Natural Language Processing: Seq2seq and Attention

HSE Faculty of Computer Science

Machine Learning and Data-Intensive Systems

Murat Khazhgeriev

Table of Content

- **Sequence to Sequence (seq2seq)**
- Attention
- Practical tips

Machine translation requires training parameters to provide results

Human Translation

$$y^* = \arg\max_y p(y|x)$$

The "probability" is intuitive and is given by a human translator's expertise

Machine Translation

model parameters

$$y' = \arg\max_y p(y|x, \theta)$$

Questions we need to answer

- **modeling**

  How does the model for $p(y|x,\theta)$ look like?

- **learning**

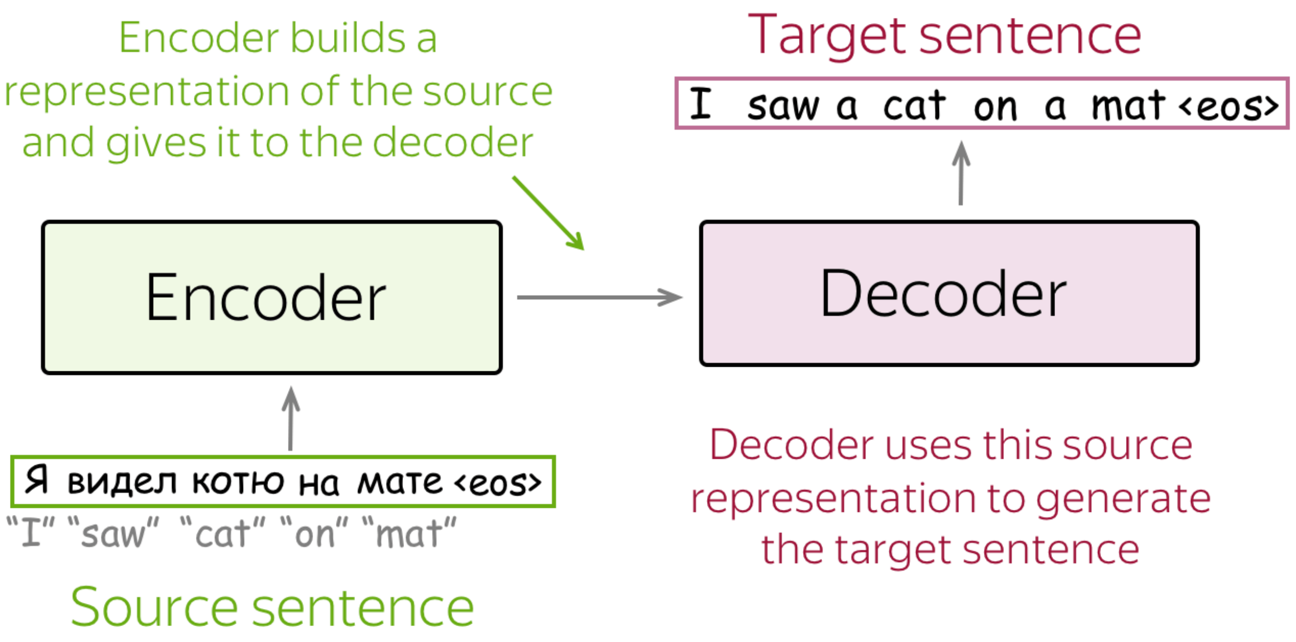
  How to find θ?

- **search**

  How to find the argmax?

