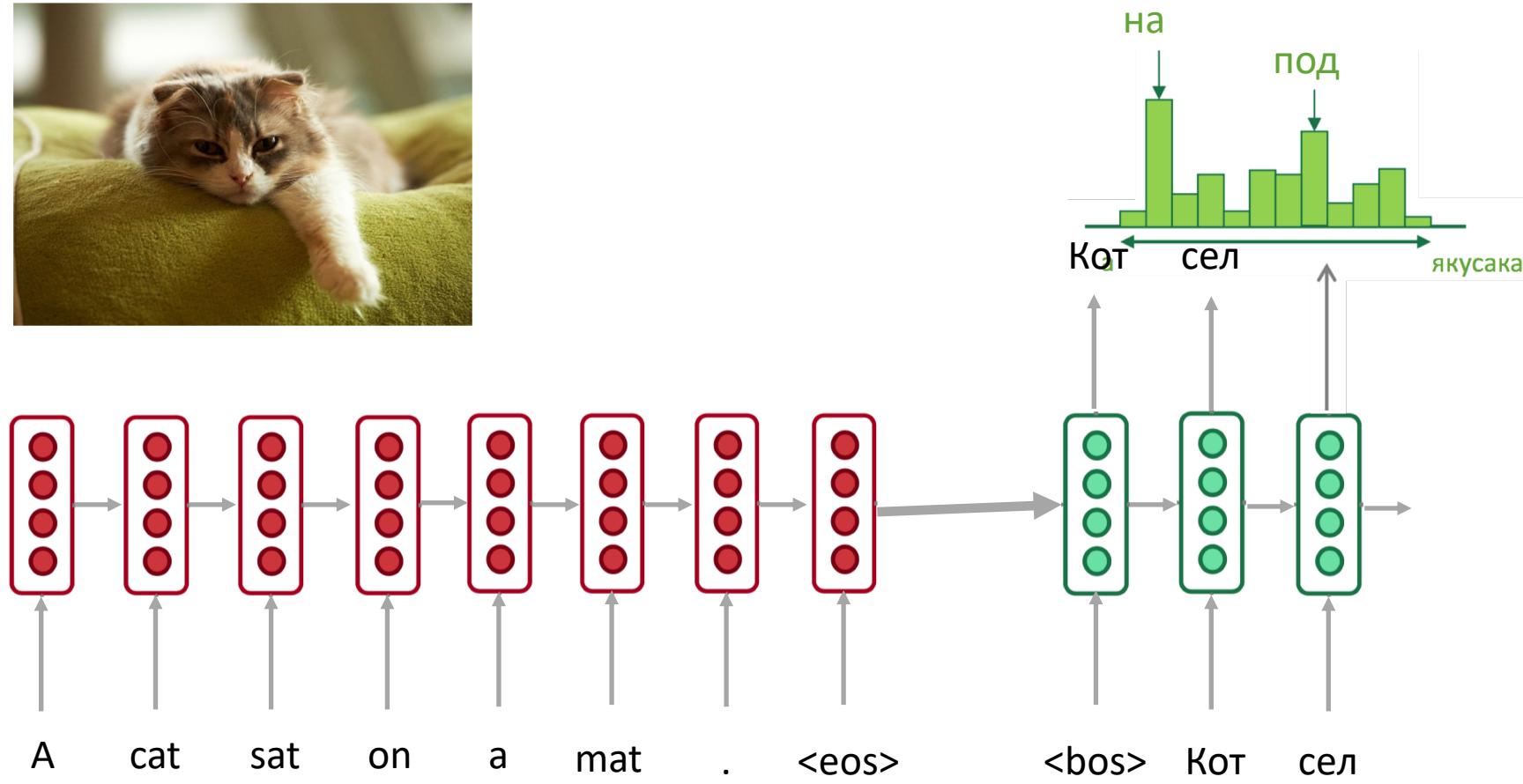
RNN Encoder-Decoder (Vanilla)



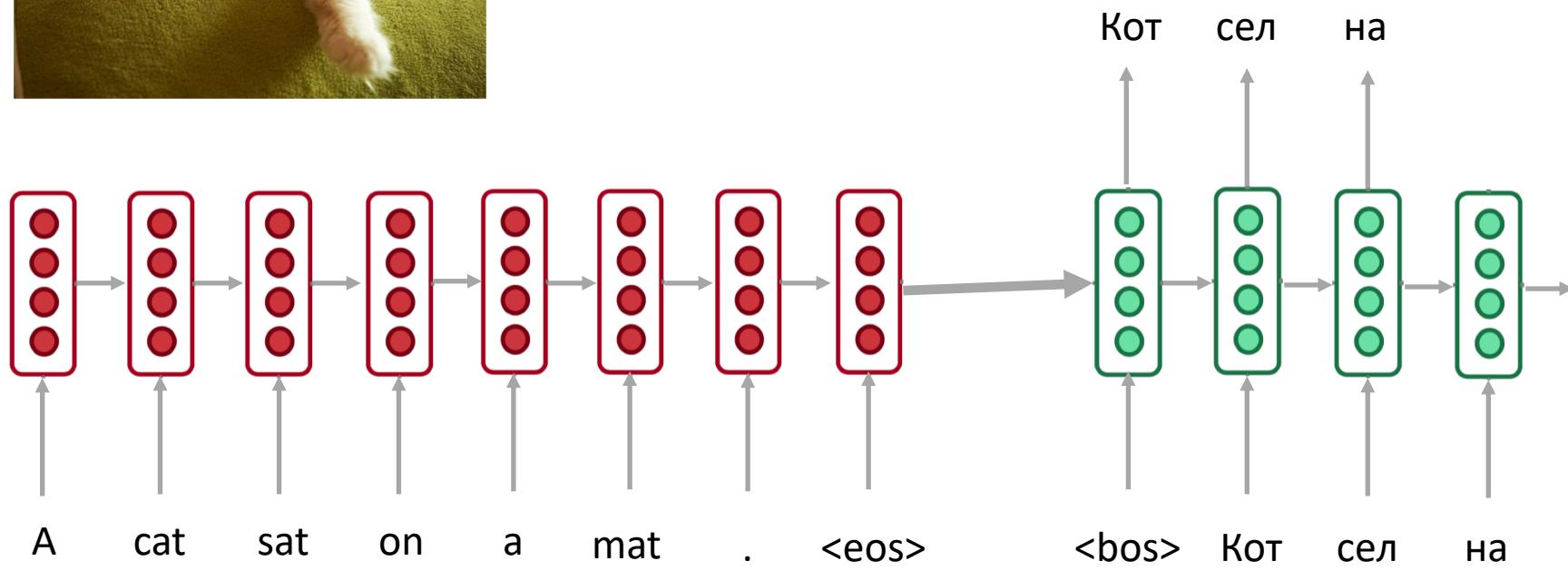
RNN Encoder-Decoder (Vanilla)



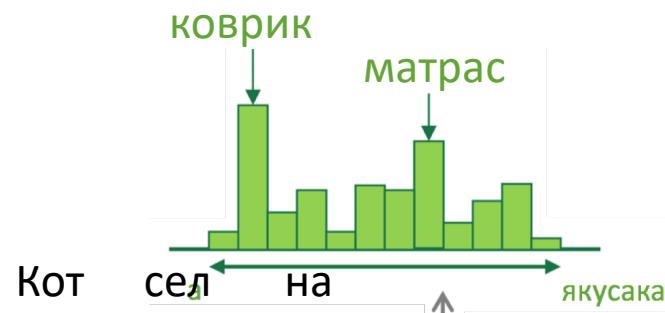
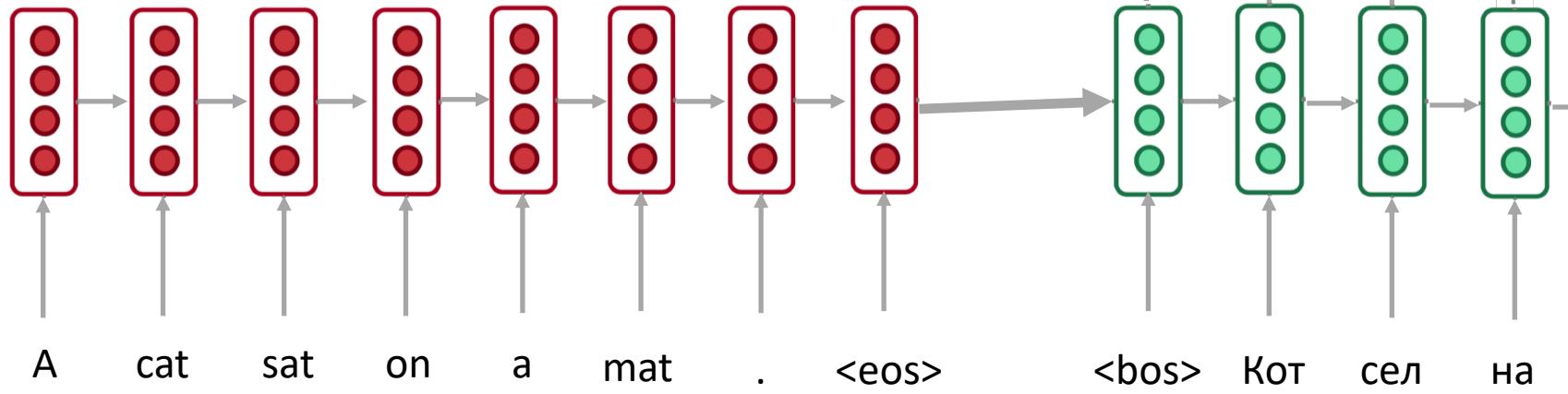
RNN Encoder-Decoder (Vanilla)



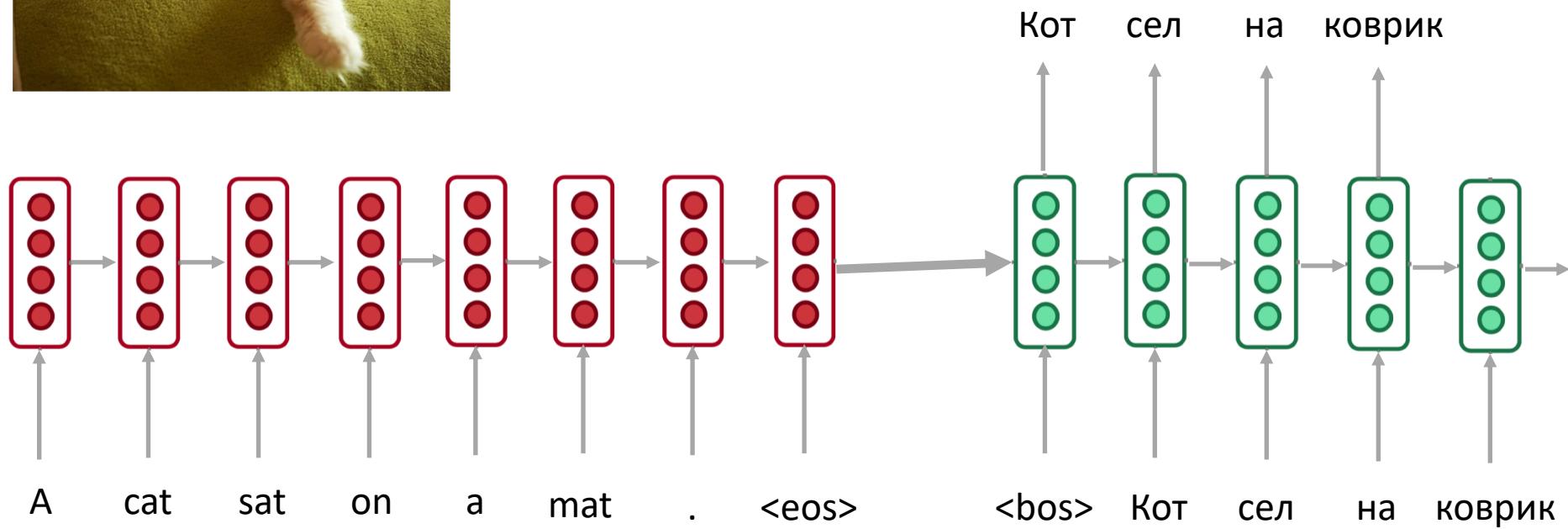
RNN Encoder-Decoder (Vanilla)



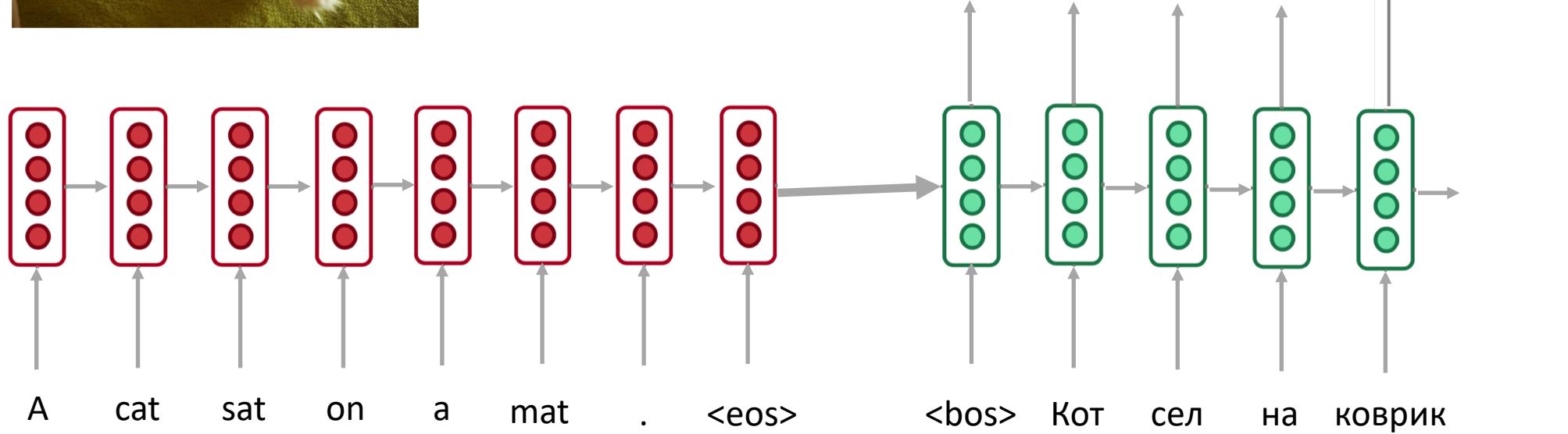
RNN Encoder-Decoder (Vanilla)



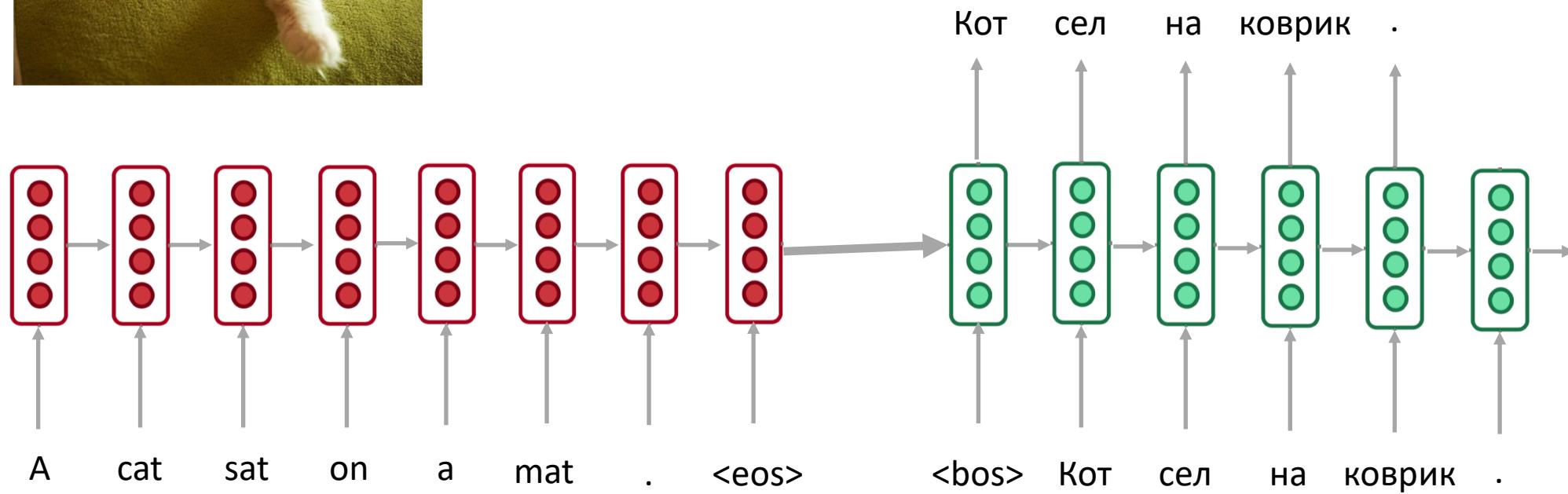
RNN Encoder-Decoder (Vanilla)



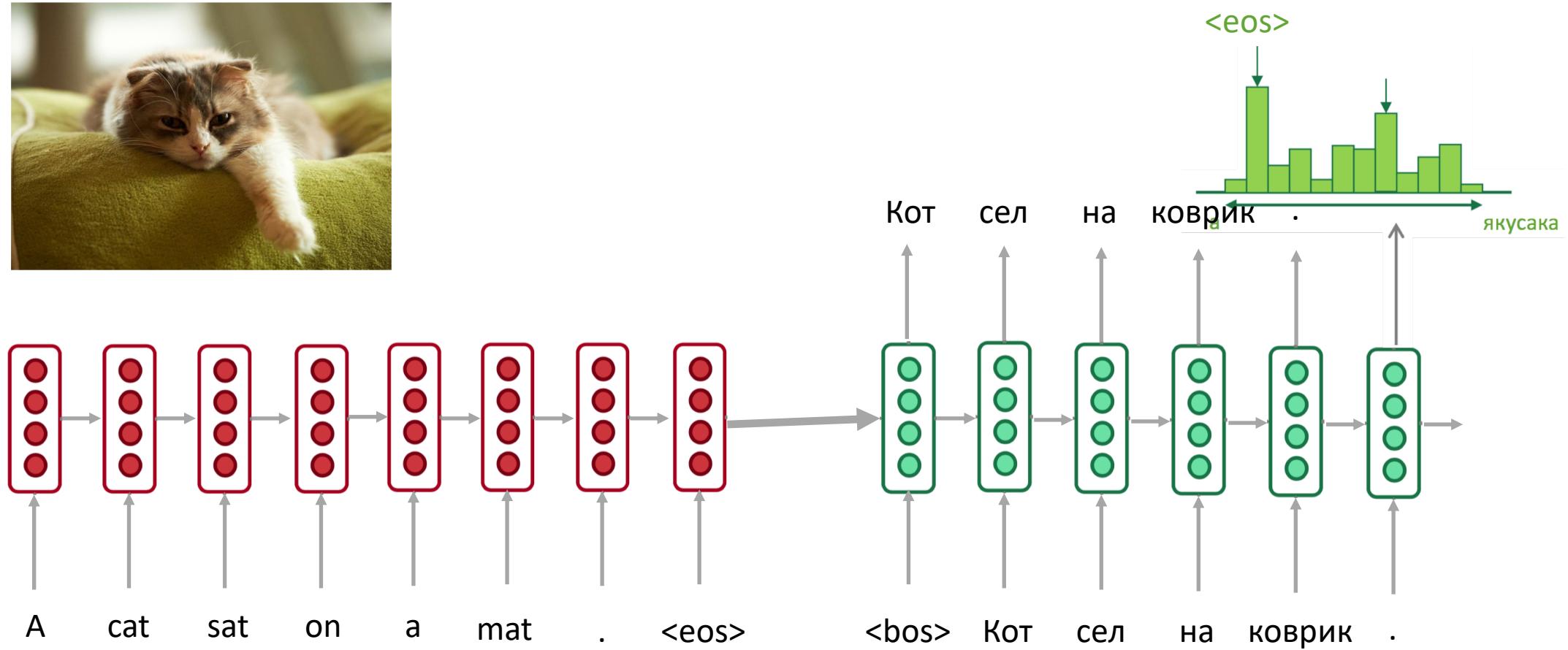
RNN Encoder-Decoder (Vanilla)



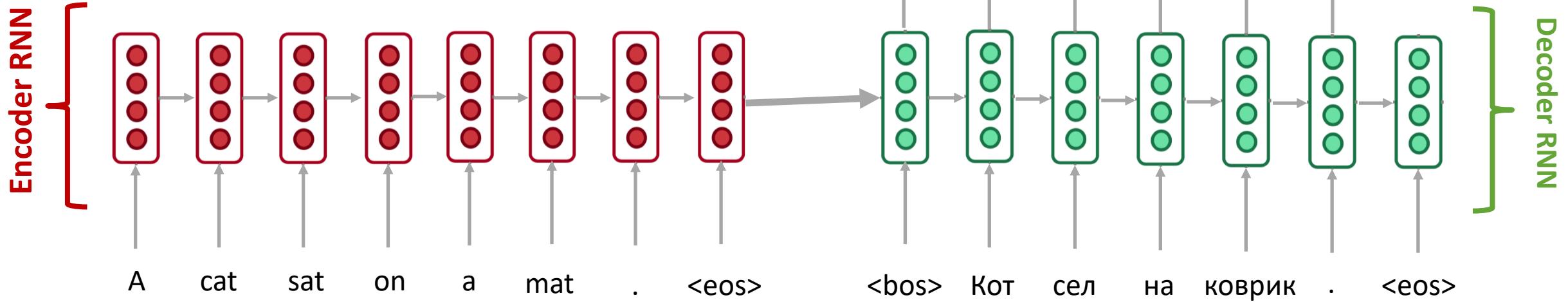
RNN Encoder-Decoder (Vanilla)



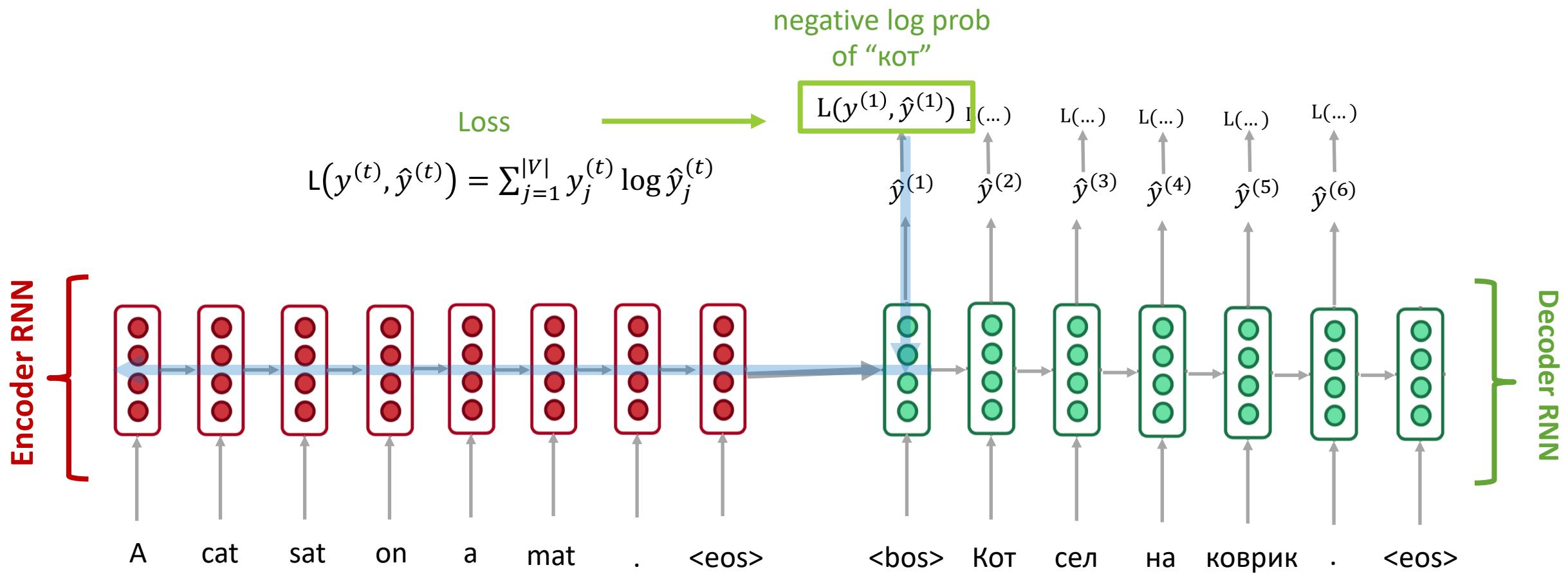
RNN Encoder-Decoder (Vanilla)



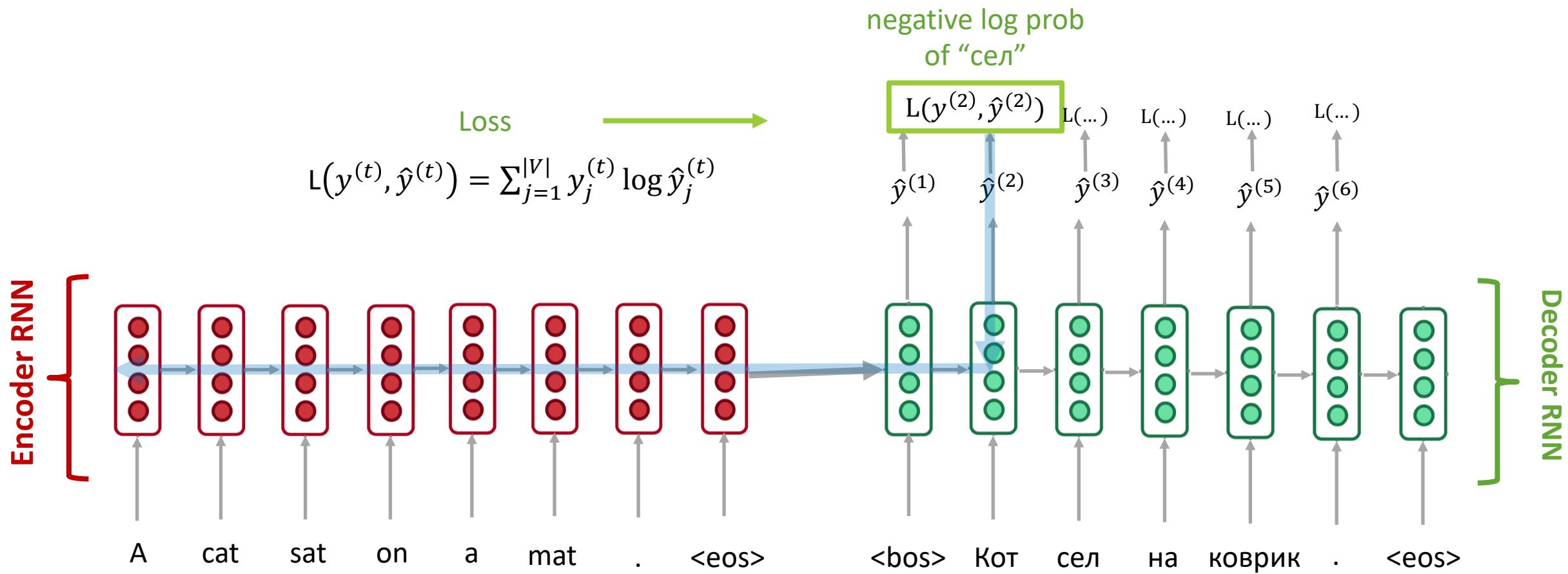
RNN Encoder-Decoder (Vanilla)



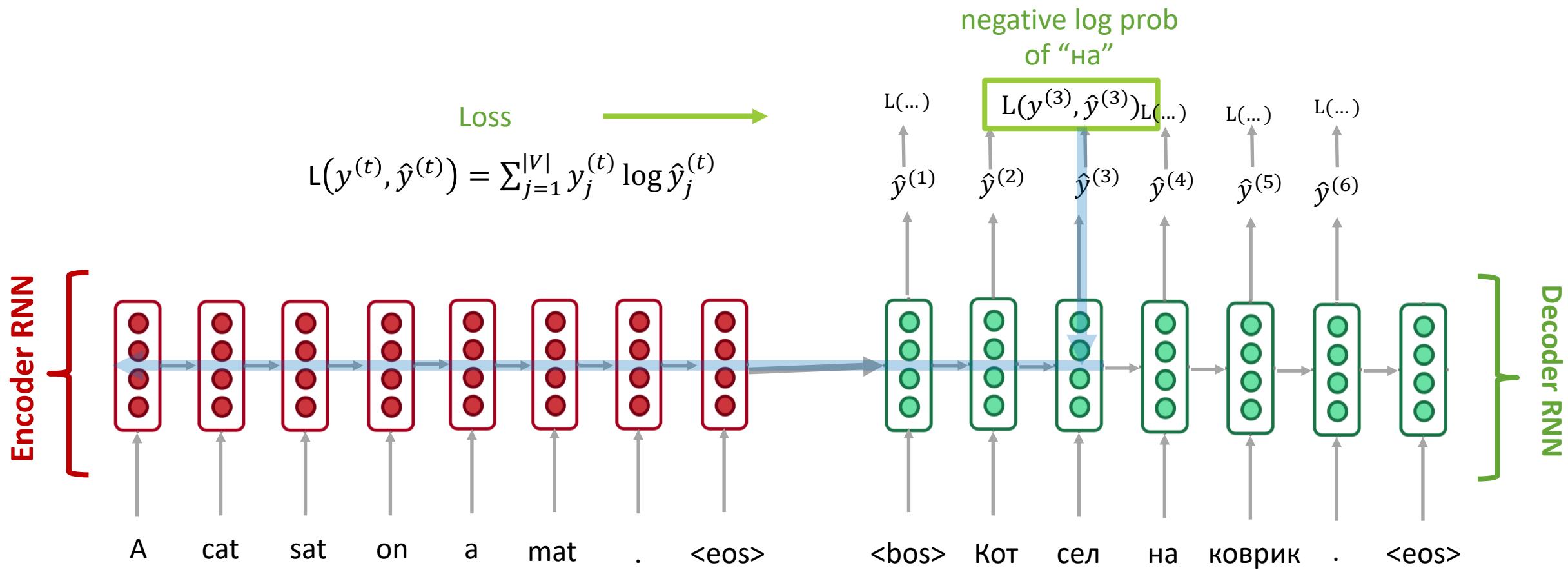
RNN Encoder-Decoder (Vanilla)



