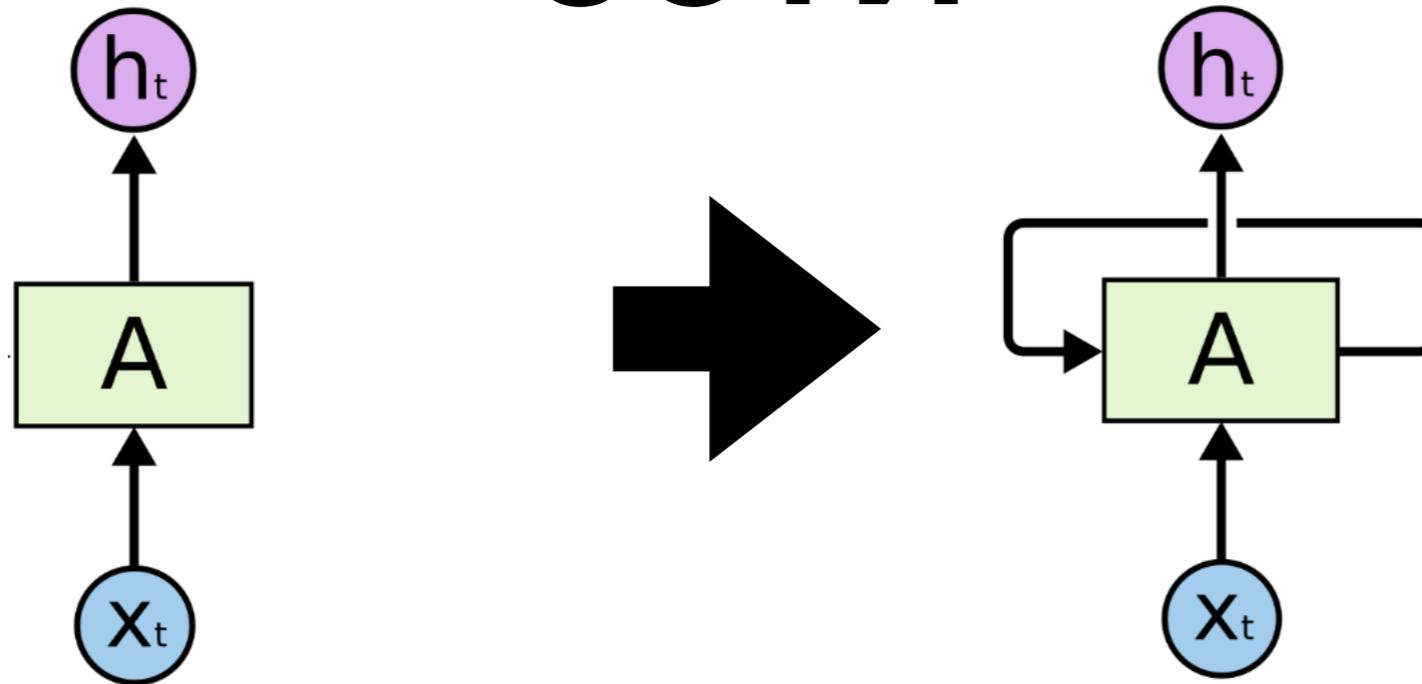


Рекуррентные нейронные

сети



- Принимают на вход упорядоченную последовательность векторов
- Внутреннее состояние (веса) зависят от предыдущих примеров, т.е. **интерпретация данных зависит от контекста**

Применение:

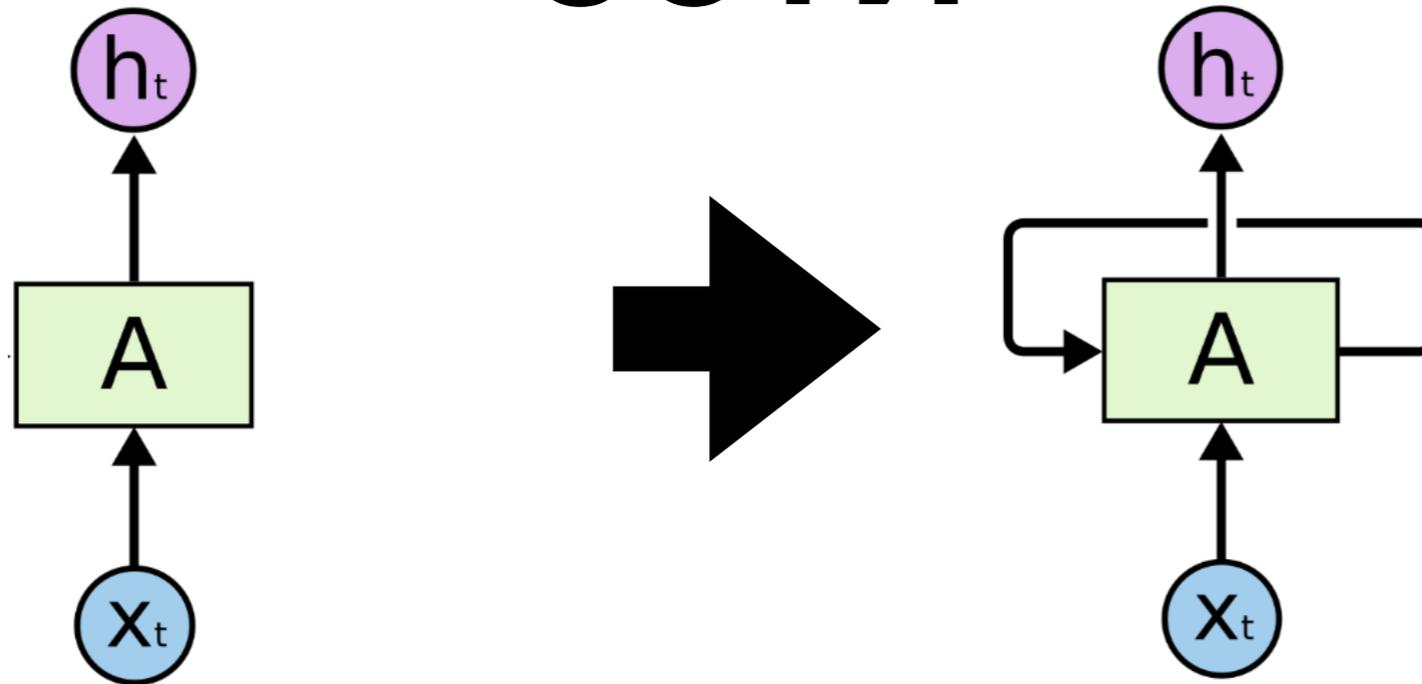
анализ временных рядов (регрессия, классификация, кластеризация, поиск аномалий)