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

Encoder-decoder architecture maps data semantics



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

Encoder encapsulates a condition for a decoder Language Model

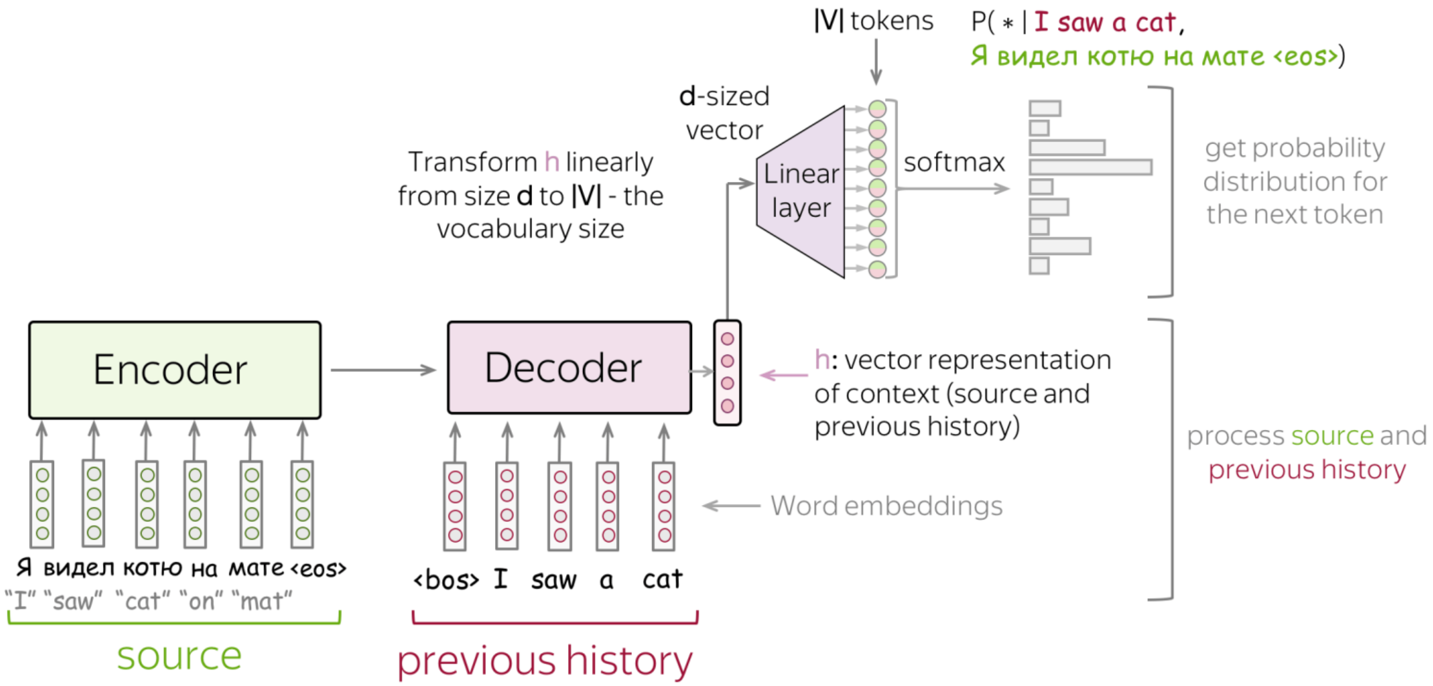
Language Models: $P(y_1, y_2, \ldots, y_n) = \prod_{t=1}^{n} p(y_t | y_{<t})$

Conditional Language Models: $P(y_1, y_2, \ldots, y_n, | x) = \prod_{t=1}^{n} p(y_t | y_{<t}, x)$

condition on source x

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

A helicopter view on the Encoder-Decoder architecture



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

Model Loss is a well-known Cross-Entropy

Source sequence:

Я видел котю на мате <eos>
"I" "saw" "cat" "on" "mat"

Target sequence:

I saw a cat on a mat <eos>

previous tokens

we want the model to predict this

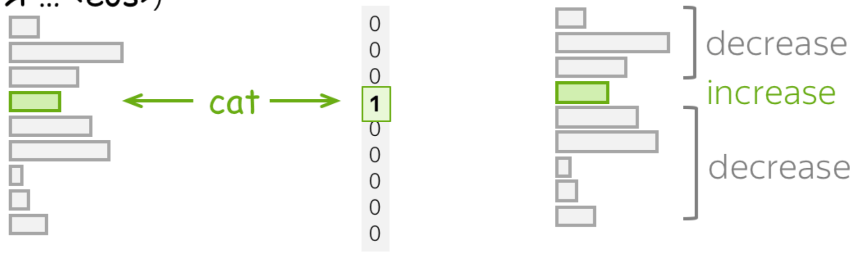
← one training example

← one step for this example

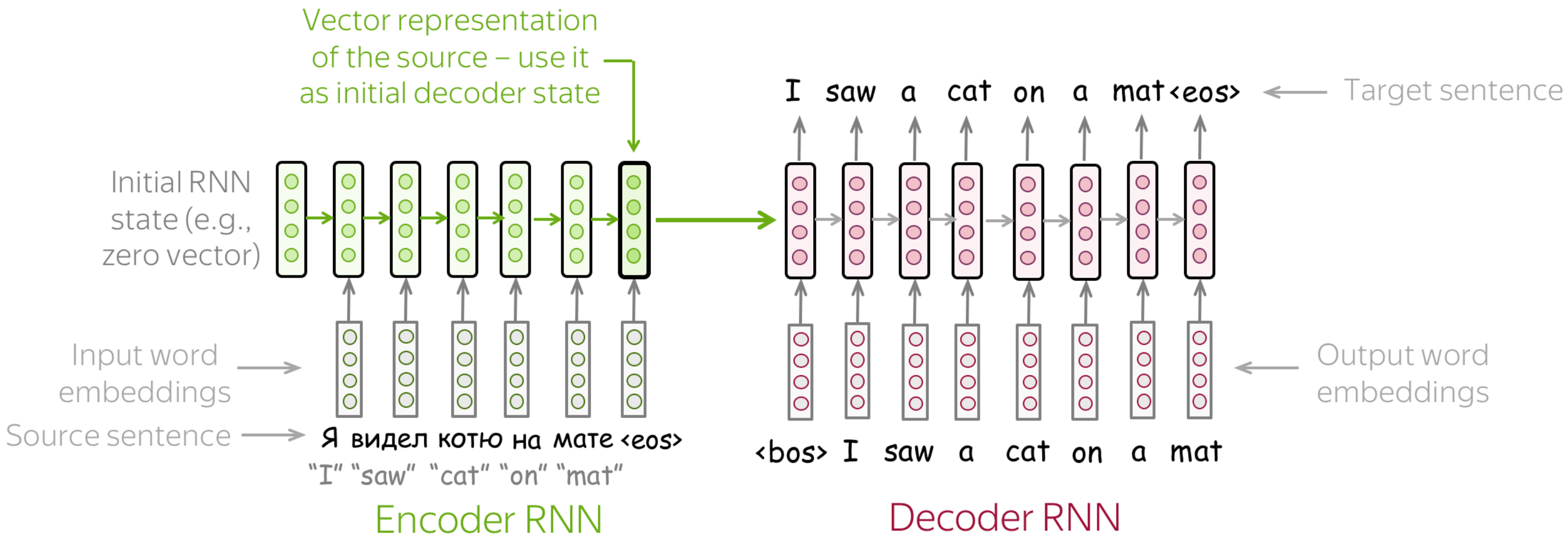
Model prediction: p(* |I saw a, Я … <eos>)