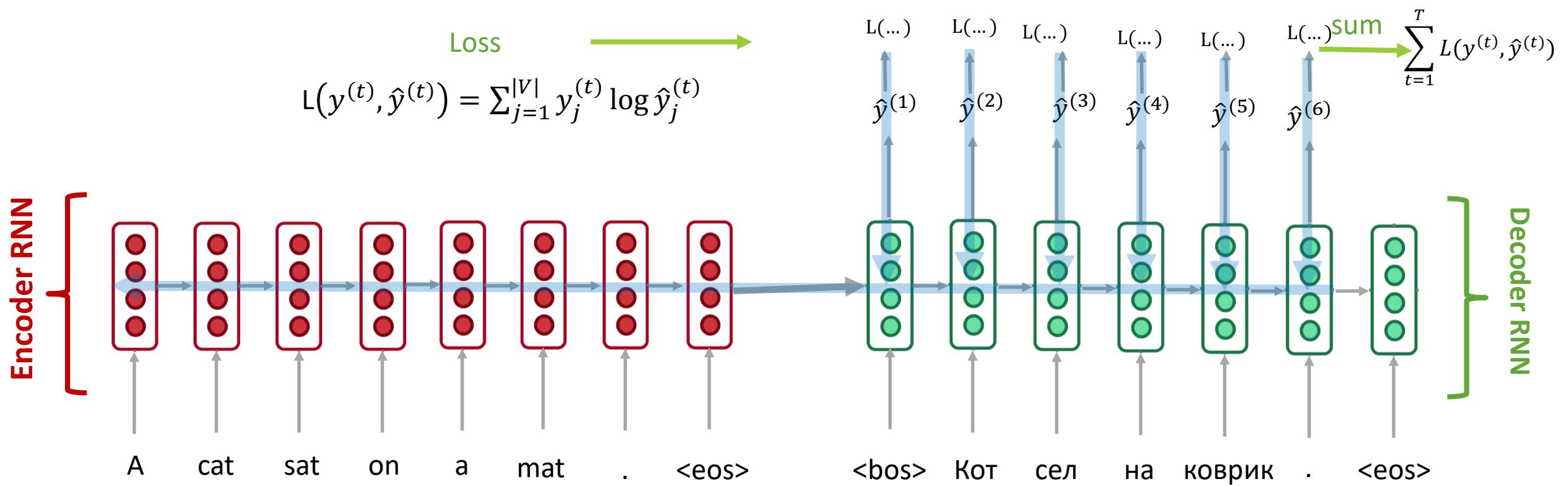
RNN Encoder-Decoder (Vanilla)



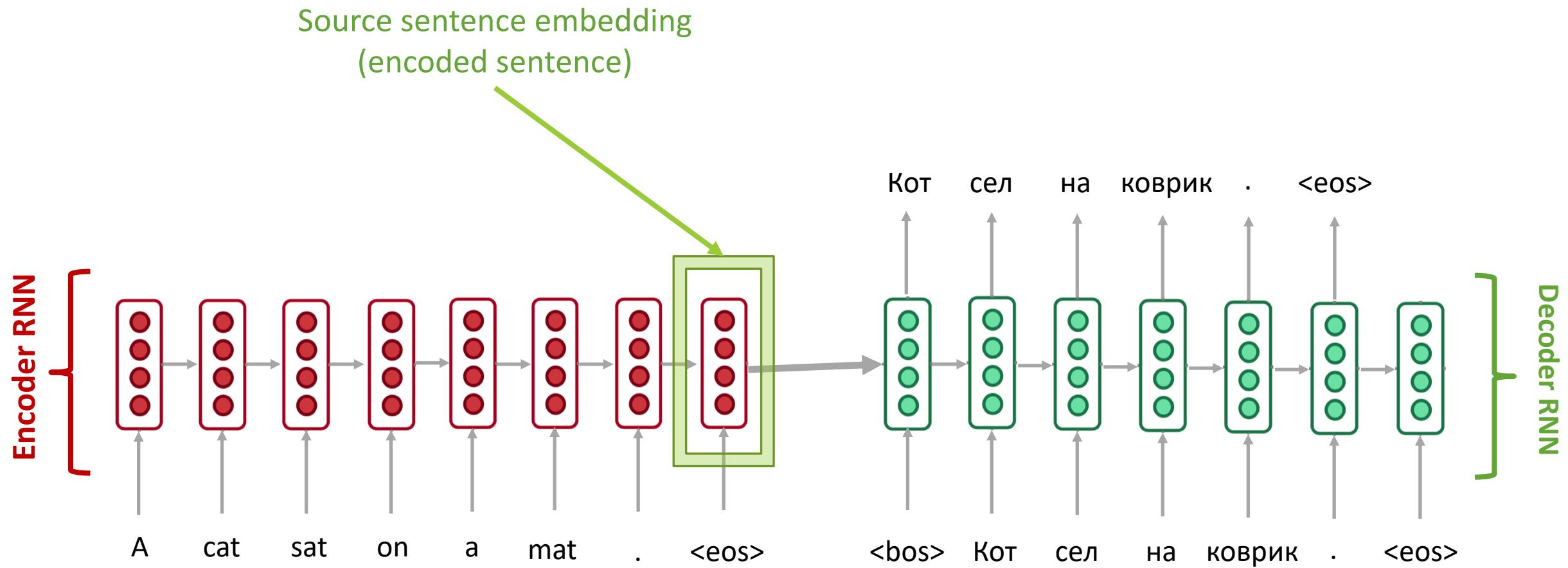
RNN Encoder-Decoder (Vanilla)



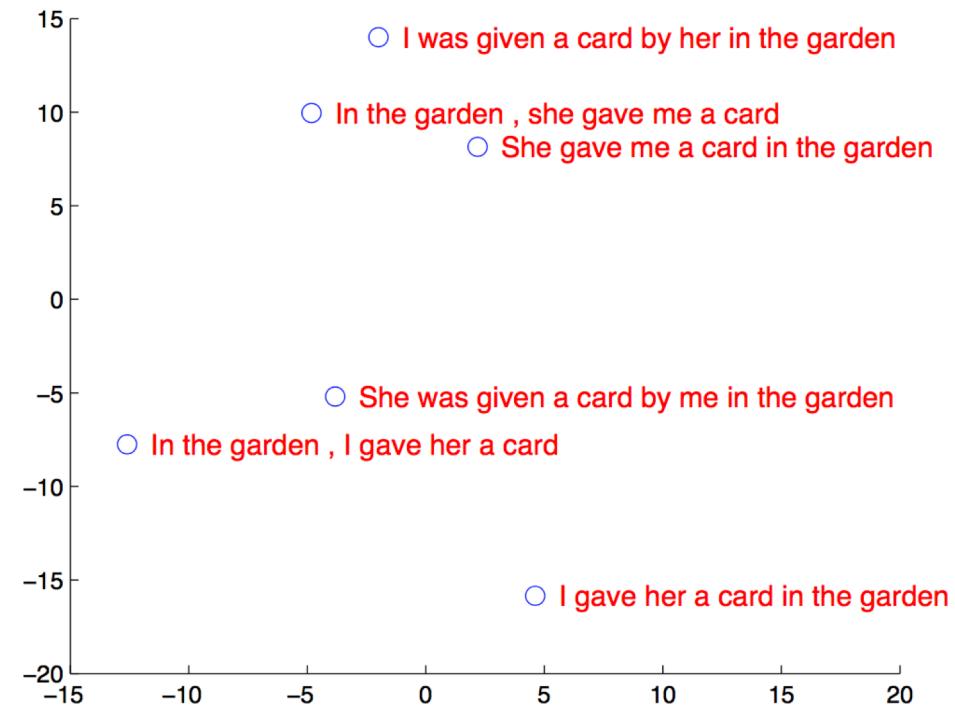
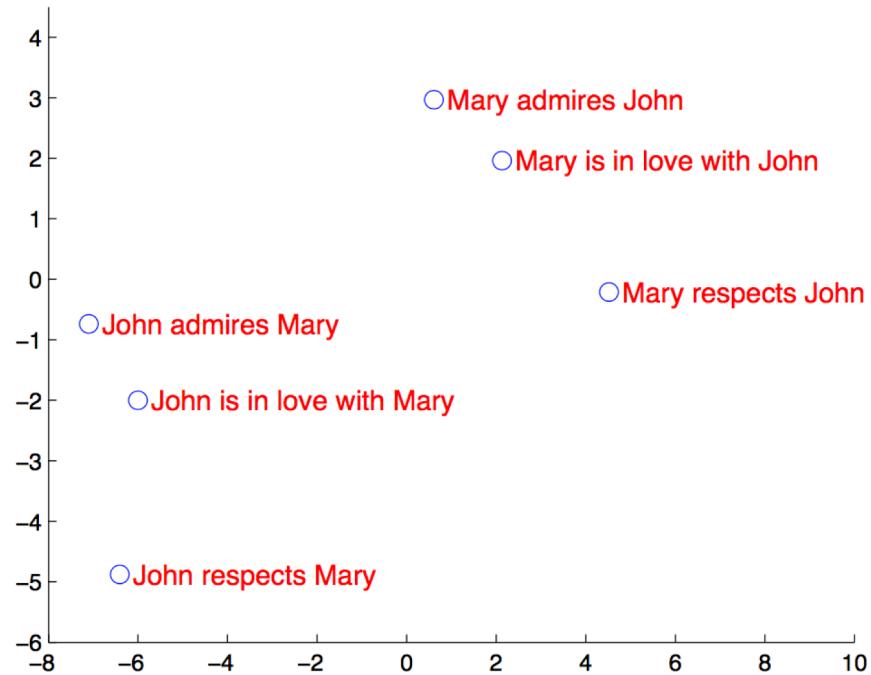
RNN Encoder-Decoder (Vanilla)



Let's look at sentence embedding

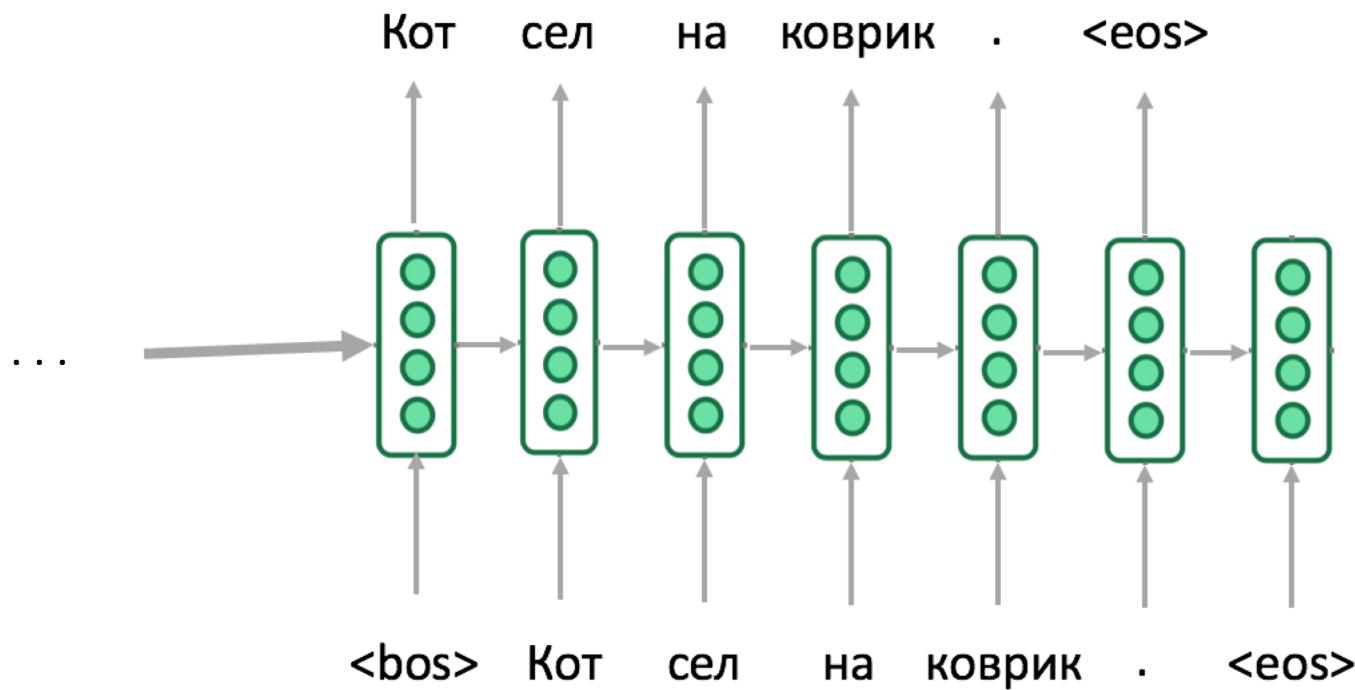


Let's look at sentence embedding



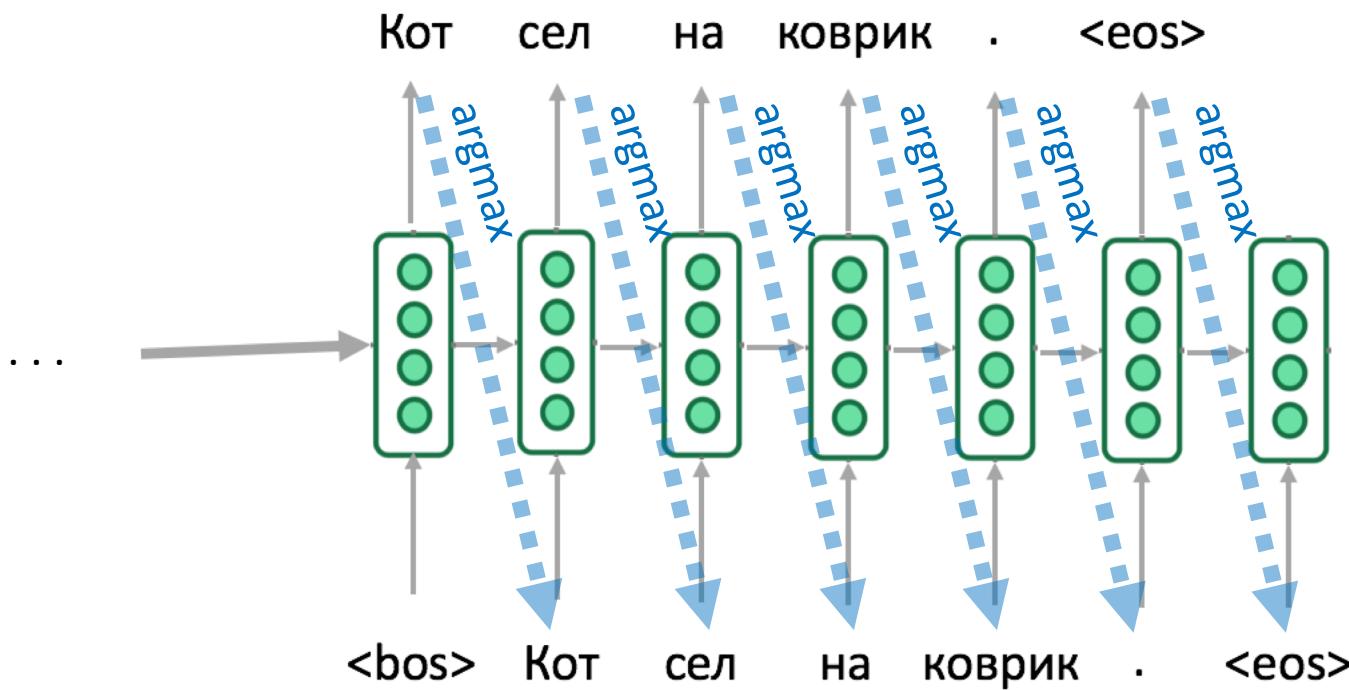
Decoding

Decoding: how to generate a sequence?



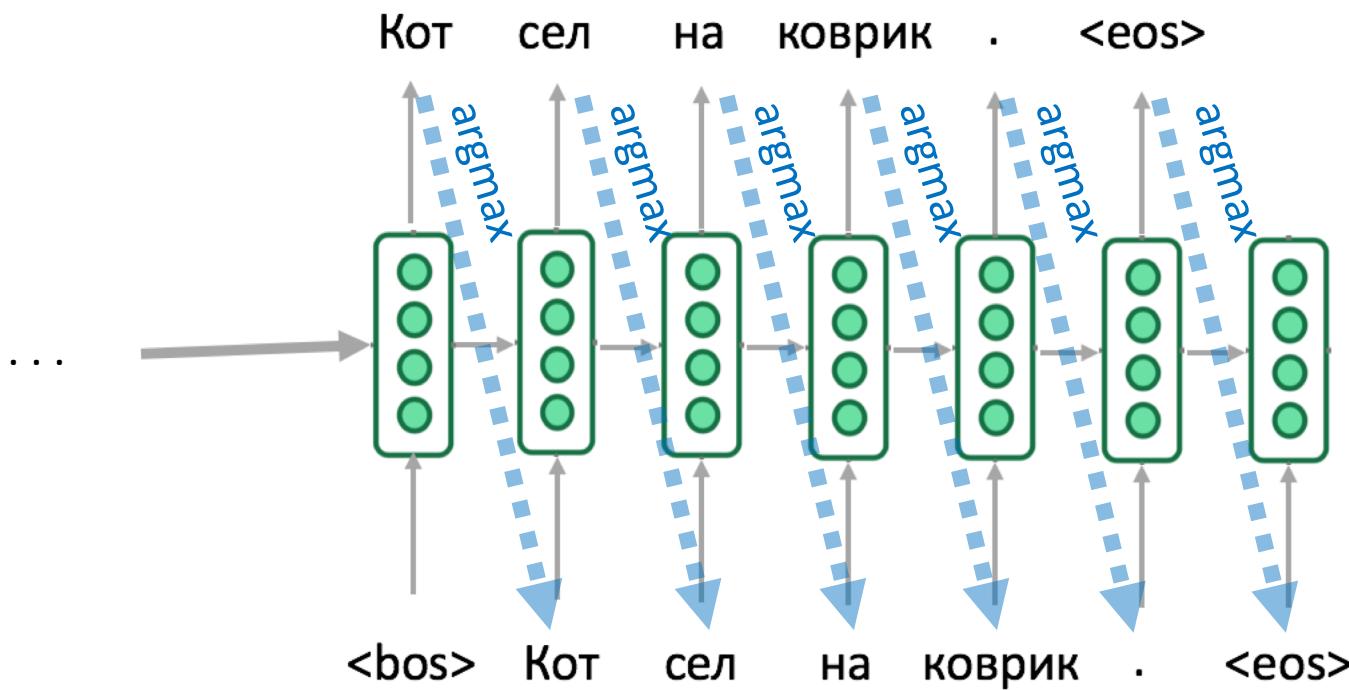
Decoding: how to generate a sequence?

1. Greedy decoding

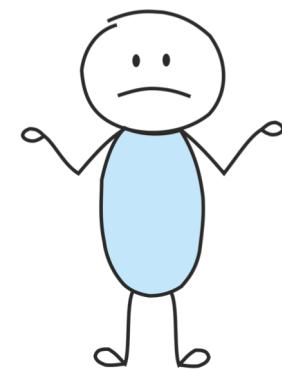


Decoding: how to generate a sequence?

1. Greedy decoding



Is it really good?



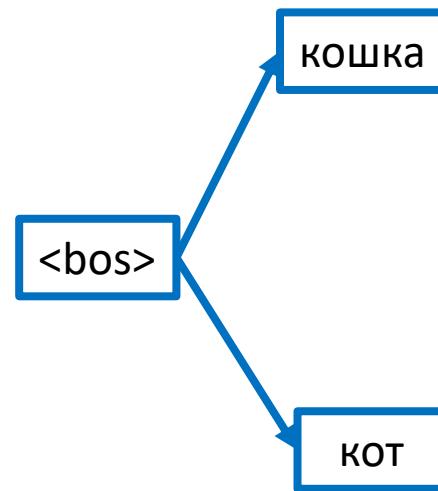
Decoding: how to generate a sequence?

2. Beam search

<bos>

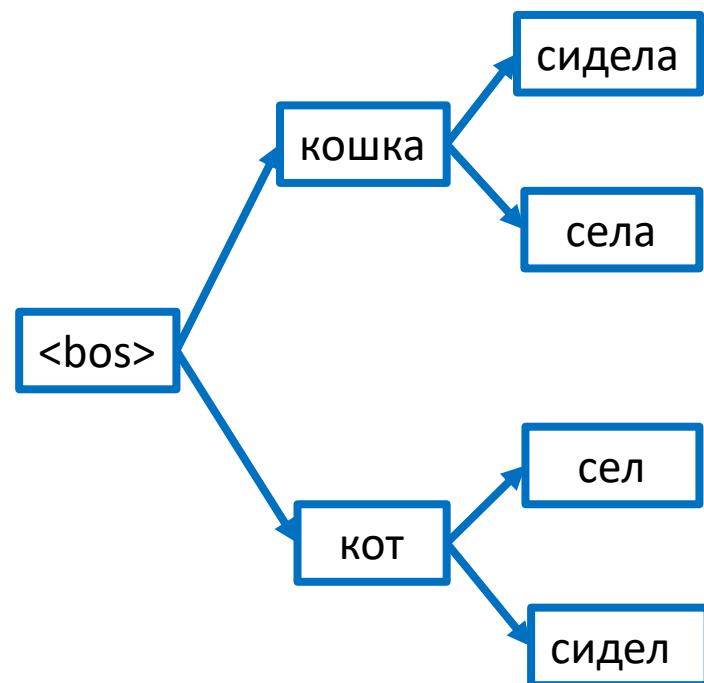
Decoding: how to generate a sequence?

2. Beam search



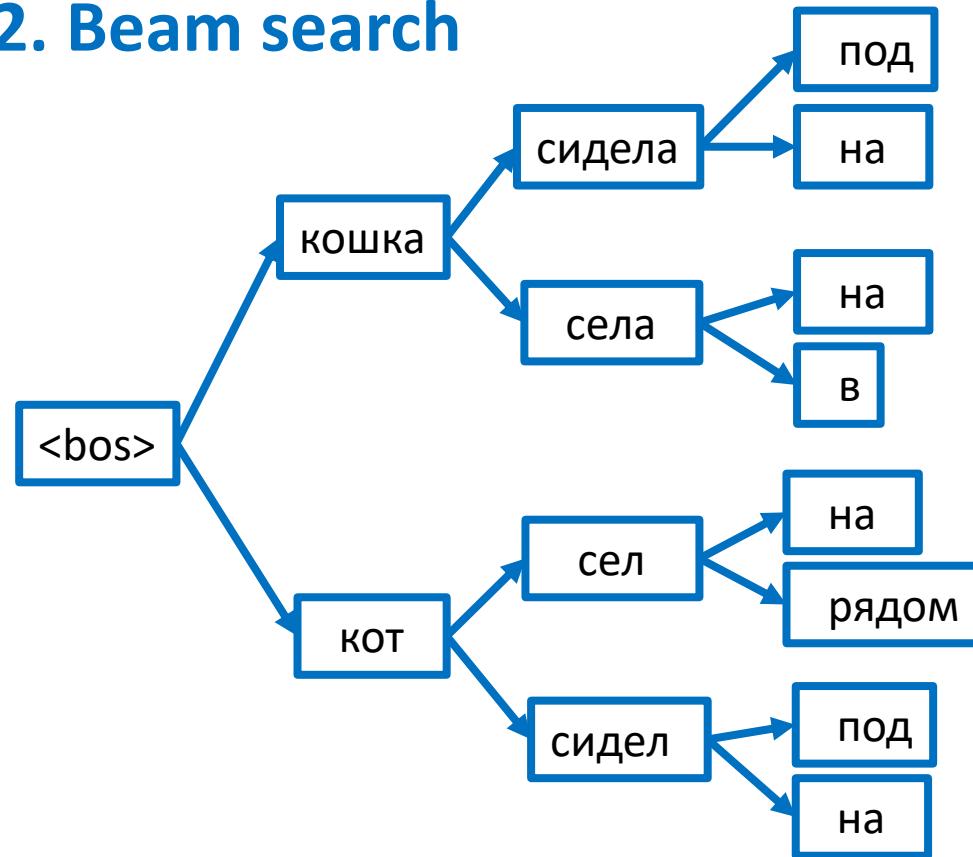
Decoding: how to generate a sequence?

2. Beam search



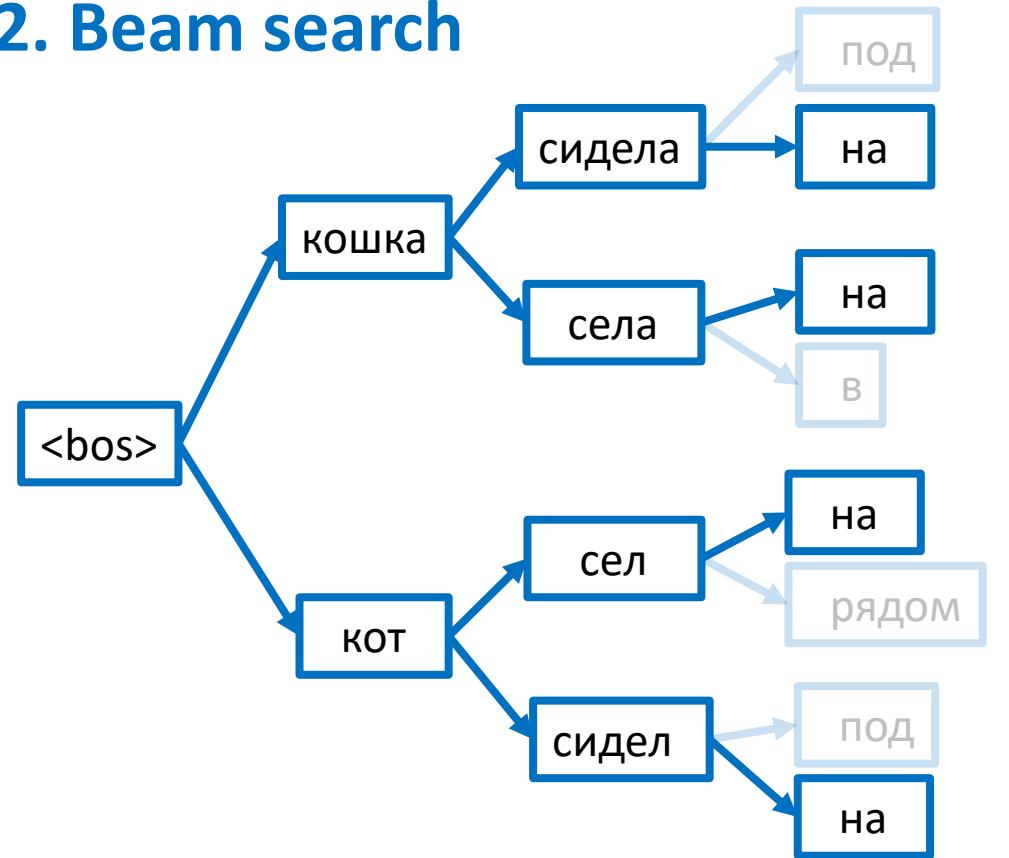
Decoding: how to generate a sequence?

2. Beam search



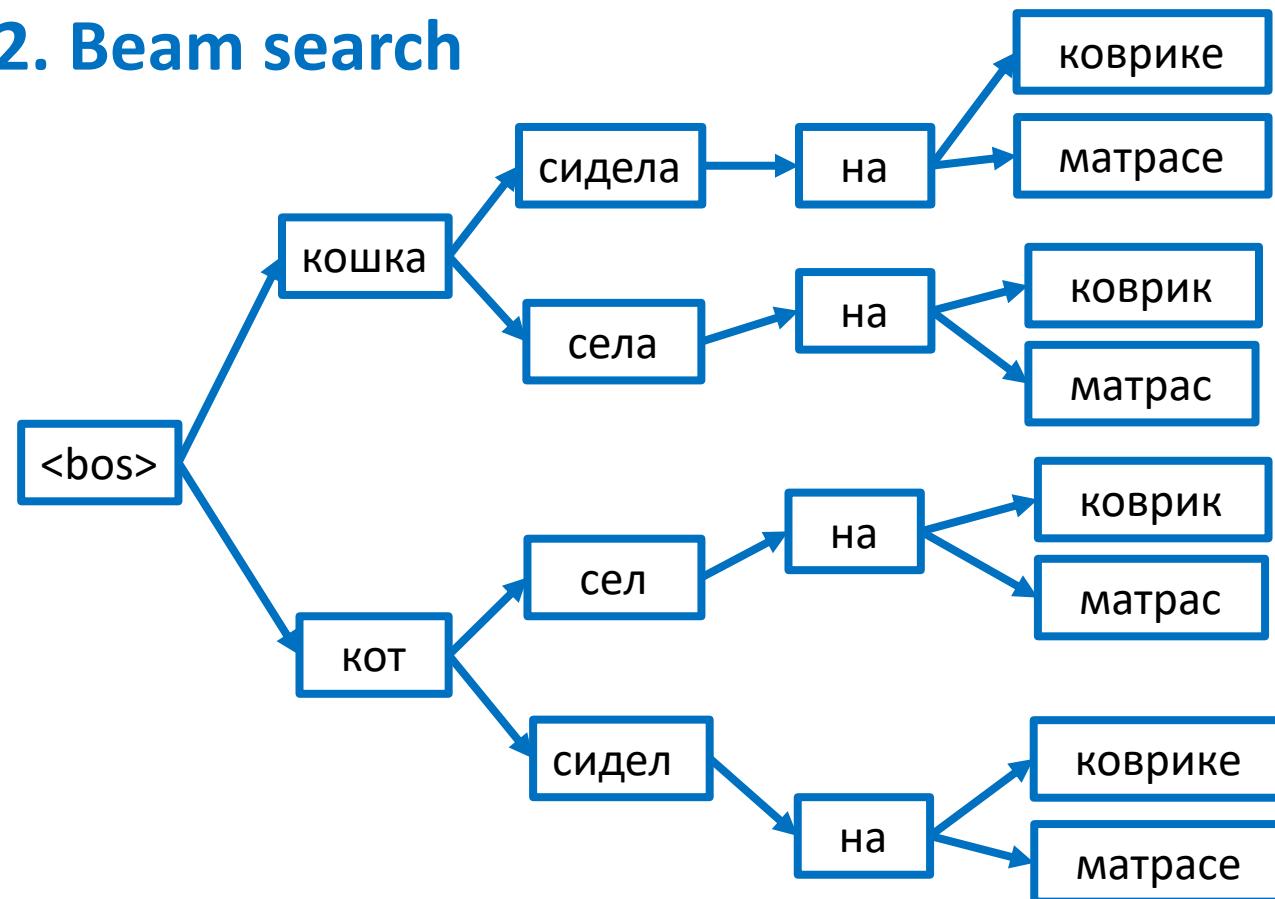
Decoding: how to generate a sequence?