распознавание естественного языка и автоматический перевод

интерпретация видео/аудио

Рекуррентные нейронные

сети



- Принимают на вход упорядоченную последовательность векторов
- Внутреннее состояние (веса) зависят от предыдущих примеров

Сеть Элмана

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h)$$

$$y_t = \sigma_y(W_y h_t + b_y)$$

h_t - скрытое состояние

Сеть Джордана

$$h_t = \sigma_h(W_h x_t + U_h y_{t-1} + b_h)$$

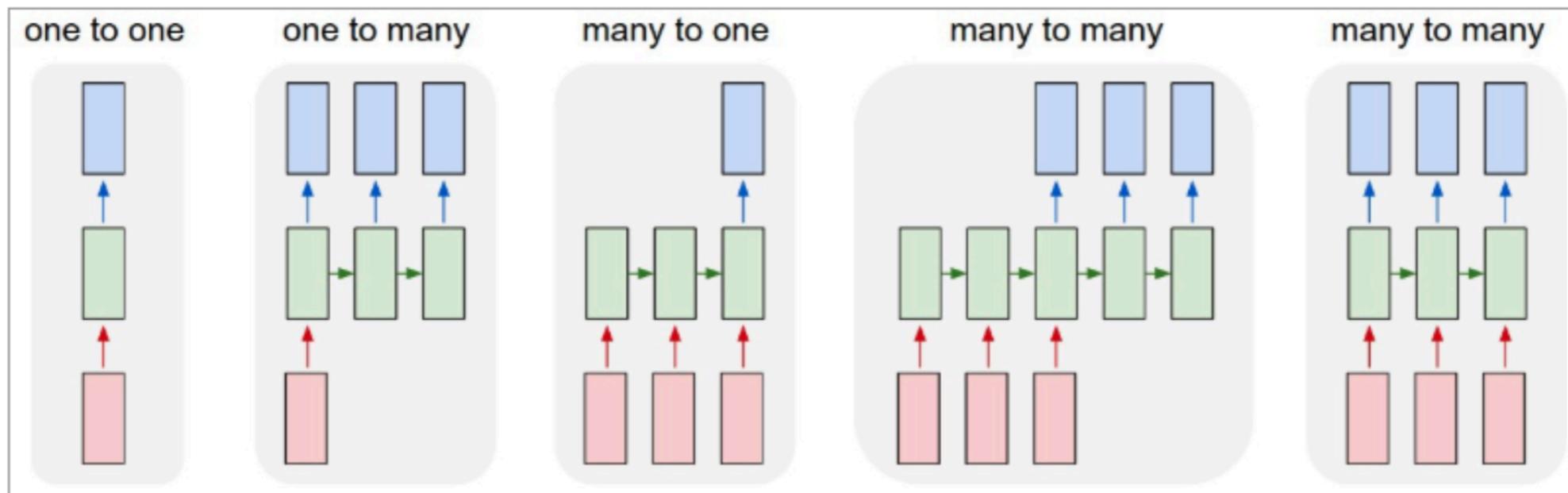
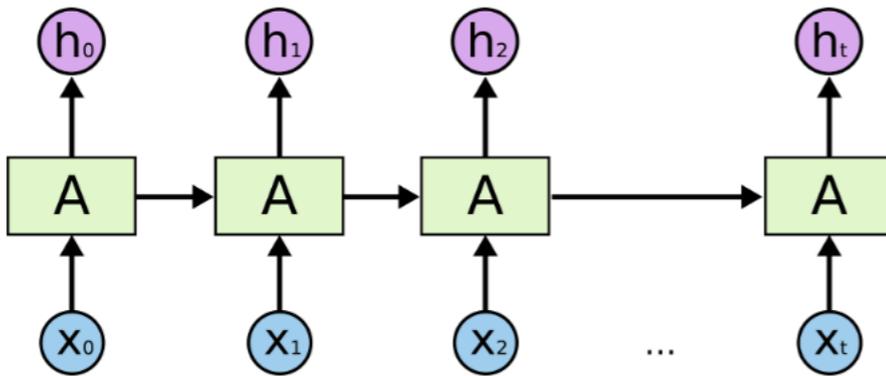
$$y_t = \sigma_y(W_y h_t + b_y)$$

