

Рекуррентные нейронные сети

Последовательность наблюдений.

$$x_1, x_2, \dots, x_{n-2}, x_{n-1}.$$

Значение x_n :

$$x_n = f(x_{n-1}, \dots, x_{n-l}).$$

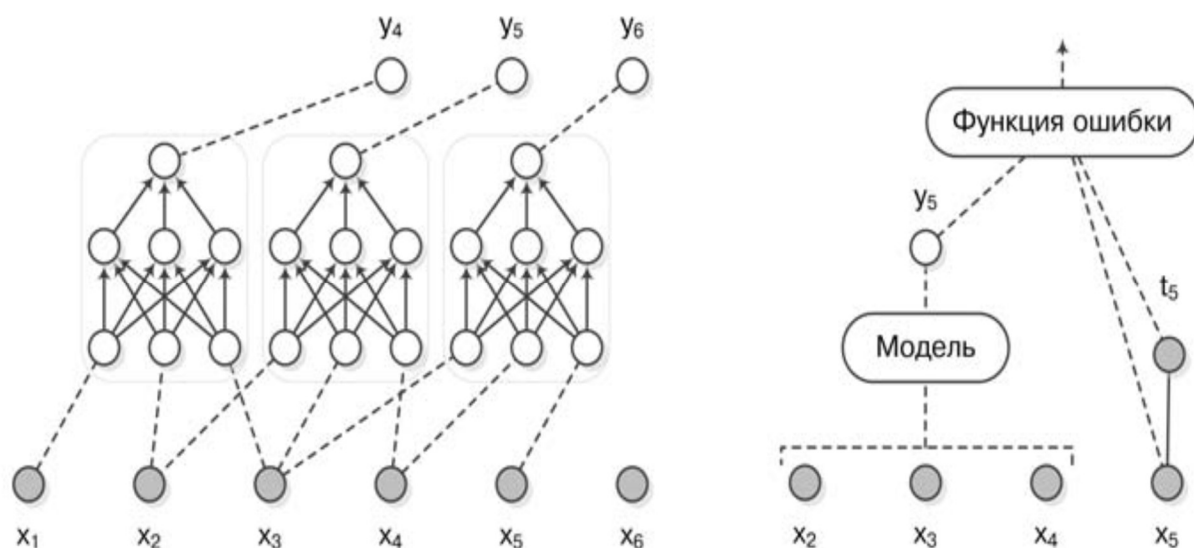


Рисунок 1. Архитектура обычной нейронной сети с фиксированным размером истории. *Слева:* одна и та же нейронная сеть применяется к последовательным окнам входа. *Справа:* результат нейронной сети сравнивается с очередным элементом последовательности

$$y_4 = f(x_1, x_2, x_3), x_5 = f(x_2, x_3, x_4), y_6 = f(x_3, x_4, x_5).$$

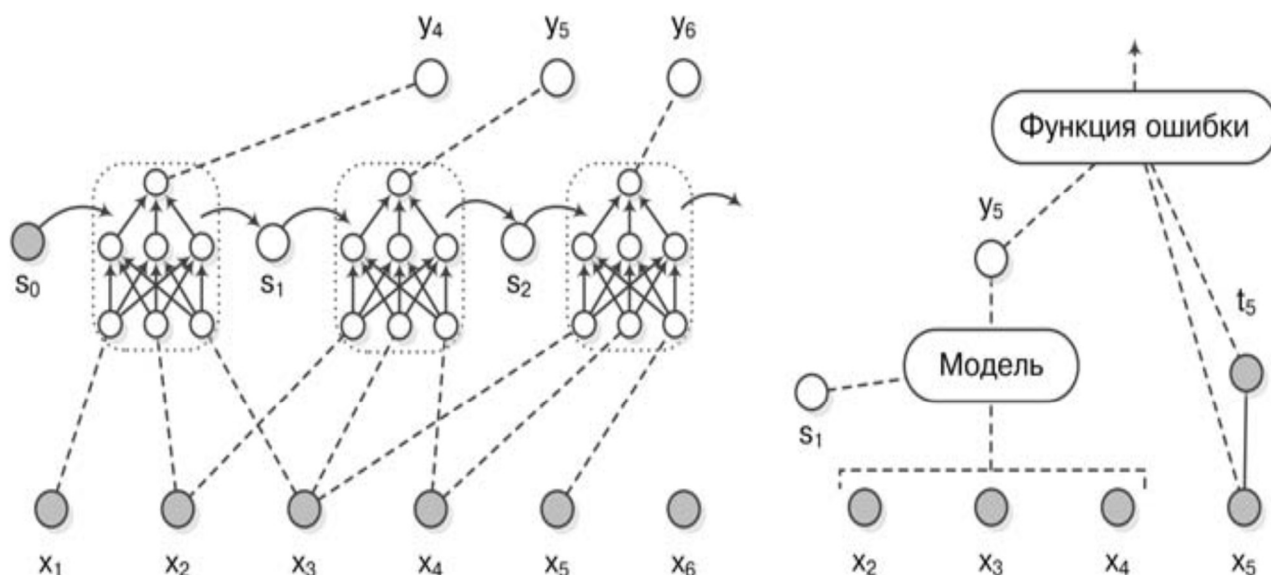


Рисунок 2. Архитектура рекуррентной нейронной сети в тех же обозначениях, что на рис 1. *Слева:* рекуррентная нейронная сеть получает на вход свое предыдущее состояние и последовательные окна входа. *Справа:* результат рекуррентной нейронной сети точно так же сравнивается с очередным элементом последовательности