2. Beam search



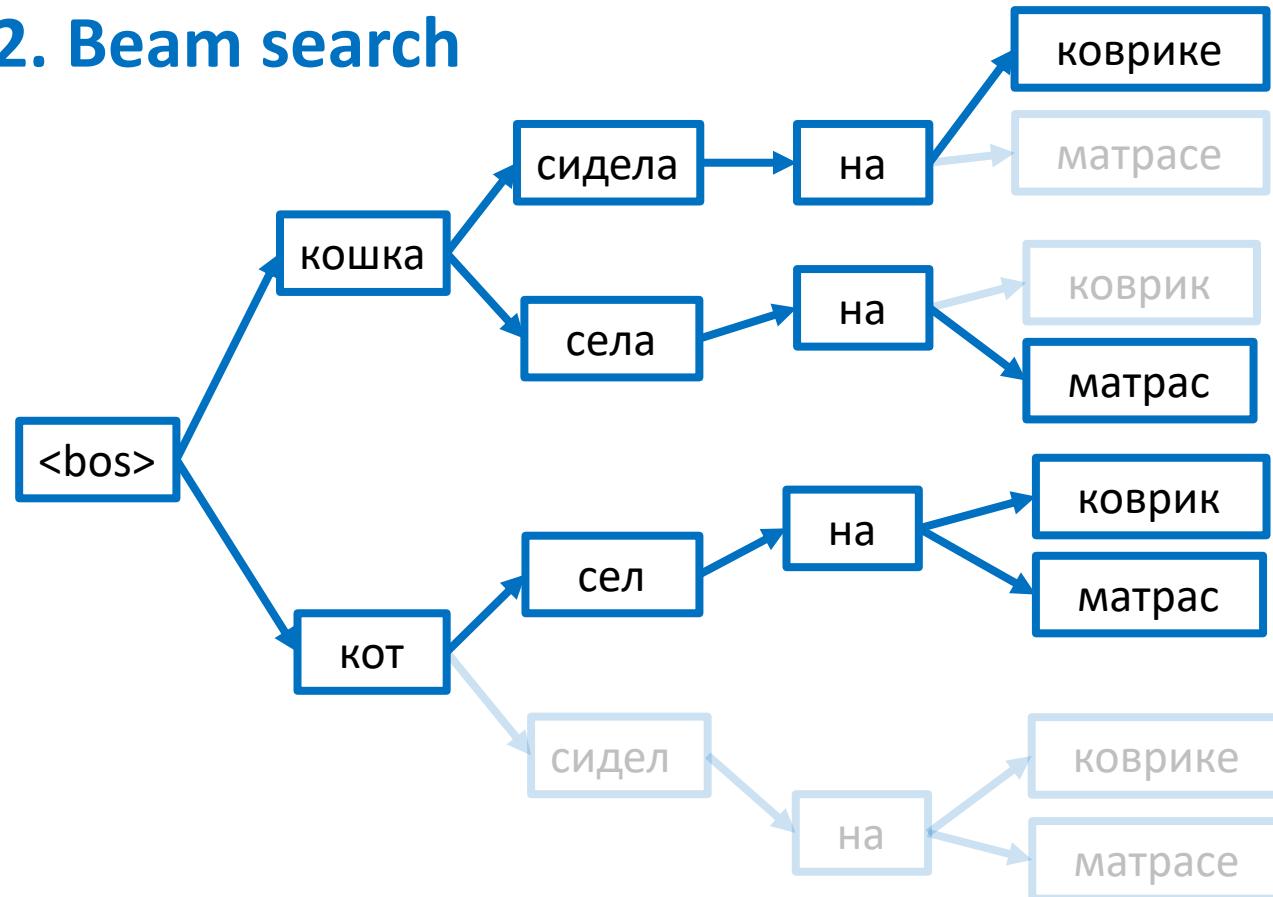
Decoding: how to generate a sequence?

2. Beam search



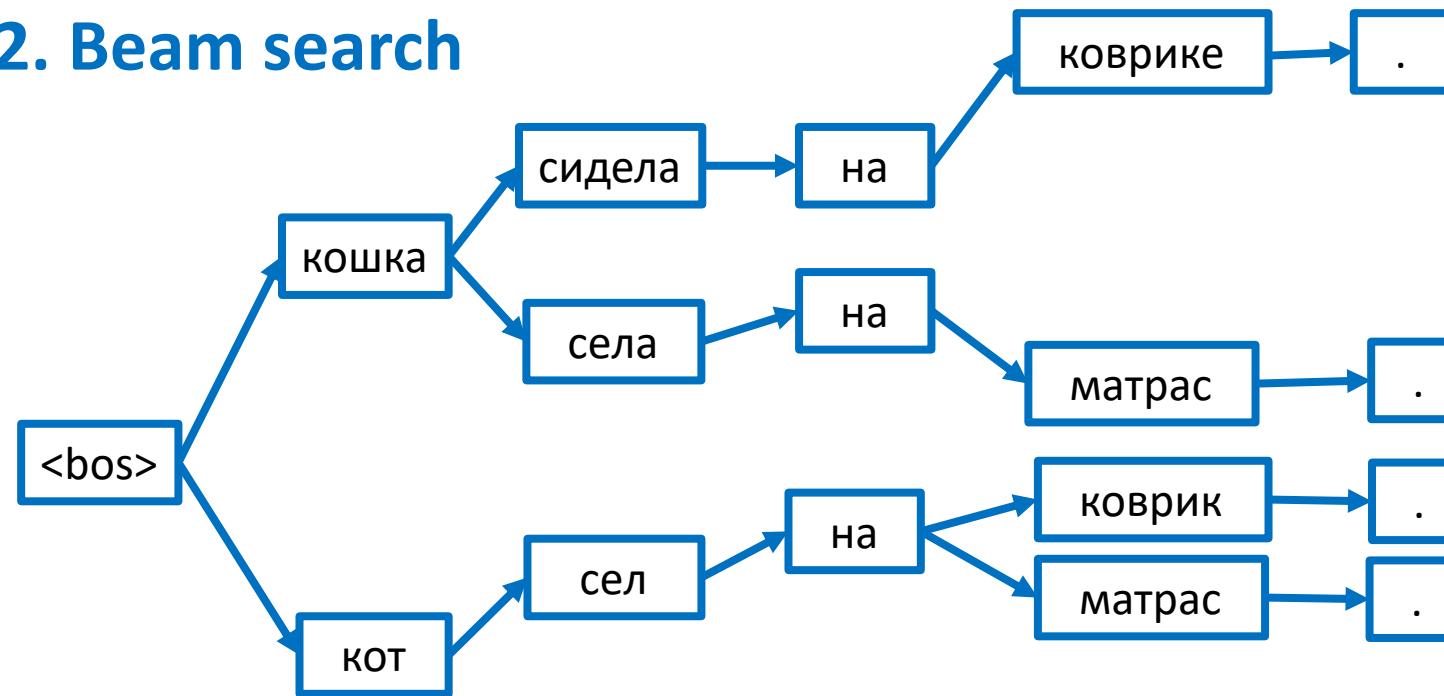
Decoding: how to generate a sequence?

2. Beam search



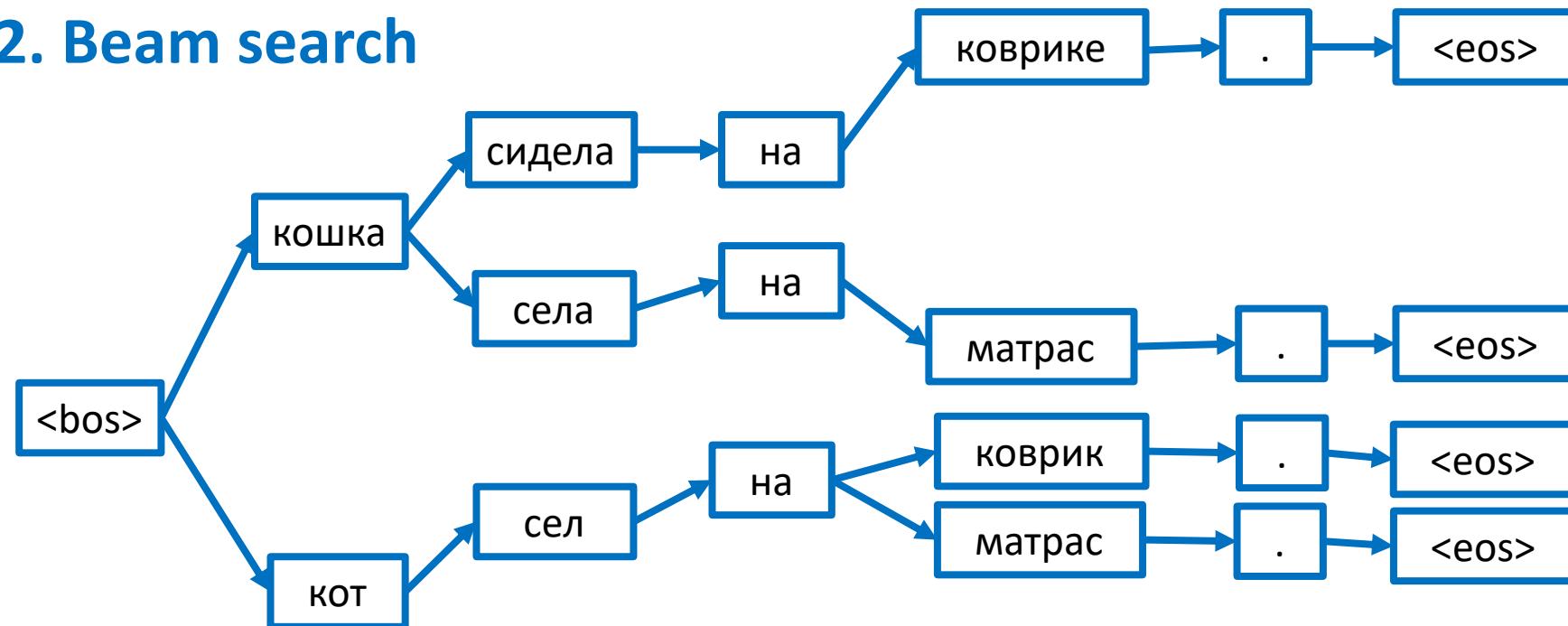
Decoding: how to generate a sequence?

2. Beam search



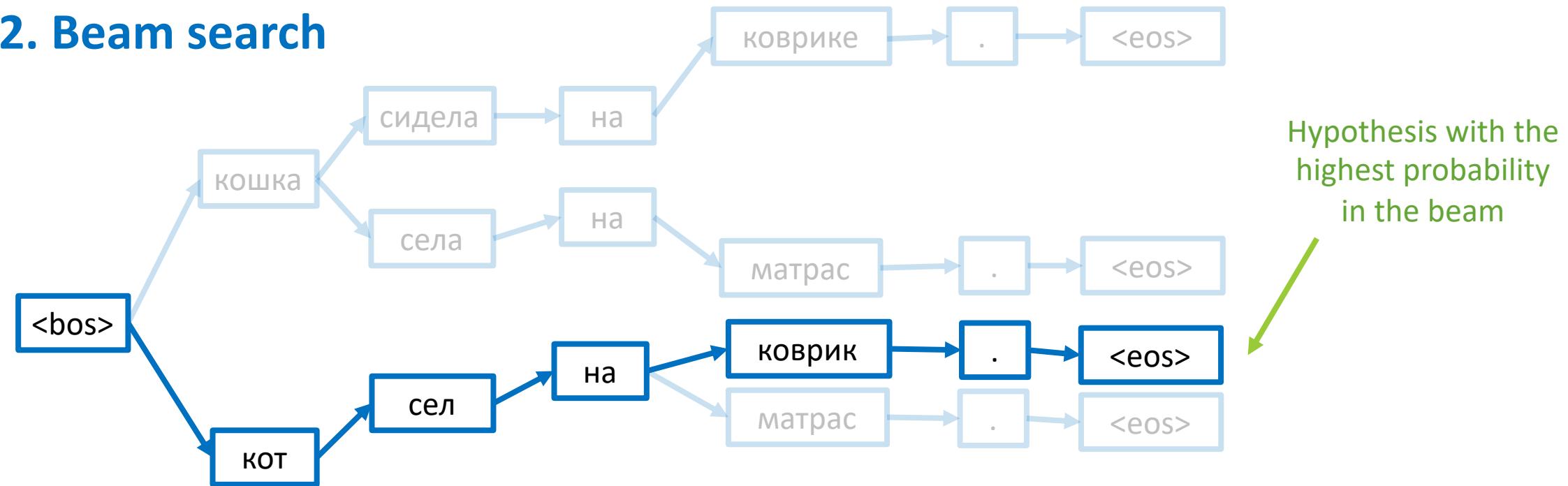
Decoding: how to generate a sequence?

2. Beam search



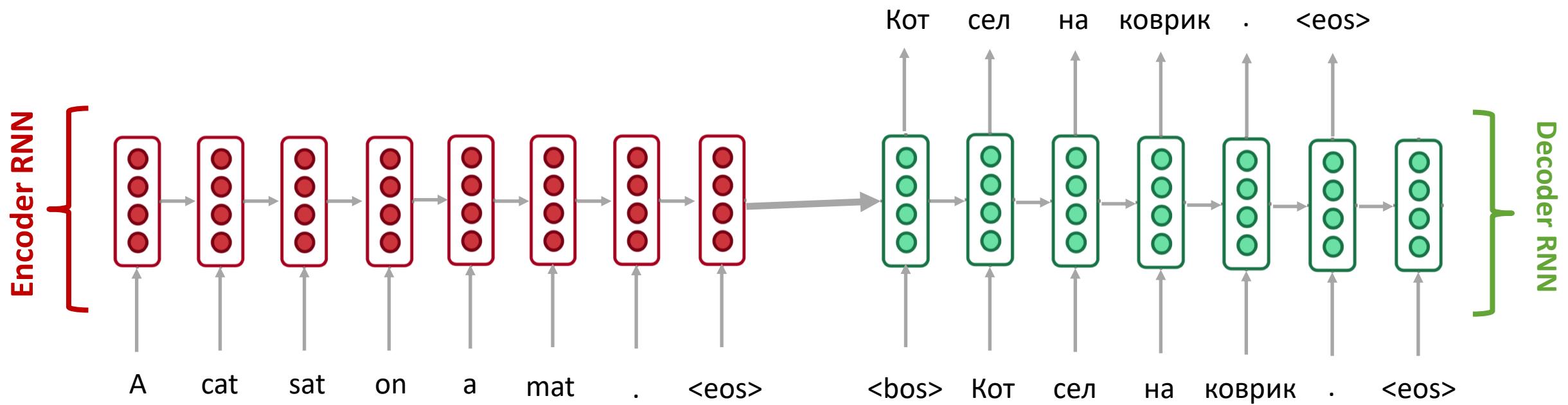
Decoding: how to generate a sequence?

2. Beam search



Attention-based models

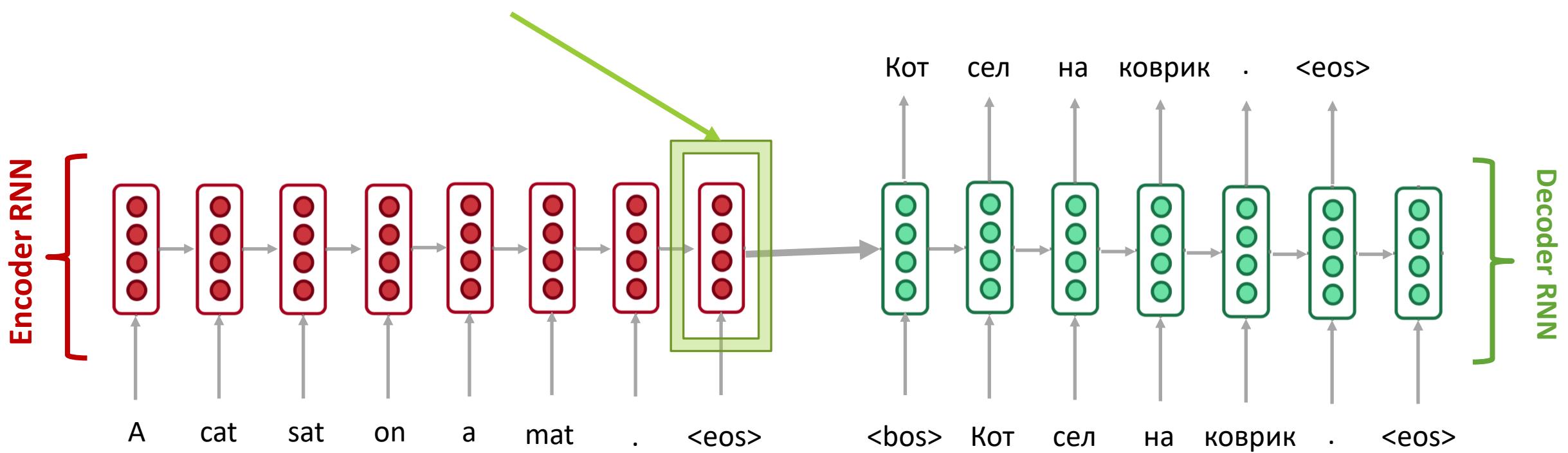
What is the problem with such models?



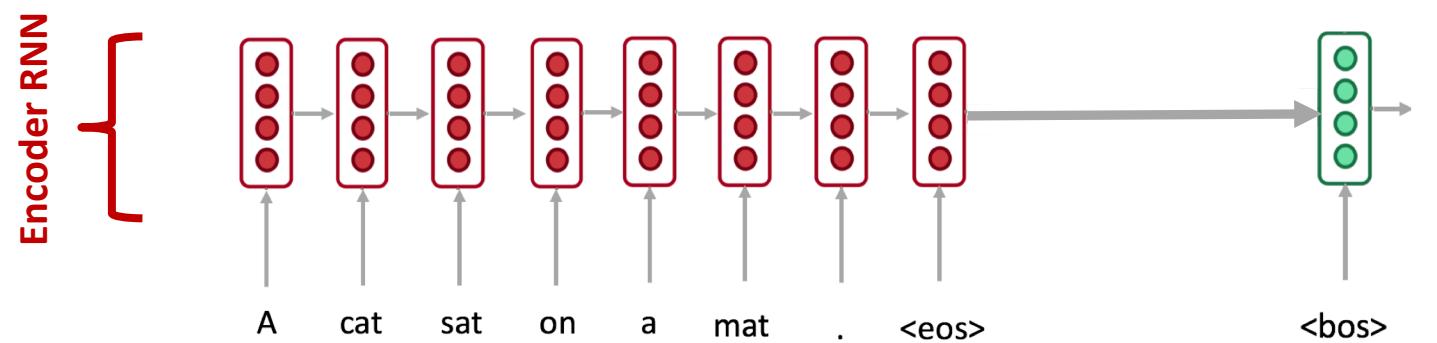
What is the problem with such models?

Information bottleneck:

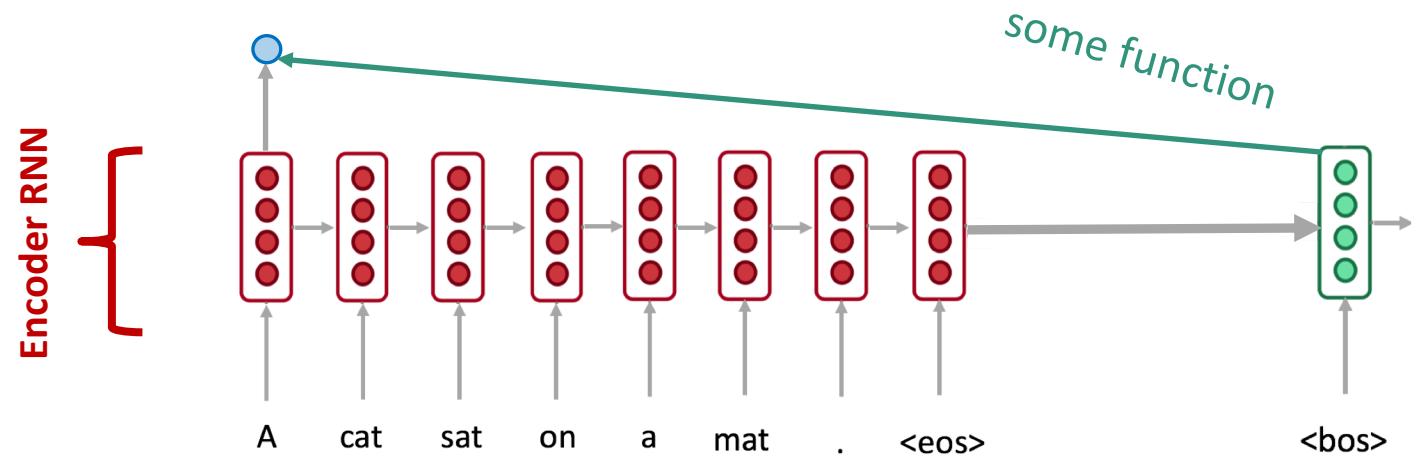
the model has to put all information
about sentence in one vector



