

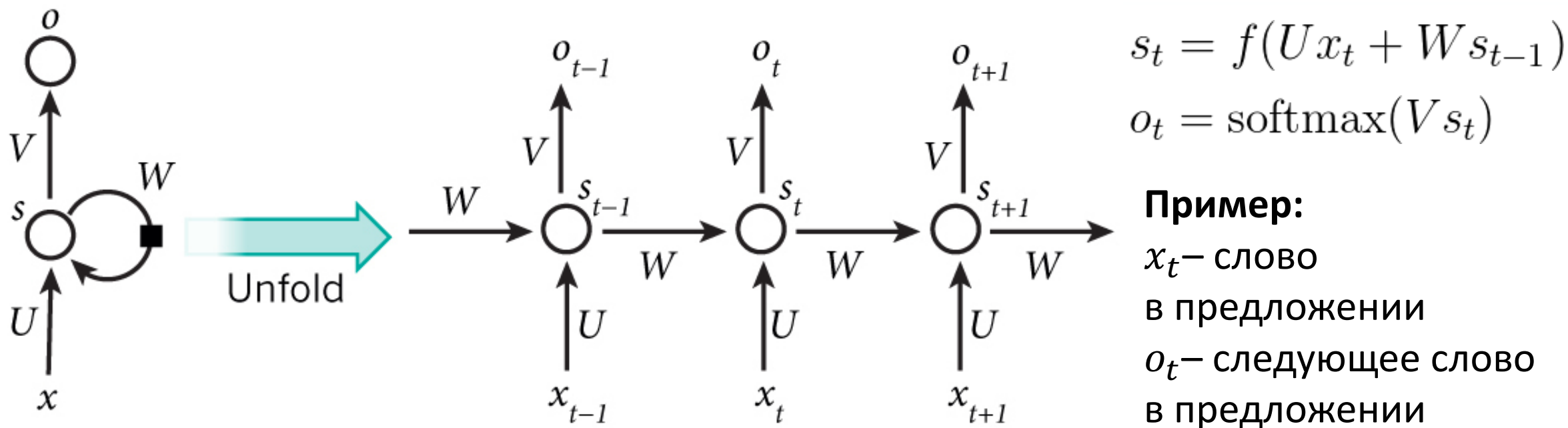
LSML #7

Рекуррентные сети

Recurrent Neural Networks (RNN)

- Работают с последовательностями
 - Слов в предложении
 - Букв в предложении
 - Отсчетов в аудио сигнале (амплитуда, частота)
 - Пикселей изображения
 - ...

Как устроена простая RNN

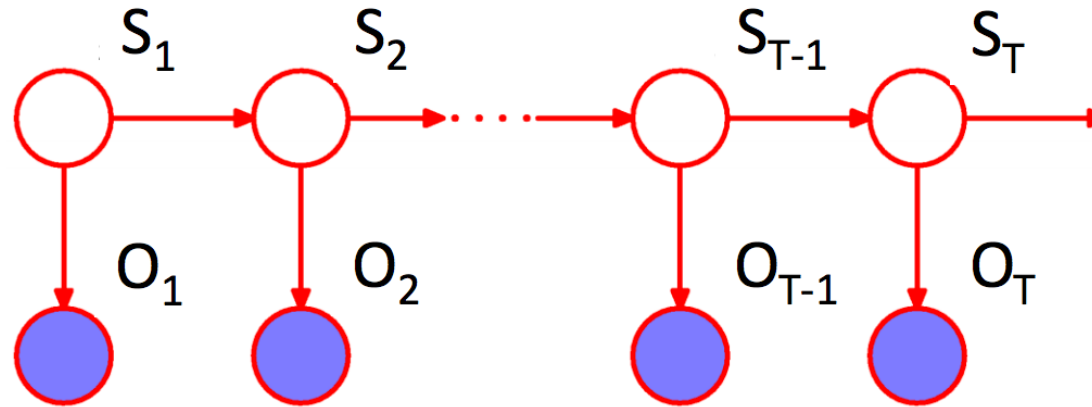


- Работает **одинаково** для каждого элемента последовательности x_t , но вычисления зависят от предыдущих элементов x_t
- Можно сказать, что у RNN есть память (**скрытое состояние s_t**), в которой хранится информация о предыдущих элементах последовательности

А как же Hidden Markov Model (HMM)?

$$O_t \in \{y_1, y_2, \dots, y_K\}$$

$$S_t \in \{1, \dots, I\}$$



Пример:

s_t – POS tag

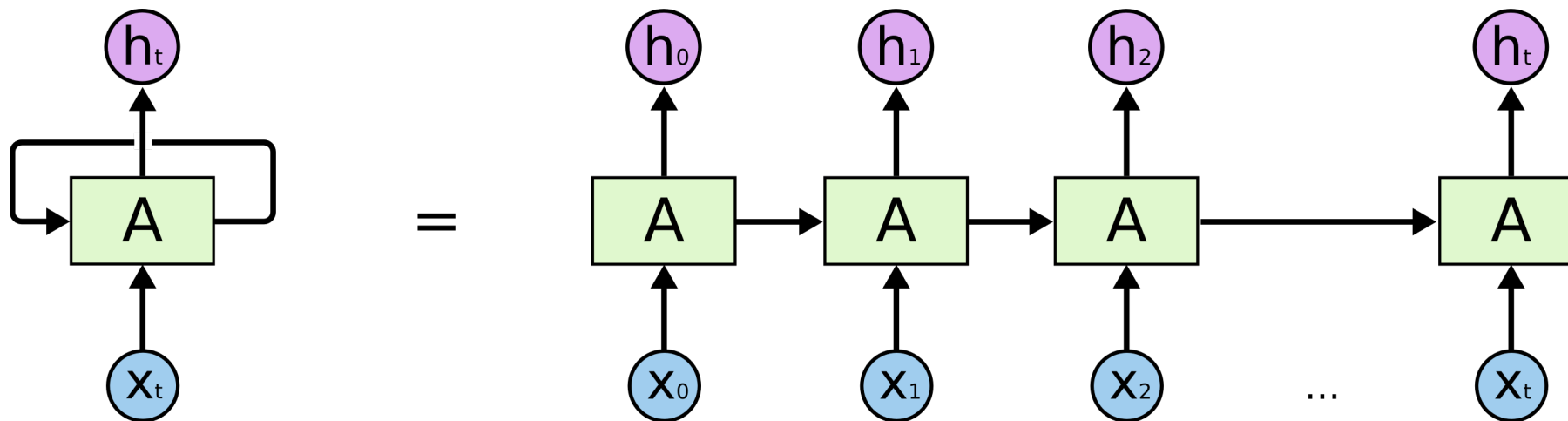
o_t – слово в предложении

$$p(S_1, \dots, S_T, O_1, \dots, O_T) = \prod_{t=1}^T p(O_t | S_t) \prod_{t=1}^T p(S_t | S_{t-1})$$

1st order Markov assumption on hidden states $\{S_t\}$ $t = 1, \dots, T$
(can be extended to higher order).

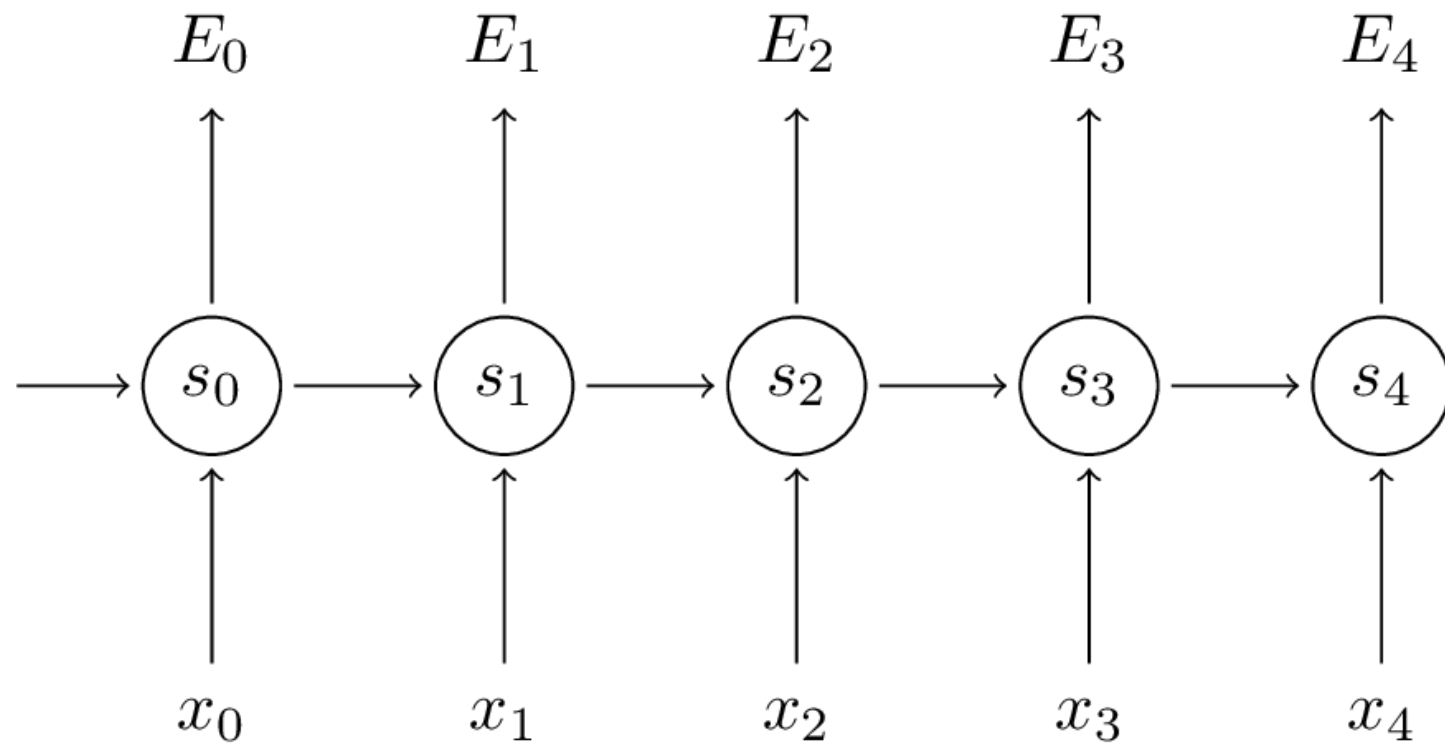
Backpropagation through time (BPTT)

