Introduction to recurrent neural networks

Skoltech

Deep Learning problems with sequential data

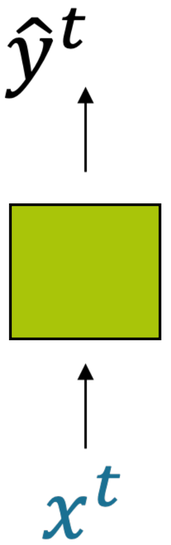
	Input	Output
• Speech recognition		“The quick brown fox jumped over the lazy dog.”
• Sentiment classification	“There is nothing to like in this movie.”	
• Machine translation	Voulez-vous chanter avec moi?	Do you want to sing with me?
• Video activity recognition		Running

Standard methods do not work when it is not clear where to get features from

Idea 1: use the previous word

Глаза Степана Аркадьича весело заблестели, и он задумался, улыбаясь

[0, 0, 0, 1, 0, …, 0]



- Features - one hot encoding for a dictionary of all words:
- The number of "features" - hundreds of thousands

Issue 1: Long-Term Dependencies

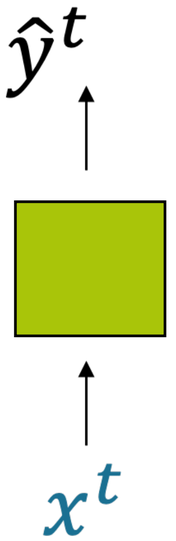
"**France** is where I grew up, but I now live in Boston. I speak fluent ____."

Skoltech

Idea 2: bag of words model

Глаза Степана Аркадьича весело заблестели, и он задумался, улыбаясь

[0, 1, 0, 1, 0, …, 1]



- Features - counting the number of words for a dictionary of all words:
- ***Bag of Words Model*** - Order Doesn't Matter

Issue 2: Use of word order information

The food was good, not bad at all.

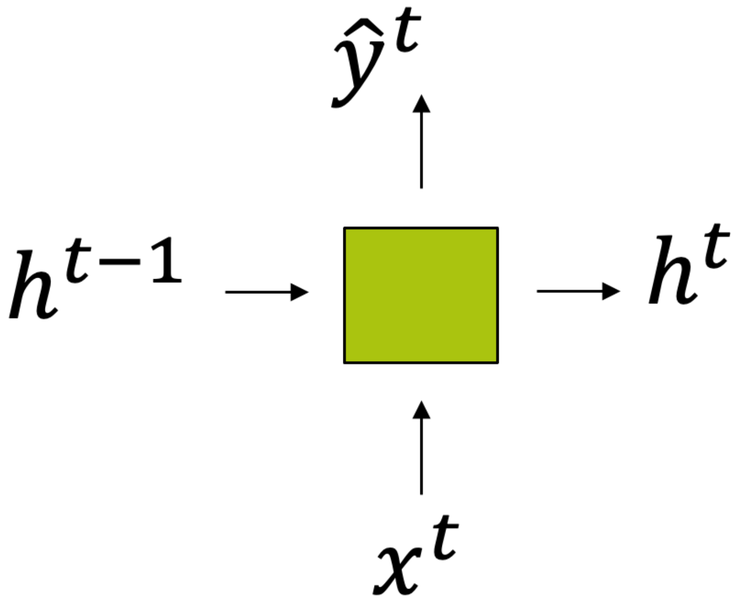
vs.

The food was bad, not good at all.

Skoltech

Model Design Criteria

- Long term memory
- Order Information
- Natural preprocessing
- Handling Variable Length Sequences



Solution: Recurrent neural networks (RNNs)

$$\boldsymbol{h}^t = f_h(\boldsymbol{x}^t, \boldsymbol{h}^{t-1})$$
$$\boldsymbol{h}^t = \tanh(V\boldsymbol{x}^t + W\boldsymbol{h}^{t-1} + b_h)$$