Target

Loss = -log (p(cat)) → min

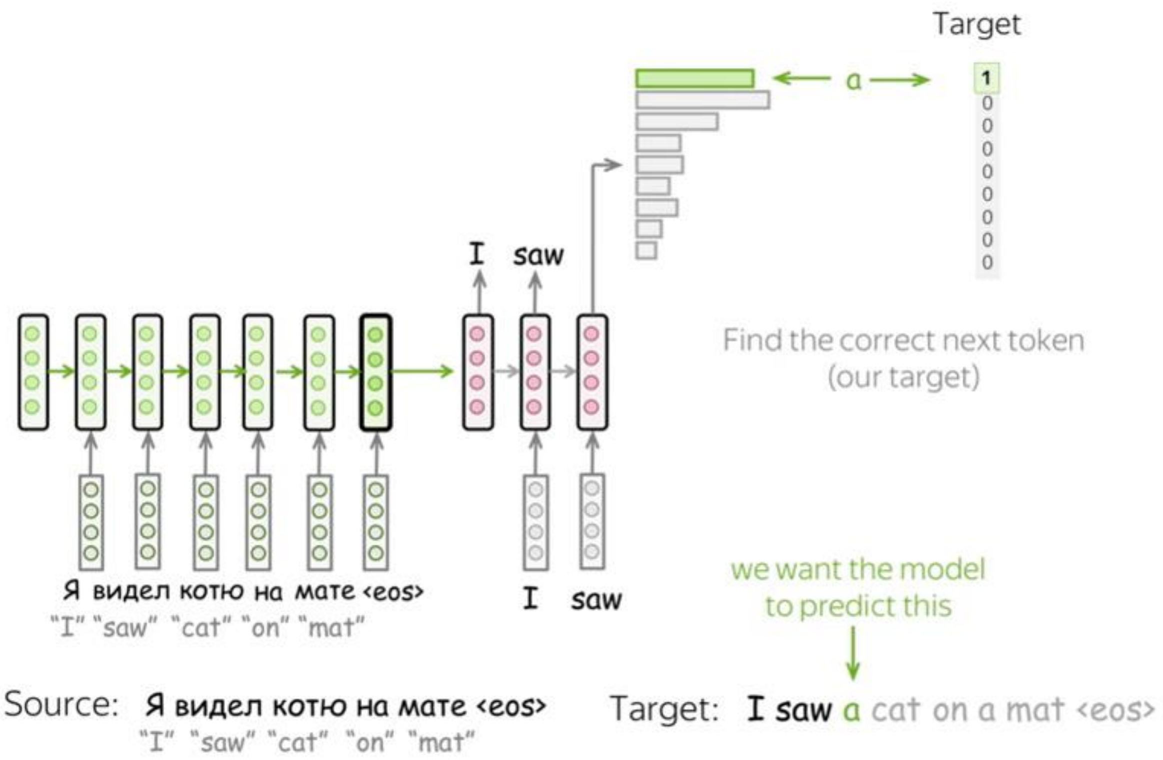


A simple RNN is a valid Encoder-Decoder model



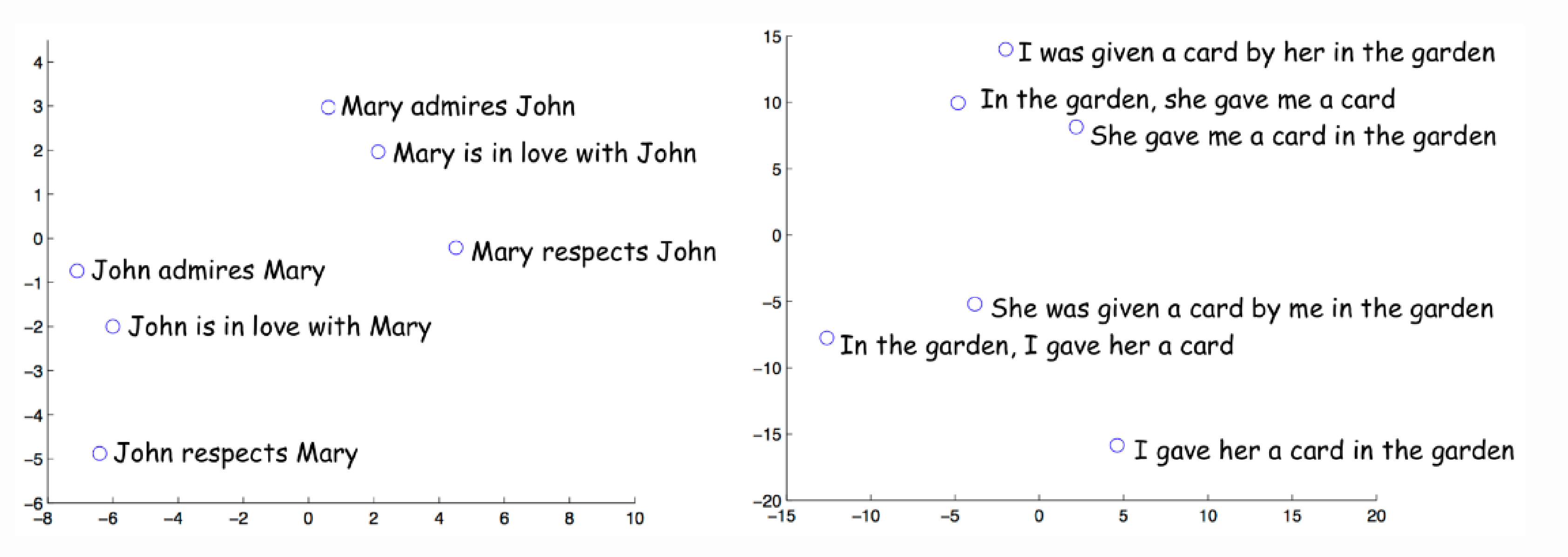
Source: https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