- Если развернуть сеть по времени t , то получим обычную feed-forward сеть с shared параметрами
- Применяем backpropagation, считаем градиенты для каждого параметра
- Далее (так же, как в случае сверток CNN) суммируем градиенты по shared параметрам и делаем шаг SGD



ВРТТ на примере

$$\begin{aligned} E_t(y_t, \hat{y}_t) &= -y_t \log \hat{y}_t \\ E(y, \hat{y}) &= \sum_t E_t(y_t, \hat{y}_t) \\ &= -\sum_t y_t \log \hat{y}_t \end{aligned}$$



$$\begin{aligned} s_t &= \tanh(Ux_t + Ws_{t-1}) \\ \hat{y}_t &= \text{softmax}(Vs_t) \end{aligned}$$

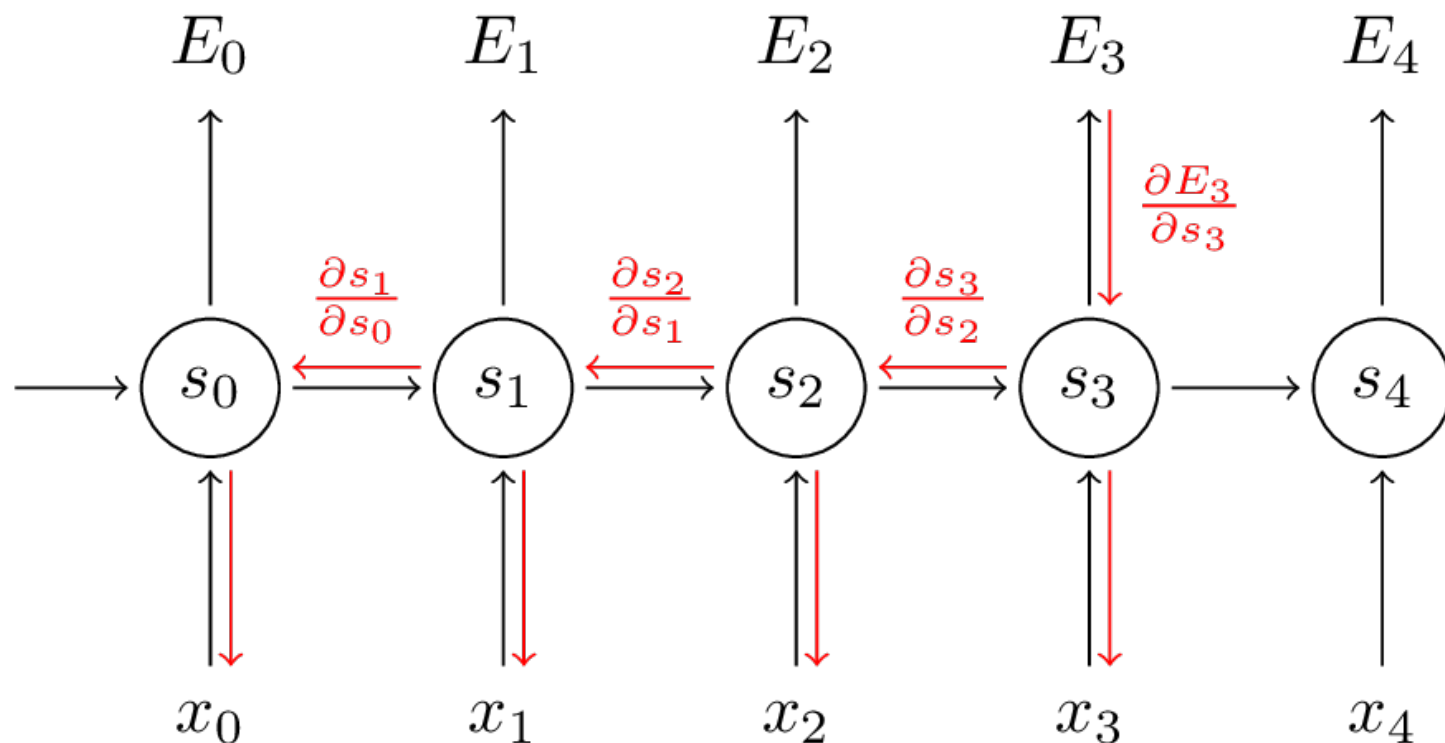
ВРТТ на примере

$$\frac{\partial E_3}{\partial W} = \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial W}$$

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial s_k} \frac{\partial s_k}{\partial W}$$

$$\frac{\partial s_3}{\partial s_1} = \frac{\partial s_3}{\partial s_2} \frac{\partial s_2}{\partial s_1}$$

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \left(\prod_{j=k+1}^3 \frac{\partial s_j}{\partial s_{j-1}} \right) \frac{\partial s_k}{\partial W}$$



$$s_t = \tanh(Ux_t + Ws_{t-1})$$

$$\hat{y}_t = \text{softmax}(Vs_t)$$

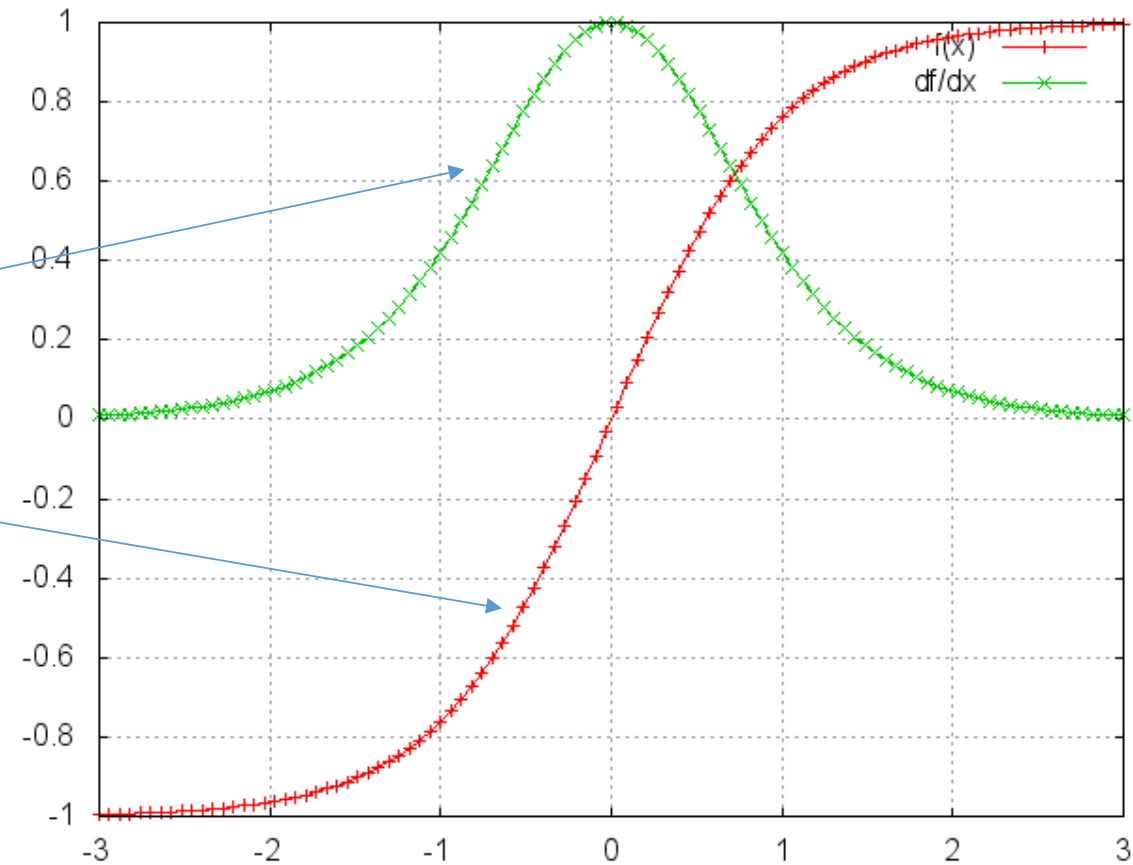
Проблема с затухающими градиентами

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \left(\prod_{j=k+1}^3 \frac{\partial s_j}{\partial s_{j-1}} \right) \frac{\partial s_k}{\partial W}$$

$$s_t = \tanh(Ux_t + Ws_{t-1})$$

$$\frac{\partial s_t}{\partial s_{t-1}} = \frac{\partial f}{\partial x} W, \quad f(x) = \tanh(x)$$