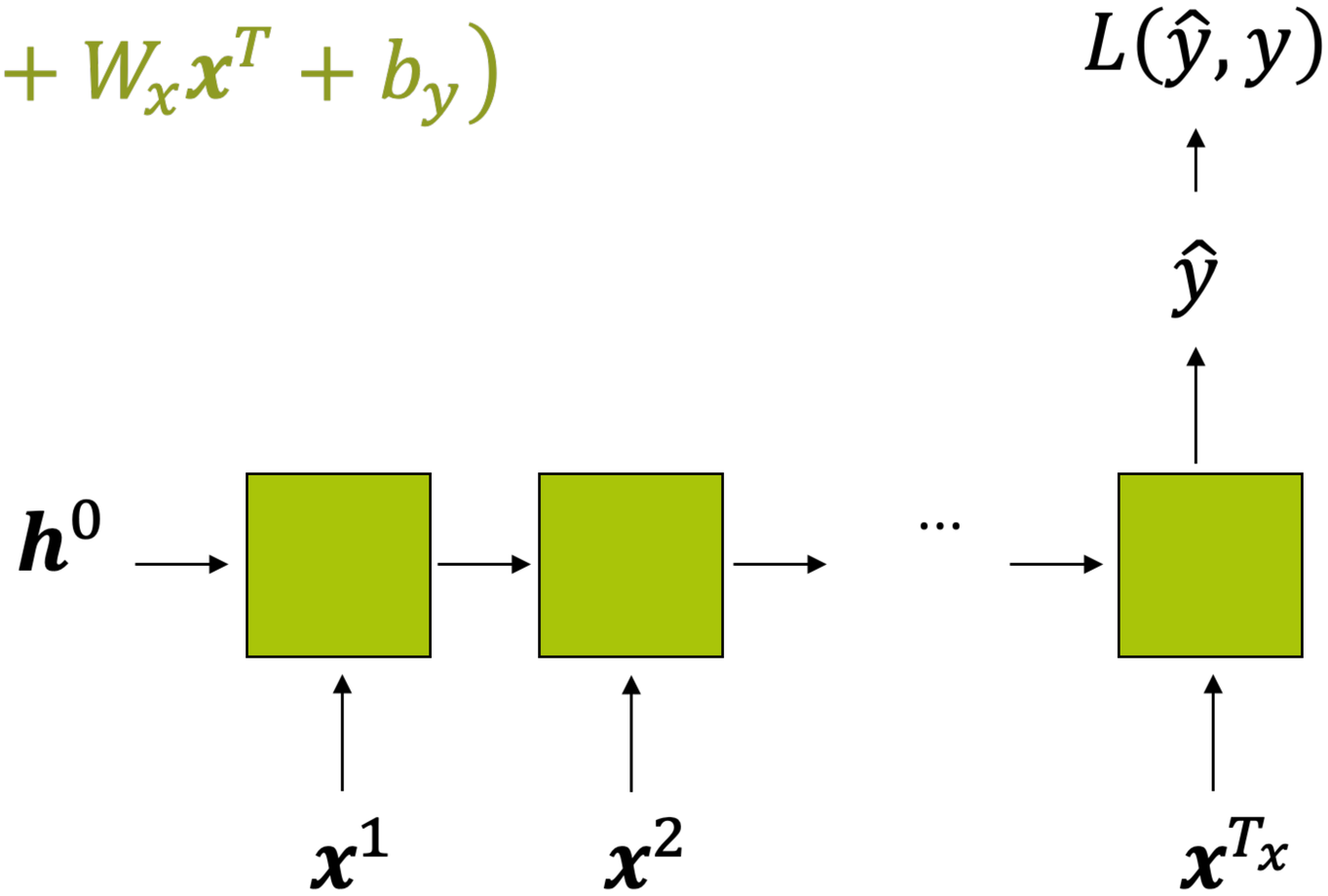
$$\hat{y}^t = f_y(\boldsymbol{h}^t)$$
$$\hat{y}^t = \mathrm{softmax}(U\boldsymbol{h}^t + b_y)$$

Skoltech

Model learning: back propagation

$$\hat{y}^t = \text{softmax}\left(W_h \boldsymbol{h}^t + W_x \boldsymbol{x}^T + b_y\right)$$

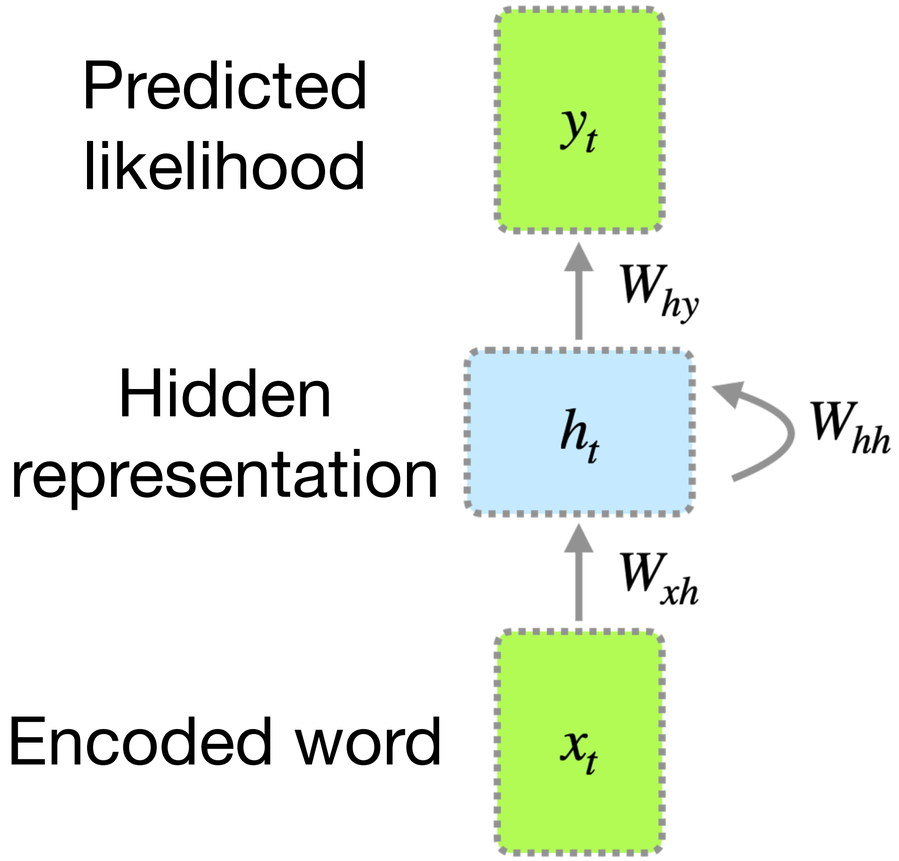
$$\frac{\partial L}{\partial W_h} = \frac{\partial L}{\partial \hat{y}^i} \frac{\partial \hat{y}^i}{\partial W_h}$$