W_h, U_h, b_h, W_y, b_y - обучаемые веса

Рекуррентные нейронные сети

сети

Операции над последовательностями

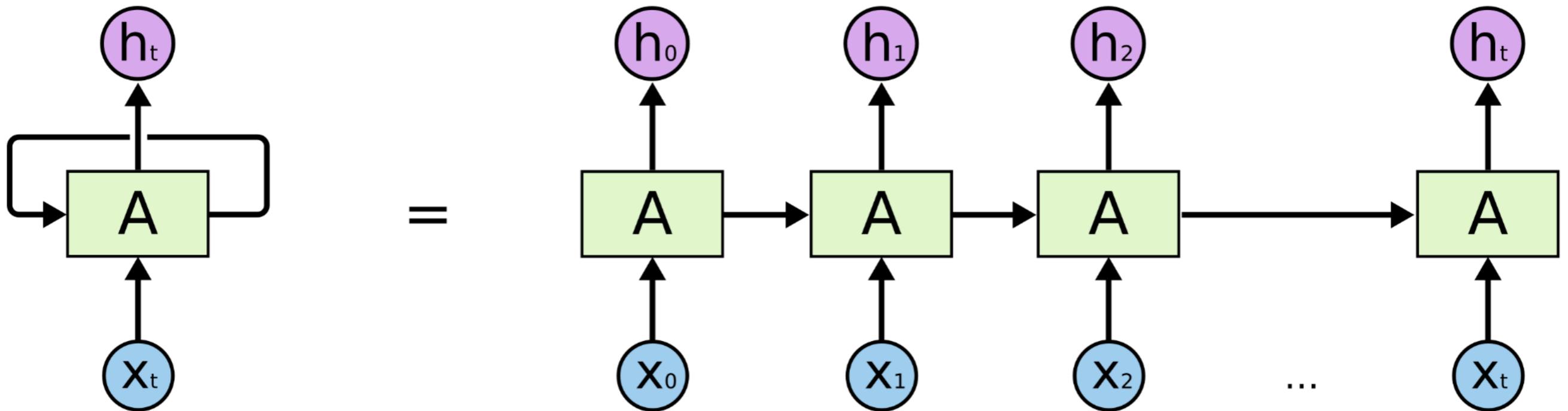


Алгоритм обратного распространения ошибки

Backpropagation Through Time

для последовательности конечной длины t

1. разворачиваем выражение

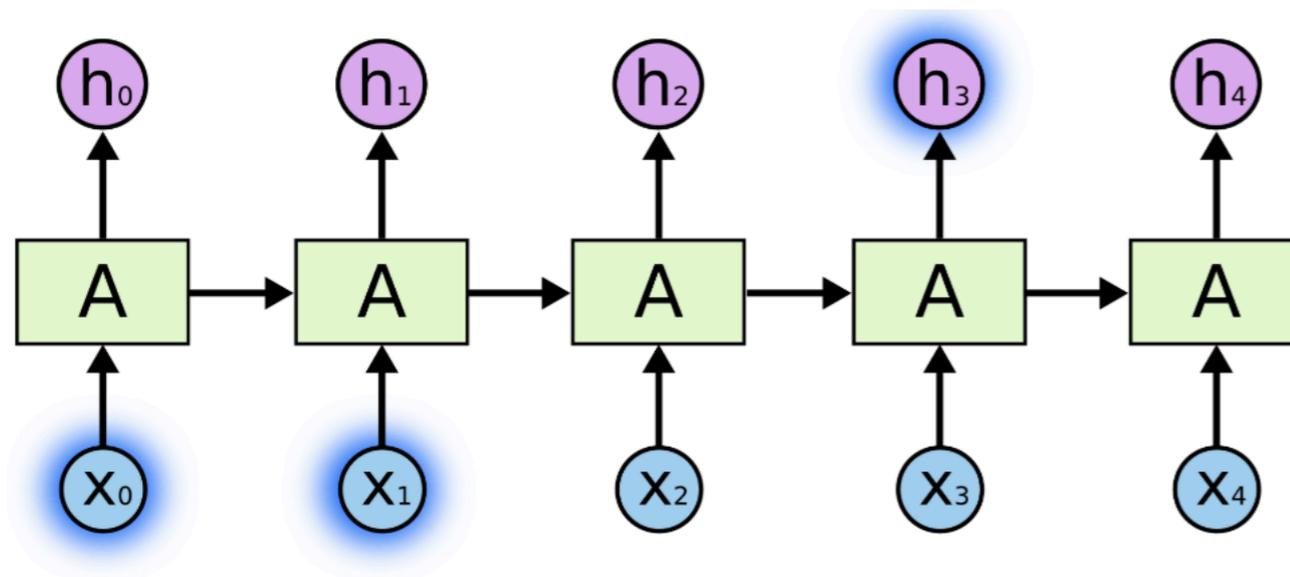


2. применяем стандартный алгоритм обратного распространения

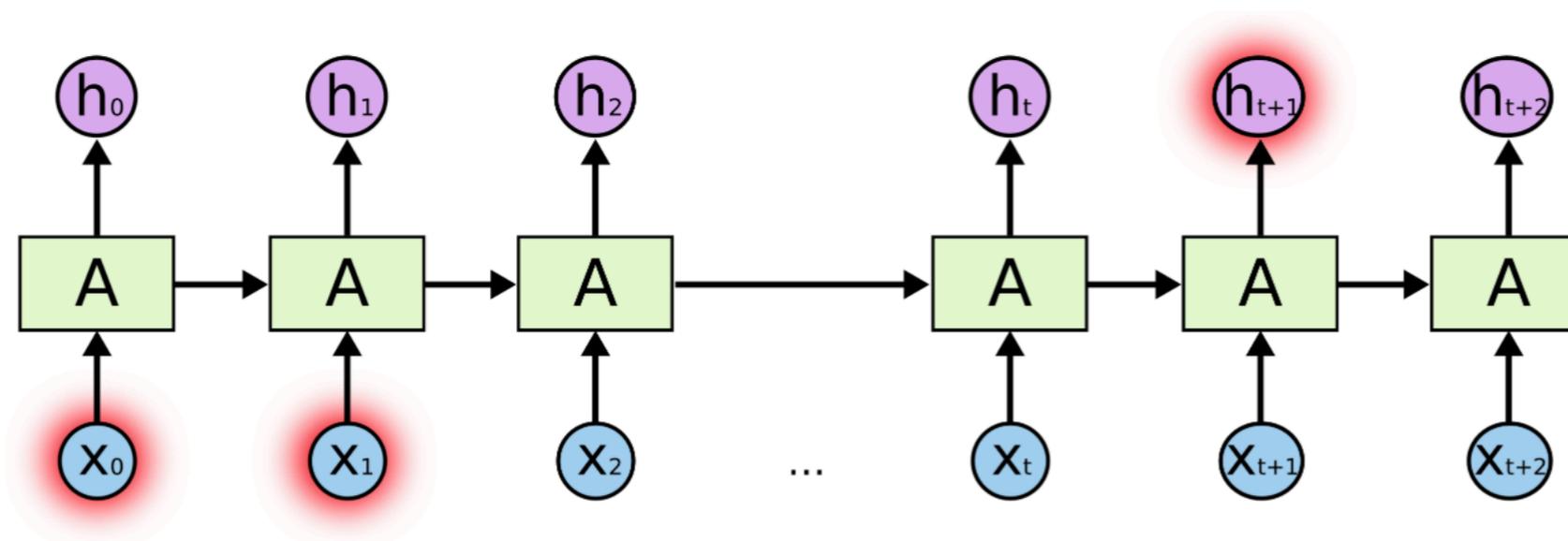
Проблема краткосрочной памяти

Только недавние примеры существенно влияют на результат

Hochreiter (1991); Bengio, et al. (1994)