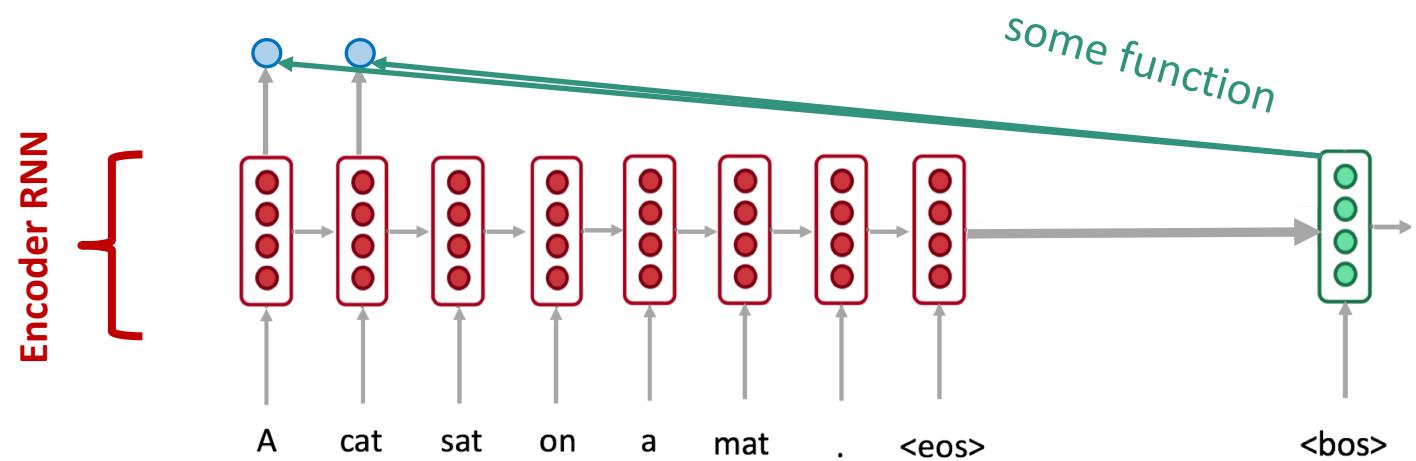
Attention



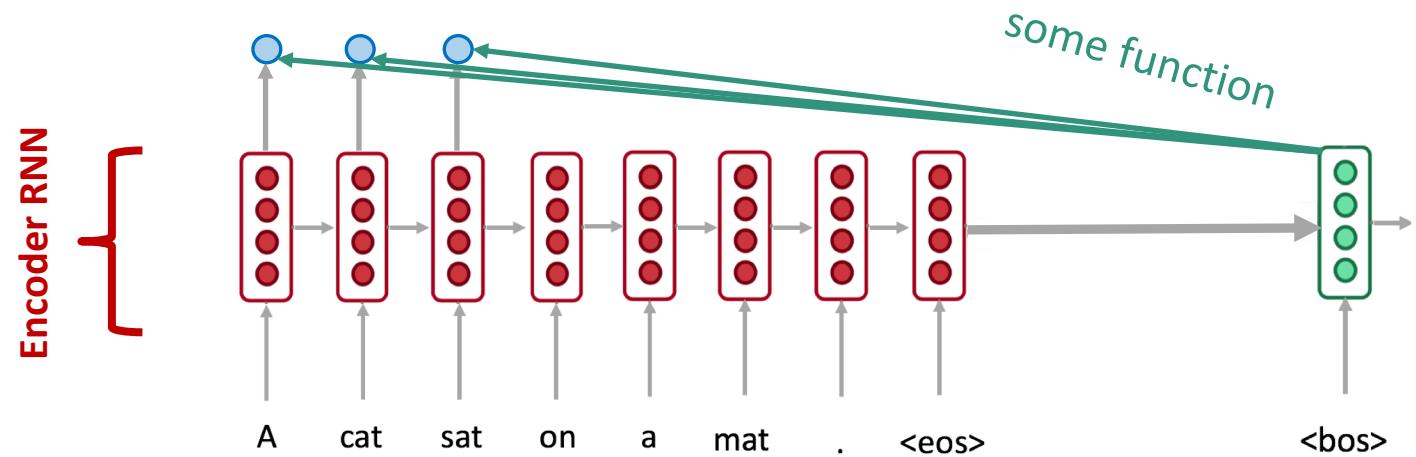
Attention



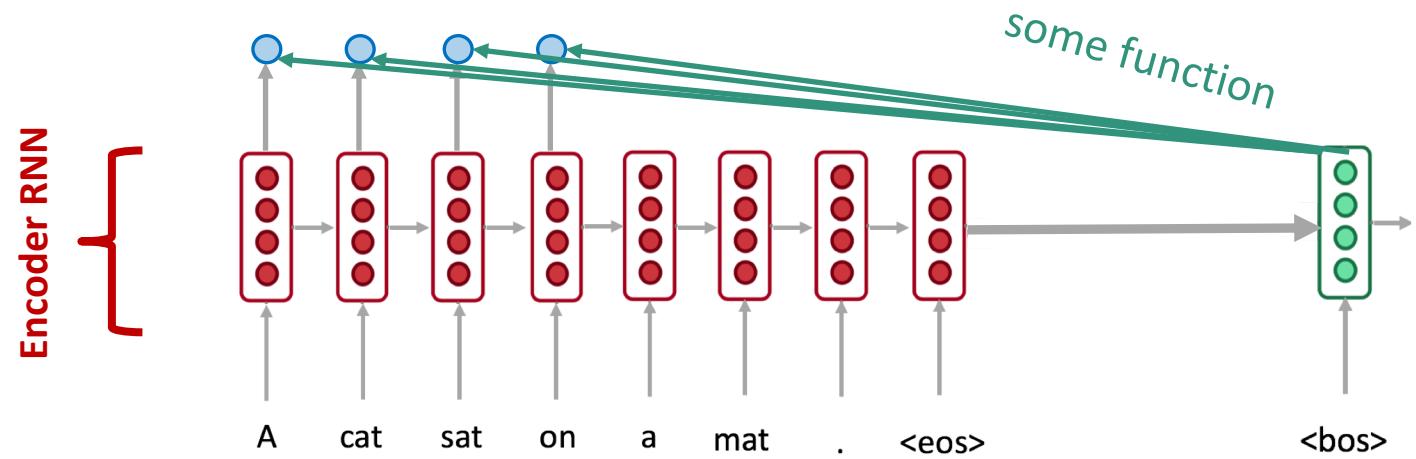
Attention



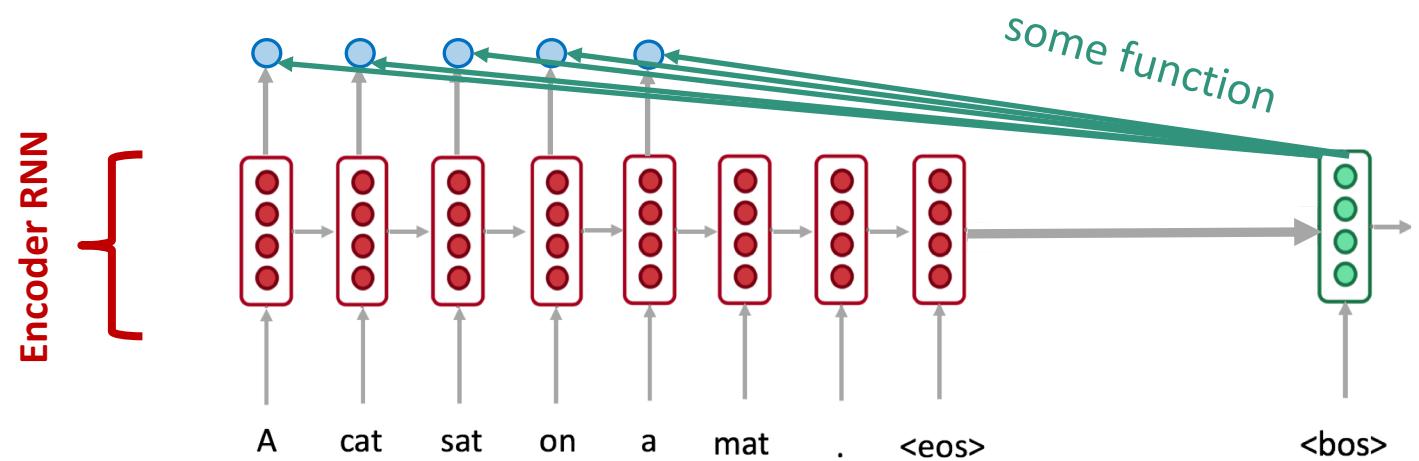
Attention



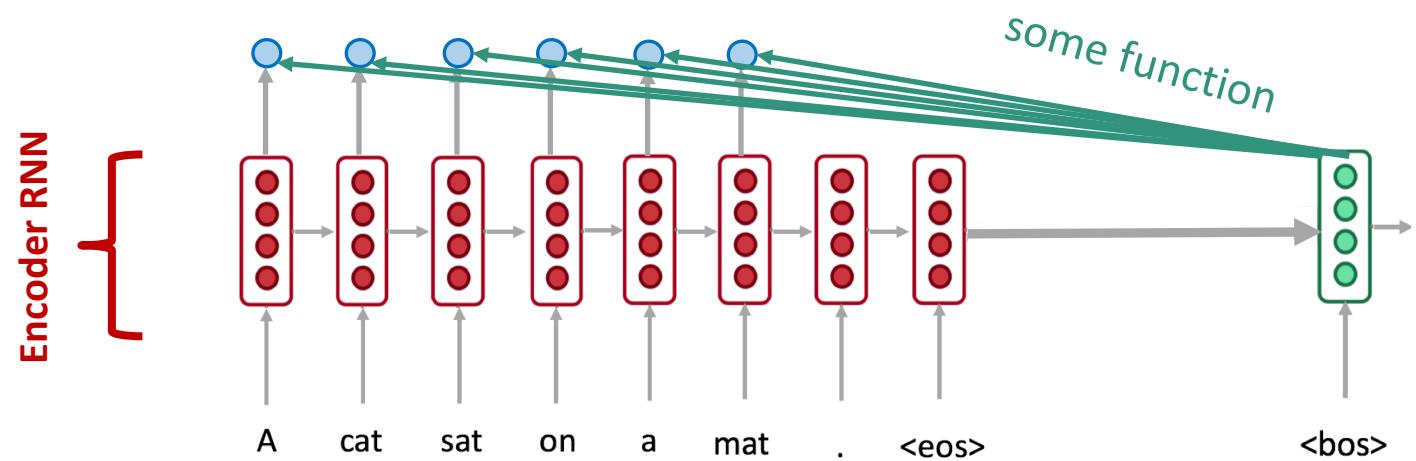
Attention



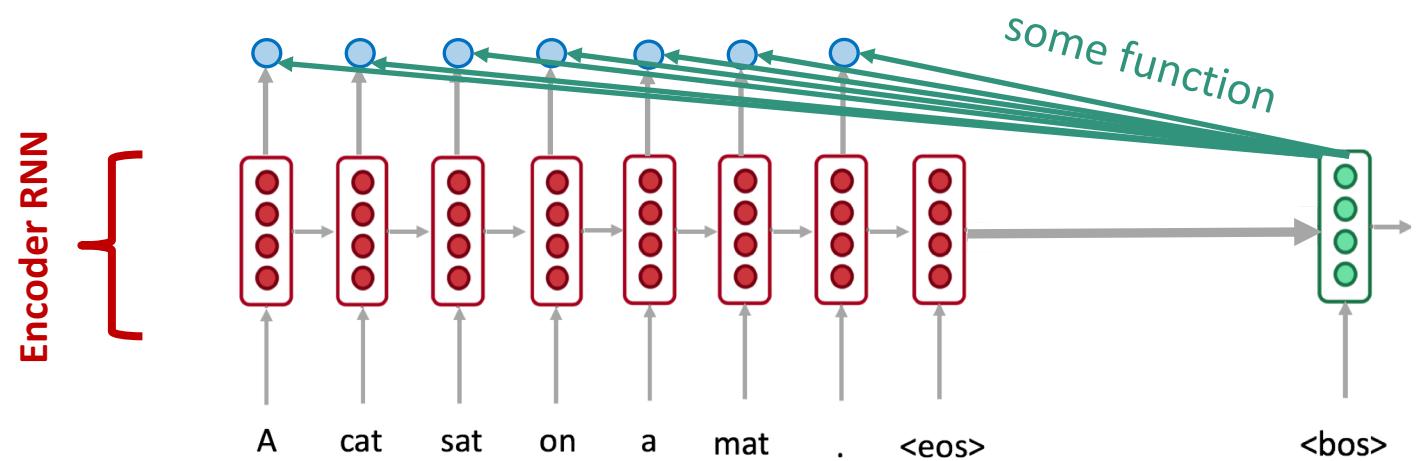
Attention



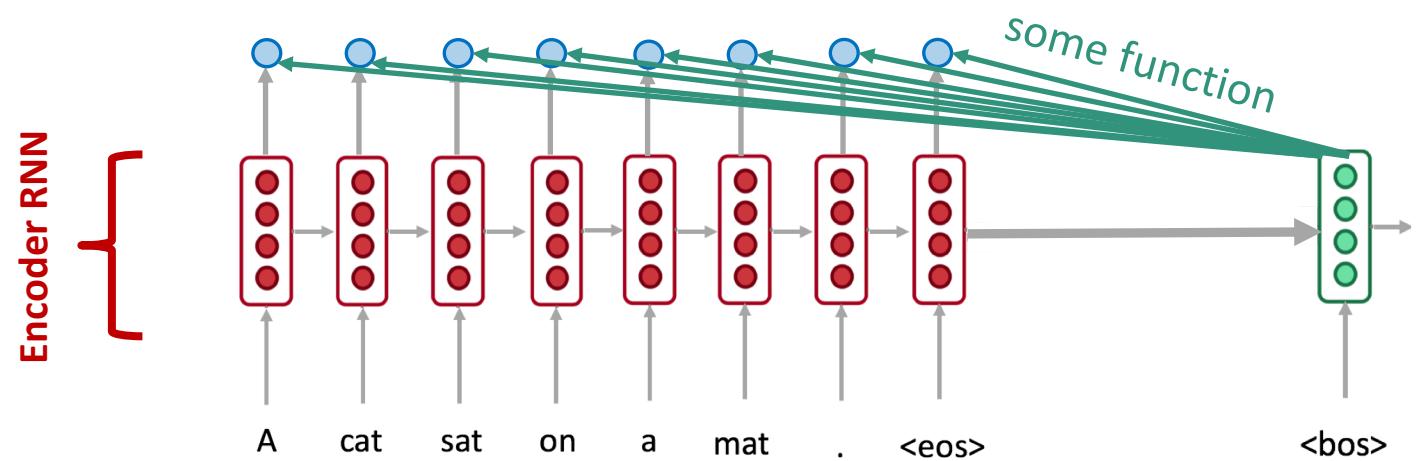
Attention



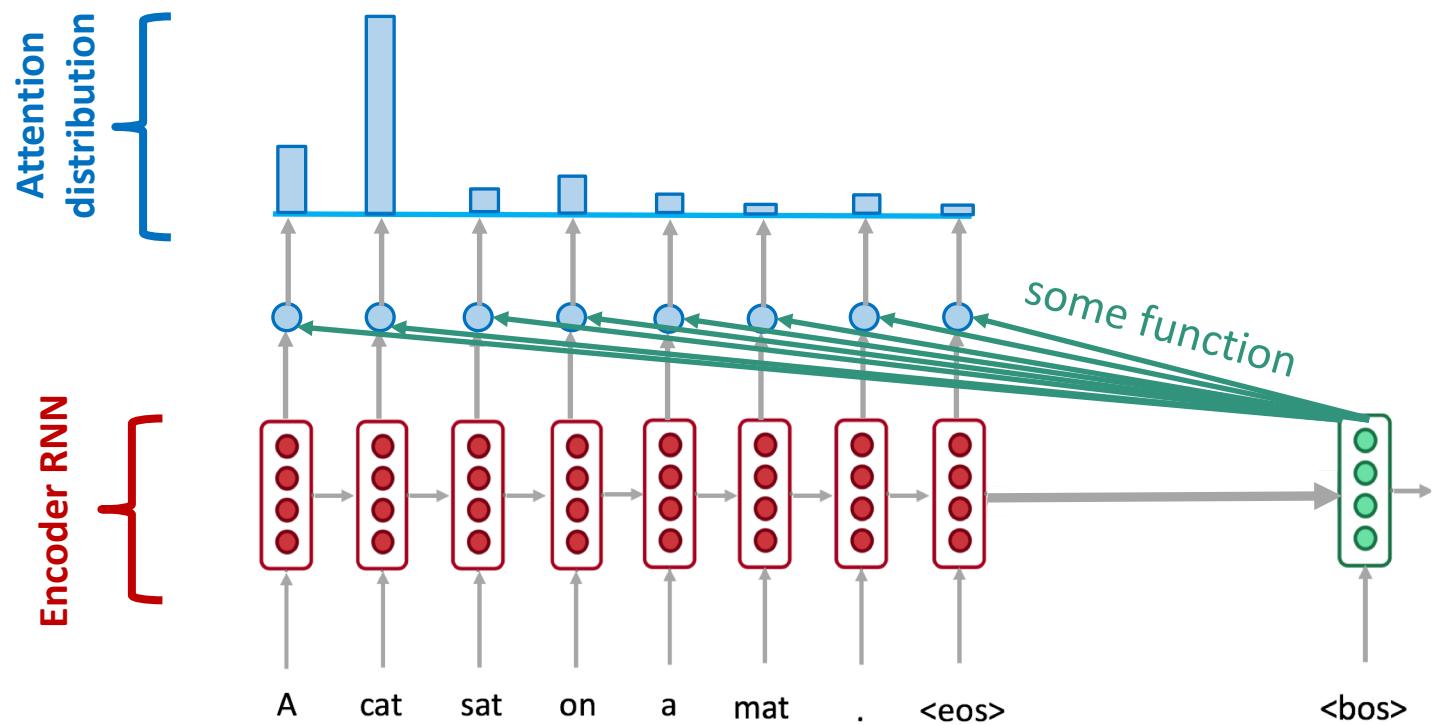
Attention



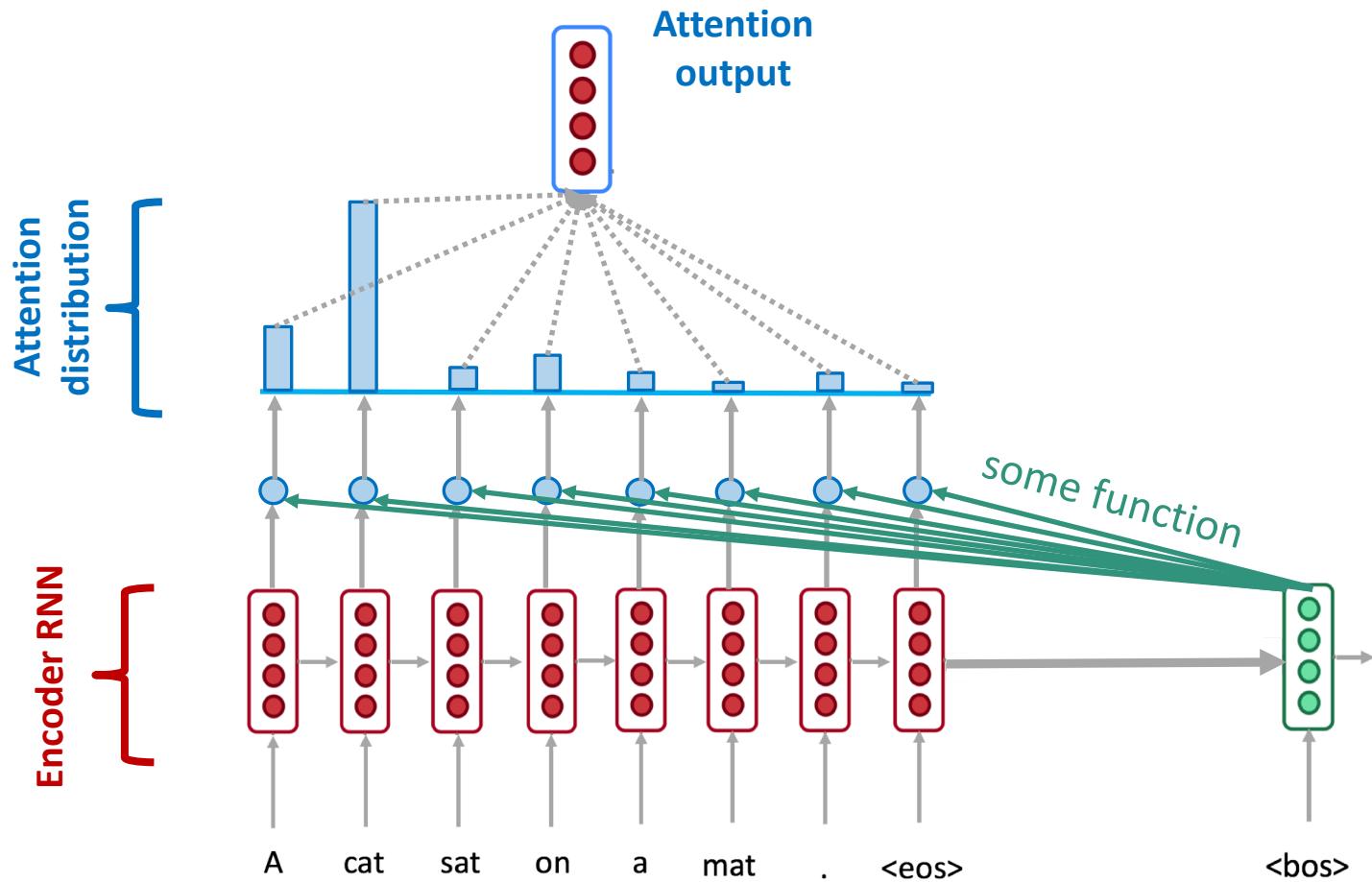
Attention



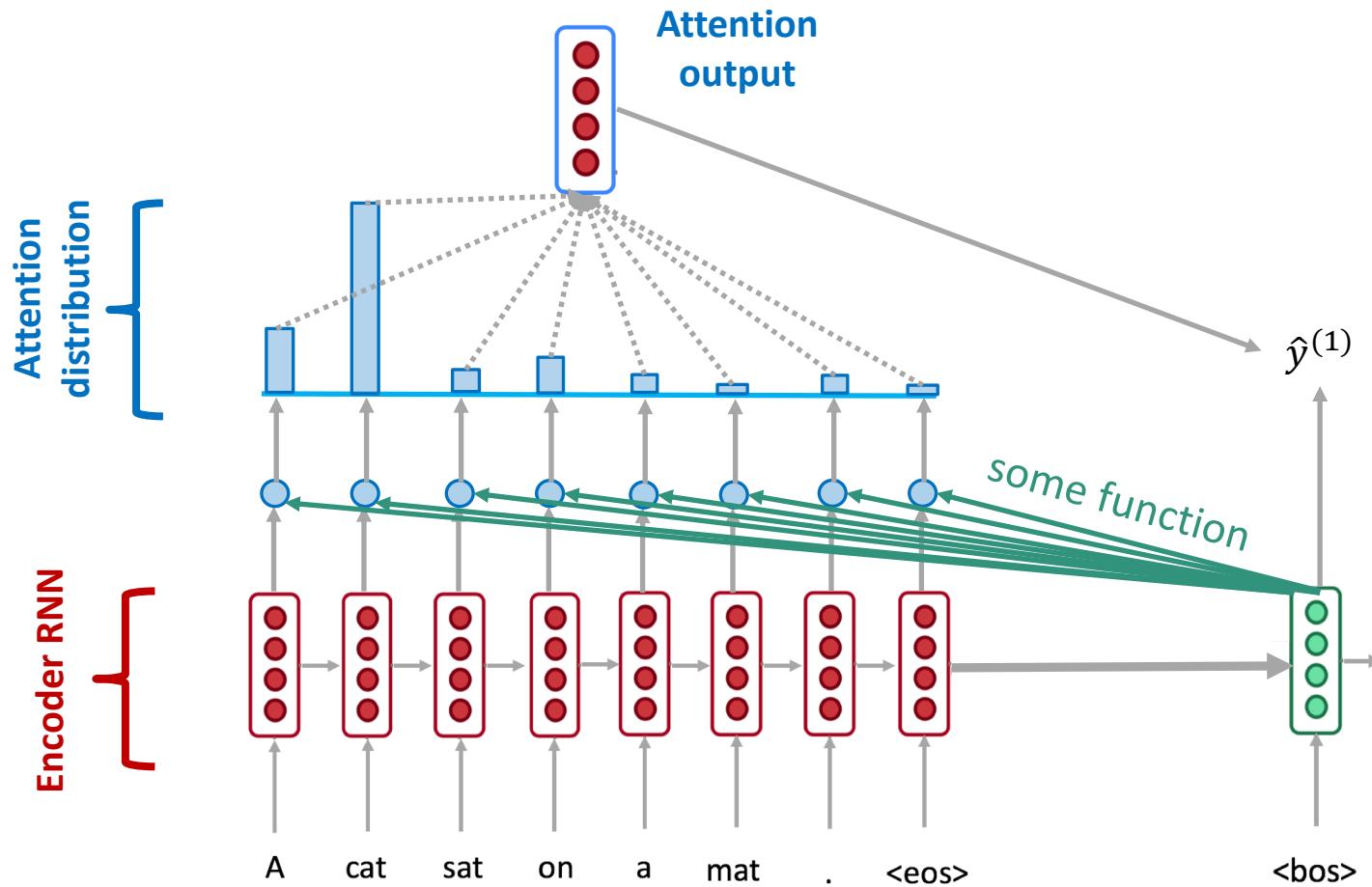
Attention



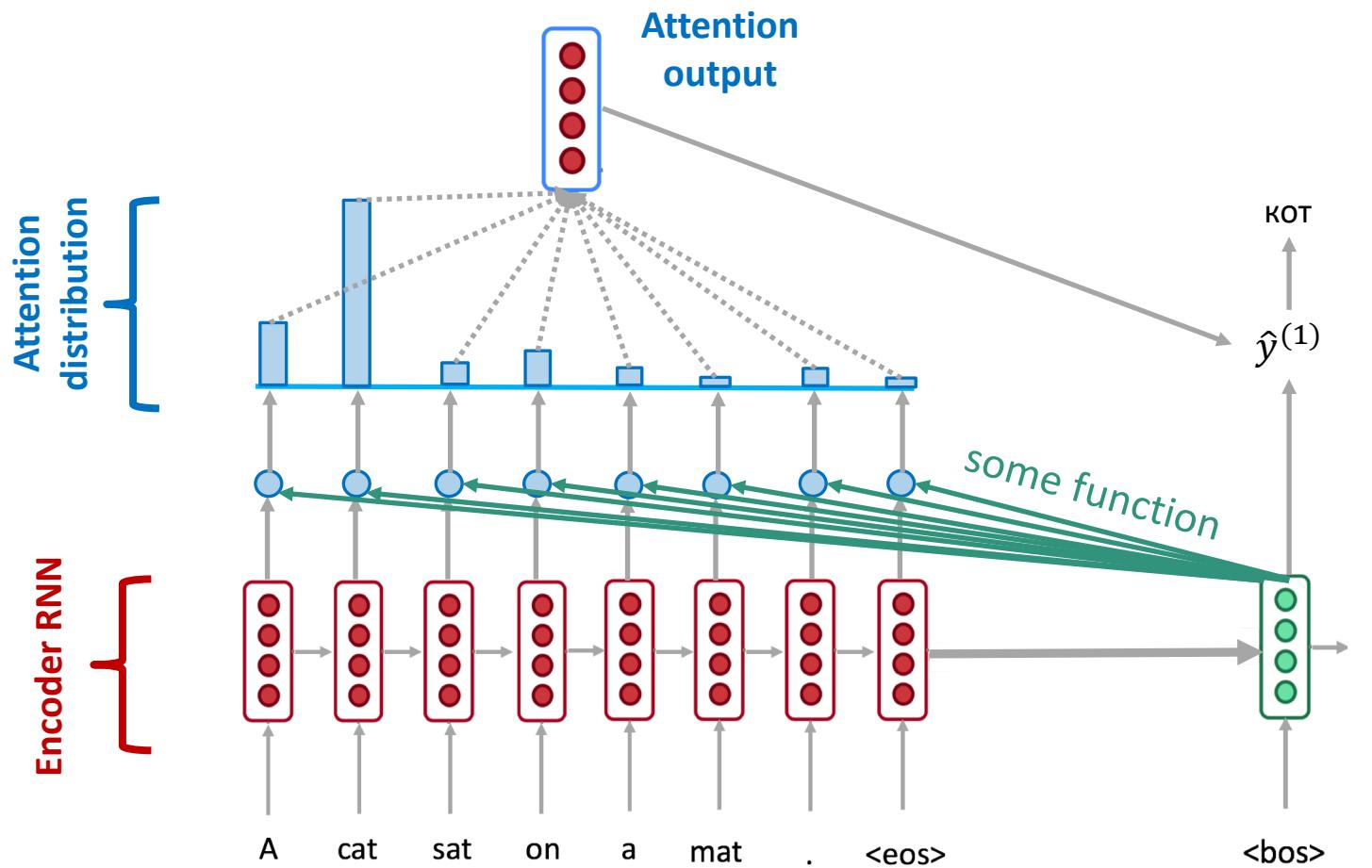
Attention



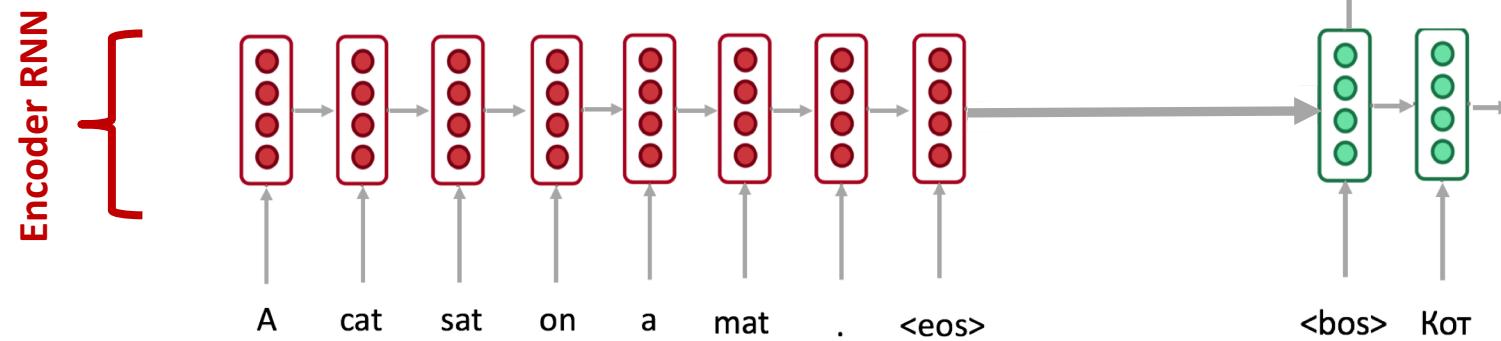
Attention



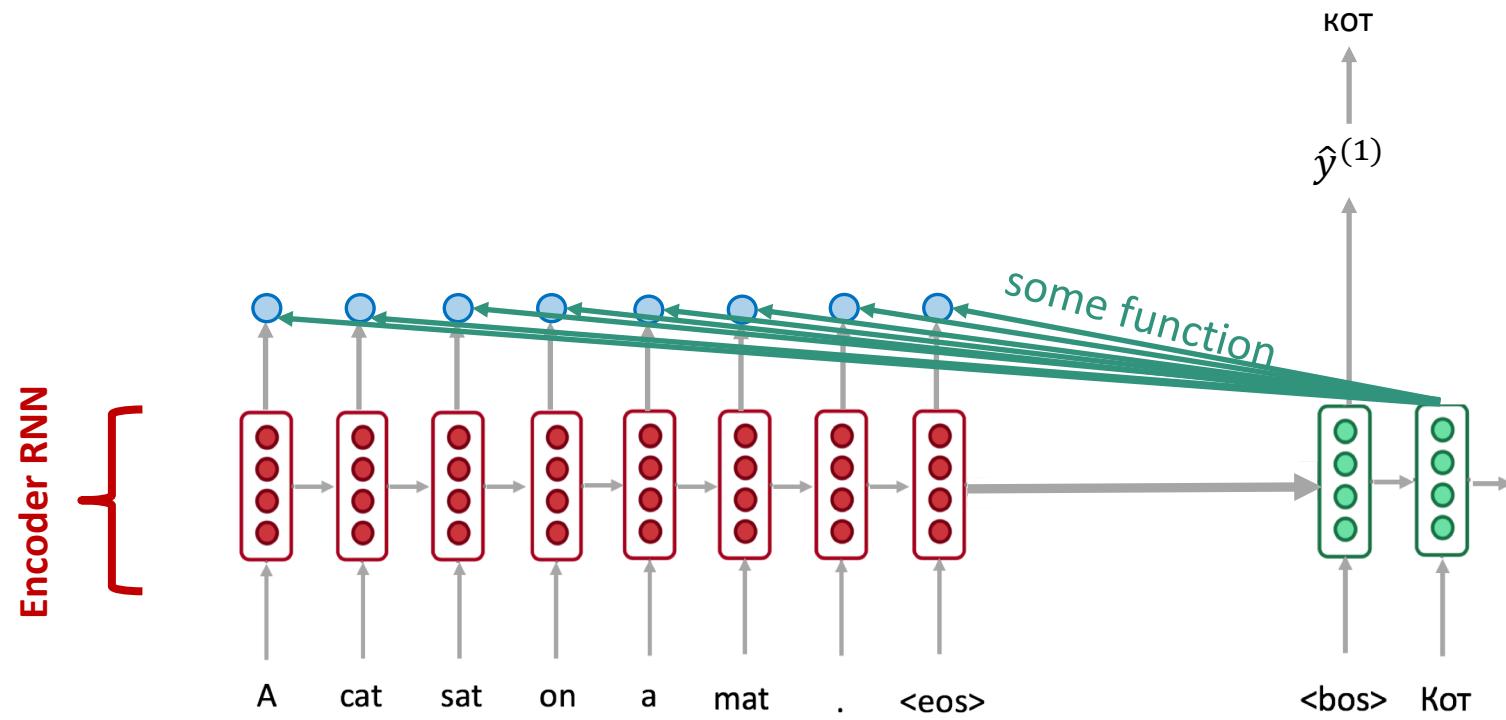
Attention



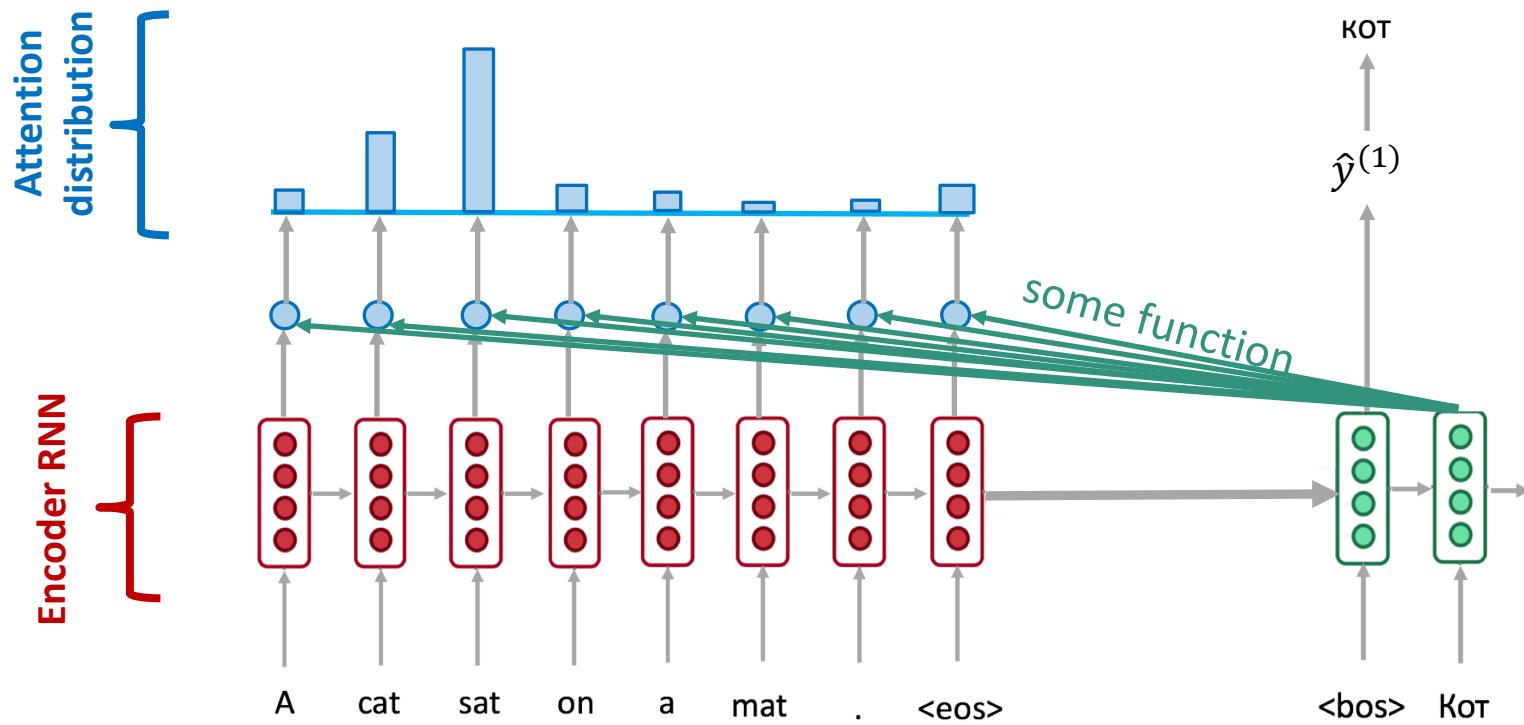
Attention



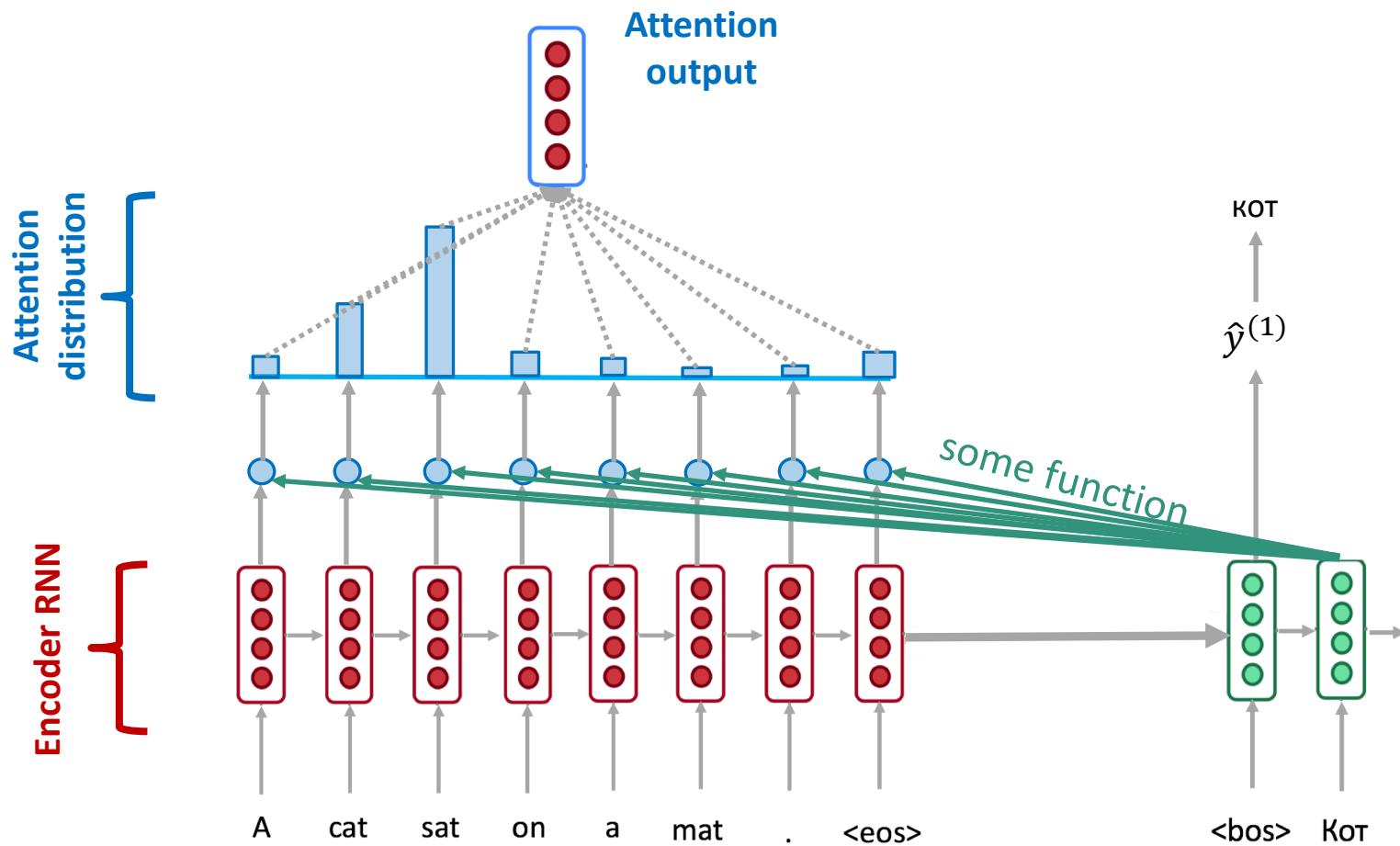
Attention



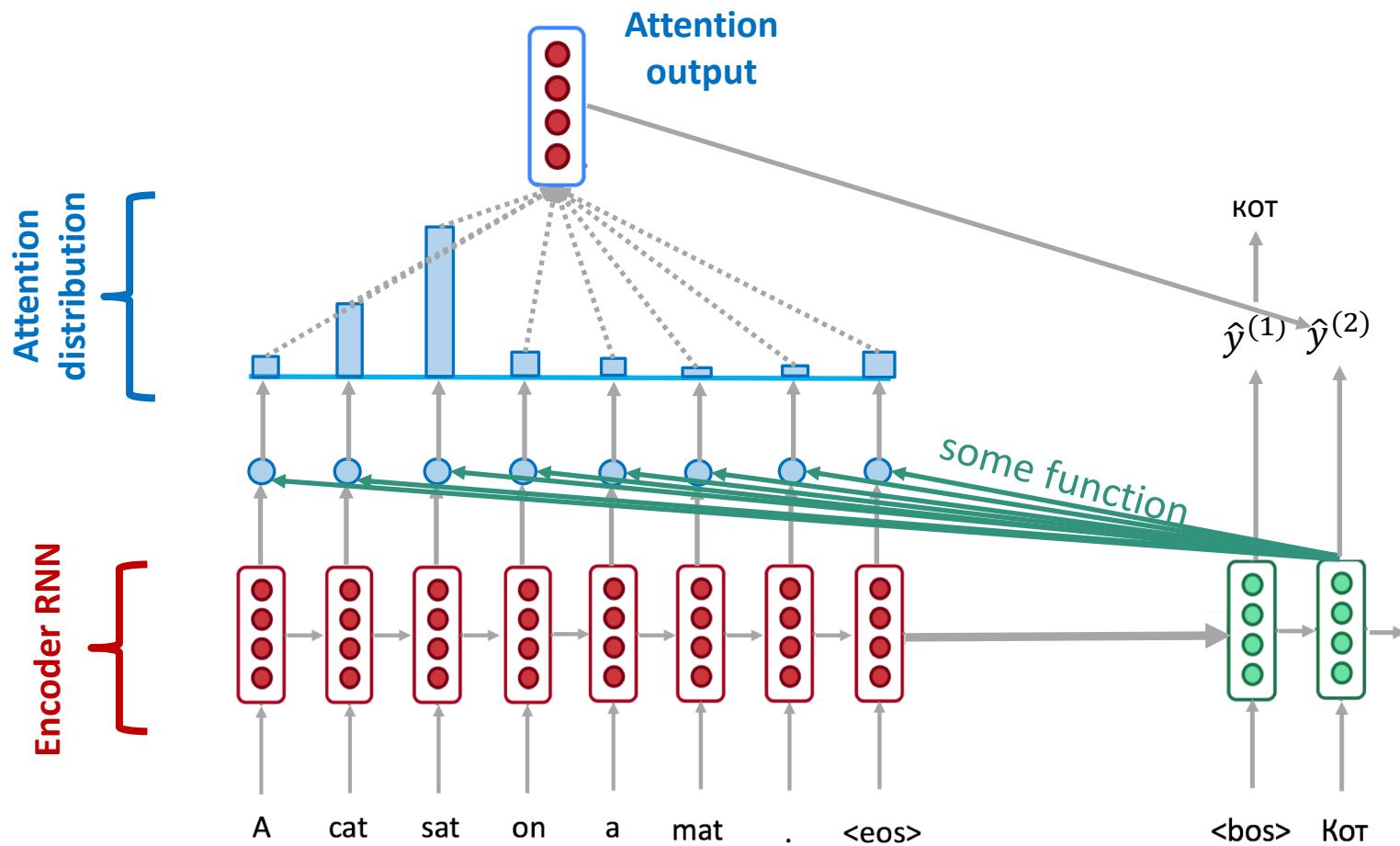