Быстро убывает
или взрывается

$$\left(\prod_{j=k+1}^3 \frac{\partial s_j}{\partial s_{j-1}} \right)$$


Gradient clipping

- Градиенты могут взорваться – ограничим их норму

```

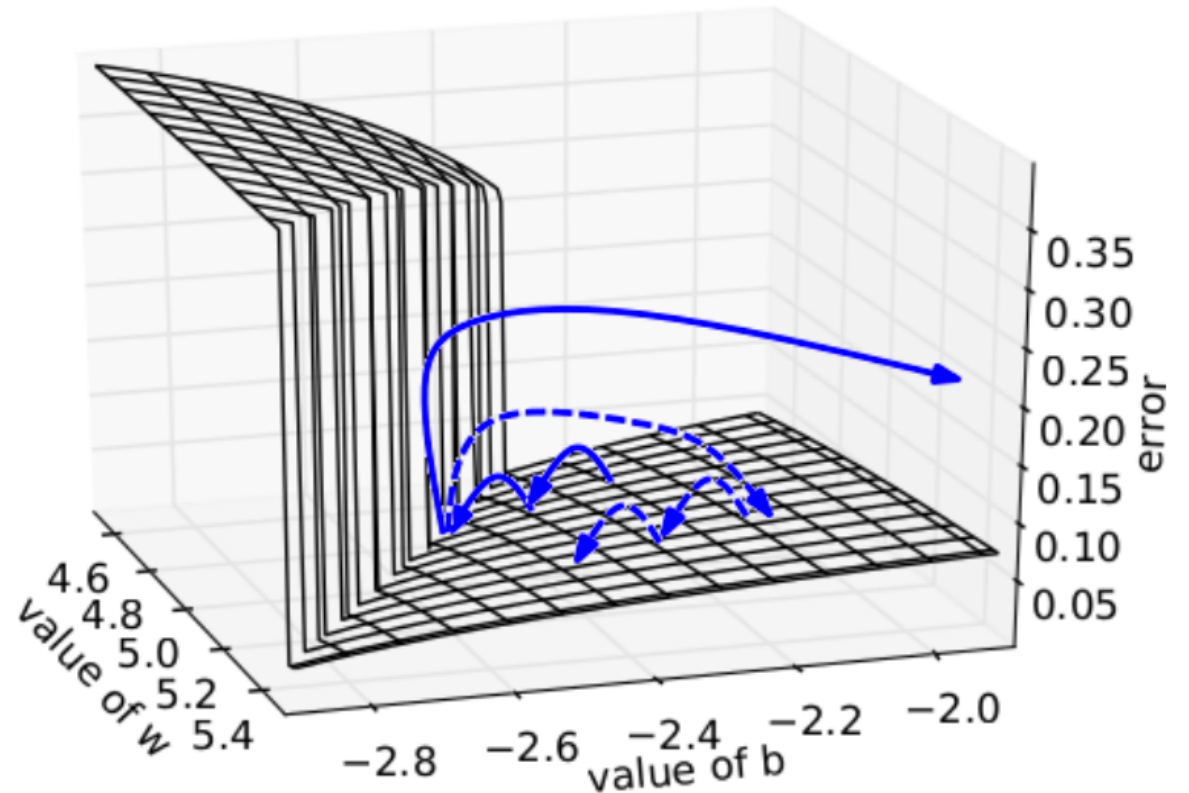
$$\hat{\mathbf{g}} \leftarrow \frac{\partial \mathcal{E}}{\partial \theta}$$


if  $\|\hat{\mathbf{g}}\| \geq threshold$  then

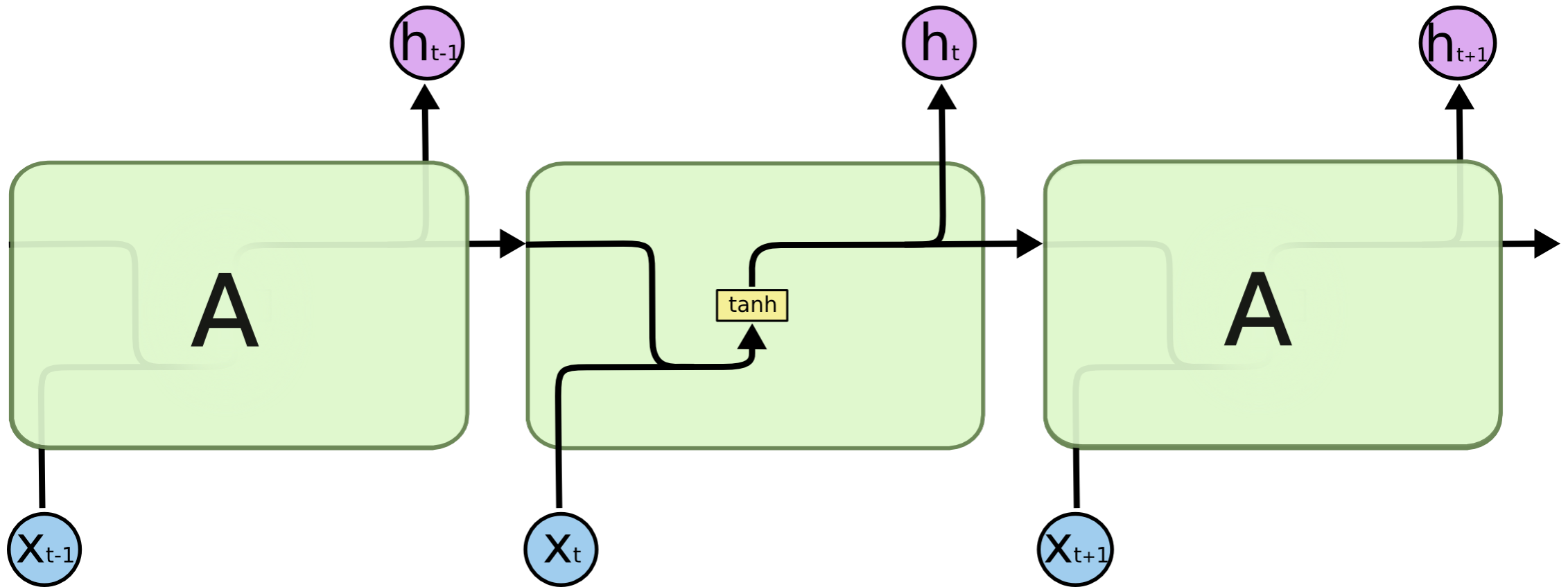

$$\hat{\mathbf{g}} \leftarrow \frac{threshold}{\|\hat{\mathbf{g}}\|} \hat{\mathbf{g}}$$


end if