- работает

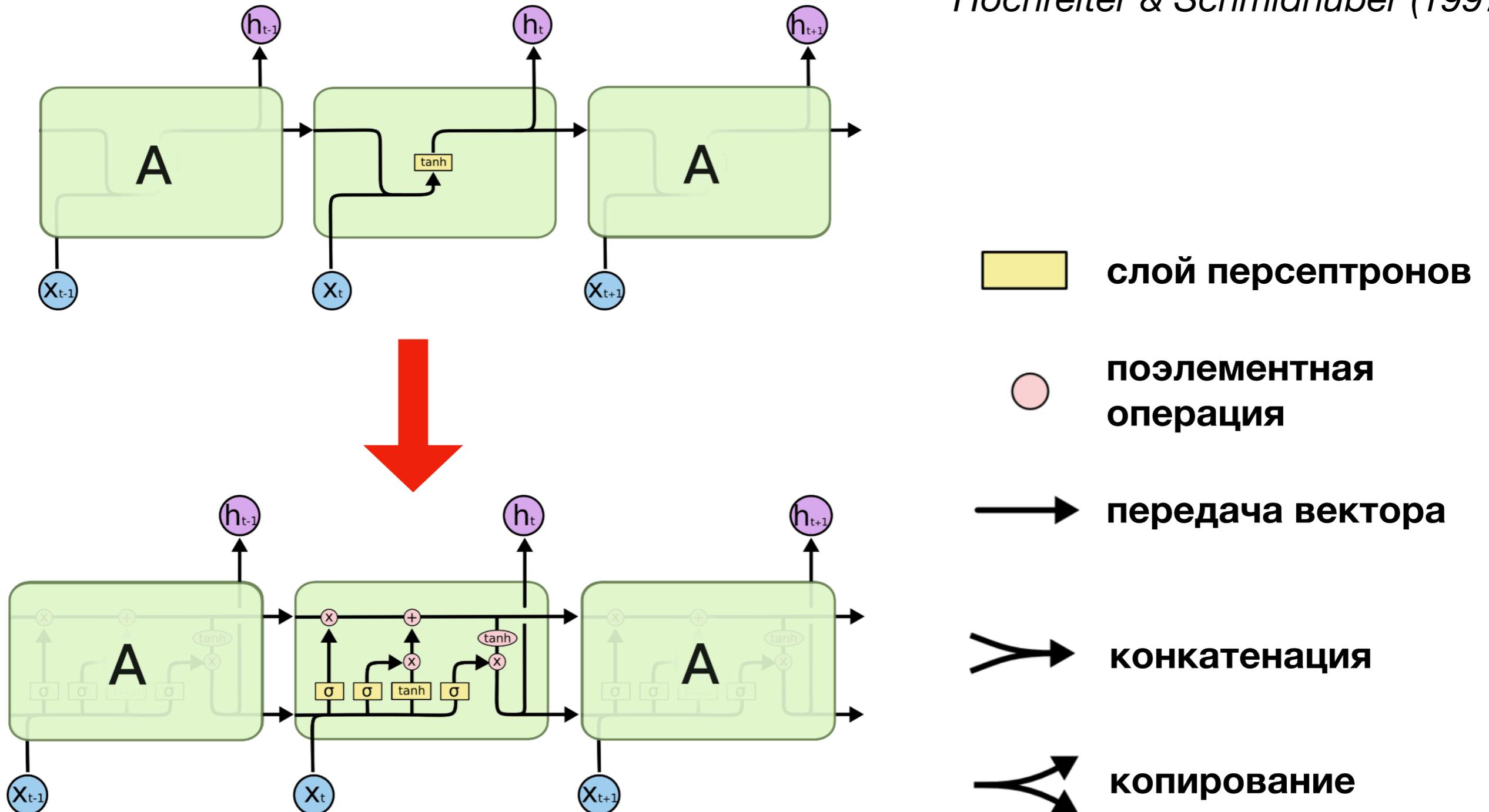


- не работает

Сеть с долговременной и кратковременной памятью

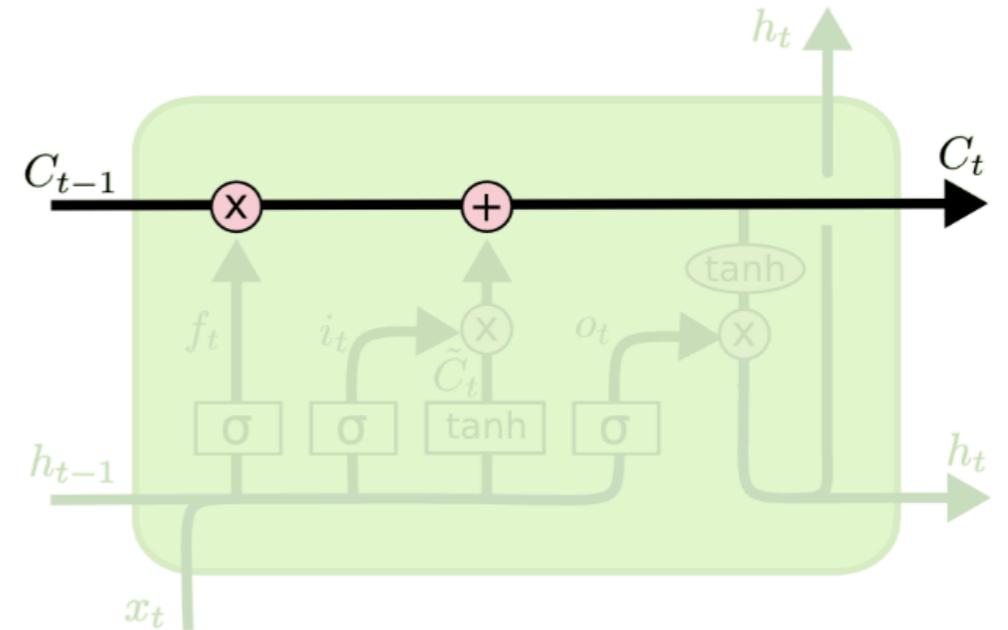