Attention



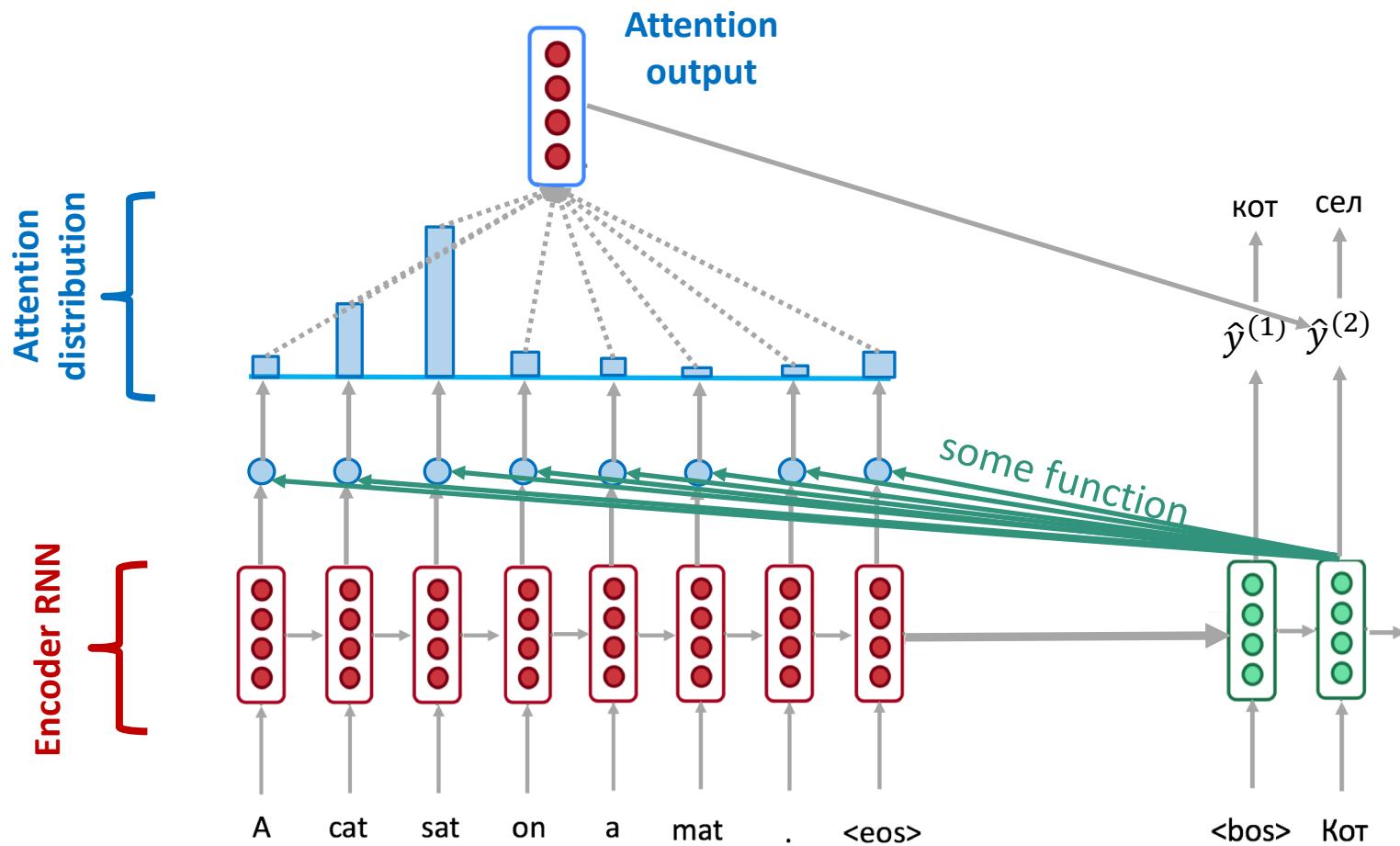
Attention



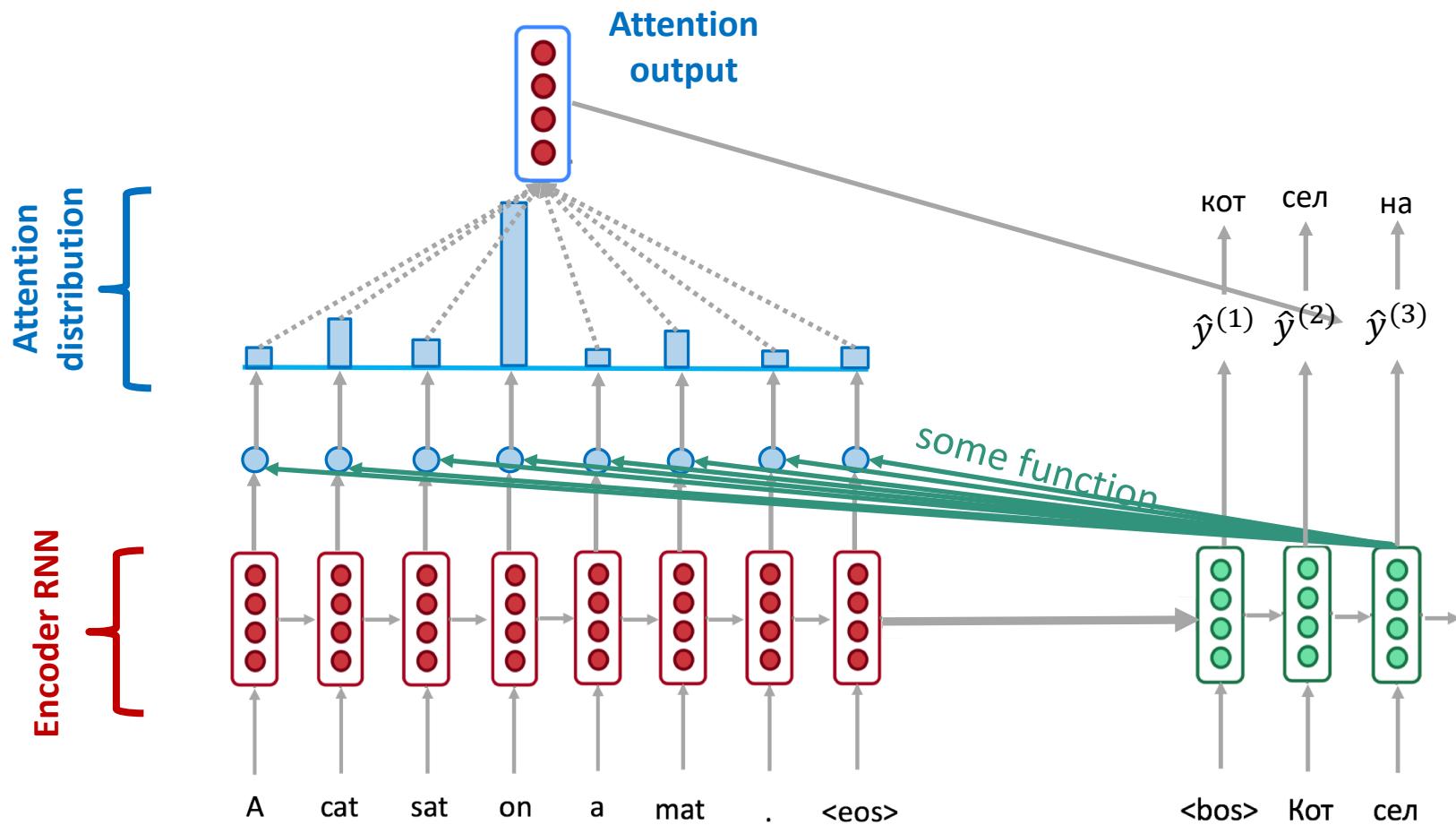
Attention



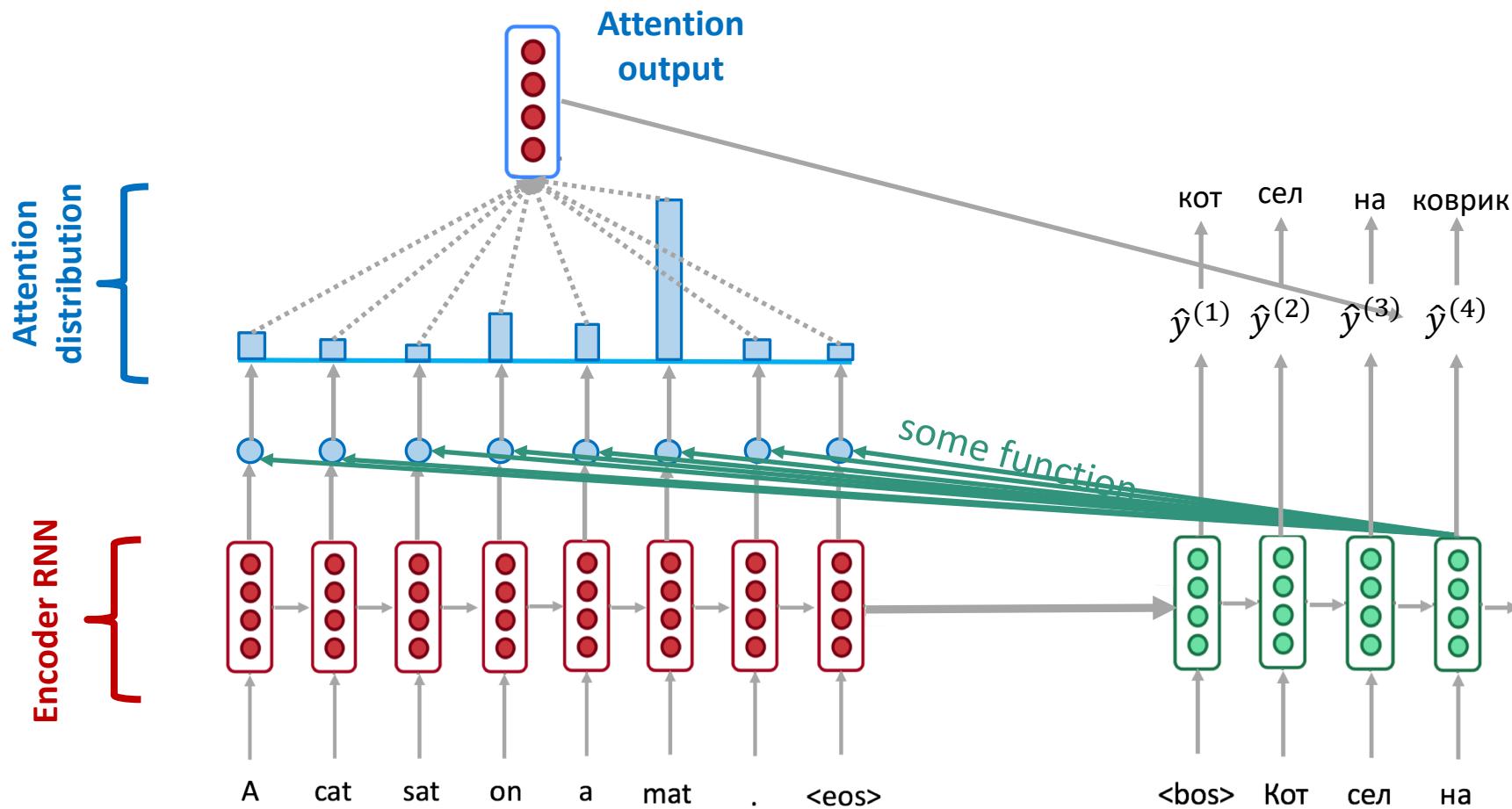
Attention



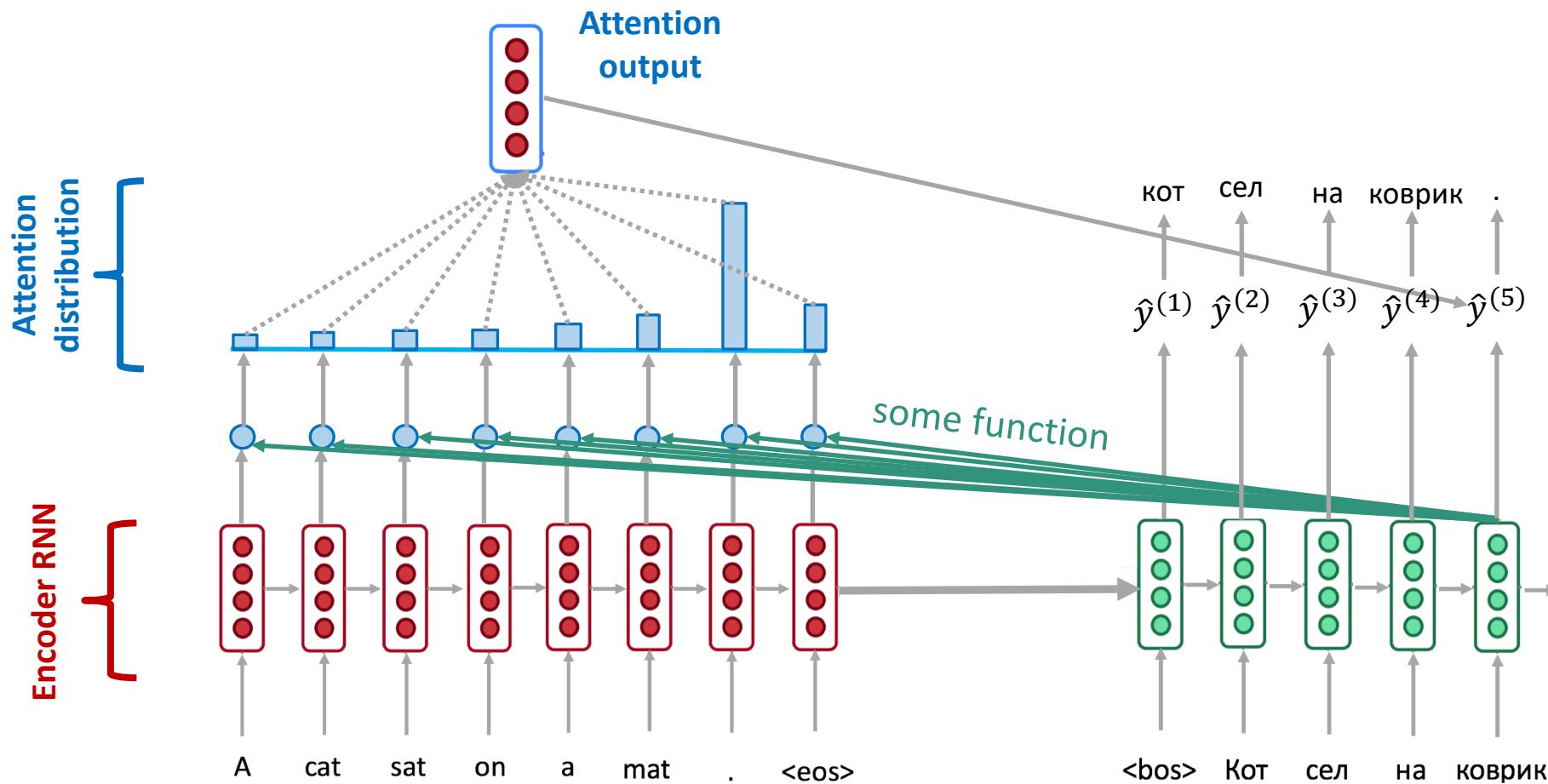
Attention



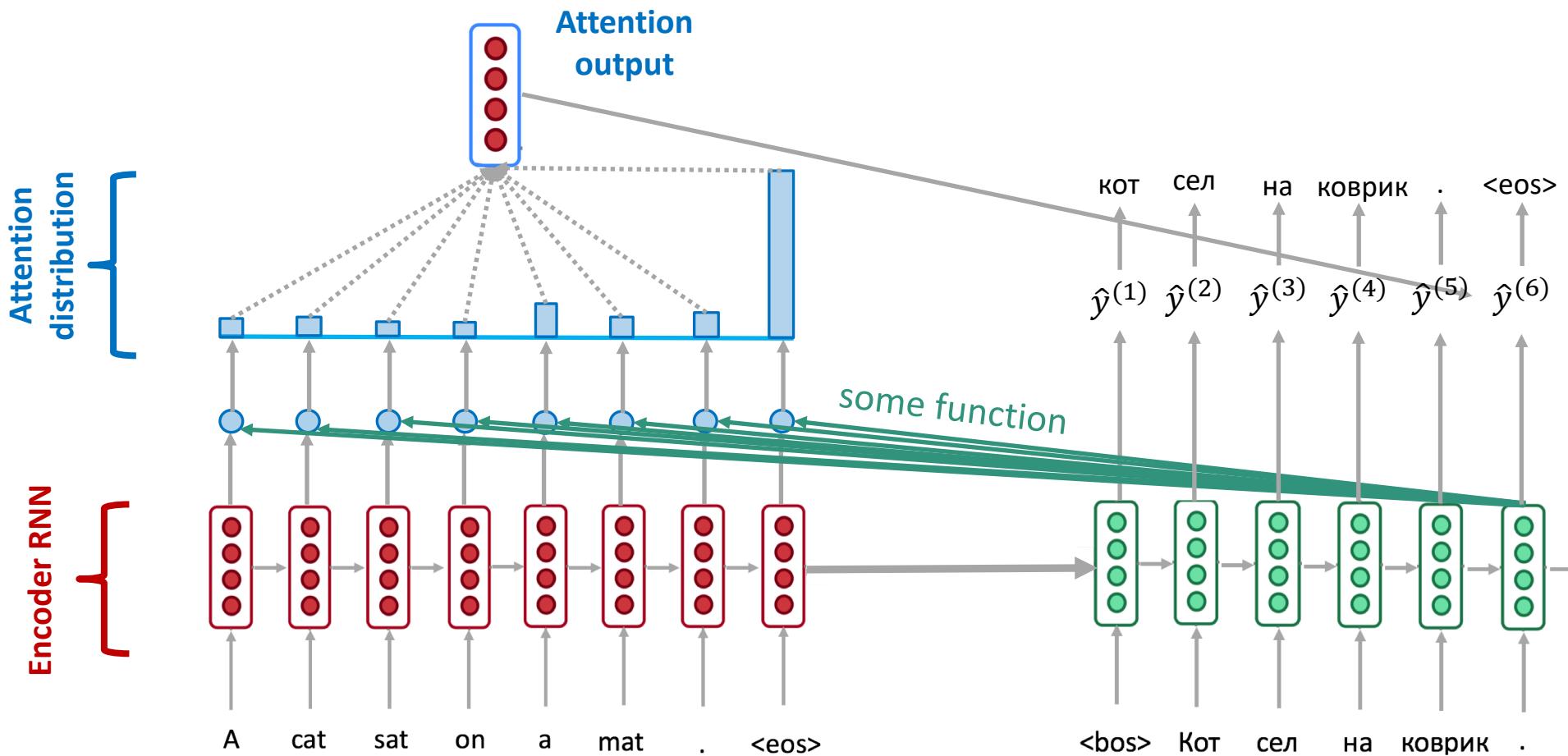
Attention



Attention



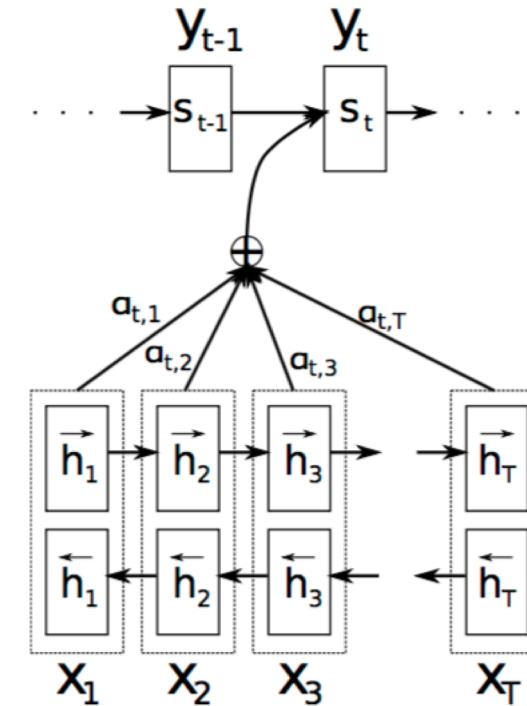
Attention



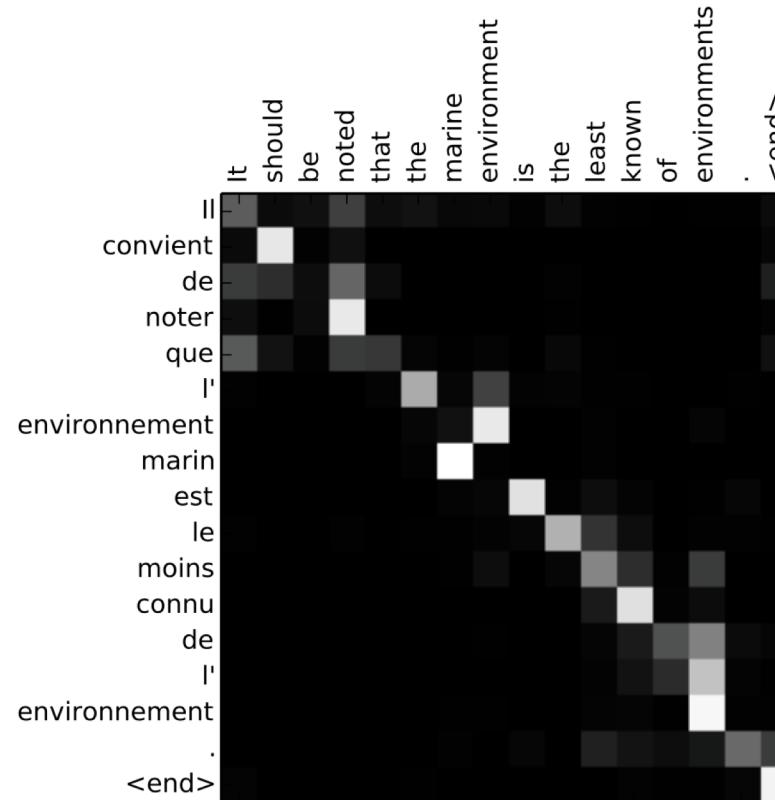
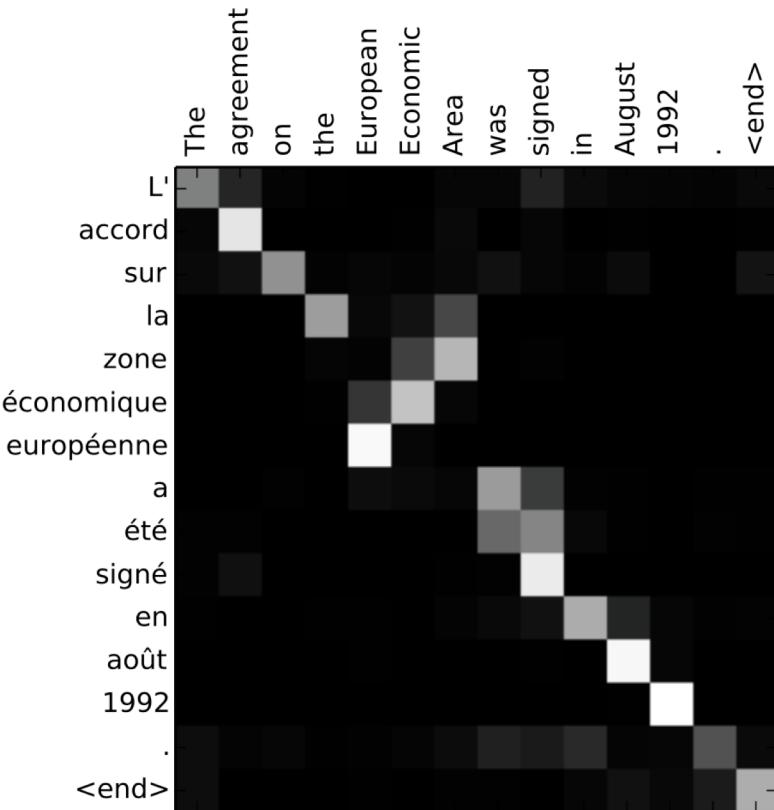
Neural Machine Translation by Jointly Learn to Align and Translate

Attention advantages:

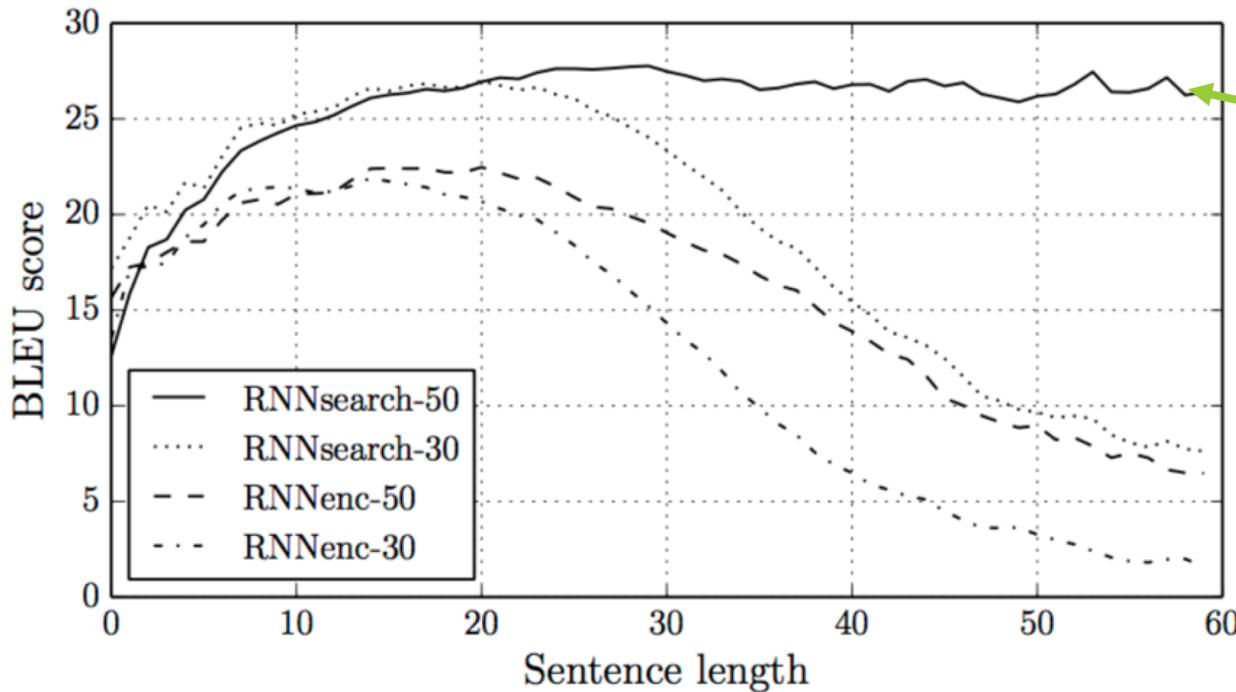
- do not have to remember the whole sentence
- no fixed size representation
- similar to humans



Neural Machine Translation by Jointly Learn to Align and Translate



Neural Machine Translation by Jointly Learn to Align and Translate



No performance drop
for long sentences!