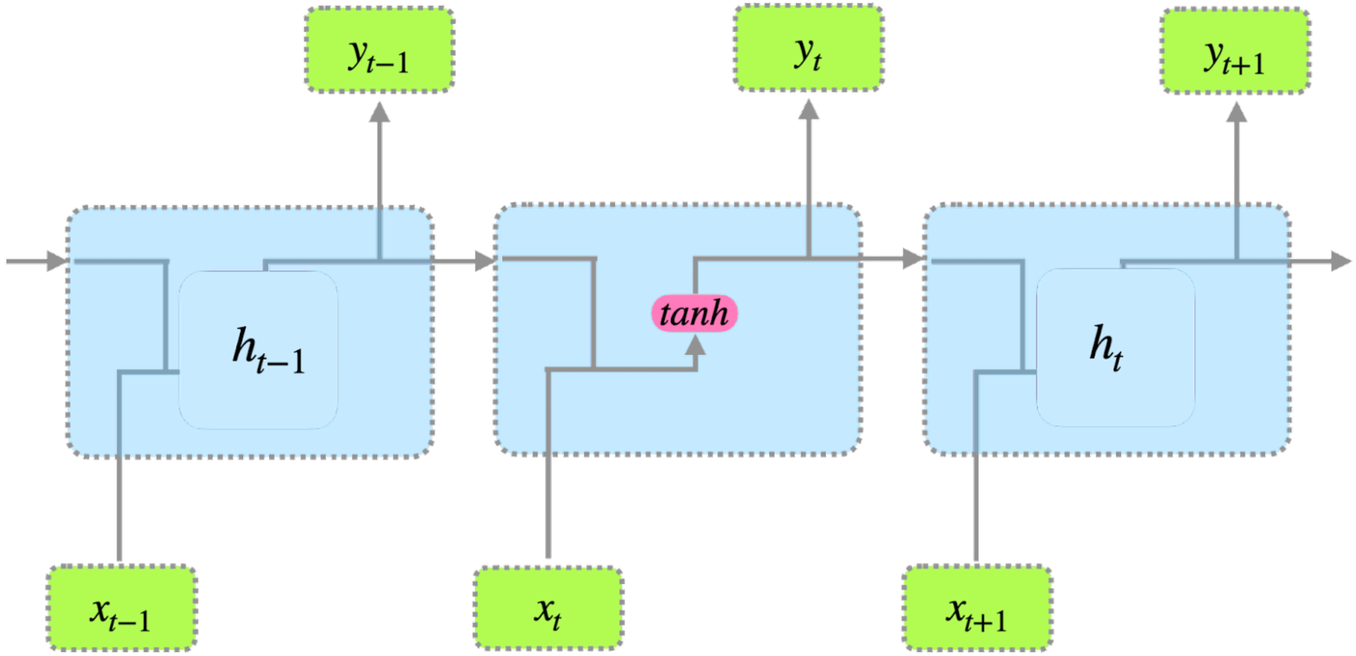


RNN intuition

Skoltech

Model Design

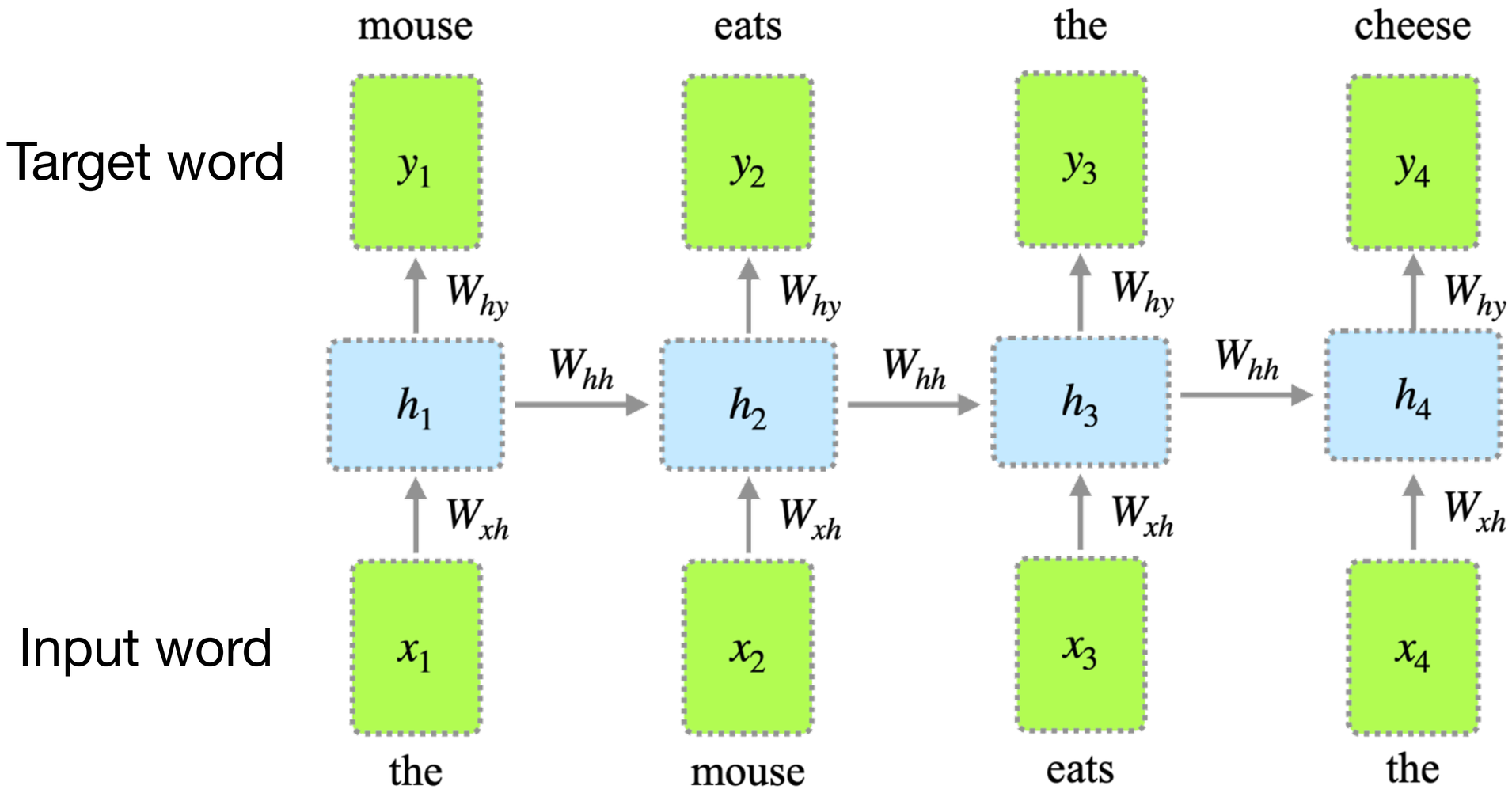