A simple RNN is a valid Encoder-Decoder model



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

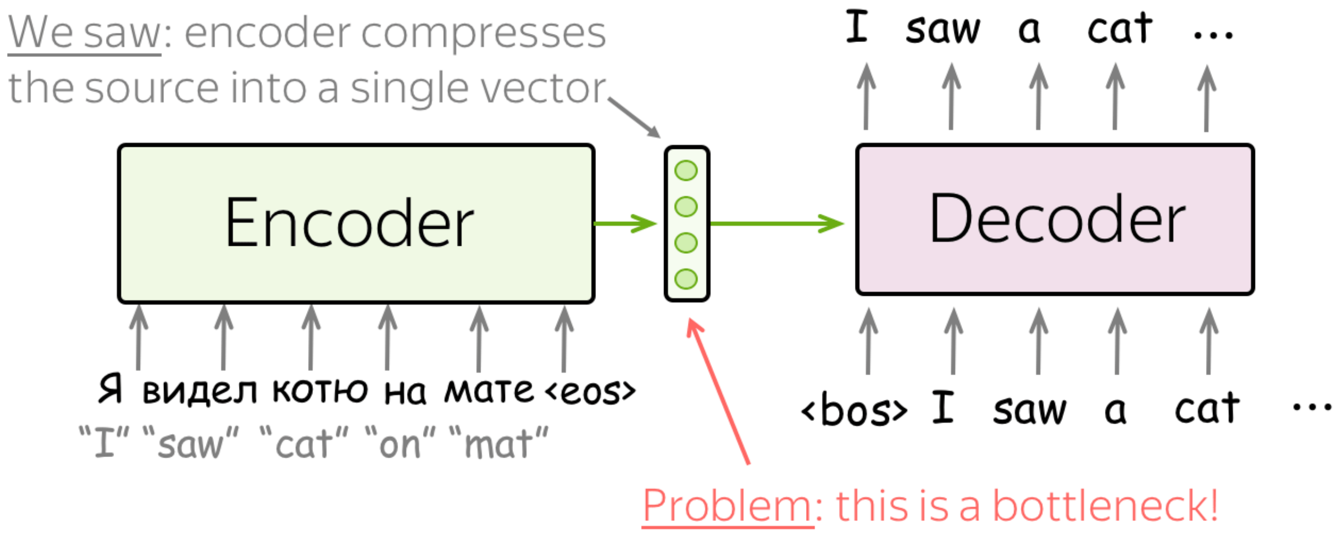
Semantic space of text embeddings



Source: Sequence to Sequence Learning with Neural Networks.