```

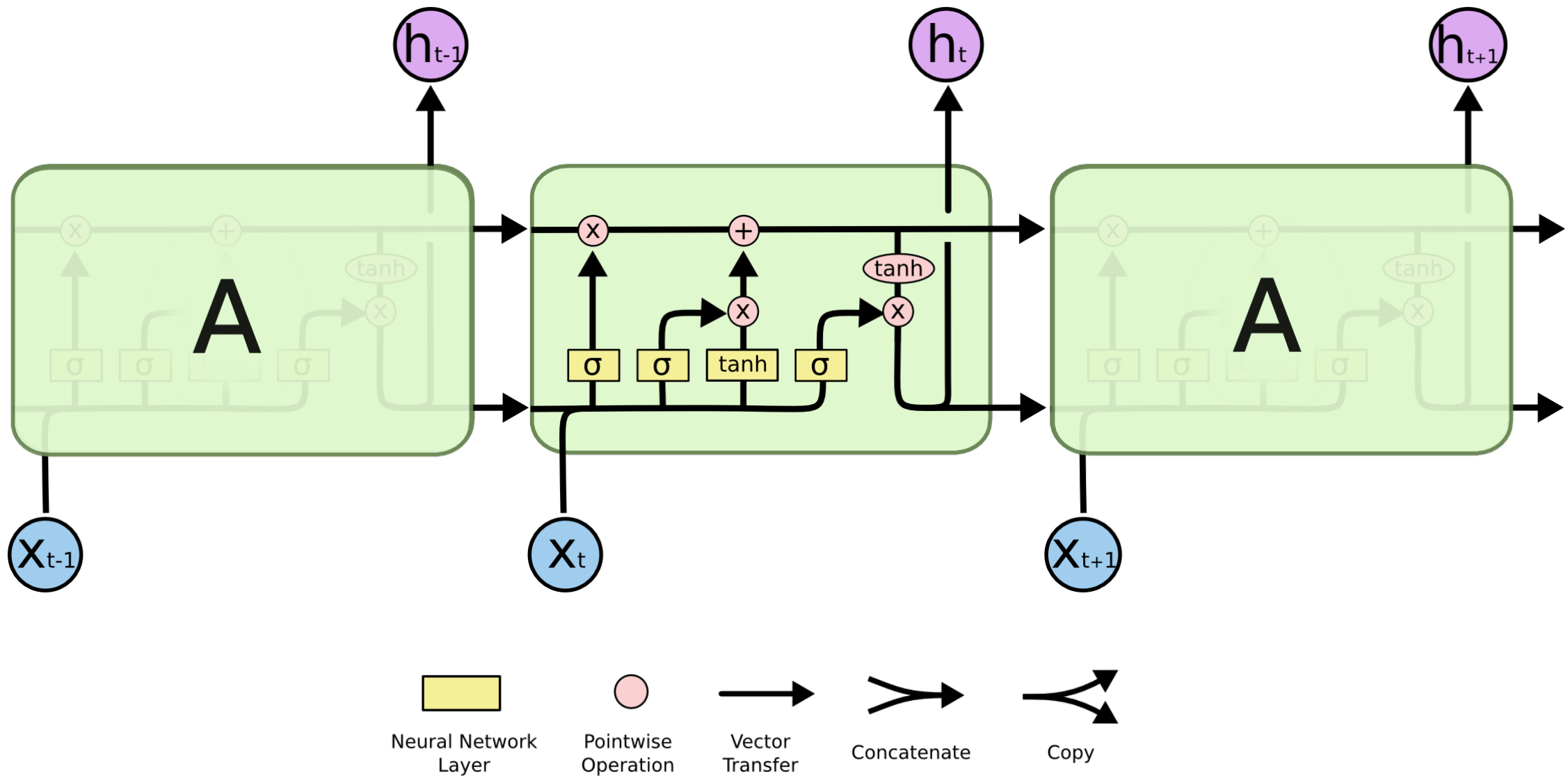


Простая RNN плохо работает



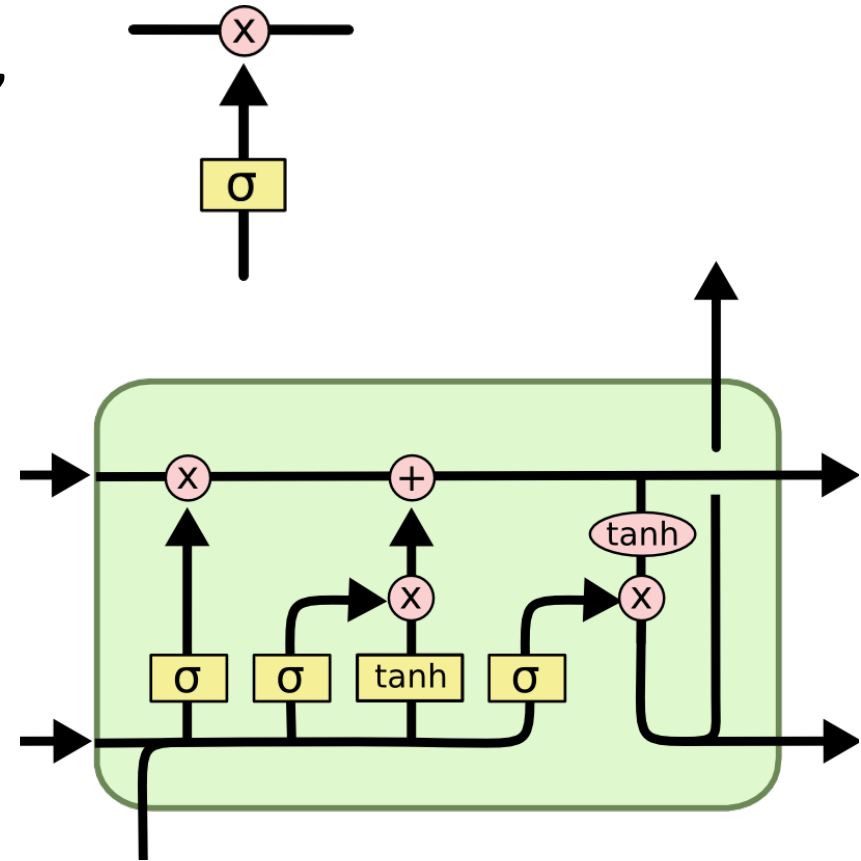
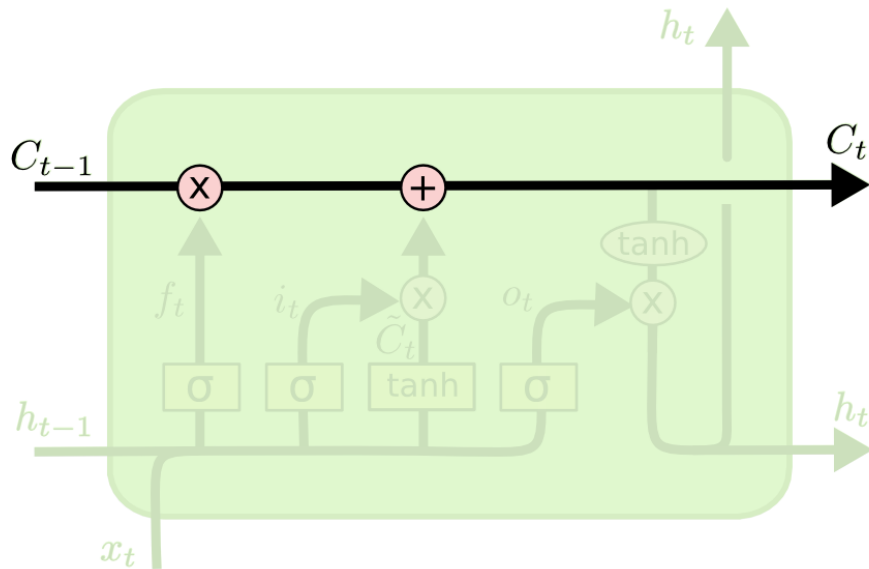
$$h_t = \tanh(Ux_t + Wh_{t-1})$$

Спасет LSTM

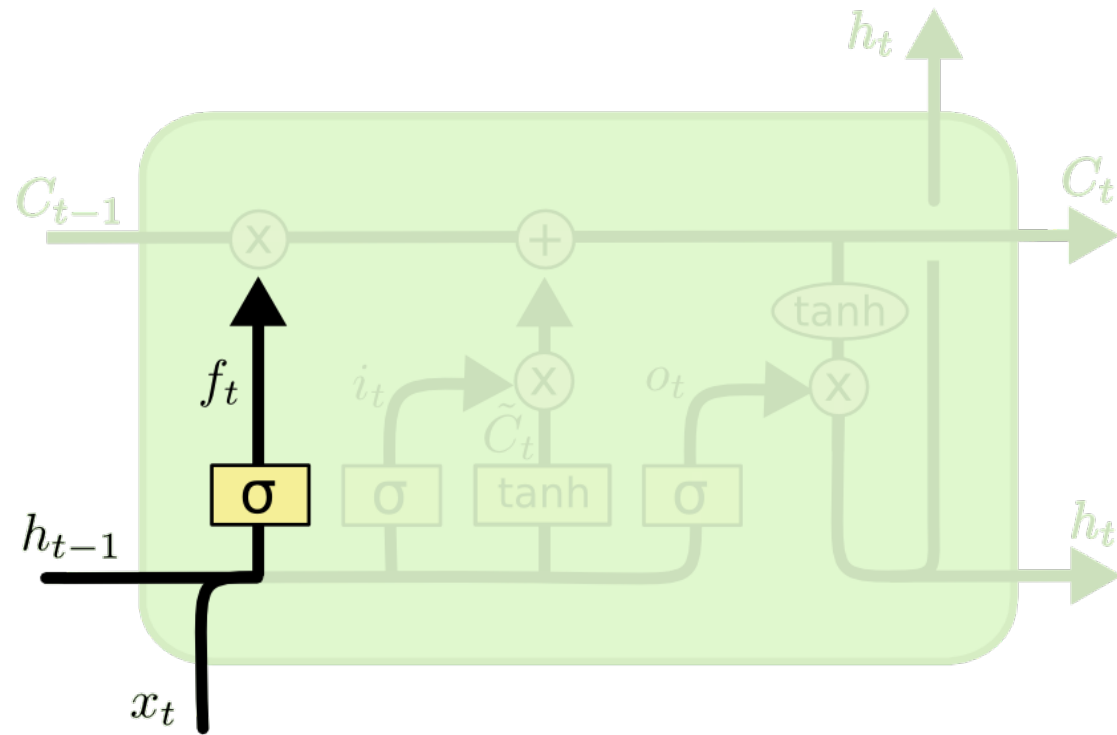


LSTM cell state

- C_t проходит через все ячейки, LSTM может забывать или добавлять информацию в C_t
- Gates учат маски для забывания (**forget**), добавления (**input**) и вывода (**output**) вектора состояний C_t