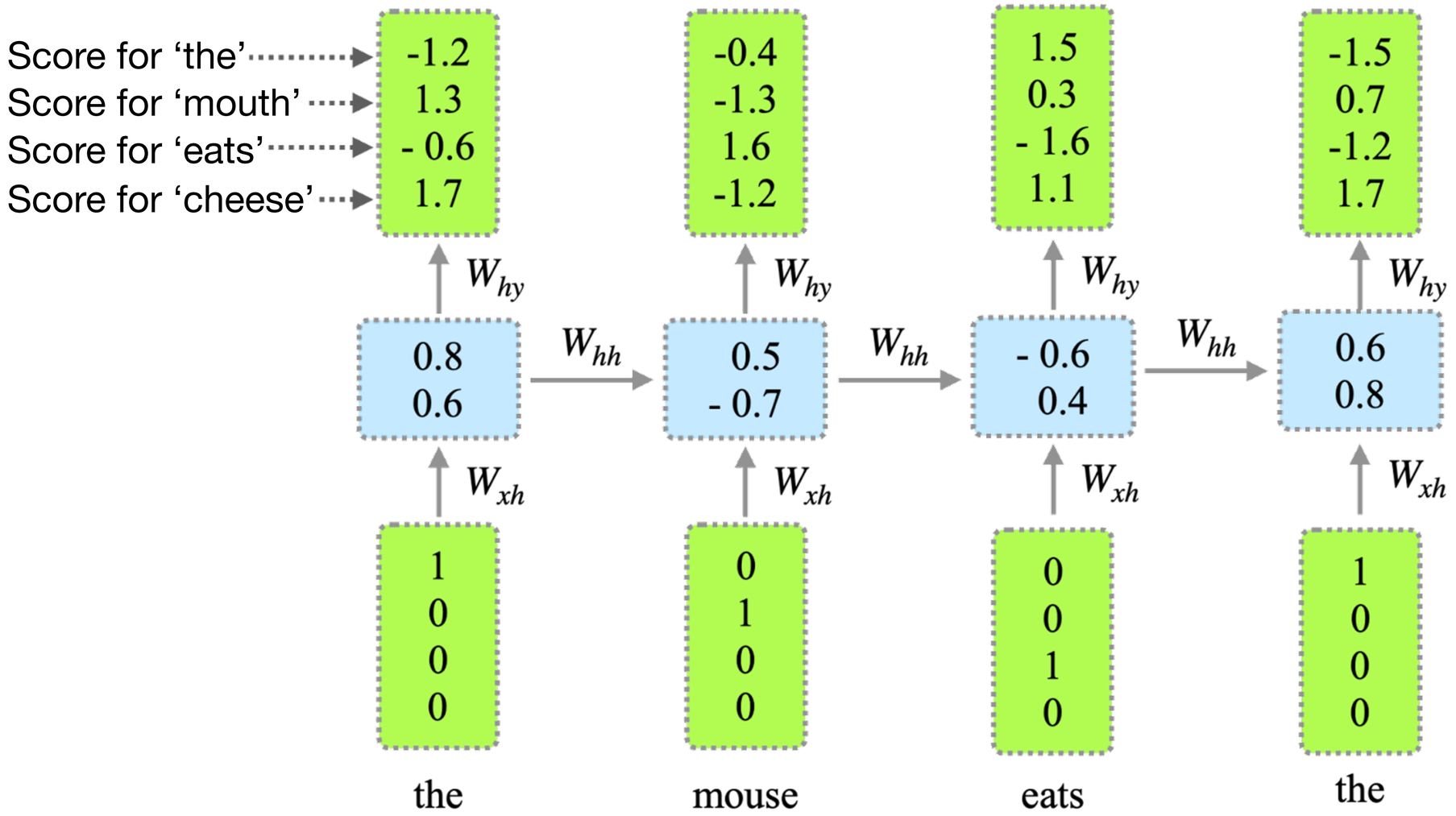
$$h_t = tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$