Attention score: how to compute it?

➤ dot product

$$f_{att}(h_i, s_j) = h_i^\top s_j$$

➤ bilinear function

$$f_{att}(h_i, s_j) = h_i^\top \mathbf{W}_a s_j$$

➤ multi-layer perceptron

$$f_{att}(\mathbf{h}_i, \mathbf{s}_j) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\mathbf{h}_i; \mathbf{s}_j])$$

➤ any, literally, ANY function you can imagine

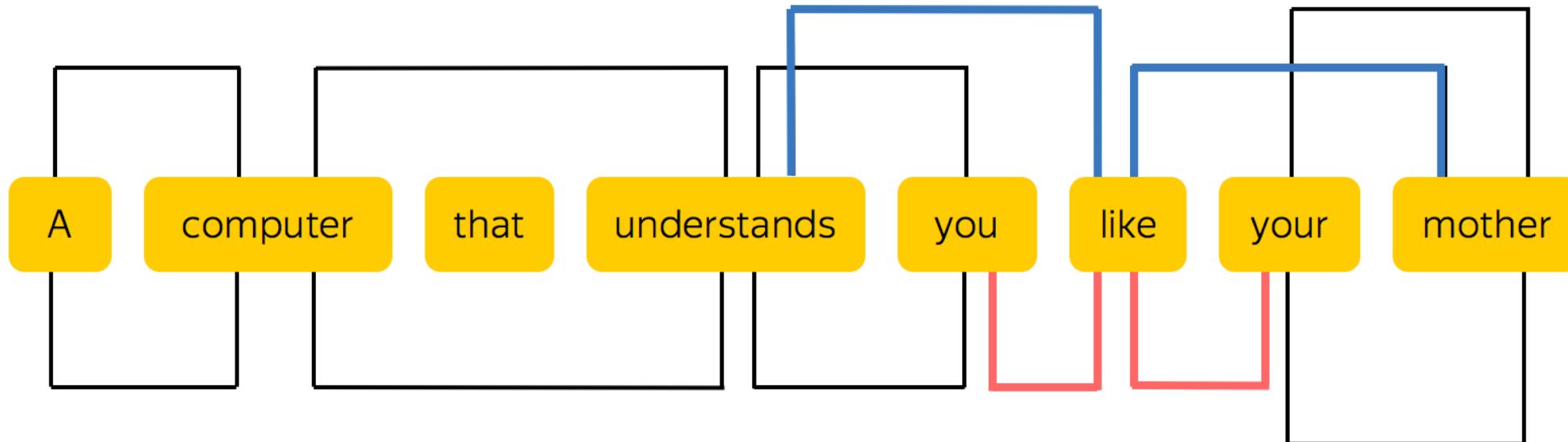
$$z(c, m, q) = [c, m, q, c \circ q, c \circ m, |c - q|, |c - m|, c^T W^{(b)} q, c^T W^{(b)} m]$$

$$G(c, m, q) = \sigma \left(W^{(2)} \tanh \left(W^{(1)} z(c, m, q) + b^{(1)} \right) + b^{(2)} \right)$$

A bit crazy, huh?

Transformer: Attention is all you need!

Self-Attention



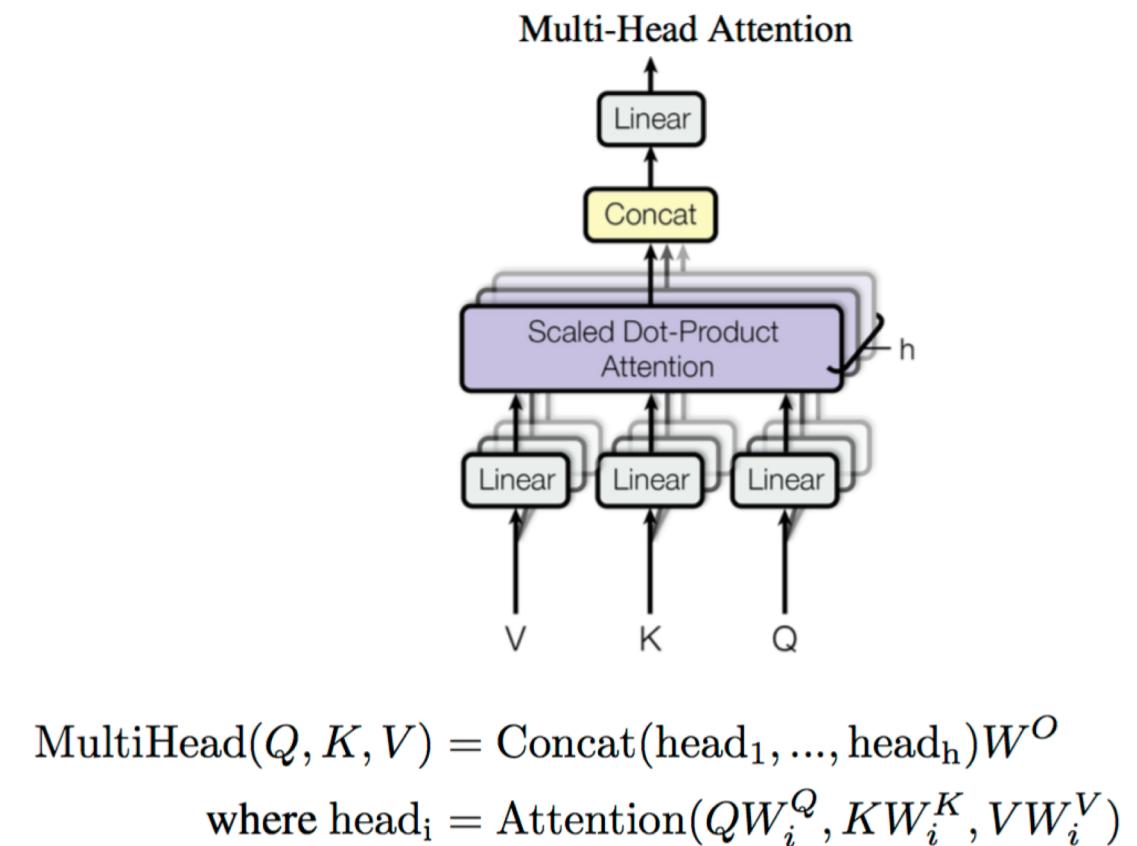
(example and picture from David Talbot)

Multi-Head Attention

Она руководит **НОВЫМ** проектом

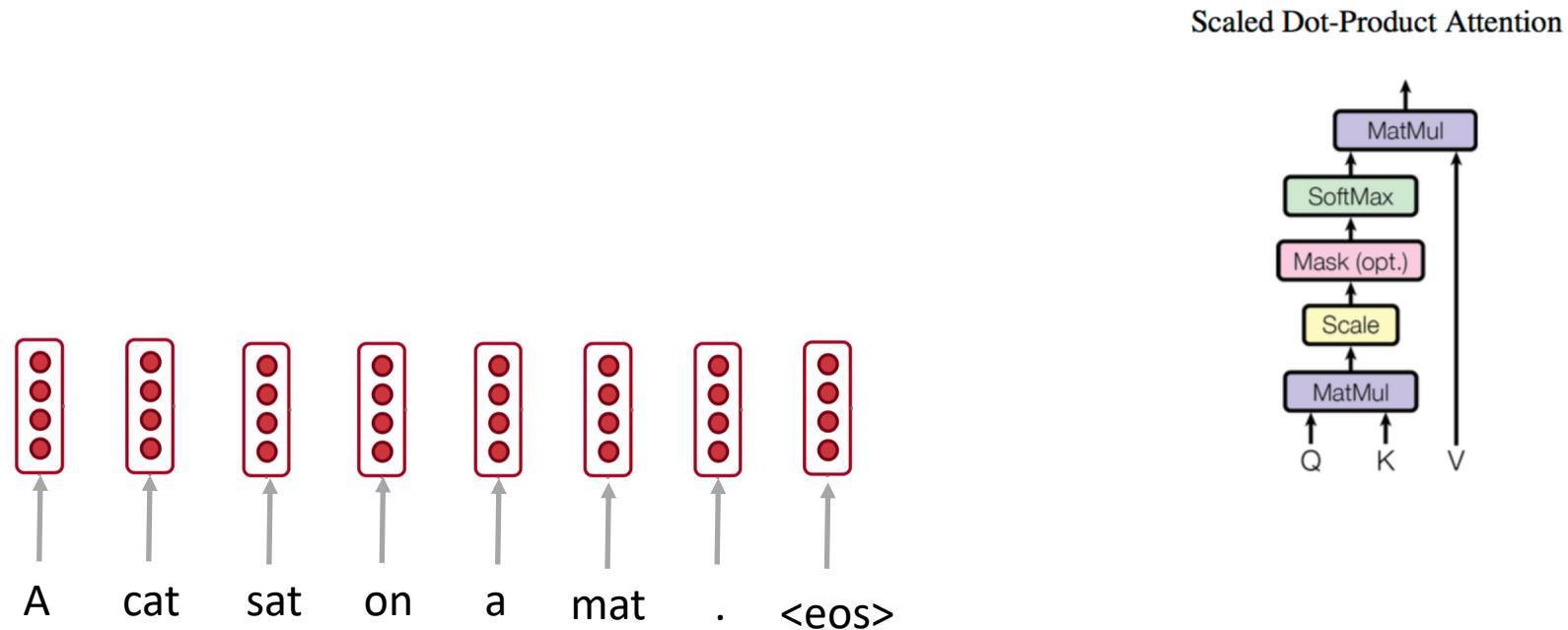
- Gender agreement
- Case government
- Lexical preferences
- ...

(example from David Talbot)



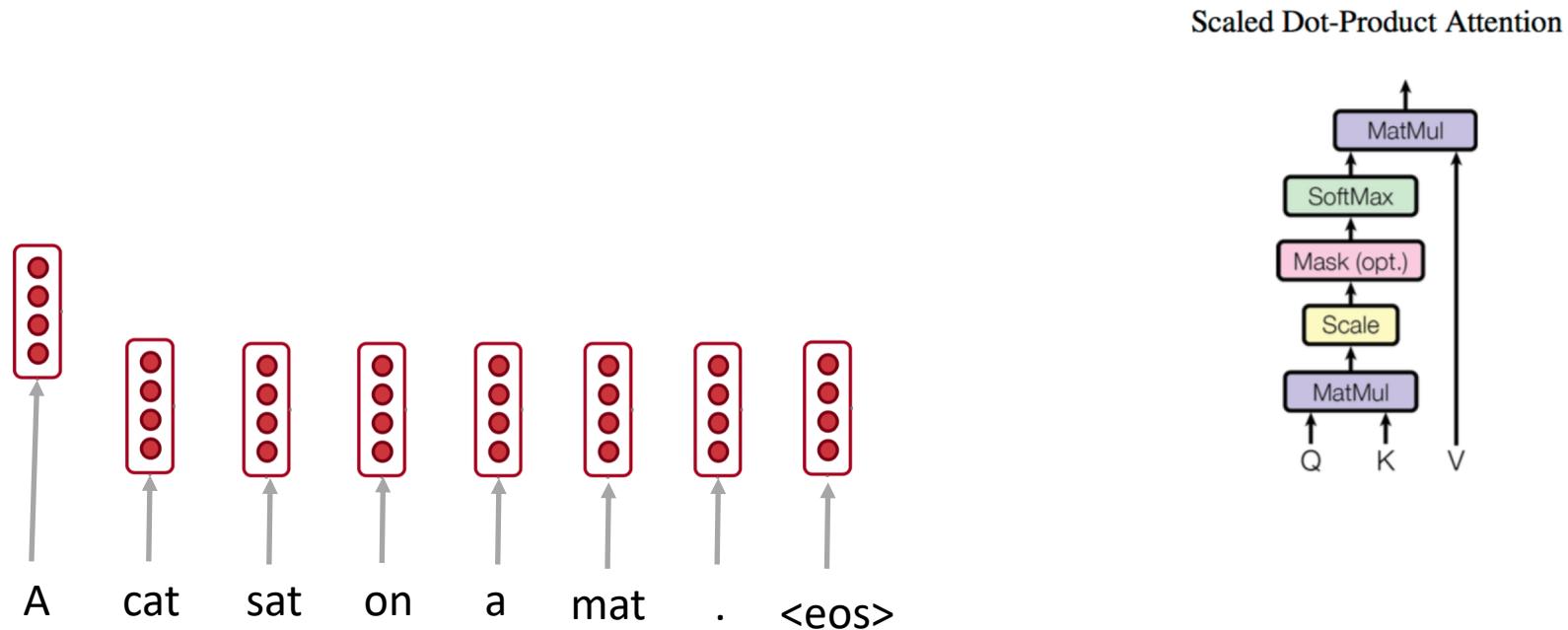
Self-Attention

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



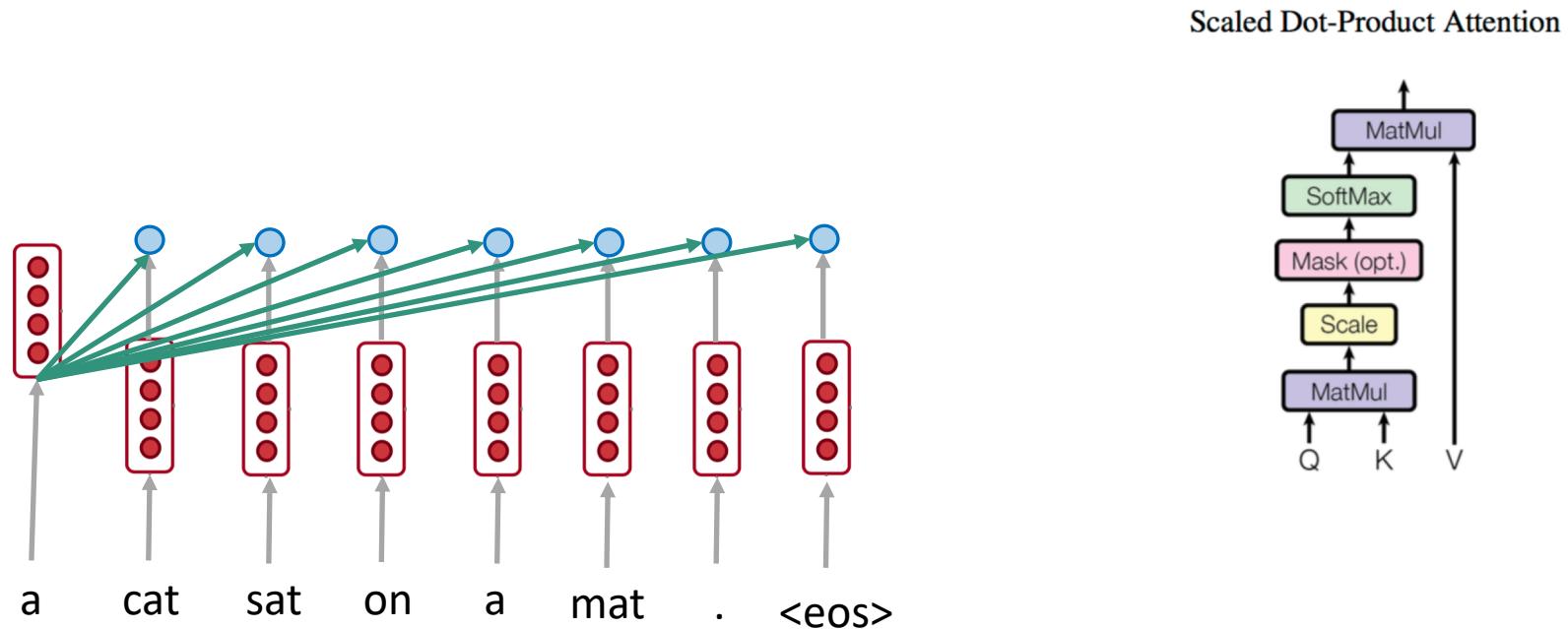
Self-Attention

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



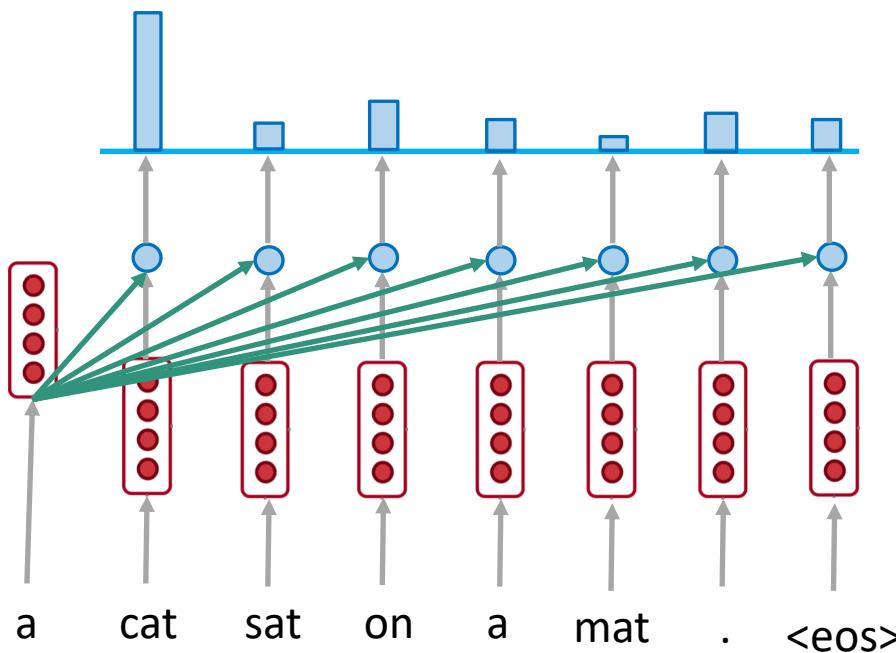
Self-Attention

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

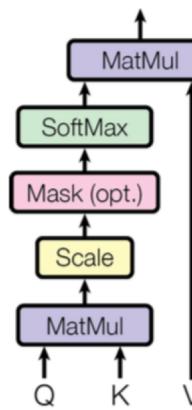


Self-Attention

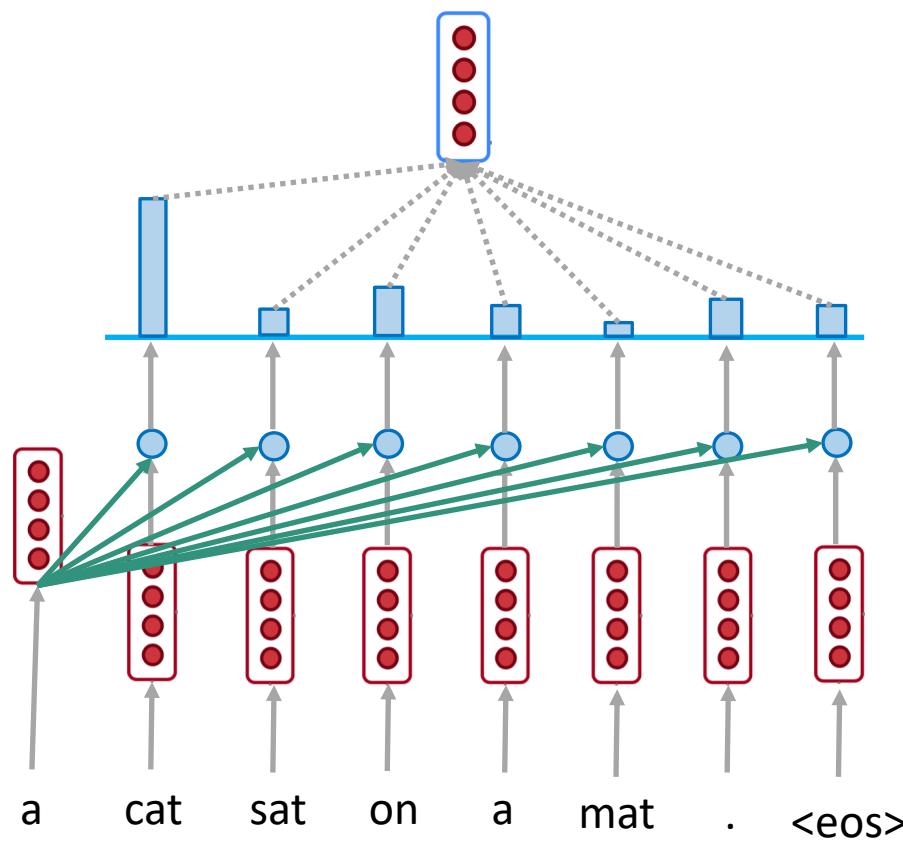
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



Scaled Dot-Product Attention

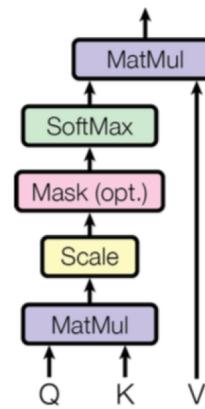


Self-Attention

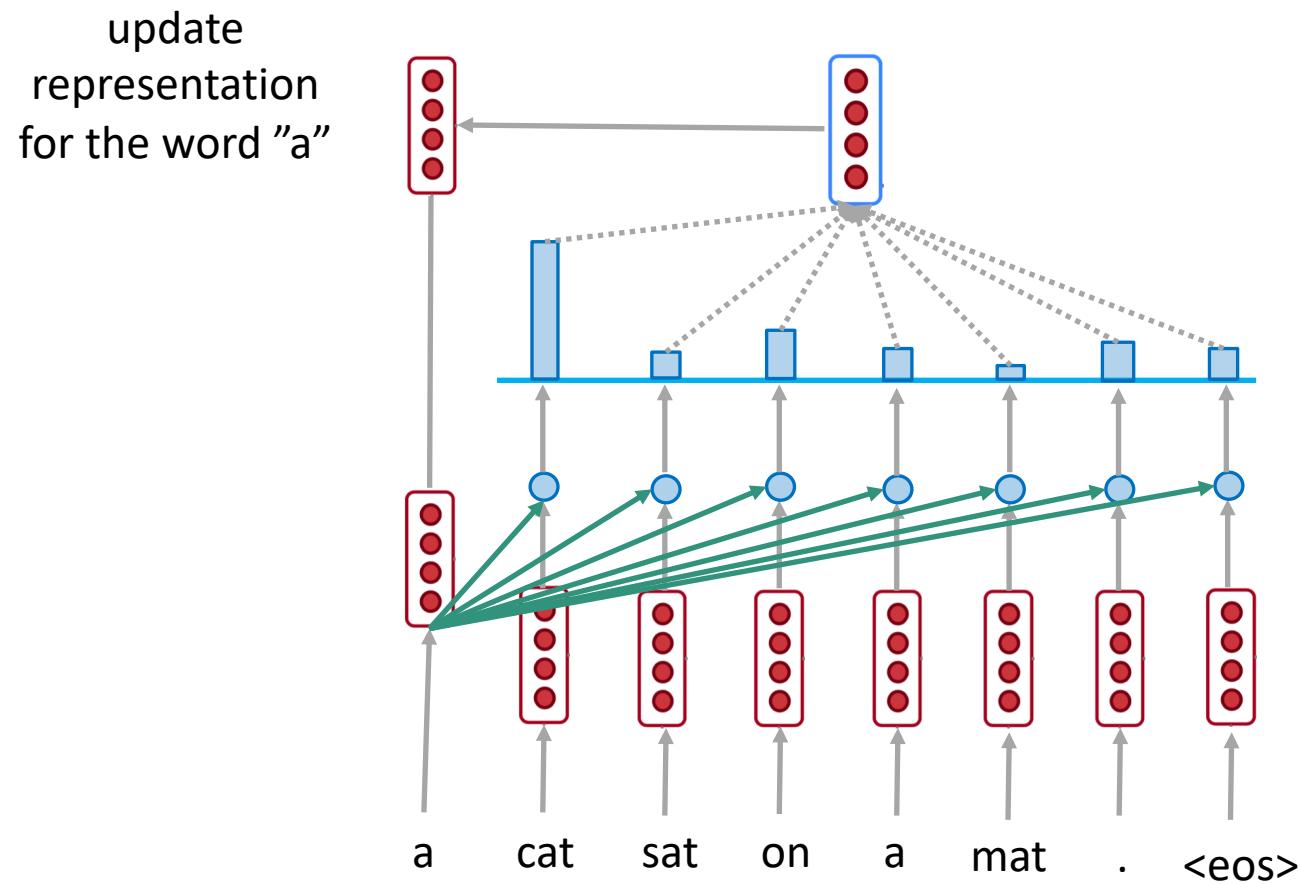


$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Scaled Dot-Product Attention

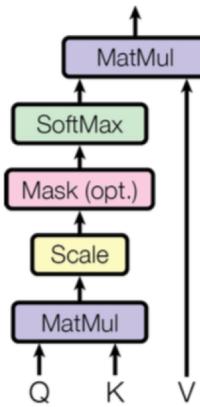


Self-Attention



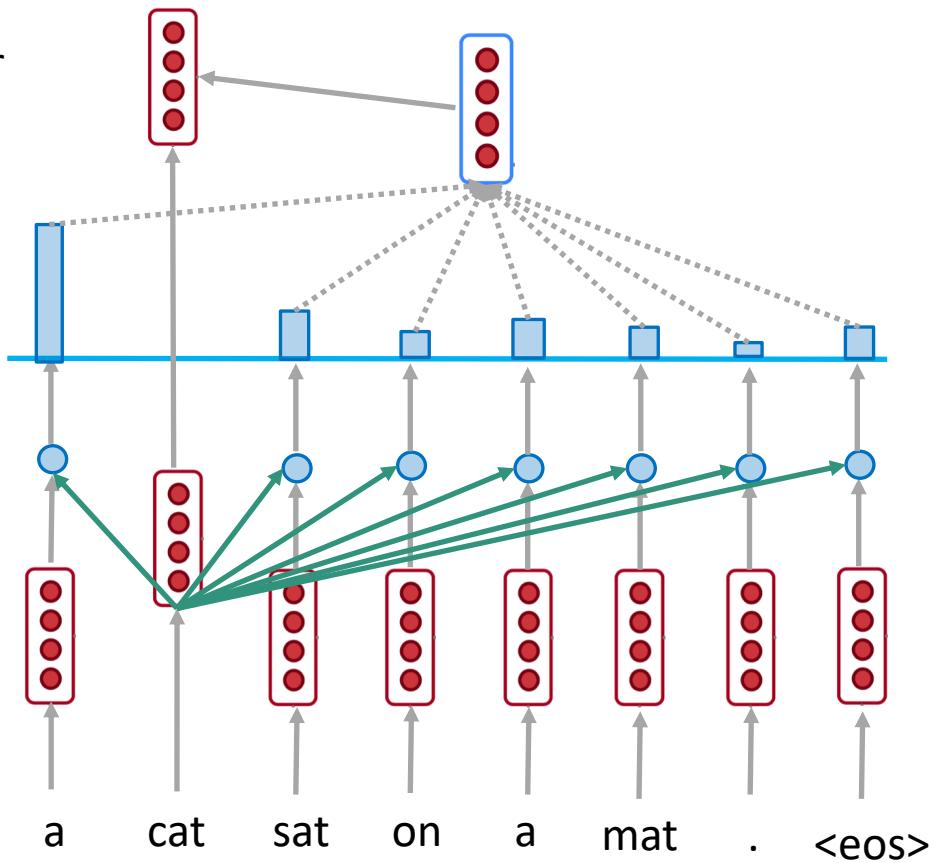
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Scaled Dot-Product Attention



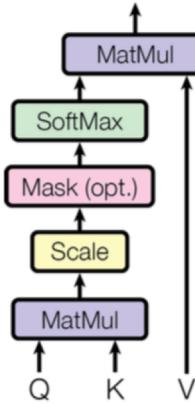
Self-Attention

update
representation for
the word "cat"



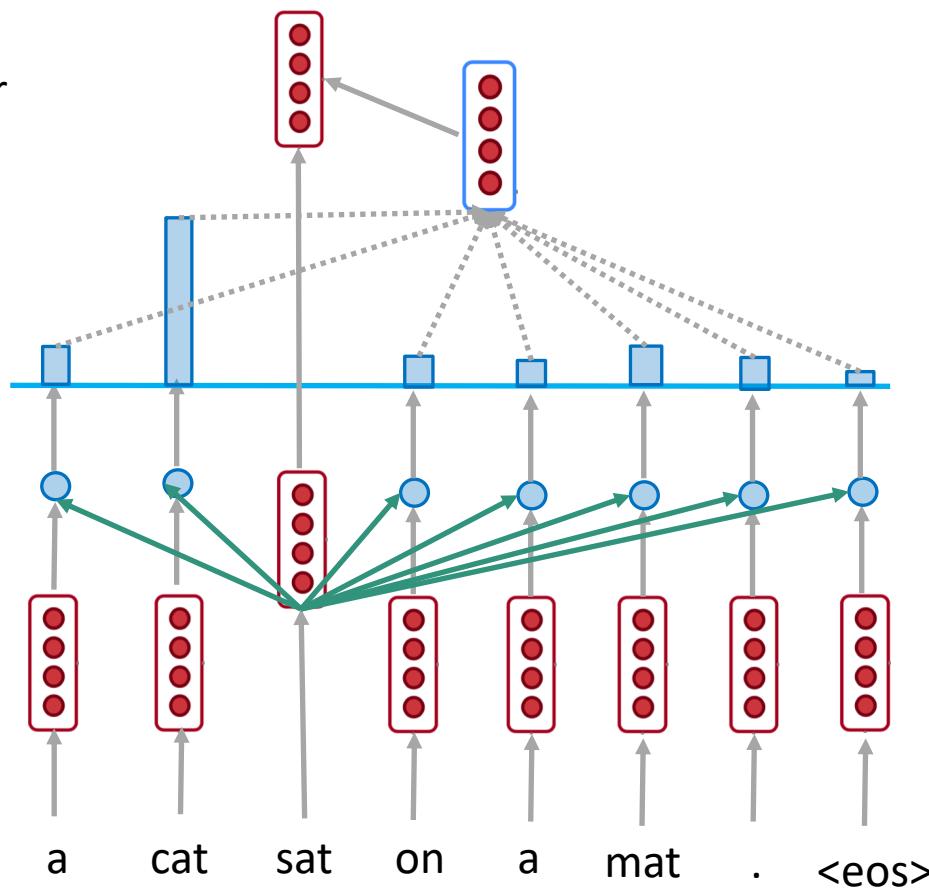
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Scaled Dot-Product Attention