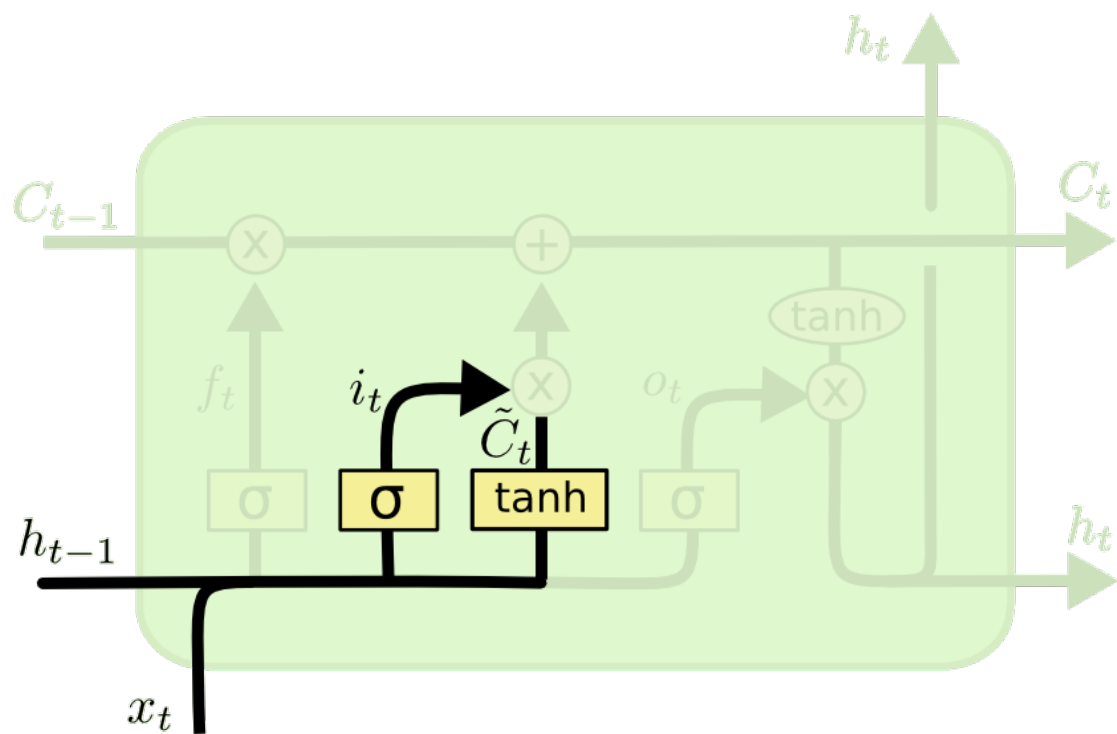


LSTM forget gate



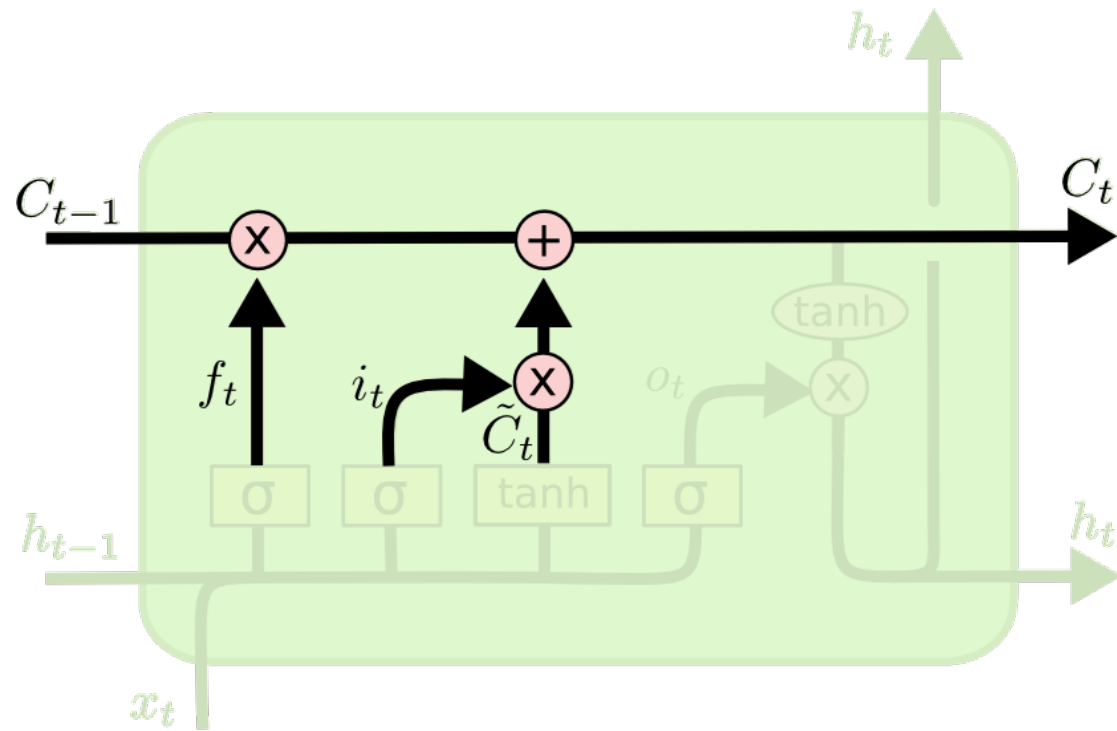
$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$

LSTM input gate и вклад ячейки в состояние



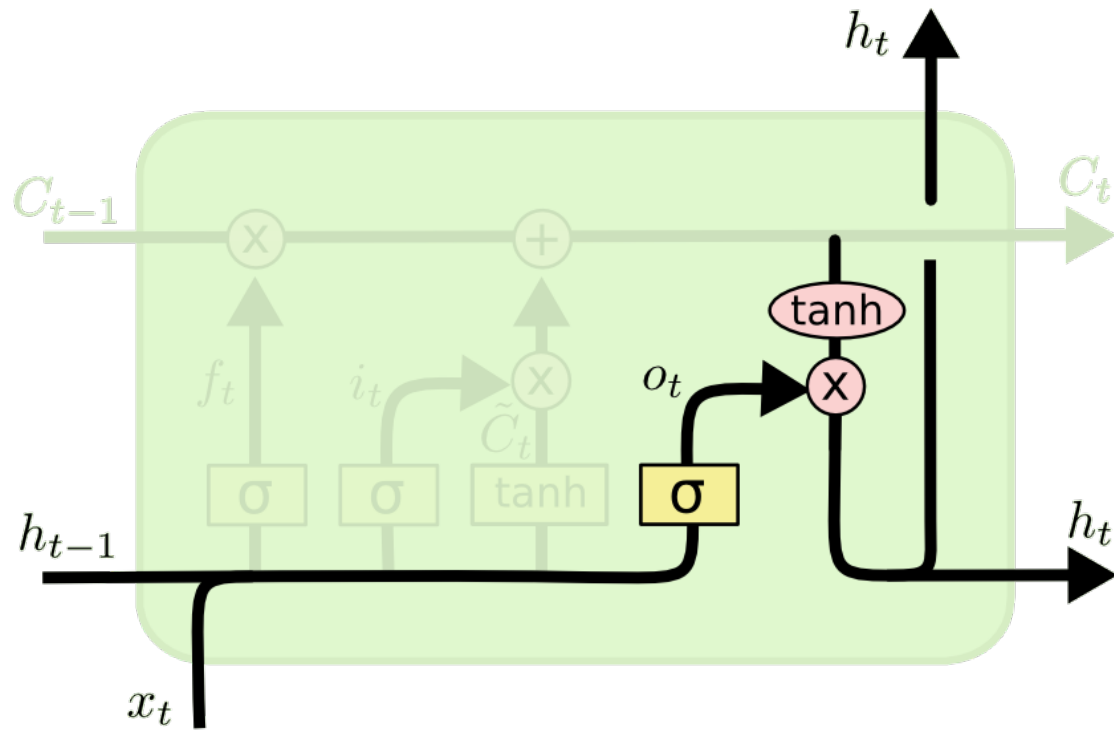
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Обновление состояния