Основные типы задач машинного обучения, связанных с последовательностями. По характеру входов и выходов задачи можно выделить такие пять вариантов:

1. Один вход, один выход (one-to-one). Здесь последовательности как бы и нет, мы просто должны независимо обработать каждый элемент входов и получить соответствующий выход, никакие скрытые состояния никуда не передаются, однако заметим, что работа с последовательностью как на рис. 1, когда окна обрабатываются независимо, вполне укладывается в эту схему, просто входы получатся достаточно сильно зависимыми.

2. Один вход, последовательность выходов (one-to-many). Здесь мы должны «развернуть» вход, который сам по себе не имеет структуры последовательности, в последовательность выходов; например, аннотирование картинок представляет собой такую задачу: на входе картинка, на выходе текст (последовательность).

3. Последовательность входов, один выход (many-to-one). В этот тип задач укладываются любые задачи классификации последовательностей, например, анализ тональности (sentiment analysis): по данному тексту (то есть последовательности) выдать, положительно или отрицательно он окрашен.

4. Последовательность входов, затем последовательность выходов (many-to-many). Здесь речь идет о том, чтобы «свернуть» входную последовательность, закодировать ее неким скрытым состоянием, а потом «развернуть» это скрытое состояние обратно в уже совершенно другую

последовательность; например, по этой общей схеме работают системы машинного перевода (вход — предложение на одном языке, выход — на другом) и диалоговые системы (вход — реплика собеседника, выход — своя собственная реплика).

5. синхронизированные последовательности входов и выходов (synchronized many-to-many). Здесь нужно снабдить своей меткой каждый элемент последовательности, но в отличие от one-to-one есть смысл также переносить на следующий временной шаг некое скрытое состояние, например, представьте, что нам нужно разметить видеопоток, в котором каждый последующий кадр, конечно, представляет собой самостоятельную картинку, но она обычно очень похожа на предыдущую и следующую.

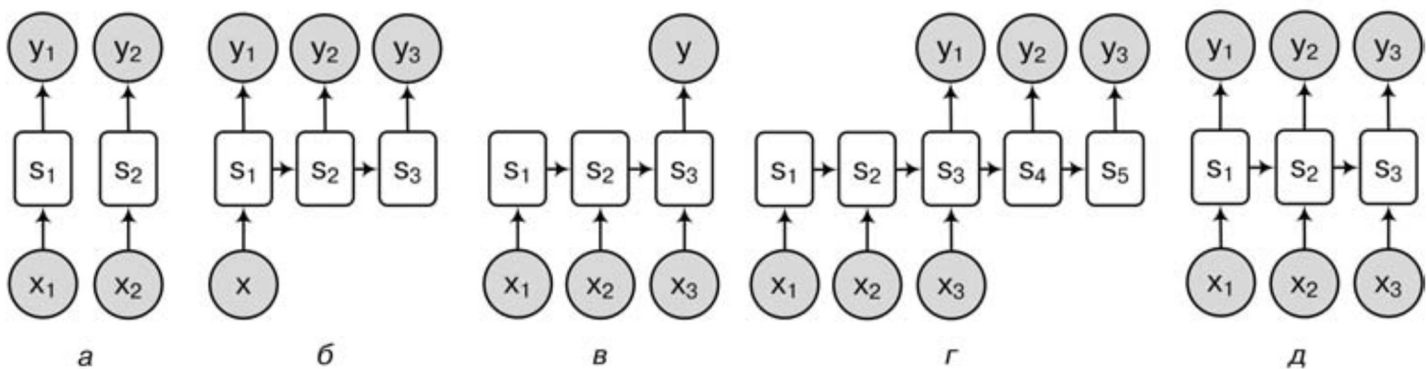


Рисунок 3. Задачи с последовательностями: *а* — один вход, один выход; *б* — один вход, последовательность выходов; *в* — последовательность входов, один выход; *г* — последовательность входов, затем последовательность выходов; *д* — синхронизированные последовательности входов и выходов