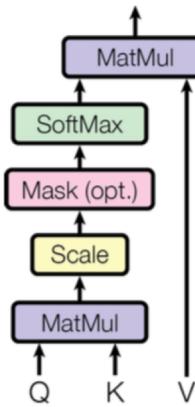
Self-Attention

update
representation for
the word "sat"



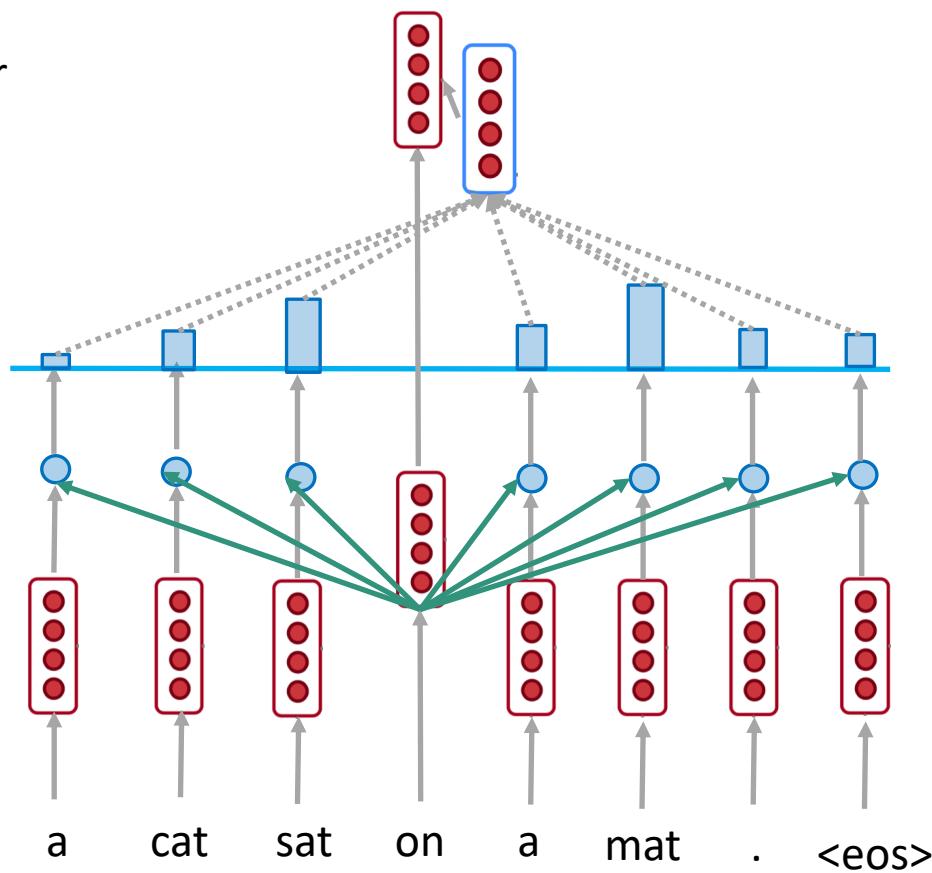
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Scaled Dot-Product Attention



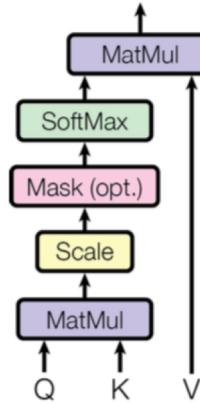
Self-Attention

update
representation for
the word "on"



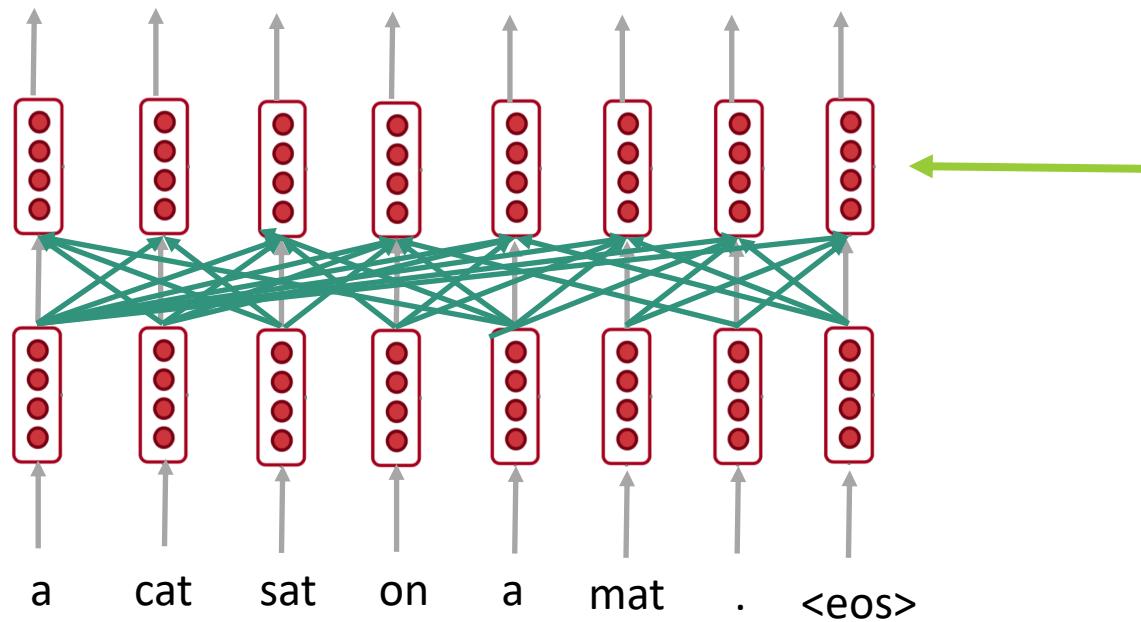
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Scaled Dot-Product Attention



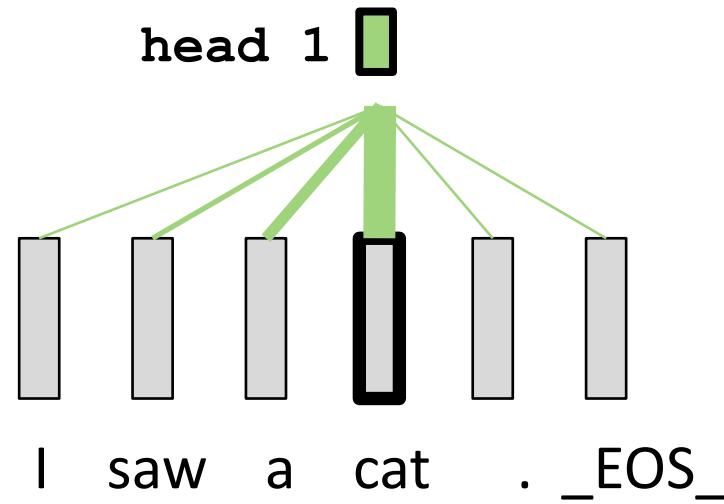
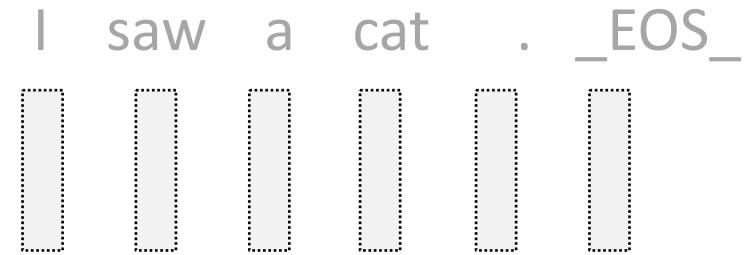
Self-Attention

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

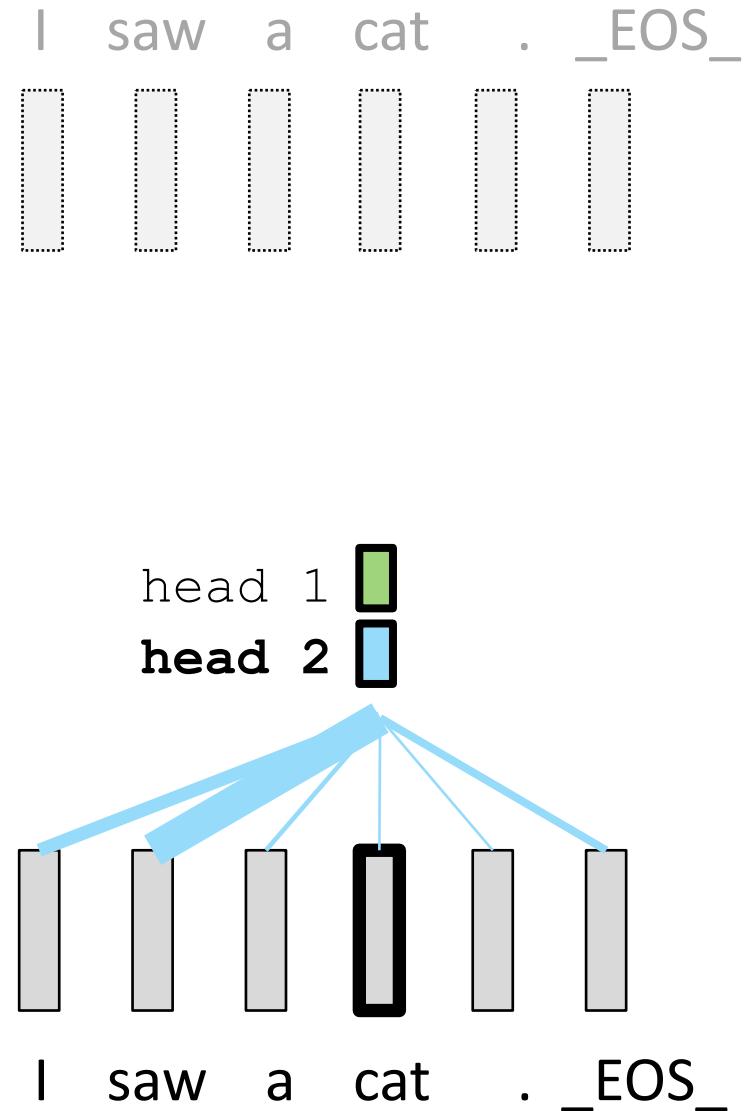


Updated representations
for each word
(one layer)

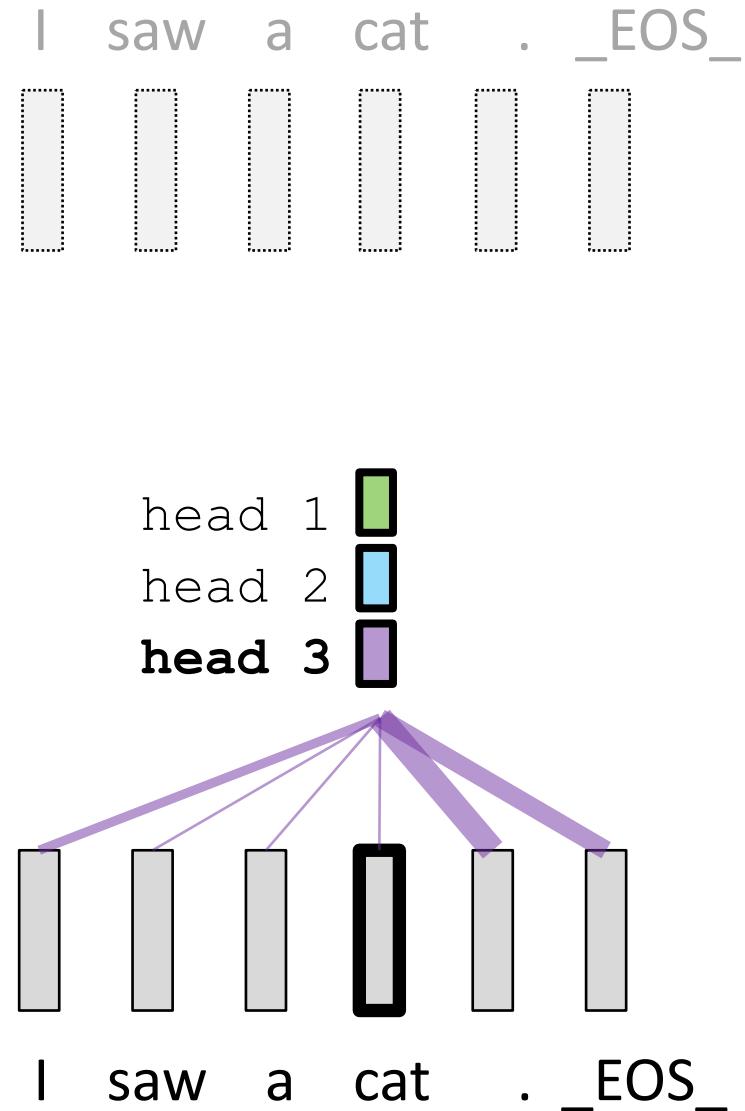
Multi-Head Attention



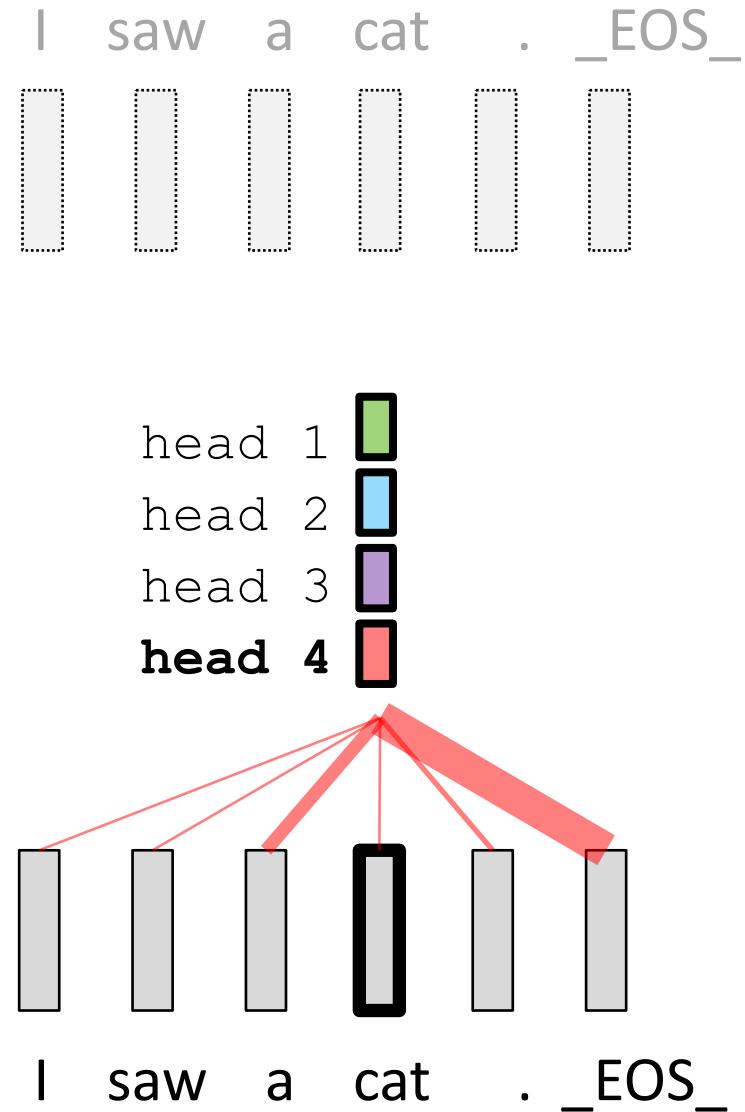
Multi-Head Attention



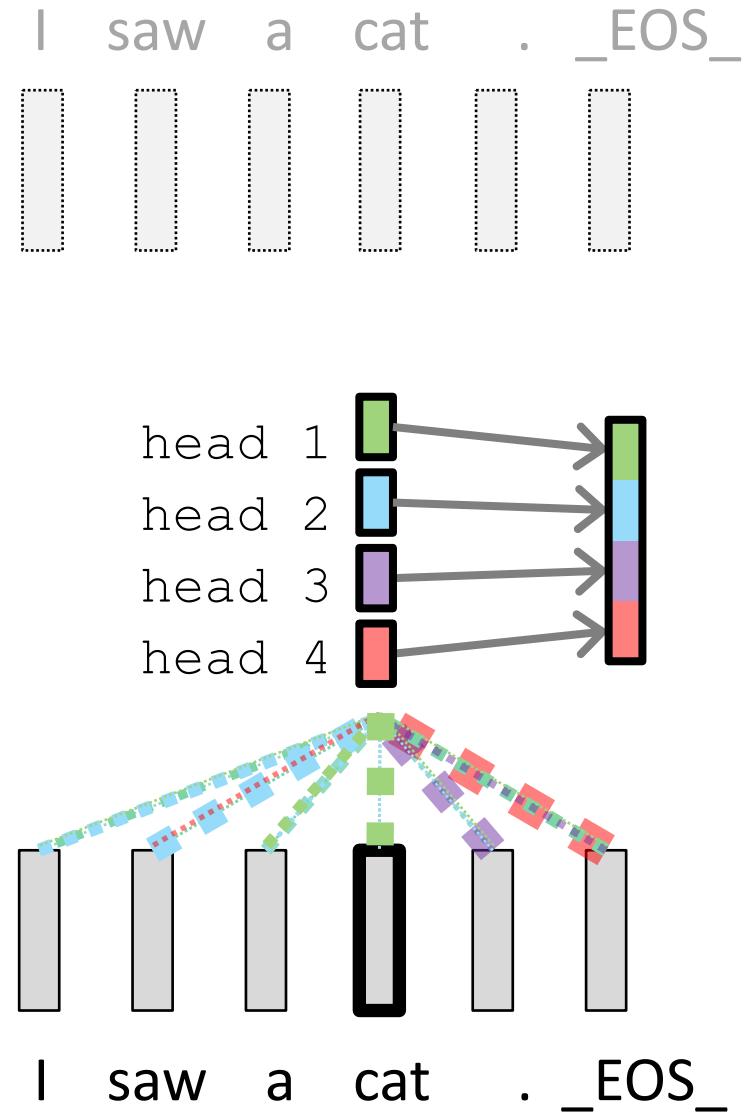
Multi-Head Attention



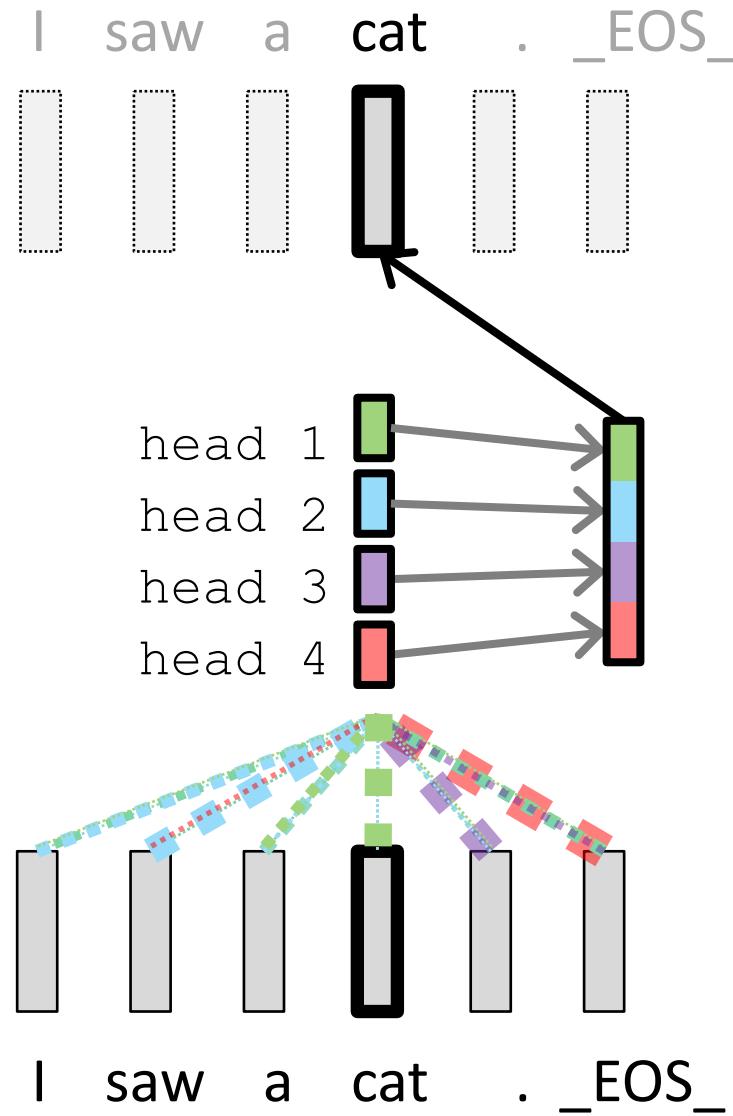
Multi-Head Attention



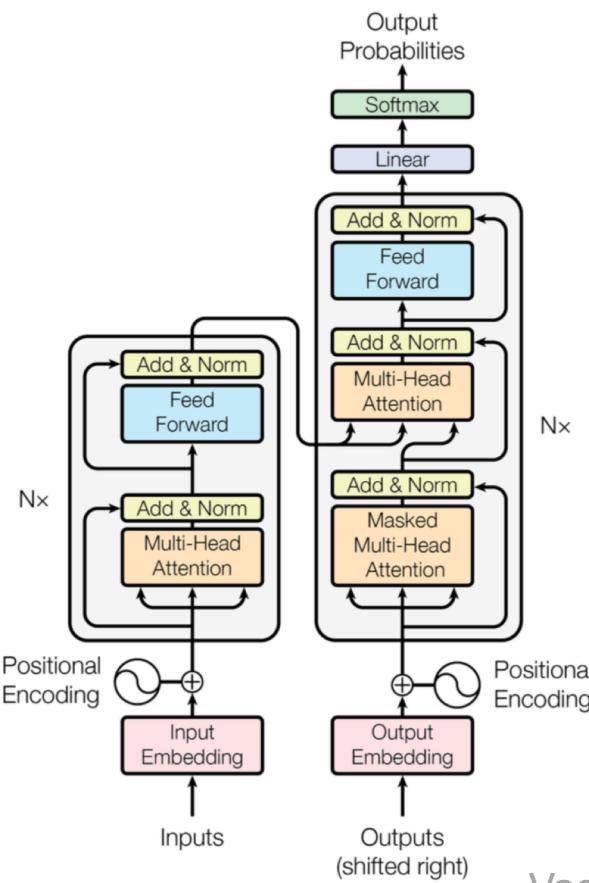
Multi-Head Attention



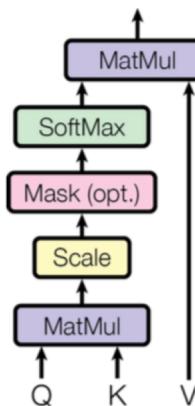
Multi-Head Attention



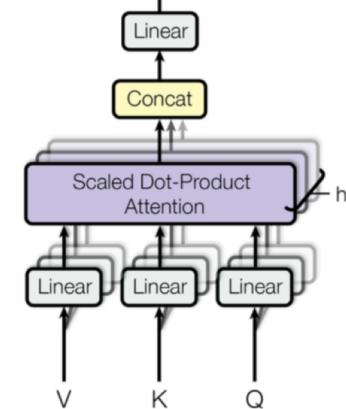
Transformer



Scaled Dot-Product Attention



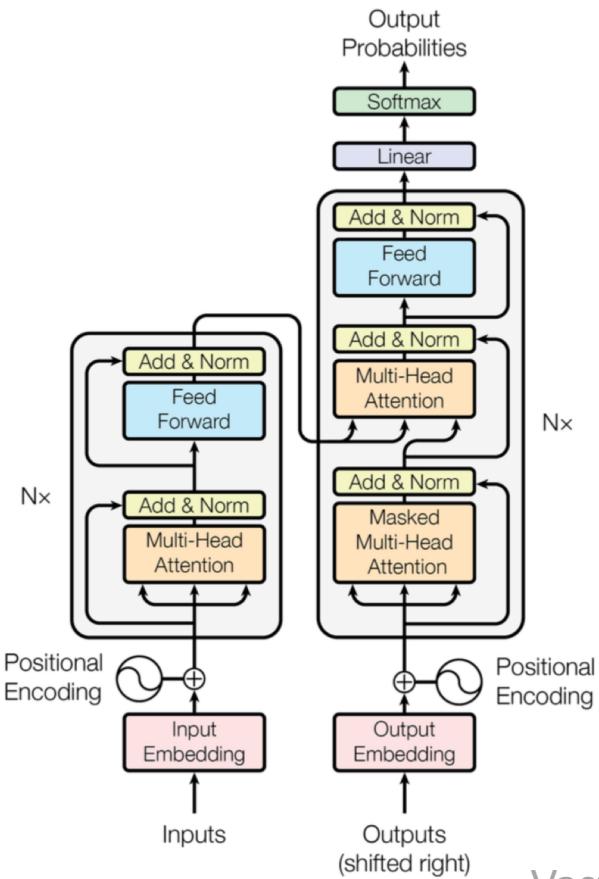
Multi-Head Attention



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

Vaswani et al, NIPS 2017, <https://papers.nips.cc/paper/7181-attention-is-all-you-need>

Transformer



Advantages:

- No recurrence: parallel encoding
- Fast training: both encoder and decoder are parallel
- No long connections: $O(1)$ for all tokens
- Three attentions: the model does not have to remember too much
- Multi-head attention allows to pay attention to different aspects

Transformer

I arrived at the

I arrived at the bank after crossing the...

... river? ...road?

Transformer

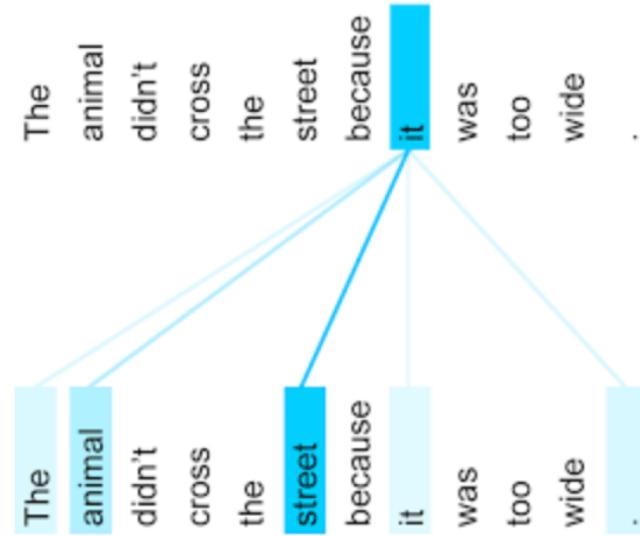
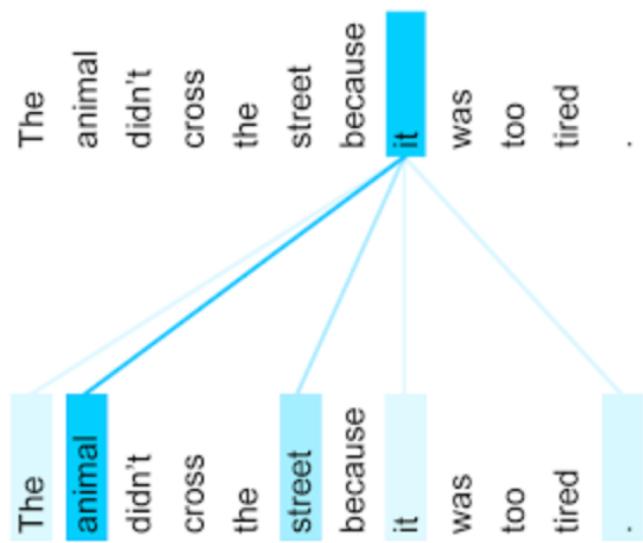
The animal didn't cross the street because it was too tired.

L'animal n'a pas traversé la rue parce qu'il était trop fatigué.

The animal didn't cross the street because it was too wide.

L'animal n'a pas traversé la rue parce qu'elle était trop large.

Transformer

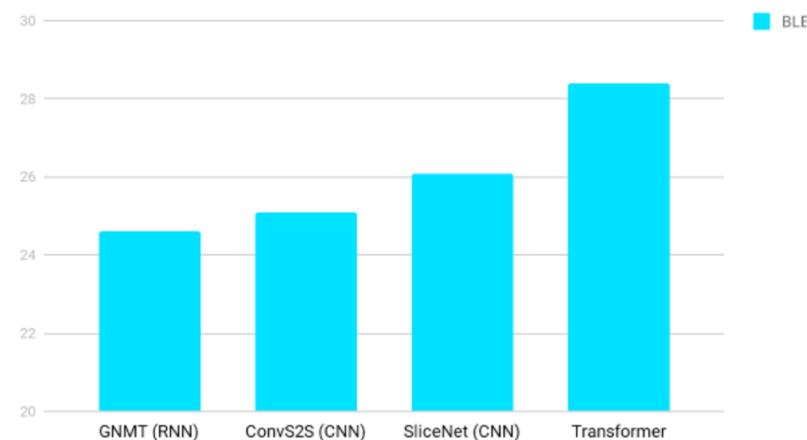


The encoder self-attention distribution for the word "it" from the 5th to the 6th layer of a Transformer trained on English to French translation (one of eight attention heads).

Vaswani et al, NIPS 2017, <https://papers.nips.cc/paper/7181-attention-is-all-you-need>

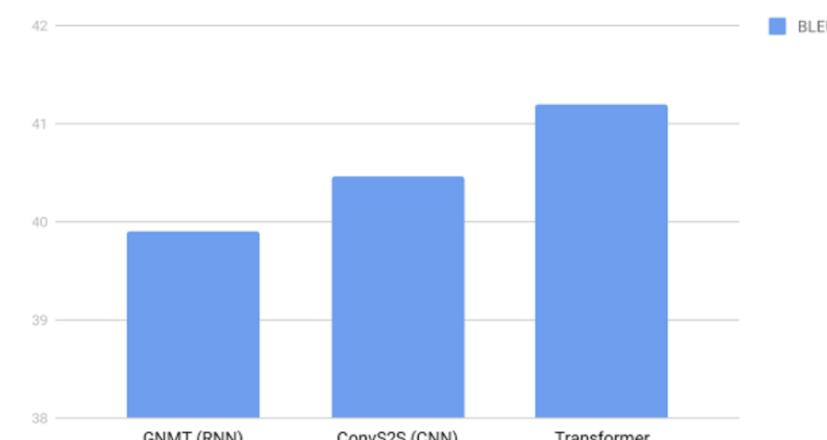
Transformer

English German Translation quality



BLEU scores (higher is better) of single models on the standard WMT newstest2014 English to German translation benchmark.