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

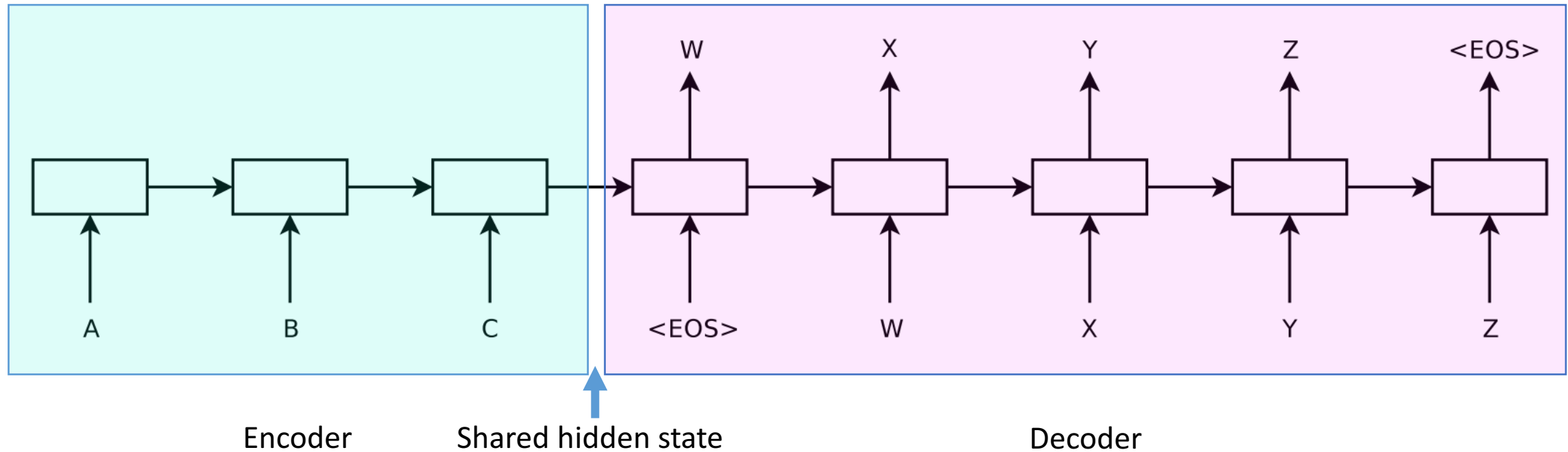
Выдача скрытого состояния наружу



$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

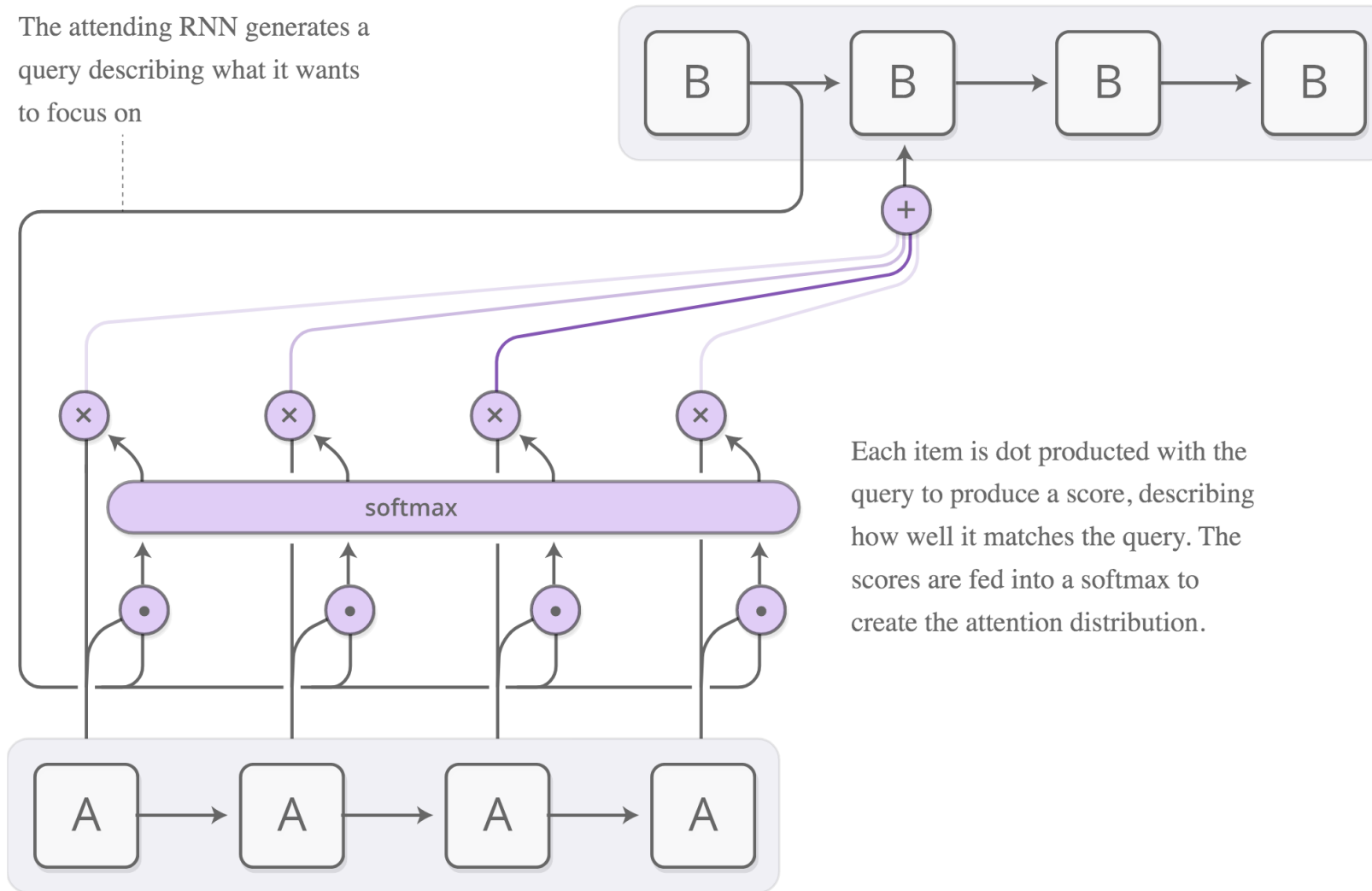
$$h_t = o_t * \tanh (C_t)$$

Encoder-decoder (seq2seq)

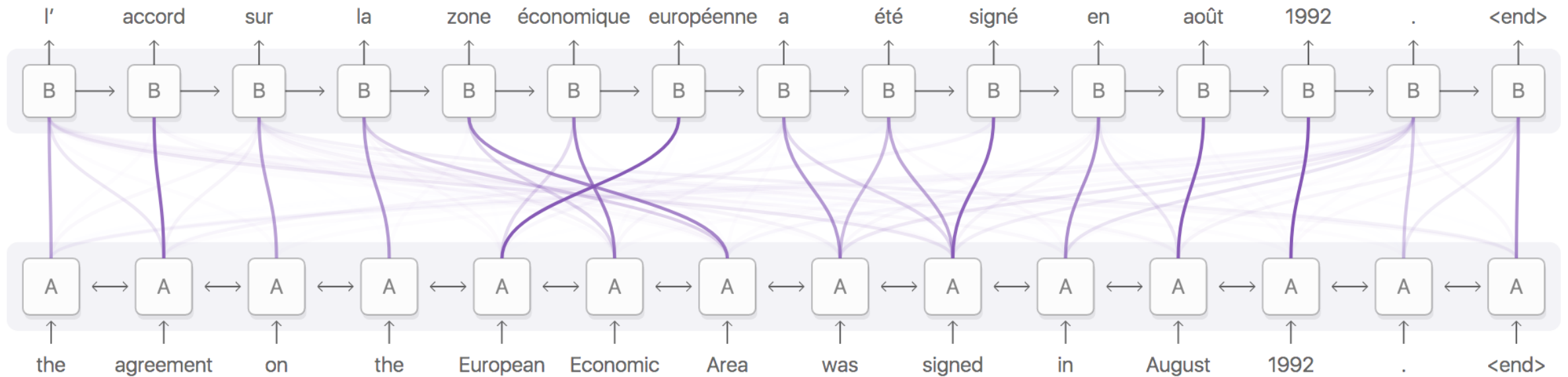


- Слабое место – связь между encoder и decoder только через последний скрытый слой encoder

Поможет attention

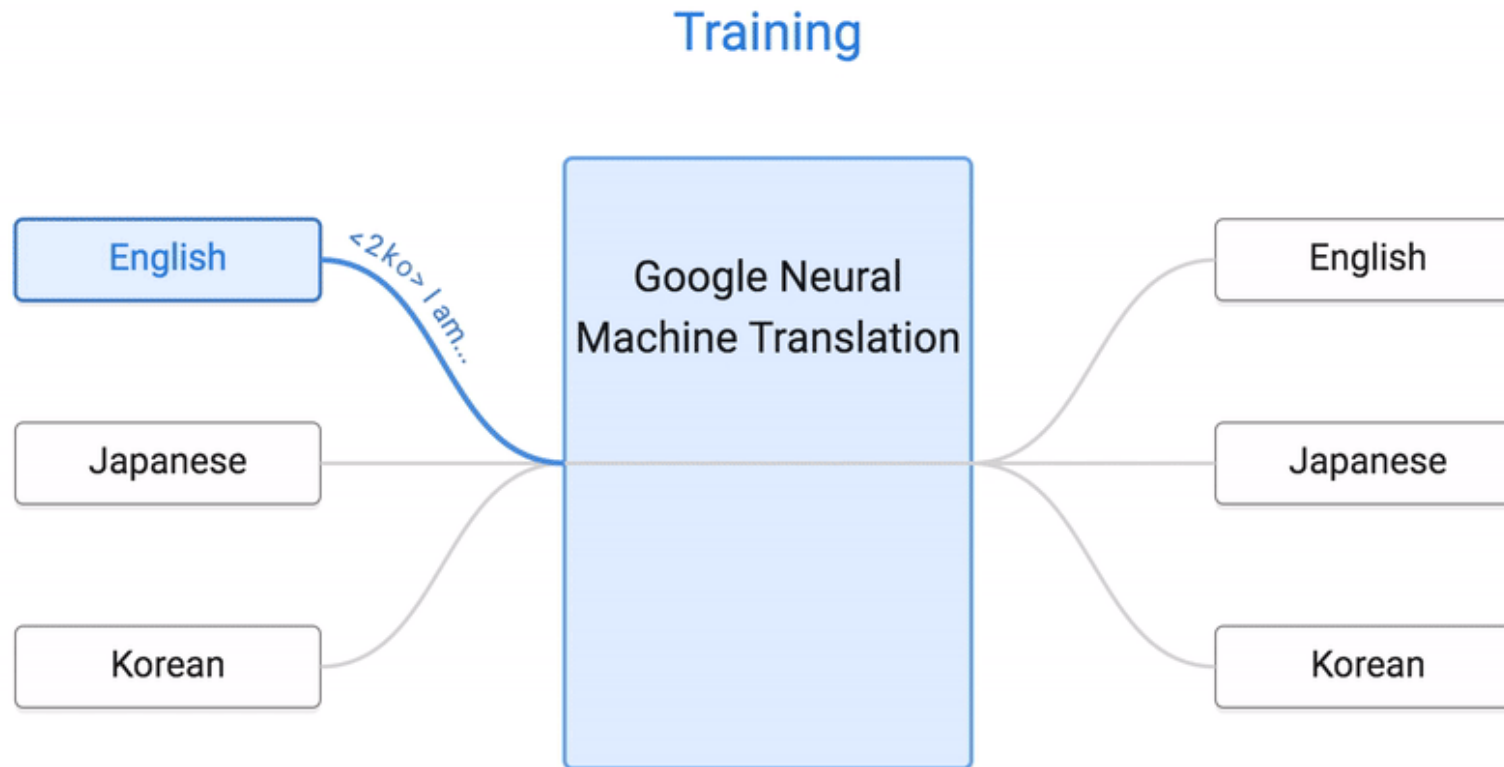


Машинный перевод (seq2seq + attention)