Long short term memory, LSTM

Hochreiter & Schmidhuber (1997)

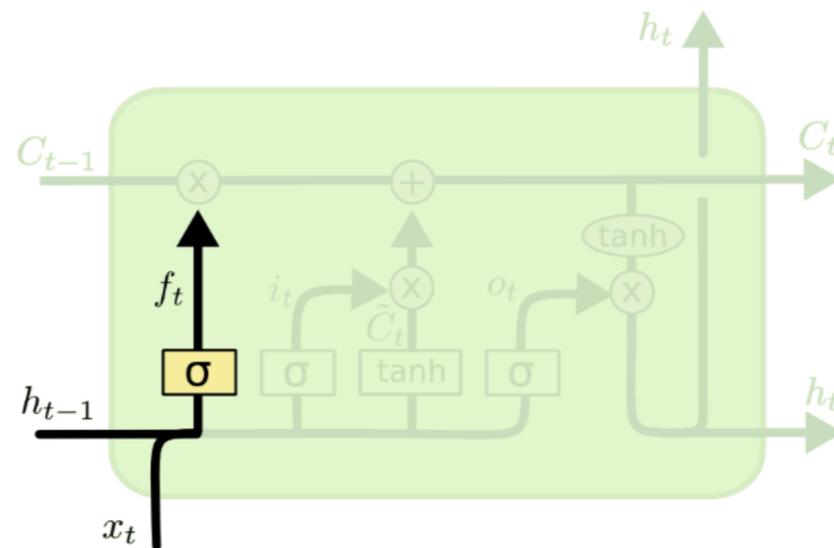
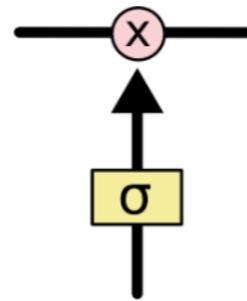