English French Translation Quality



BLEU scores (higher is better) of single models on the standard WMT newstest2014 English to French translation benchmark.

What do encoder self-attention heads learn?

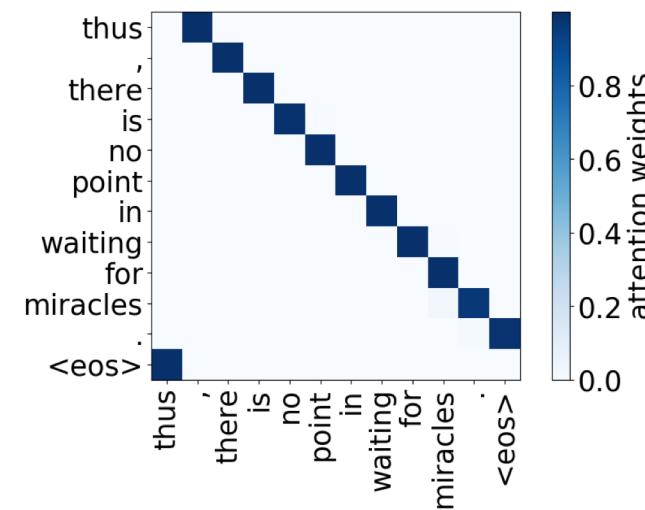
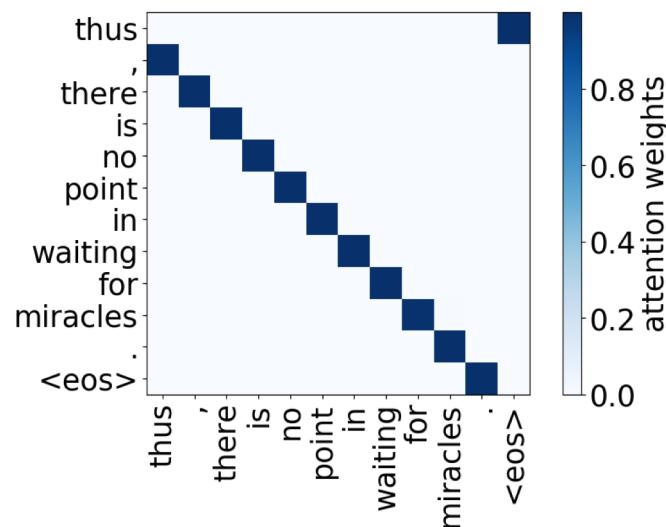
Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned

- Only a small subset of heads are important for translation;
- Important heads have one or more specialized and interpretable functions in the model;
- The vast majority of heads can be removed from the trained model without seriously affecting performance.

Voita et al, ACL 2019, <https://www.aclweb.org/anthology/P19-1580/>
Blog post: https://lena-voita.github.io/posts/acl19_heads.html

Role 1: positional heads

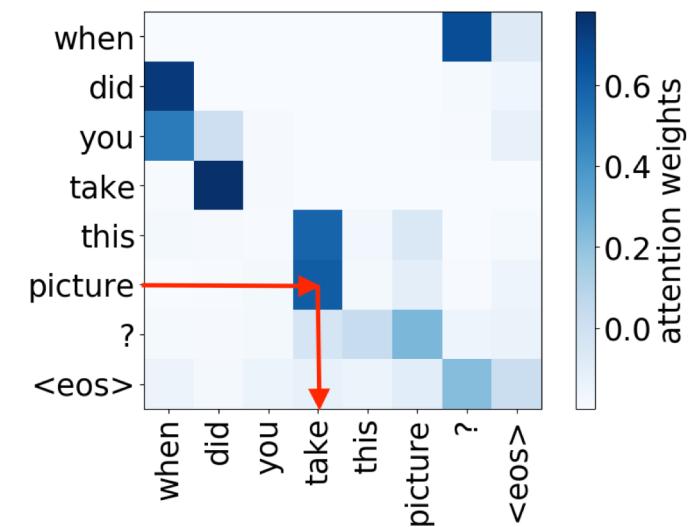
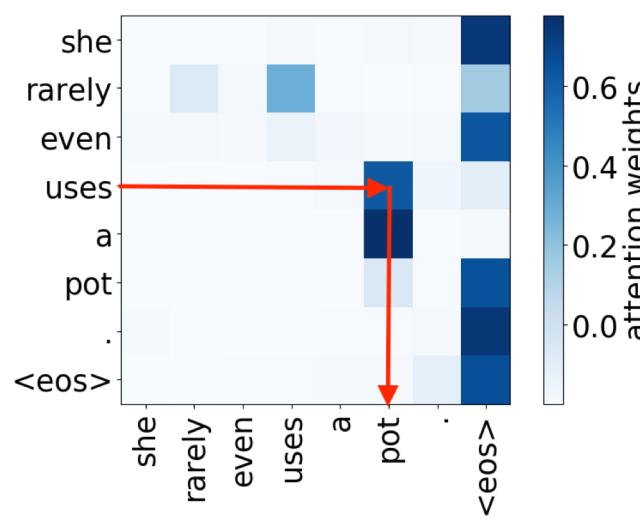
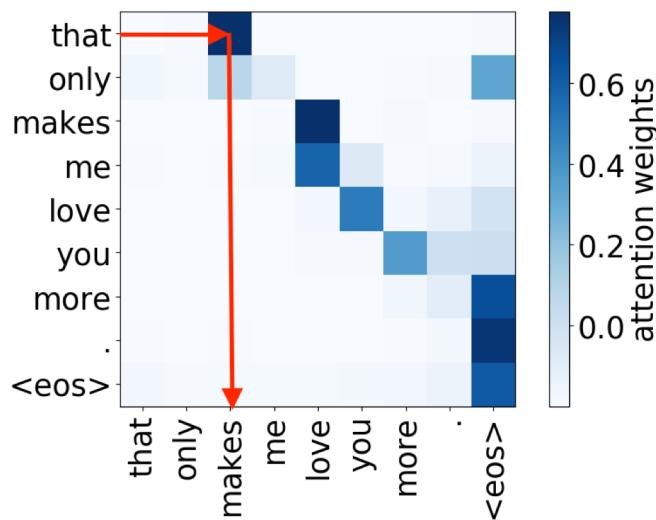
- at least 90% of the time maximum attention weight is assigned to a specific relative position (in practice either -1 or +1)



Voita et al, ACL 2019, <https://www.aclweb.org/anthology/P19-1580/>
Blog post: https://lena-voita.github.io/posts/acl19_heads.html

Role 2: syntactic heads

- heads which track syntactic relations

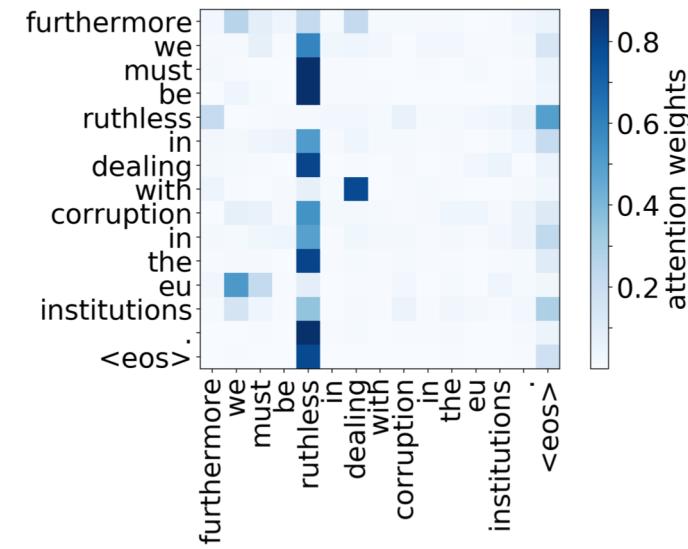
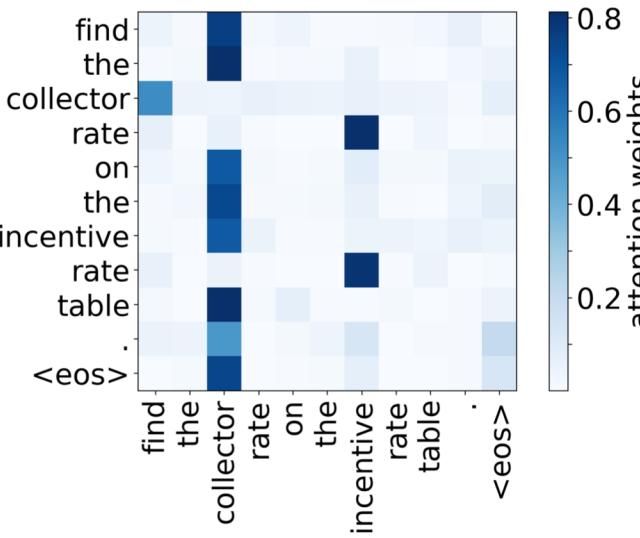
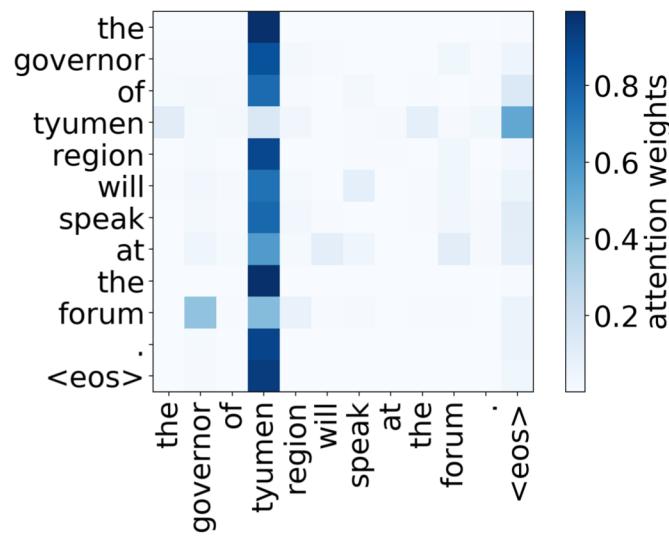


Voita et al, ACL 2019, <https://www.aclweb.org/anthology/P19-1580/>
Blog post: https://lena-voita.github.io/posts/acl19_heads.html

Role 3: attention to rare tokens

- In more than 50% of the cases, maximum attn weight points to one of the two least frequent tokens

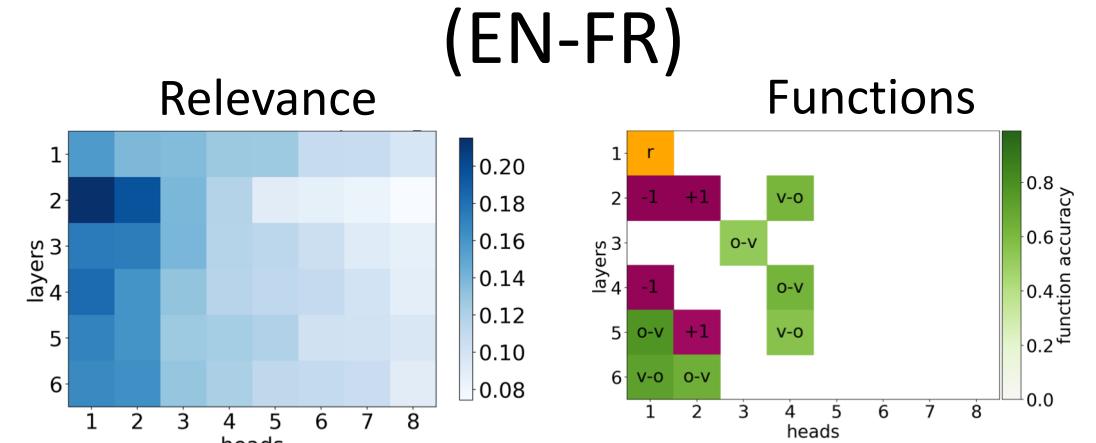
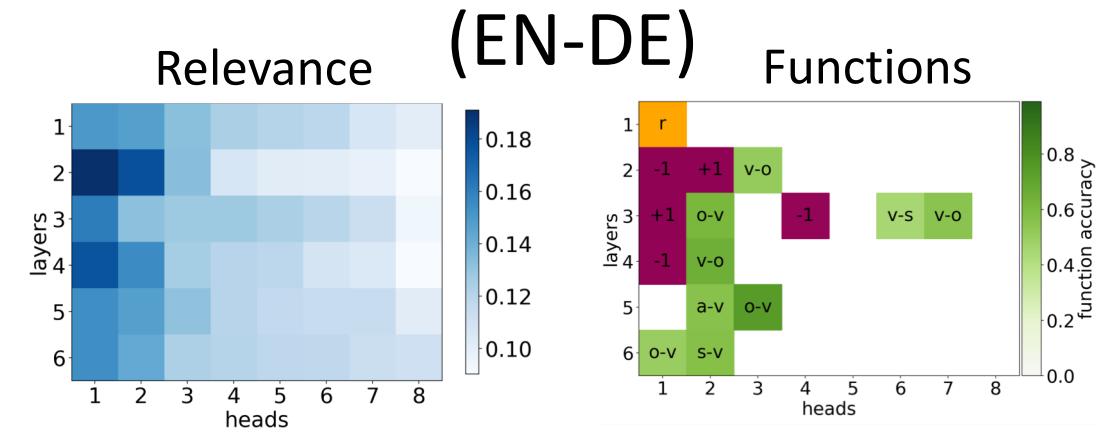
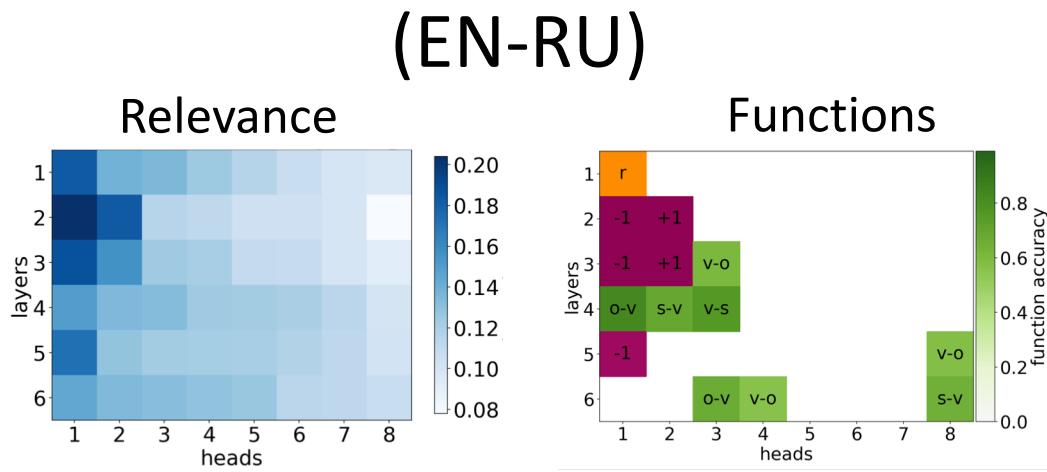
Models trained on WMT: (a) EN-RU, (b) EN-DE, (c) EN-FR



Voita et al, ACL 2019, <https://www.aclweb.org/anthology/P19-1580/>
Blog post: https://lena-voita.github.io/posts/acl19_heads.html

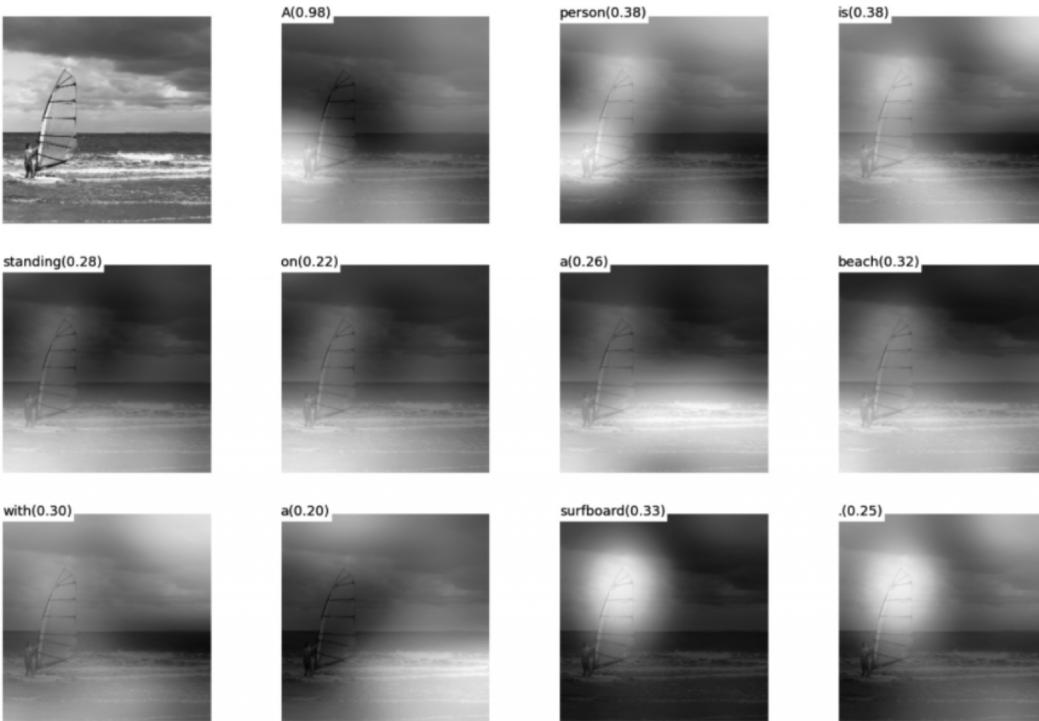
Head roles: different language pairs

- positionnal
- syntactic
- rare tokens



Attention: Other use cases

Image caption generation



(b) A person is standing on a beach with a surfboard.

Use a Convolutional Neural Network to “encode” the image, and a Recurrent Neural Network with attention mechanisms to generate a description.

Question answering task

by *ent423* ,*ent261* correspondent updated 9:49 pm et ,thu march 19 ,2015 (*ent261*) a *ent114* was killed in a parachute accident in *ent45* ,*ent85* ,near *ent312* ,a *ent119* official told *ent261* on wednesday .he was identified thursday as special warfare operator 3rd class *ent23* ,29 ,of *ent187* ,*ent265* .`` *ent23* distinguished himself consistently throughout his career .he was the epitome of the quiet professional in all facets of his life ,and he leaves an inspiring legacy of natural tenacity and focused

...

ent119 identifies deceased sailor as **X** ,who leaves behind a wife

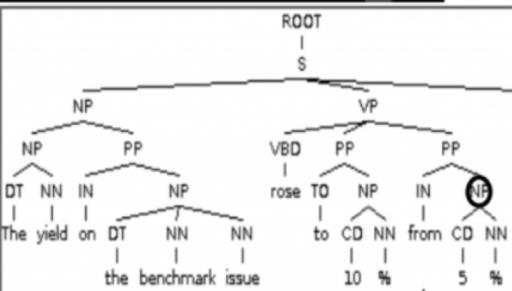
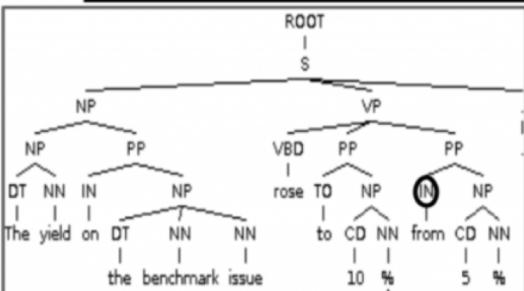
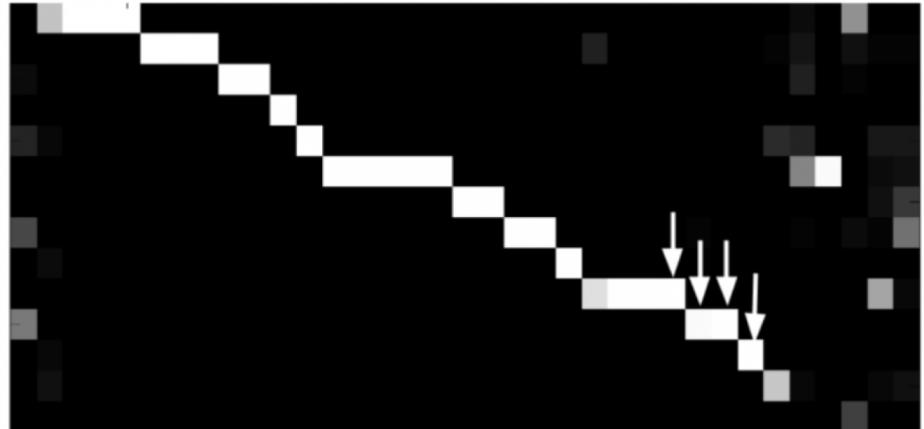
by *ent270* ,*ent223* updated 9:35 am et ,mon march 2 ,2015 (*ent223*) *ent63* went familial for fall at its fashion show in *ent231* on sunday ,dedicating its collection to `` mamma '' with nary a pair of `` mom jeans '' in sight .*ent164* and *ent21* ,who are behind the *ent196* brand ,sent models down the runway in decidedly feminine dresses and skirts adorned with roses ,lace and even embroidered doodles by the designers ' own nieces and nephews .many of the looks featured saccharine needlework phrases like `` i love you ,

...

X dedicated their fall fashion show to moms

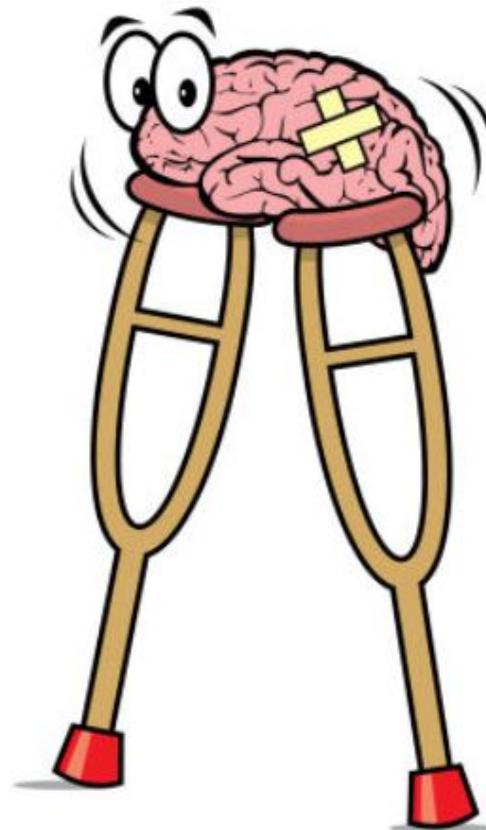
Use a RNN to read a text, read a (synthetically generated) question, and then produce an answer.

Generate sentence parse trees

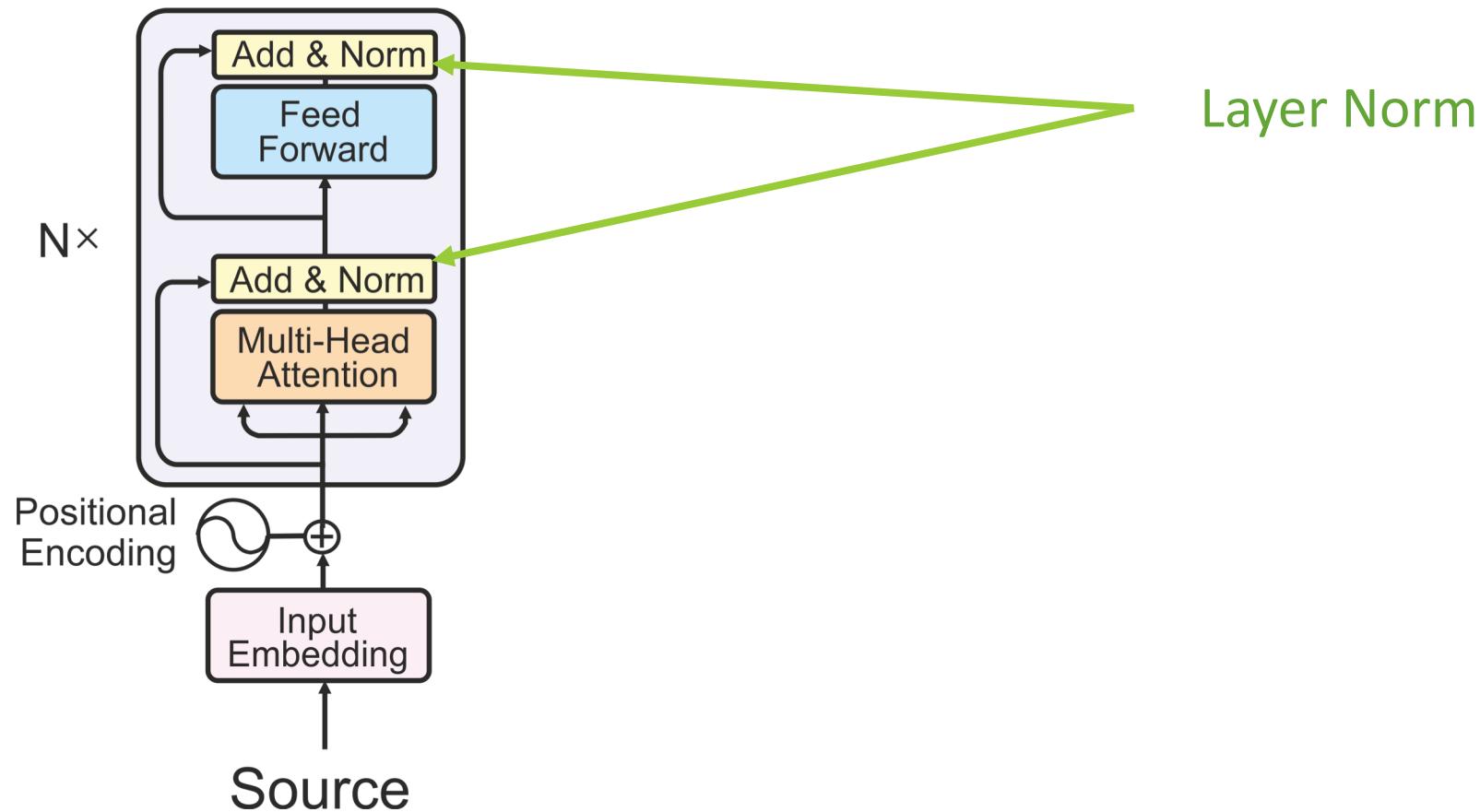


Use a Recurrent Neural Network with attention mechanism to generate sentence parse trees

Hack of the day



Layer Norm



Layer norm

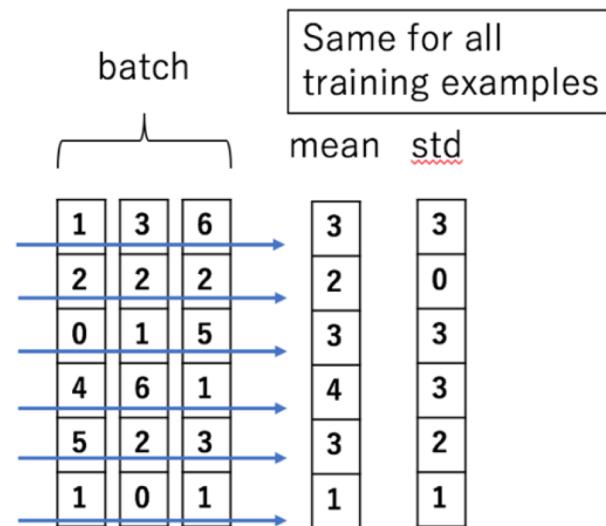
Batch normalization

$$\begin{aligned}\mu_j &= \frac{1}{m} \sum_{i=1}^m x_{ij} \\ \sigma_j^2 &= \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2 \\ \hat{x}_{ij} &= \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}\end{aligned}$$

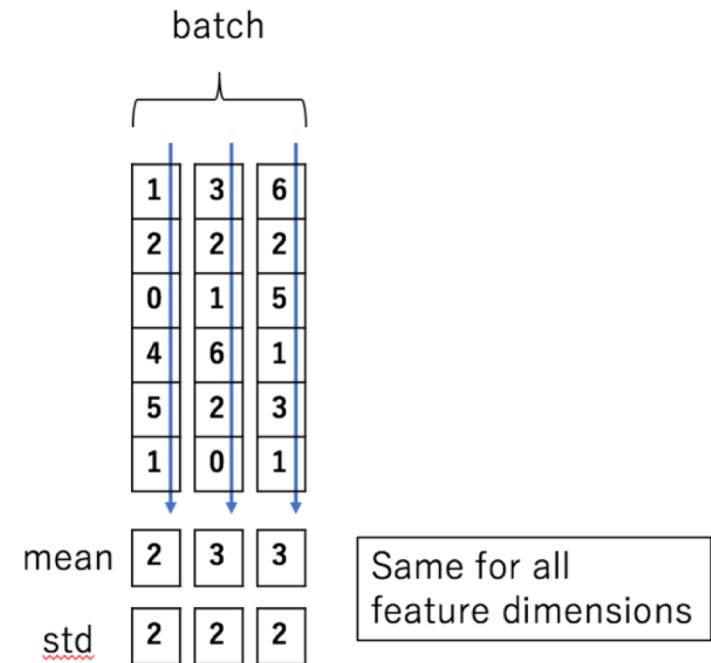
Layer normalization:

$$\begin{aligned}\mu_i &= \frac{1}{m} \sum_{j=1}^m x_{ij} \\ \sigma_i^2 &= \frac{1}{m} \sum_{j=1}^m (x_{ij} - \mu_i)^2 \\ \hat{x}_{ij} &= \frac{x_{ij} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}\end{aligned}$$

Batch Normalization



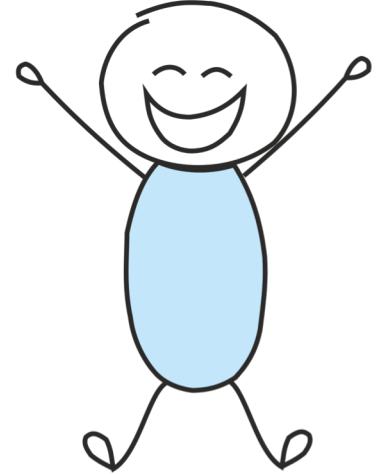
Layer Normalization



Paper: Ba et al, 2016, <https://arxiv.org/abs/1607.06450>

Pictures: <http://mlexplained.com/2018/01/13/weight-normalization-and-layer-normalization-explained-normalization-in-deep-learning-part-2/>

That's all for today!



Coming soon:

- EM, Machine Translation (specific)
- Structured prediction
- Speech
- Dialogue systems
- and much, much more

Sincerely yours,
Yandex Research