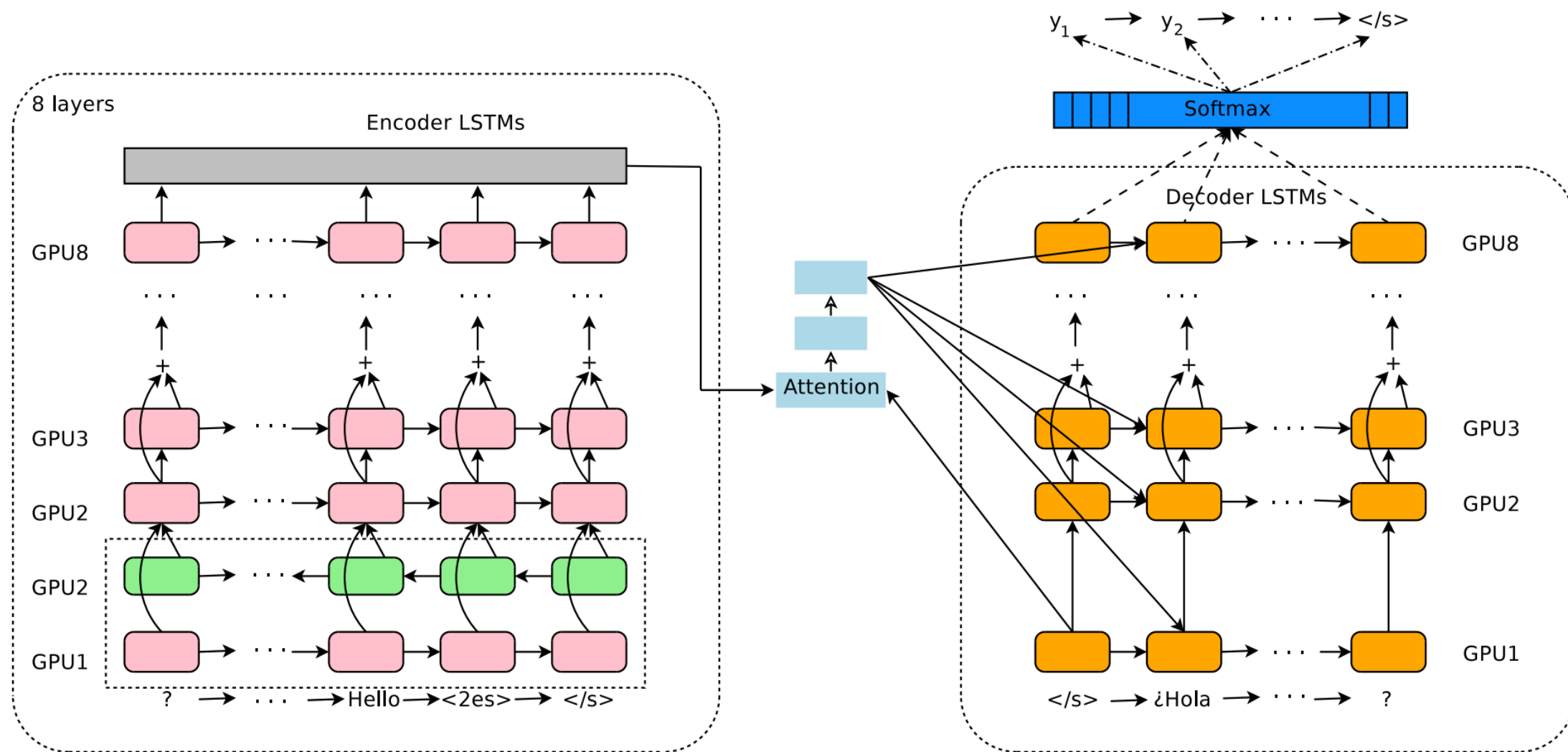


Multi-language перевод от Google

- Обучение encoder и decoder для каждого языка на разных парах языков одновременно



seq2seq + attention на 8 GPU



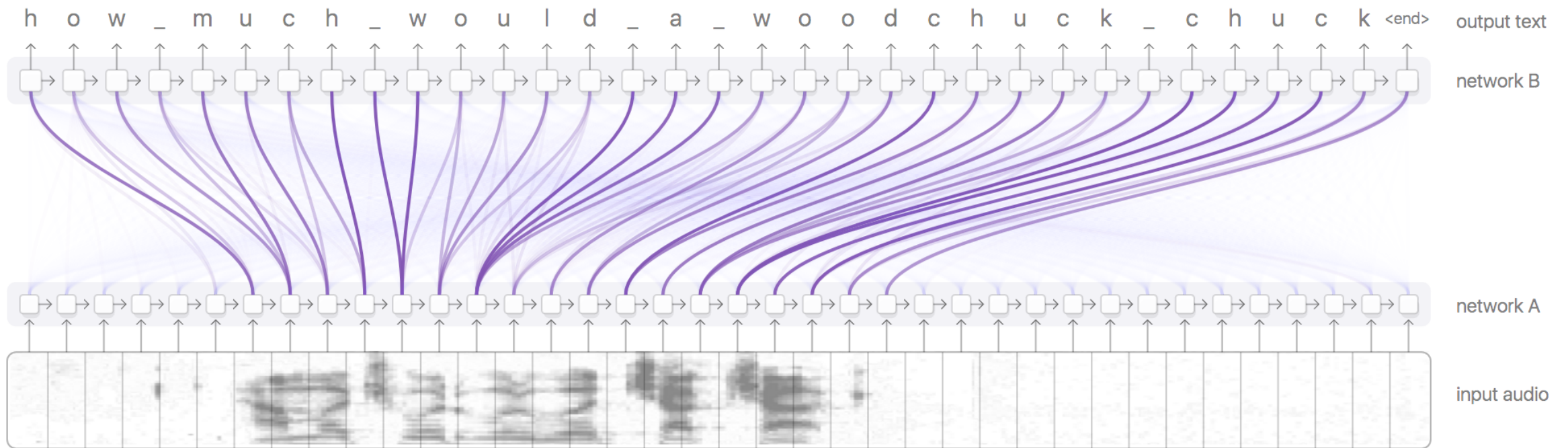
Zero-shot learning

- Система начинает неплохо переводить даже для тех пар языков, для которых не было пар примеров в обучении
 - **Zero-Shot:** обучили на English \leftrightarrow {Belarusian, Russian, Ukrainian}
 - **From-Scratch:** добавили еще Russian \leftrightarrow {Belarusian, Ukrainian}
 - **Incremental:** fine-tuning zero-shot на $\sim 3\%$ Russian \leftrightarrow {Belarusian, Ukrainian}

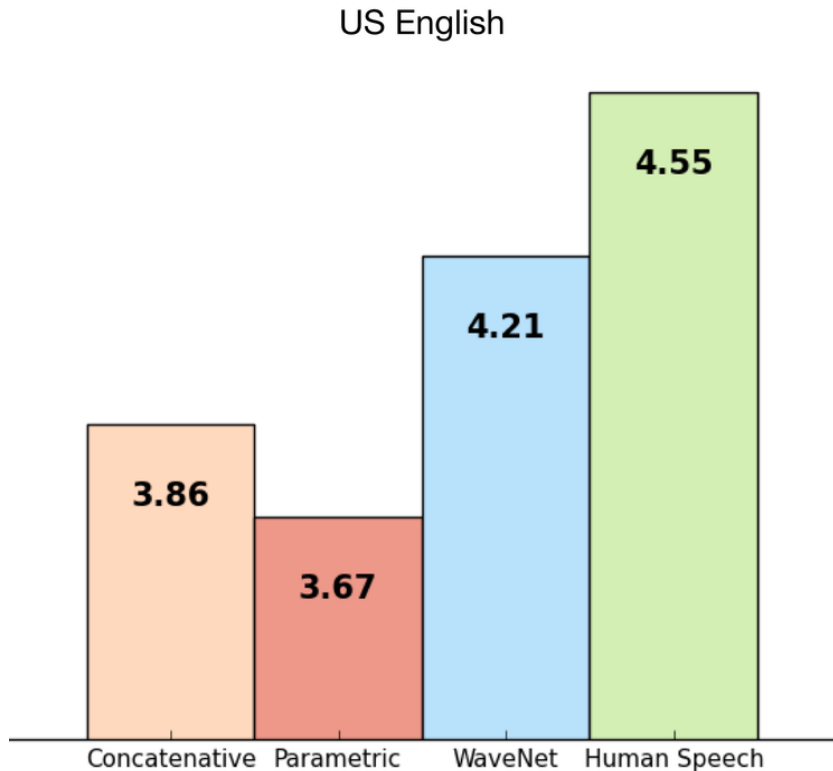
Table 6: BLEU scores for English \leftrightarrow {Belarusian, Russian, Ukrainian} models.