LSTM в деталях

C_t - внутреннее состояние блока LSTM
изменяется сравнительно медленно



Механизм (шлюз) забывания
forget gate

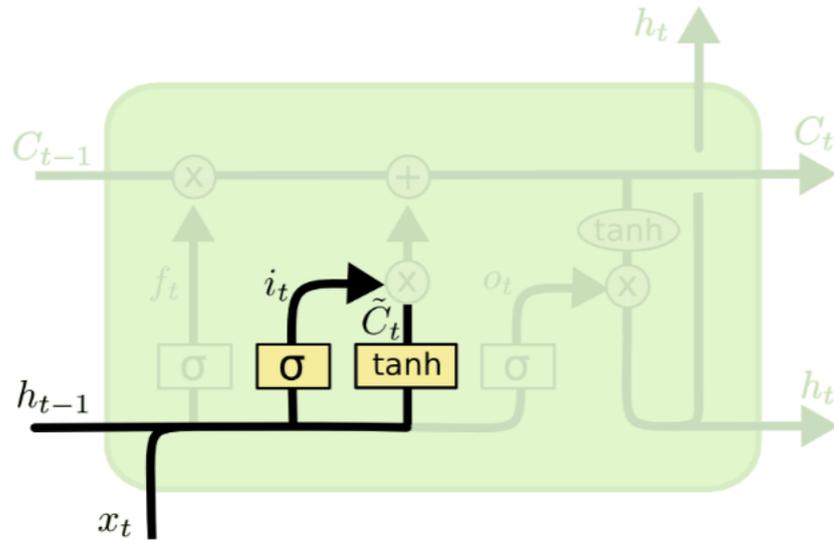


$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

LSTM в деталях

Механизм (шлюз) накопления информации

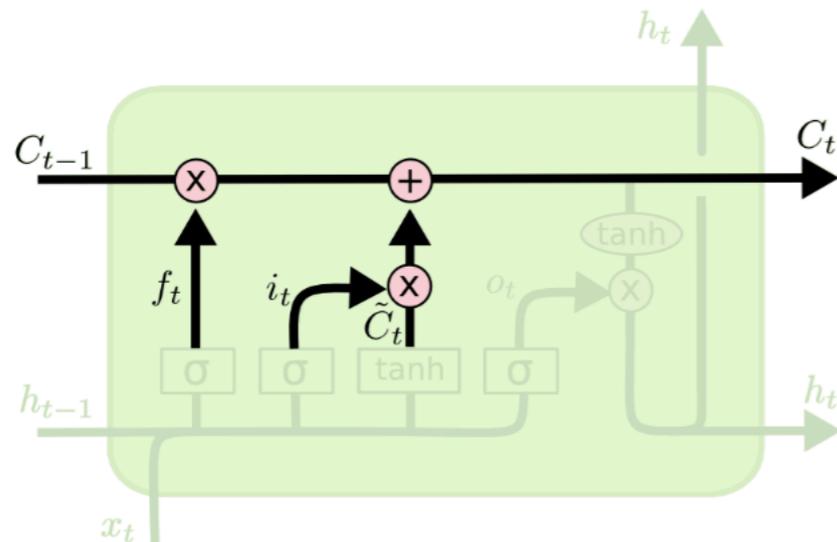
input gate



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

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

Обновление внутреннего состояния

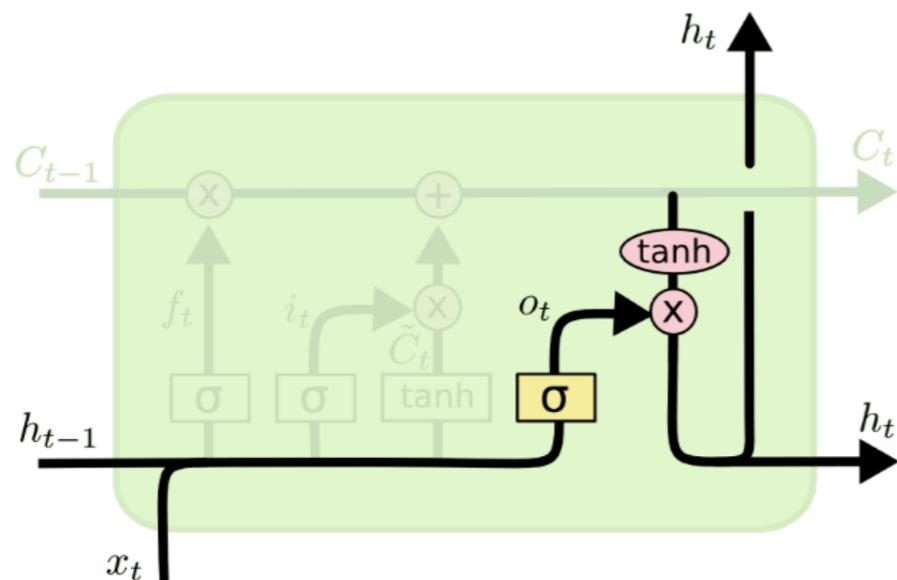


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

LSTM в деталях