Table of Content

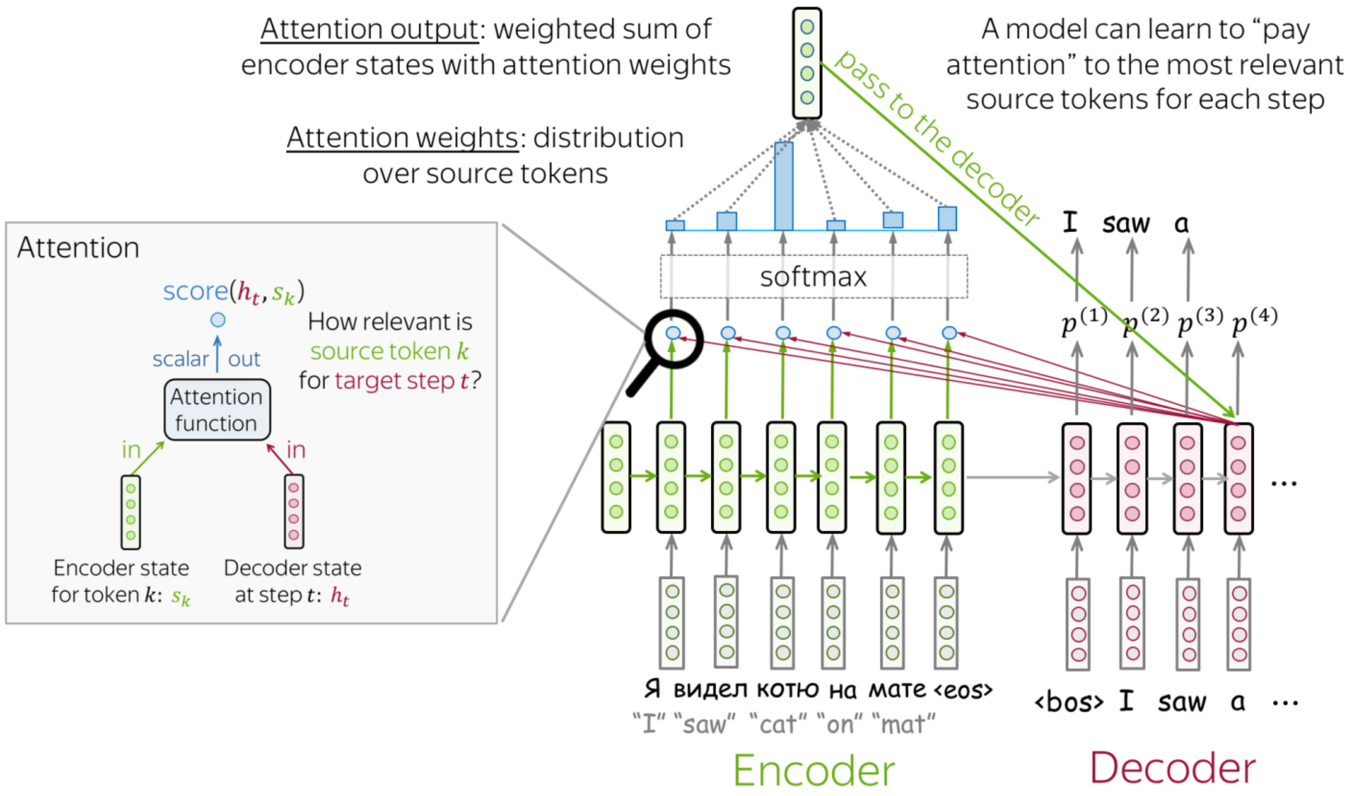
The final hidden state is a bottleneck



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

A simple Attention overview

$$\begin{bmatrix} - h_1^e - \\ - h_i^e - \\ \vdots \\ - h_t^e - \end{bmatrix} \quad d \quad h_t$$



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

Encoder hidden states are weighed according to their attention score

Attention output

$$c^{(t)} = a_1^{(t)} s_1 + a_2^{(t)} s_2 + \dots + a_m^{(t)} s_m = \sum_{k=1}^{m} a_k^{(t)} s_k$$