	Zero-Shot	From-Scratch	Incremental
English \rightarrow Belarusian	16.85	17.03	16.99
English \rightarrow Russian	22.21	22.03	21.92
English \rightarrow Ukrainian	18.16	17.75	18.27
Belarusian \rightarrow English	25.44	24.72	25.54
Russian \rightarrow English	28.36	27.90	28.46
Ukrainian \rightarrow English	28.60	28.51	28.58
Belarusian \rightarrow Russian	56.53	82.50	78.63
Russian \rightarrow Belarusian	58.75	72.06	70.01
Russian \rightarrow Ukrainian	21.92	25.75	25.34
Ukrainian \rightarrow Russian	16.73	30.53	29.92

Распознавание голоса (seq2seq + attention)



Text to speech (WaveNet)



Mean Opinion Scores (MOS) are a standard measure for subjective sound quality tests, and were obtained in blind tests with human subjects (from over 500 ratings on 100 test sentences)

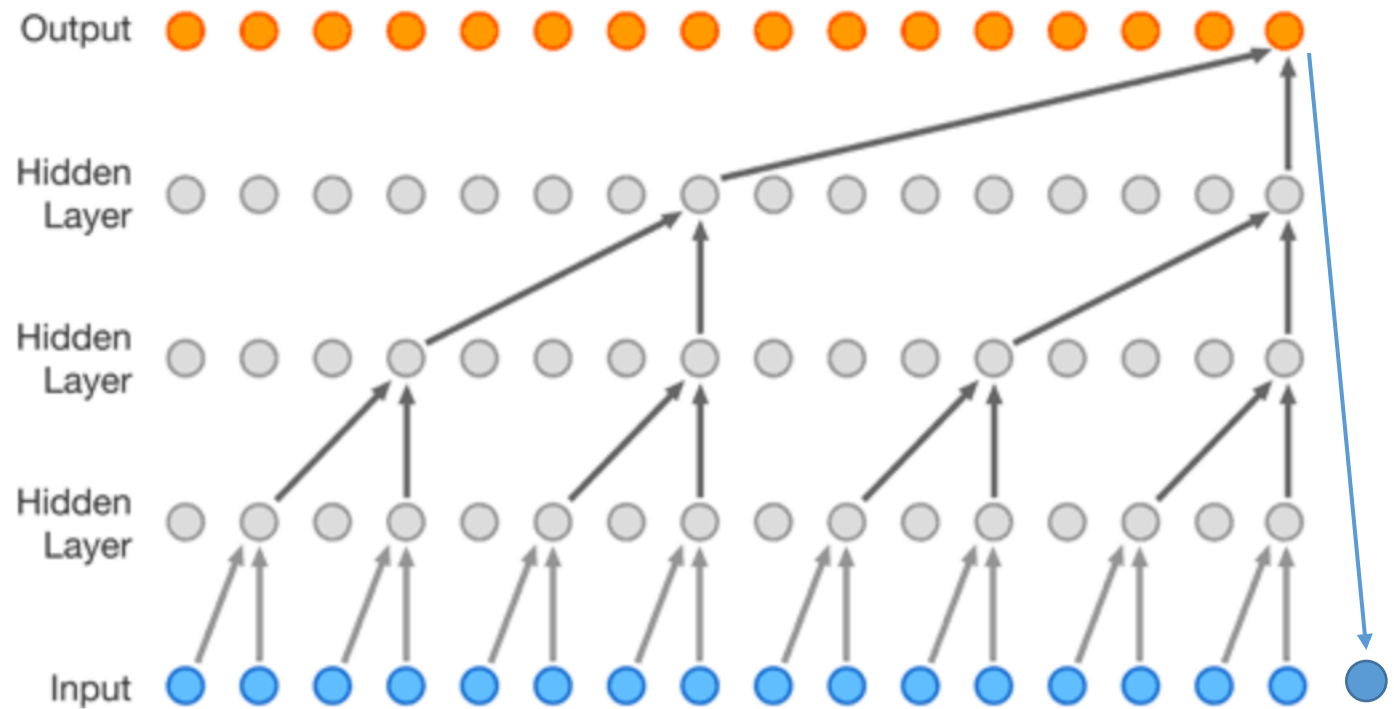
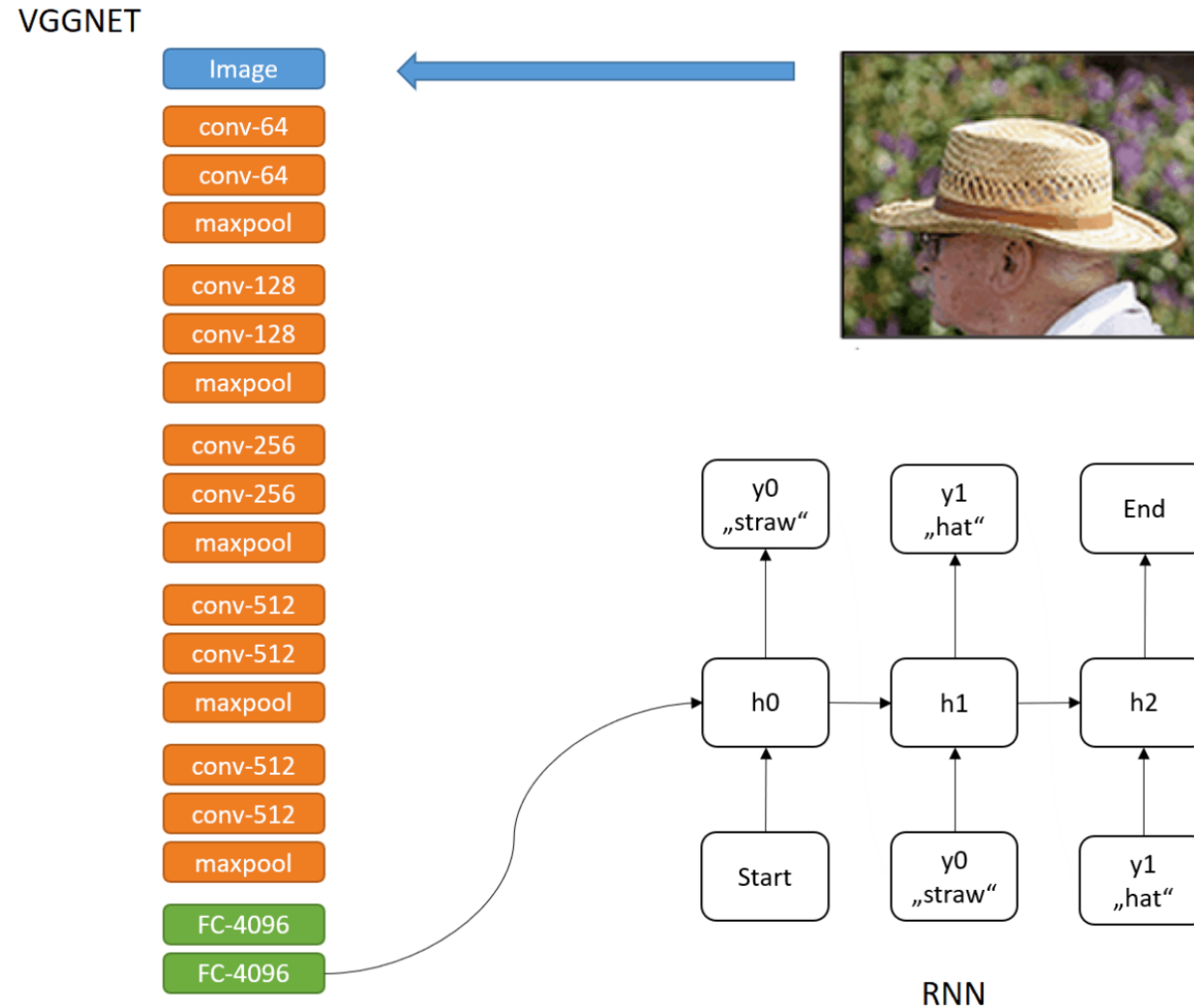


Image captioning (CNN encoder – RNN decoder)



Ссылки

- <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/>
- http://www.fit.vutbr.cz/research/groups/speech/publi/2010/mikolov_interspeech_2010_IS100722.pdf
- <http://distill.pub/2016/augmented-rnns/>
- Multilingual Neural Machine Translation <https://arxiv.org/abs/1611.04558>
- Pixel Recurrent Neural Networks <https://arxiv.org/abs/1601.06759>
- <https://deepmind.com/blog/wavenet-generative-model-raw-audio/>
- <https://www.tensorflow.org/tutorials/recurrent>
- <https://www.tensorflow.org/tutorials/seq2seq>