x_t and y_t — n-dimensional for dictionary of size t

Model Design Example



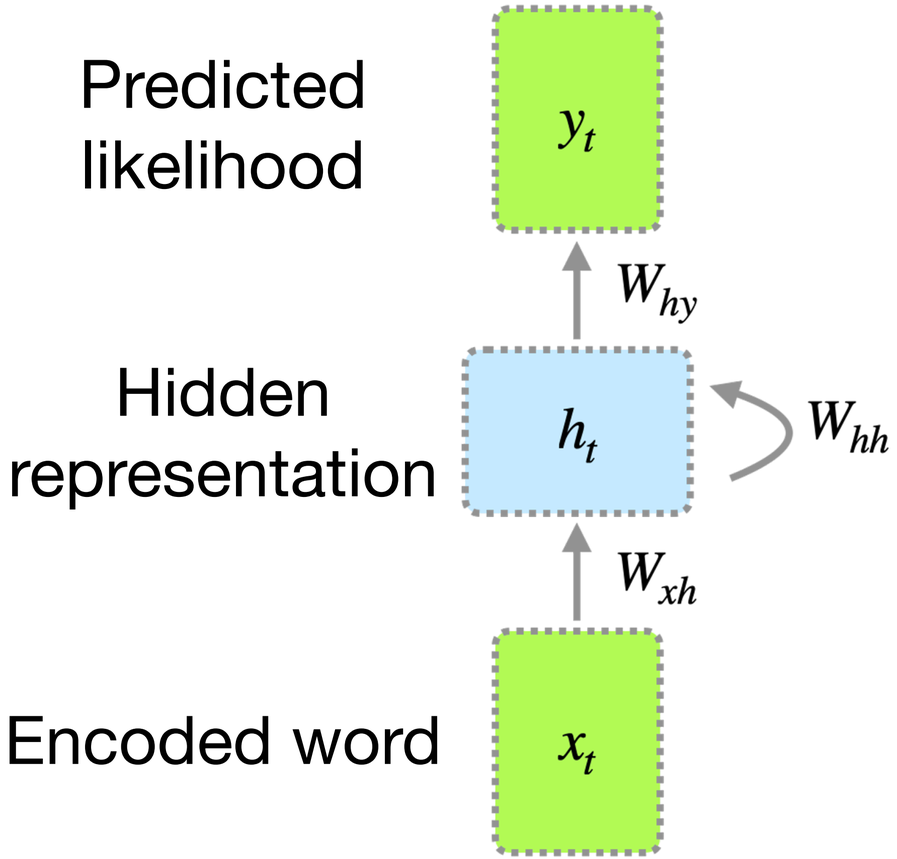
Model Design Example



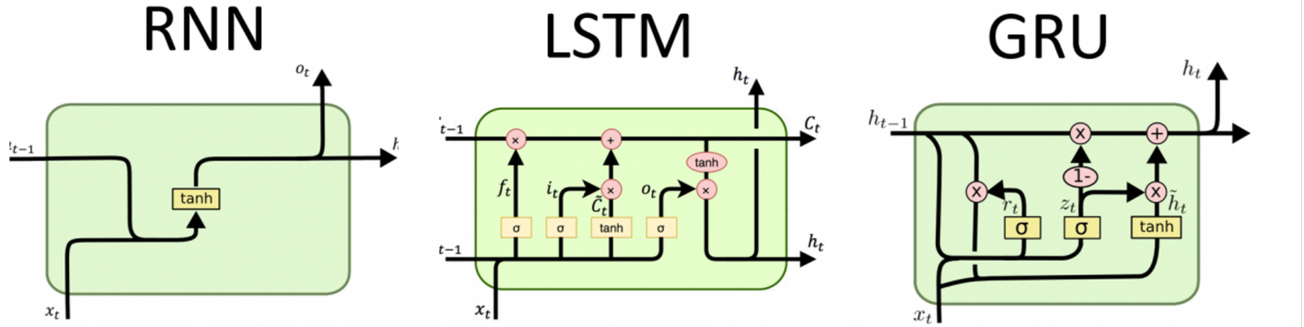
LSTM and GRU

Skoltech

Proper configuration of the recurrent block ensures long-term dependencies

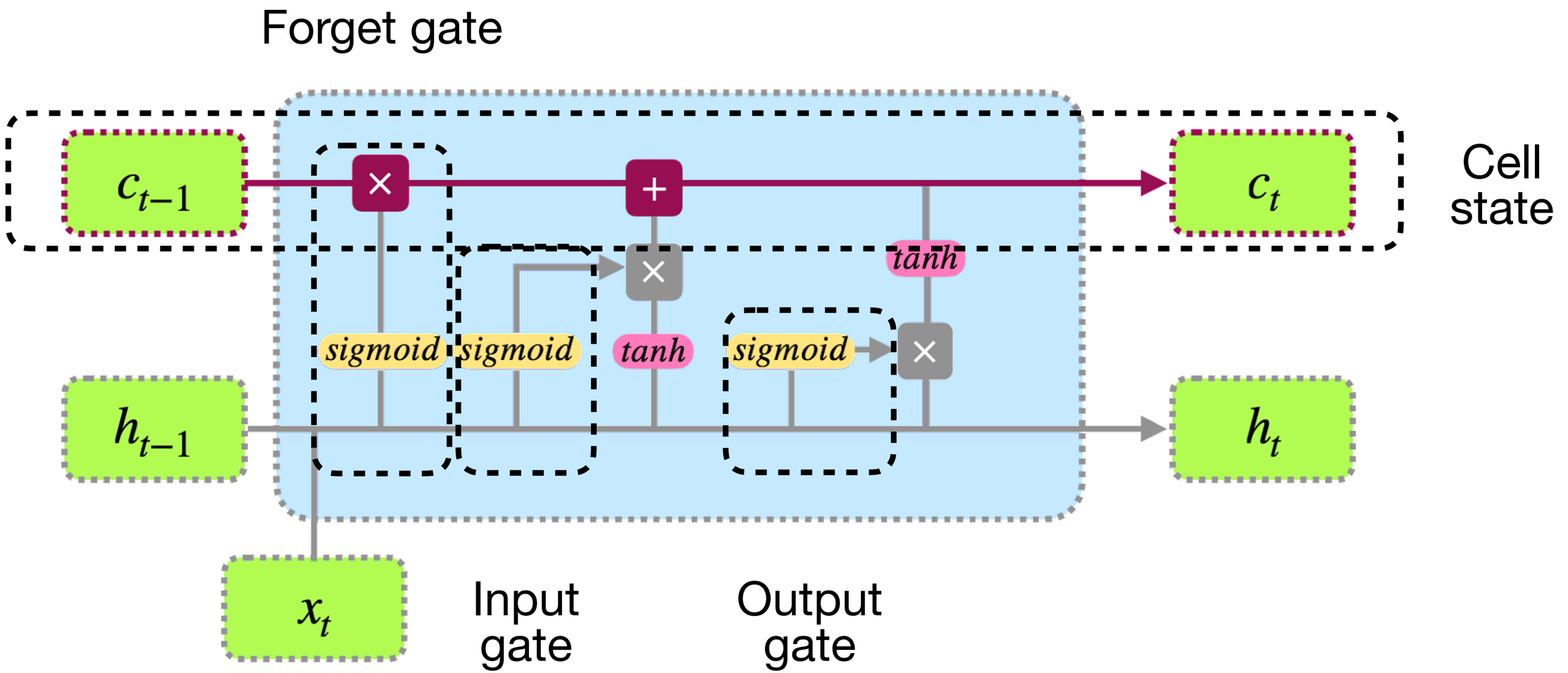


- Plain RNN is not a good choice
- LSTM and GRU cells work better
- LSTM is slightly better
- GRU is a little faster