"source context for decoder step t"

(weighted sum)

Attention weights

$$a_k^{(t)} = \frac{\exp(\text{score}(h_t, s_k))}{\sum_{i=1}^{m} \exp(\text{score}(h_t, s_i))}, \mathrm{k} = 1..\mathrm{m}$$

"attention weight for source token k at decoder step t"

(softmax)

Attention scores

$$\text{score}(h_t, s_k), \mathrm{k} = 1..\mathrm{m}$$

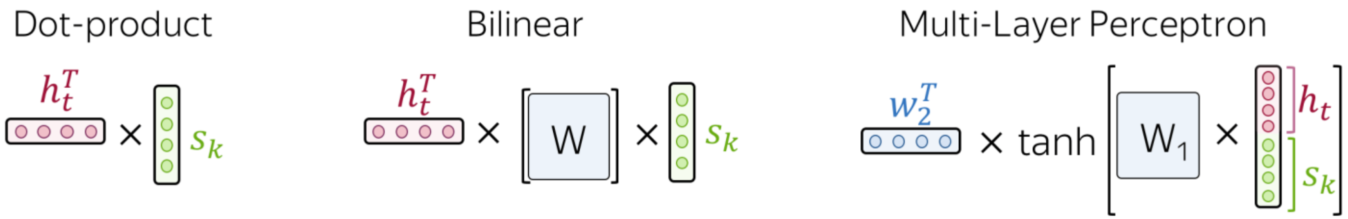
"How relevant is source token k for target step t?"