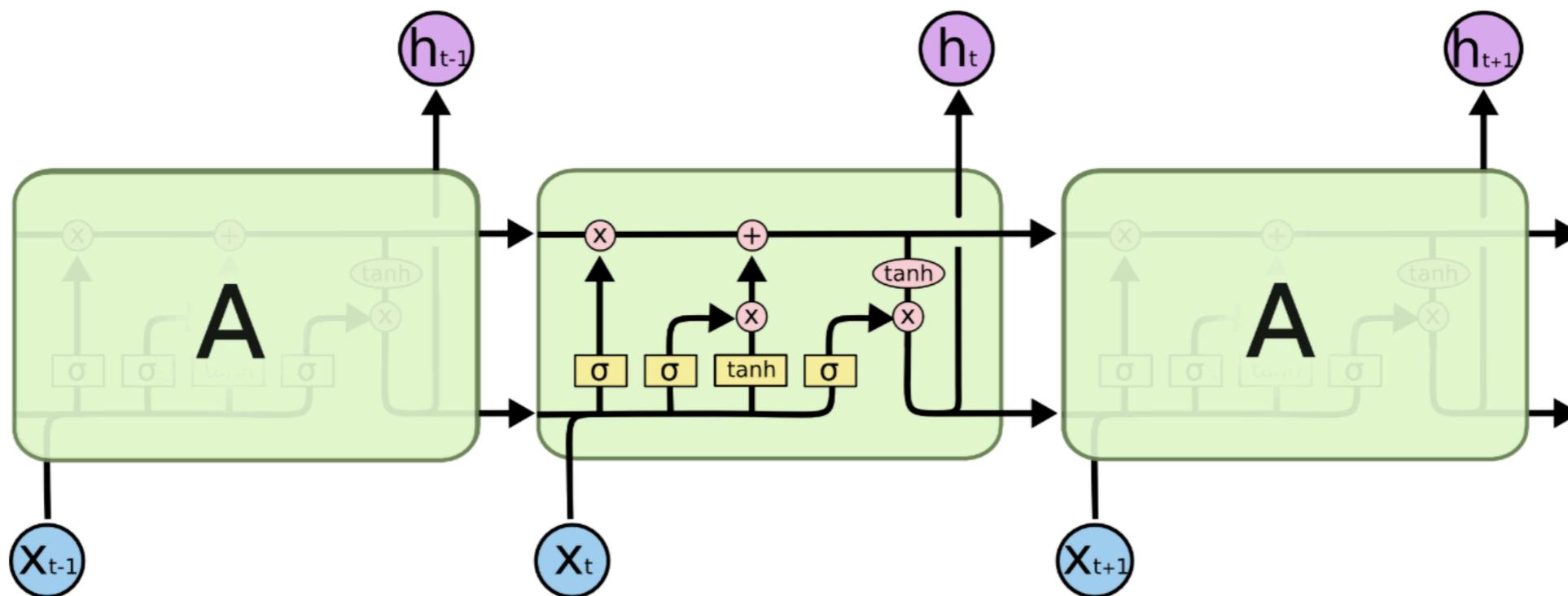
Вычисление выходного сигнала

output gate



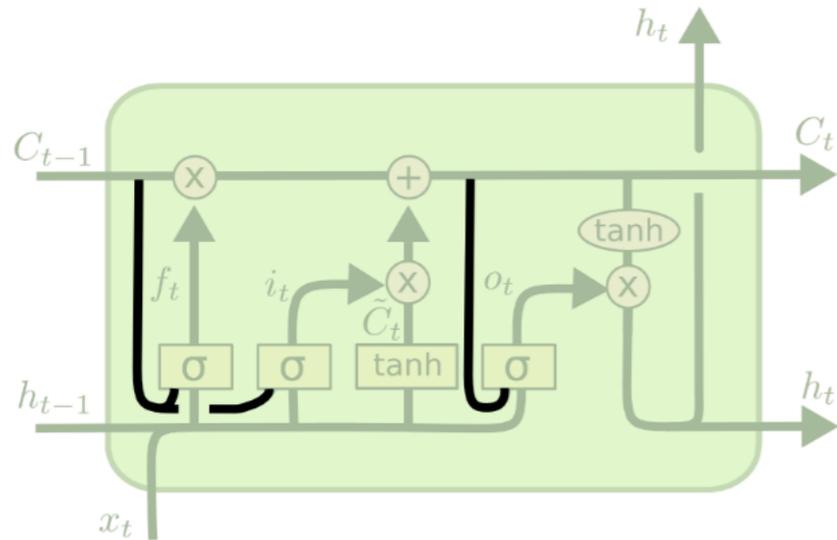
$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

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



LSTM модификации

“peerhole connections.” Gers & Schmidhuber (2000)



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

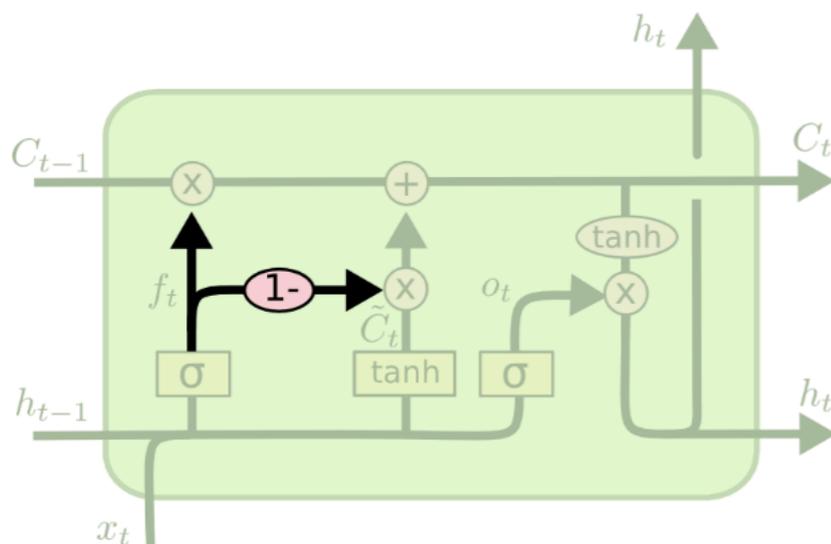
$$i_t = \sigma (W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i)$$

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

внутреннее состояние влияет на шлюзы

Объединение forget и input gates.