Skoltech

Long Short-Term Memory (LSTM)



LSTM gates

- ***Forget gate*** decides what information should be thrown away or kept
- ***Input Gate:*** update the cell state
- ***Cell state:***

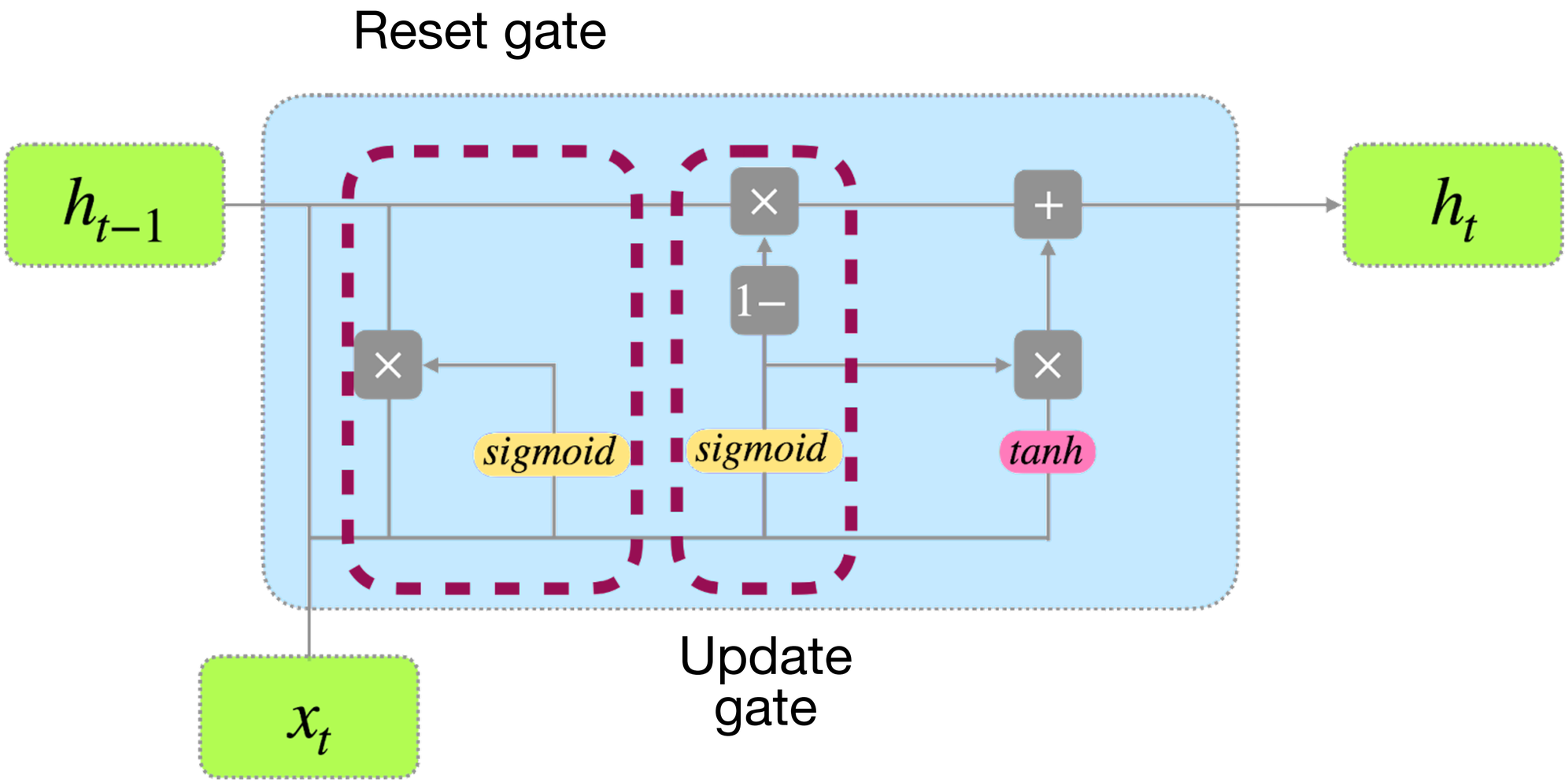
1) Cell state gets pointwise multiplied by the forget vector: drop values in the cell state if it gets multiplied by values near 0

2) Take the output from the input gate and do a pointwise addition which updates the cell state to new values that the NN finds relevant

3) Get a new cell state

- ***Output Gate*** decides what the next hidden state should be

Gated Recurrent Unit (GRU)

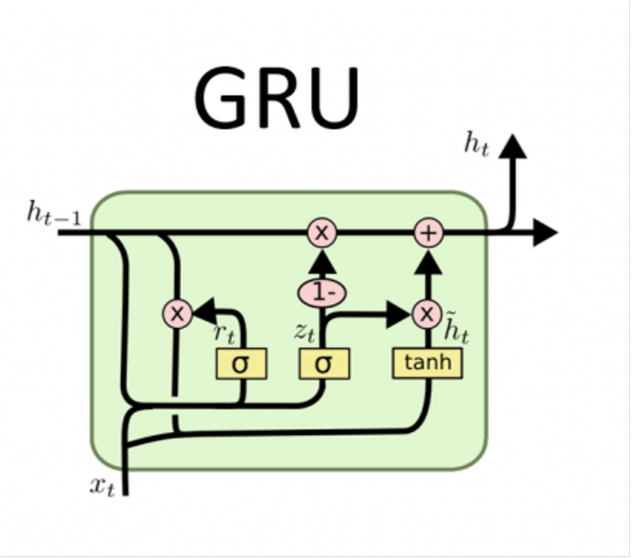


GRU gates

- ***Reset gate*** is used to decide how much past information to forget
- ***Update Gate*** acts similar to the forget and input gate of an LSTM