Attention input

$s_1, s_2, \dots, s_m$ — all encoder states

h_t — one decoder state

Encoder hidden states are weighed according to their attention score



score$(h_t, s_k) = h_t^T\ s_k$

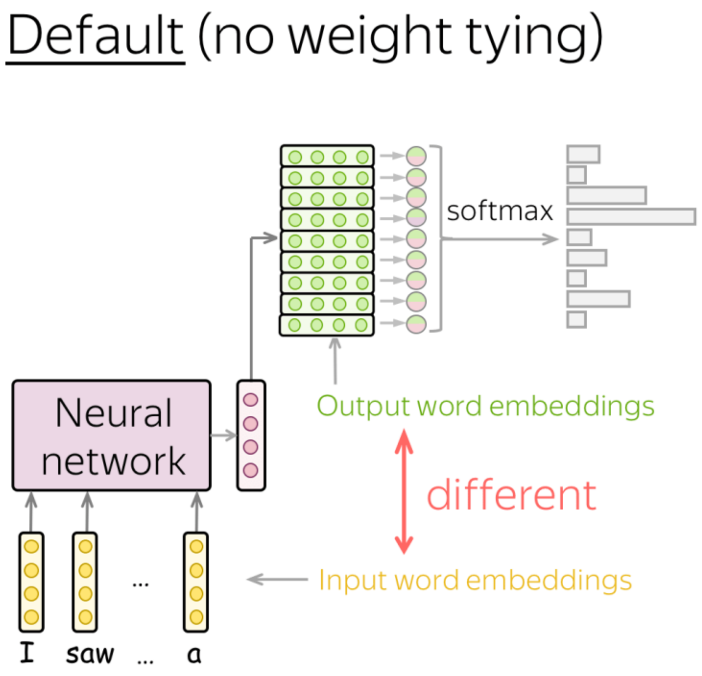
score$(h_t, s_k) = h_t^T\ W s_k$

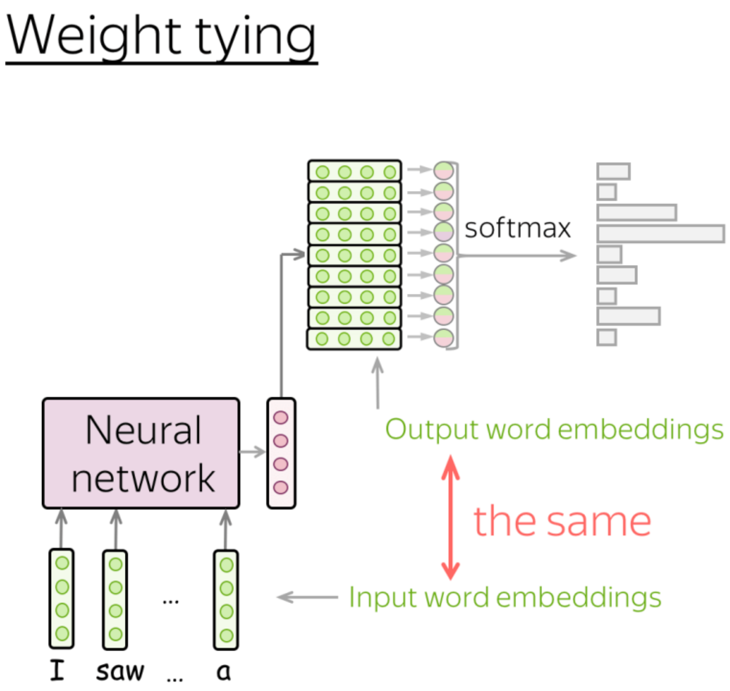
score$(h_t, s_k) = w_2^T \cdot \tanh(W_1[h_t, s_k])$

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

Table of Content

Weight tying is a way to significantly reduce the amount of parameters





Source: https://lena-voita.github.io/nlp_course/language_modeling.html#weight_tying_trick

Finding the next token is not that trivial

$$y' = \arg\max_{y} p(y|x) = \arg\max_{y} \prod_{t=1}^{n} p(y_t|y_{<t}, x)$$

How to find the argmax?

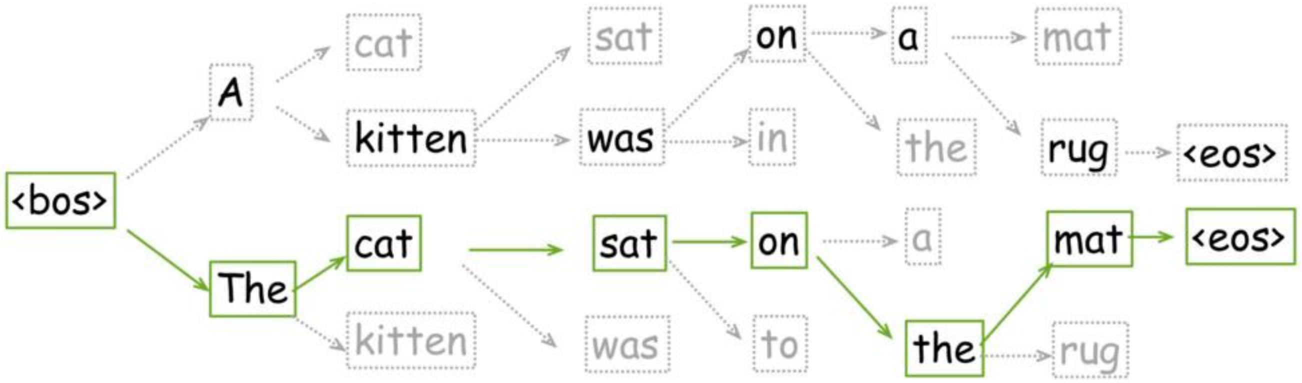
Source: https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

Greedy Decoding: At each step, pick the most probable token

$$\arg\max_{y} \prod_{t=1}^{n} p(y_t | y_{<t}, x) \neq \prod_{t=1}^{n} \arg\max_{y_t} p(y_t | y_{<t}, x)$$

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

A beam search illustration



Pick the hypothesis with the highest probability

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

Temperature: the higher, the more chaotic the choice becomes

$$w_{next} \sim \frac{P(w_{next}|X)^{1/\tau}}{\sum_{\hat{w}} P(\hat{w}|X)^{1/\tau}}$$

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

HSE UNIVERSITY • EST 1992 •