объем забываемой информации “примерно равен” объему поступающей

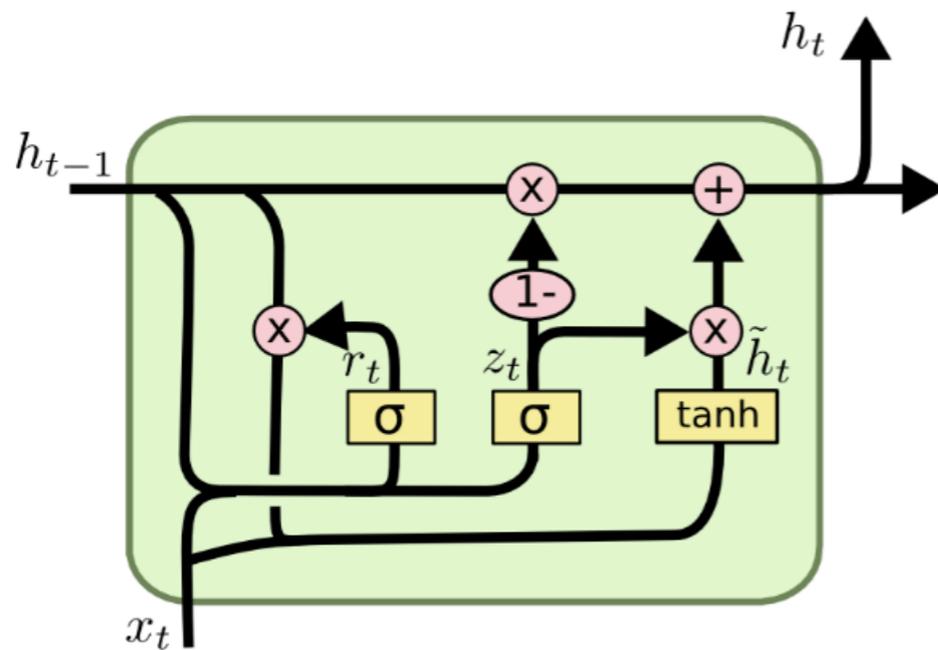


$$C_t = f_t * C_{t-1} + (1 - f_t) * \tilde{C}_t$$

Gated Recurrent Unit (GRU)

Cho, et al. (2014)

Объединение C_t и h_t



$$z_t = \sigma (W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma (W_r \cdot [h_{t-1}, x_t])$$

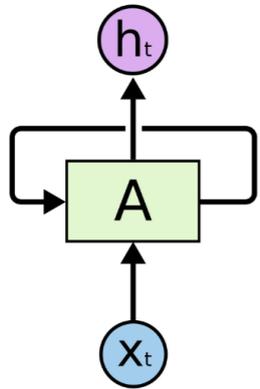
$$\tilde{h}_t = \tanh (W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

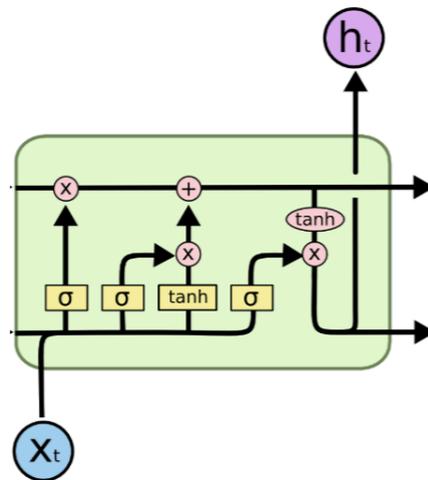
Реализация RNN в Keras

<https://keras.io/layers/recurrent/>

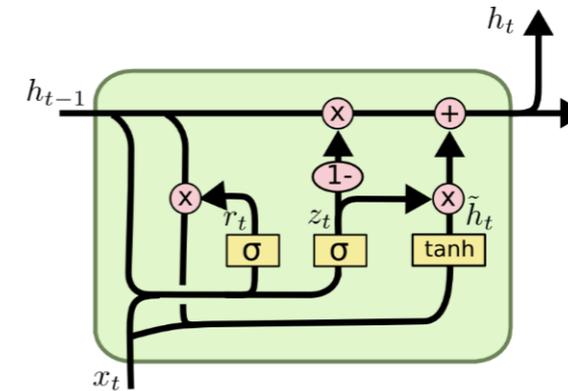
пакет keras.layers



SimpleRNN



LSTM, ConvLSTM2D



GRU

Пример (автокодировщик):

```
from keras.layers import Input, LSTM, RepeatVector
from keras.models import Model
```

```
inputs = Input(shape=(timesteps, input_dim))
encoded = LSTM(latent_dim)(inputs)
```

```
decoded = RepeatVector(timesteps)(encoded)
decoded = LSTM(input_dim, return_sequences=True)(decoded)
```

```
sequence_autoencoder = Model(inputs, decoded)
encoder = Model(inputs, encoded)
```