$$\boldsymbol{r}^t = \sigma(W_{xr}\boldsymbol{x}^t + W_{hr}\boldsymbol{h}^{t-1} + b_r)$$
$$\boldsymbol{z}^t = \sigma(W_{xz}\boldsymbol{x}^t + W_{hz}\boldsymbol{h}^{t-1} + b_z)$$

$$\tilde{\boldsymbol{h}}^t = \tanh(W_{xh}\boldsymbol{x}^t + W_{hr}(\boldsymbol{r}^t \odot \boldsymbol{h}^{t-1}) + b_h)$$
$$\boldsymbol{h}^t = \boldsymbol{z}^t \odot \boldsymbol{h}^{t-1} + (1 - \boldsymbol{z}^t) \odot \tilde{\boldsymbol{h}}^t$$



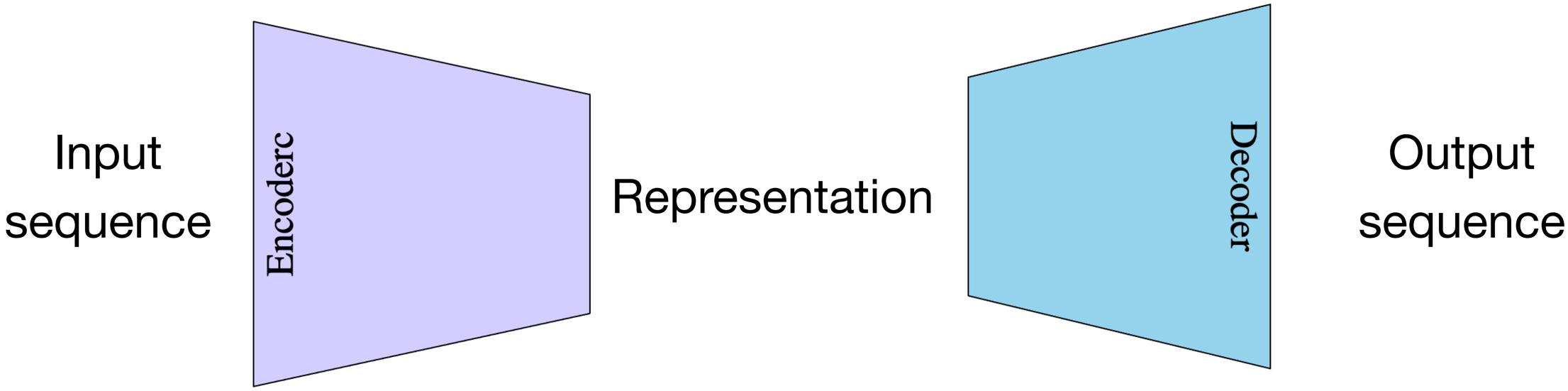
Skoltech

Machine translation: application example

Skoltech

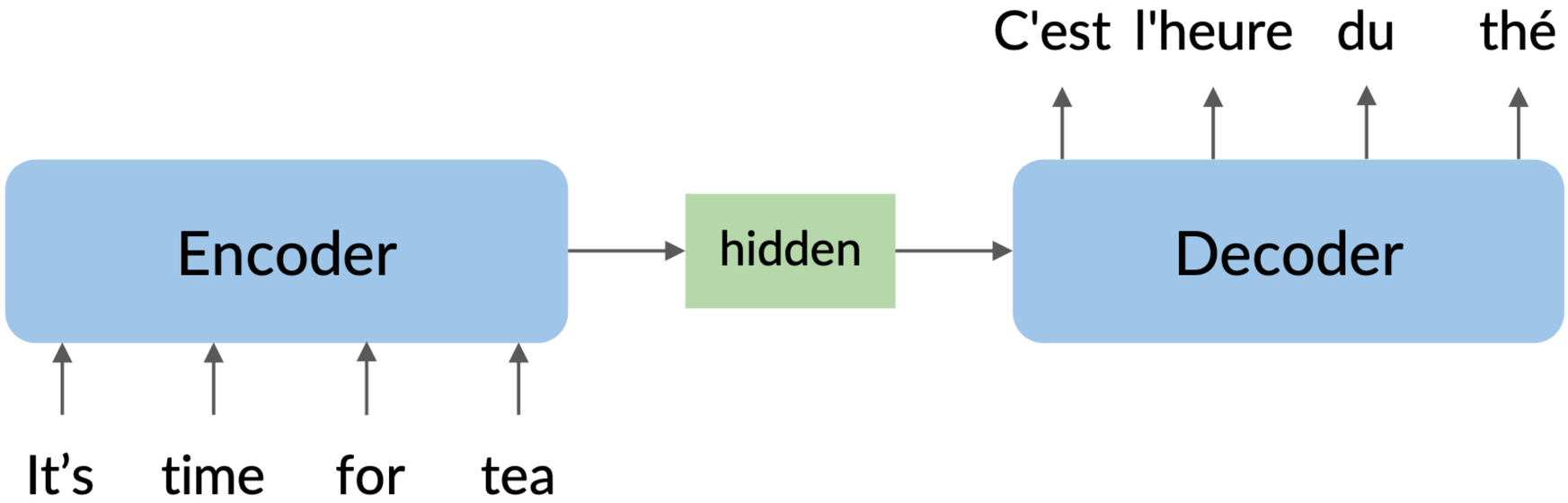
Neural network for machine translation

- Let’s bring two RNNs
- Let's make **seq2seq (sequence to sequence)** architecture out of them

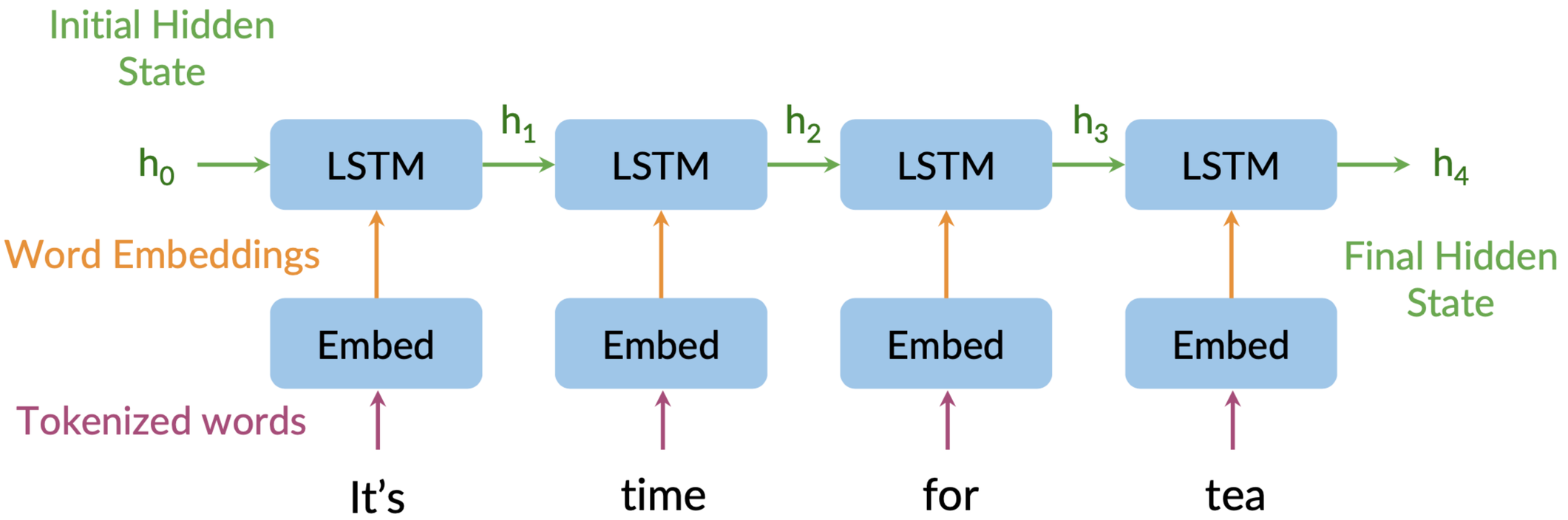


Neural network for machine translation

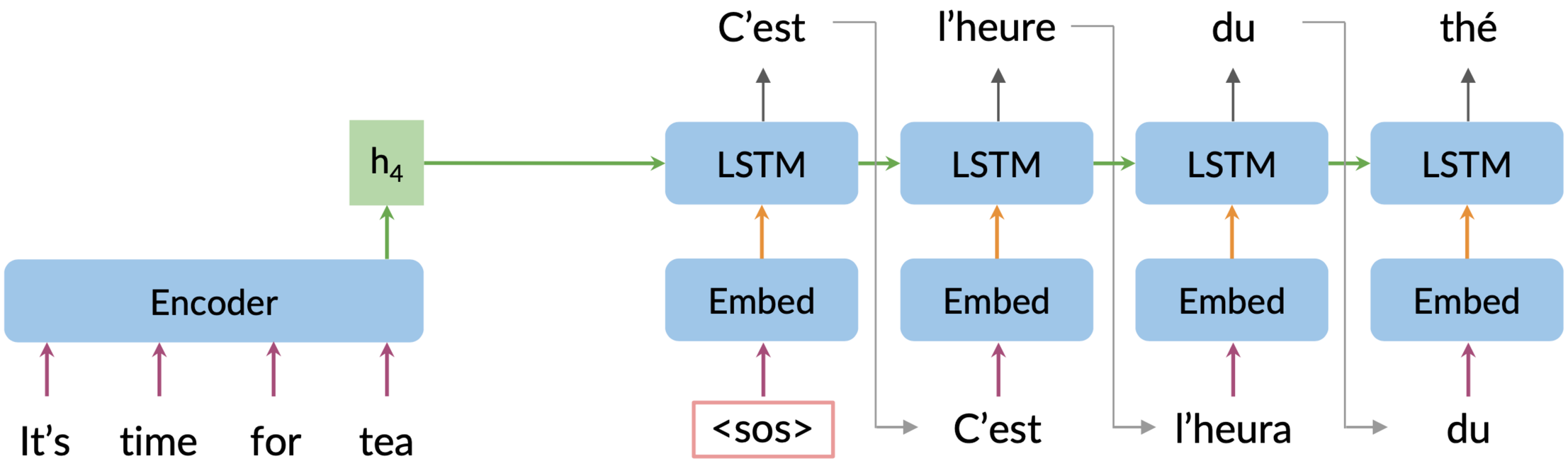
- Let’s bring two RNNs
- Let's make **seq2seq (sequence to sequence)** architecture out of them



Encoder


Initial Hidden State
h_0
LSTM
h_1
LSTM
h_2
LSTM
h_3
LSTM
h_4
Word Embeddings
Final Hidden State
Embed
Embed
Embed
Embed
Tokenized words
It's
time
for
tea

Decoder



Main issue: bottleneck, the network will have time to forget everything

Conclusion

- RNN’s are good for processing sequence data for predictions but suffers from short-term memory
- LSTM’s and GRU’s were created as a method to mitigate short-term memory using mechanisms called gates
- RNNs struggle dealing with long sequences

Next lecture: Models with attention

Skoltech