«Простая» рекуррентная нейронная сеть

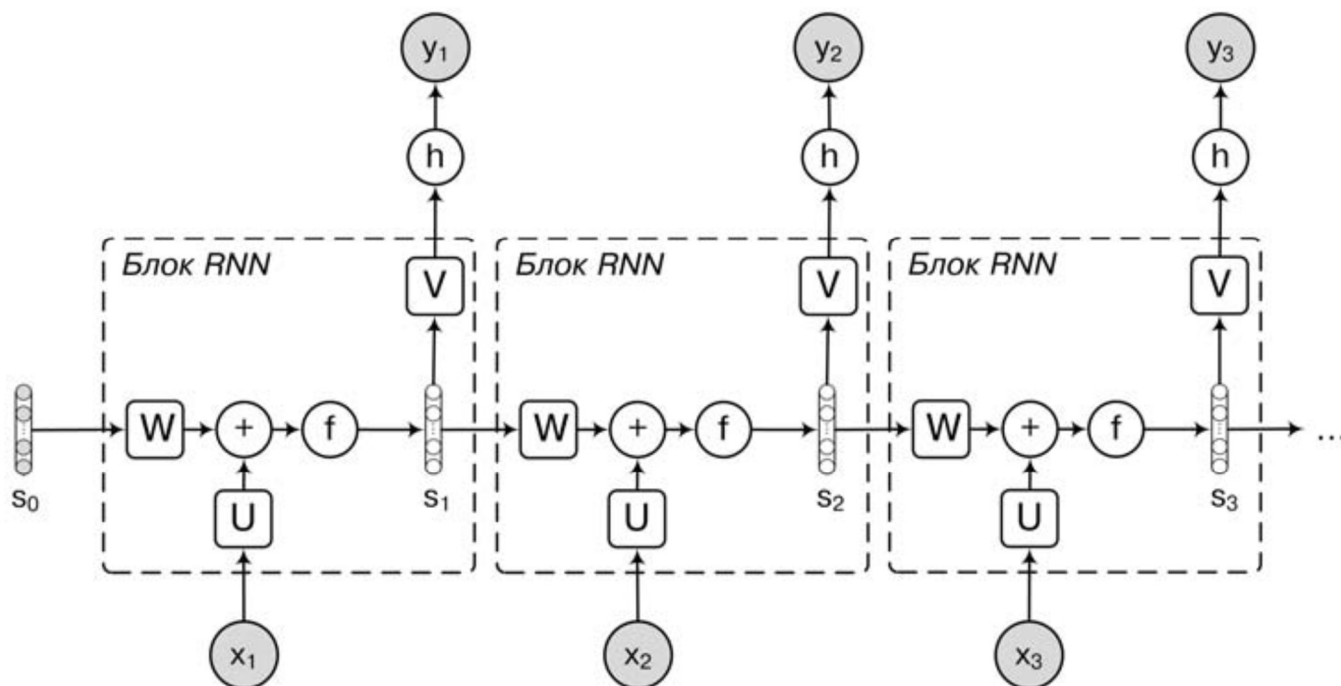


Рисунок 4. Архитектура «простой» рекуррентной нейронной сети

W – матрица весов для перехода между скрытыми состояниями,

U – матрица весов для входов,

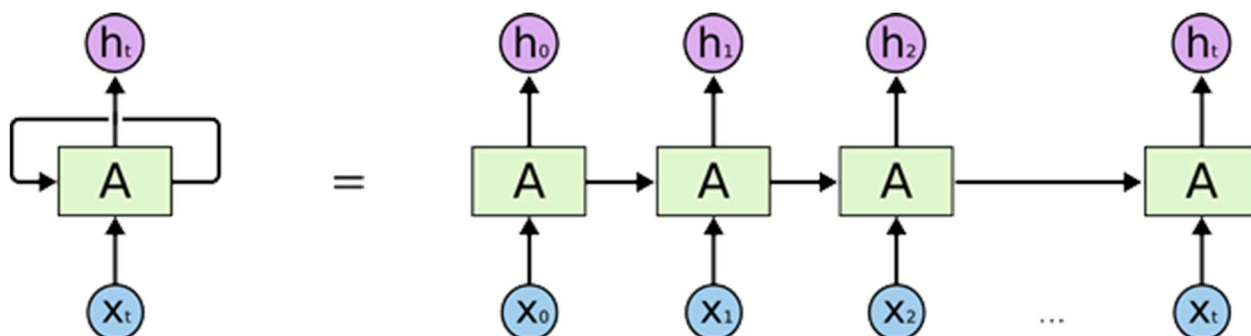
V – матрица весов выходов.

$$\mathbf{a}_t = \mathbf{b} + W_{s_{t-1}} + U_{x_t}, \mathbf{s}_t = f(\mathbf{a}_t);$$

$$\mathbf{o}_t = \mathbf{c} + V_{s_t}, \mathbf{y}_t = h(\mathbf{o}_t);$$

$$s_i = h(x_i, x_{i+1}, x_{i+2}, s_{i+2})$$

$$\begin{aligned} y_6 &= f(x_3, x_4, x_5, s_2) = f(x_3, x_4, x_5, h(x_2, x_3, x_4, s_1)) = \\ &= f(x_3, x_4, x_5, h(x_2, x_3, x_4, h(x_1, x_2, x_3, s_0))). \end{aligned}$$



LSTM сеть

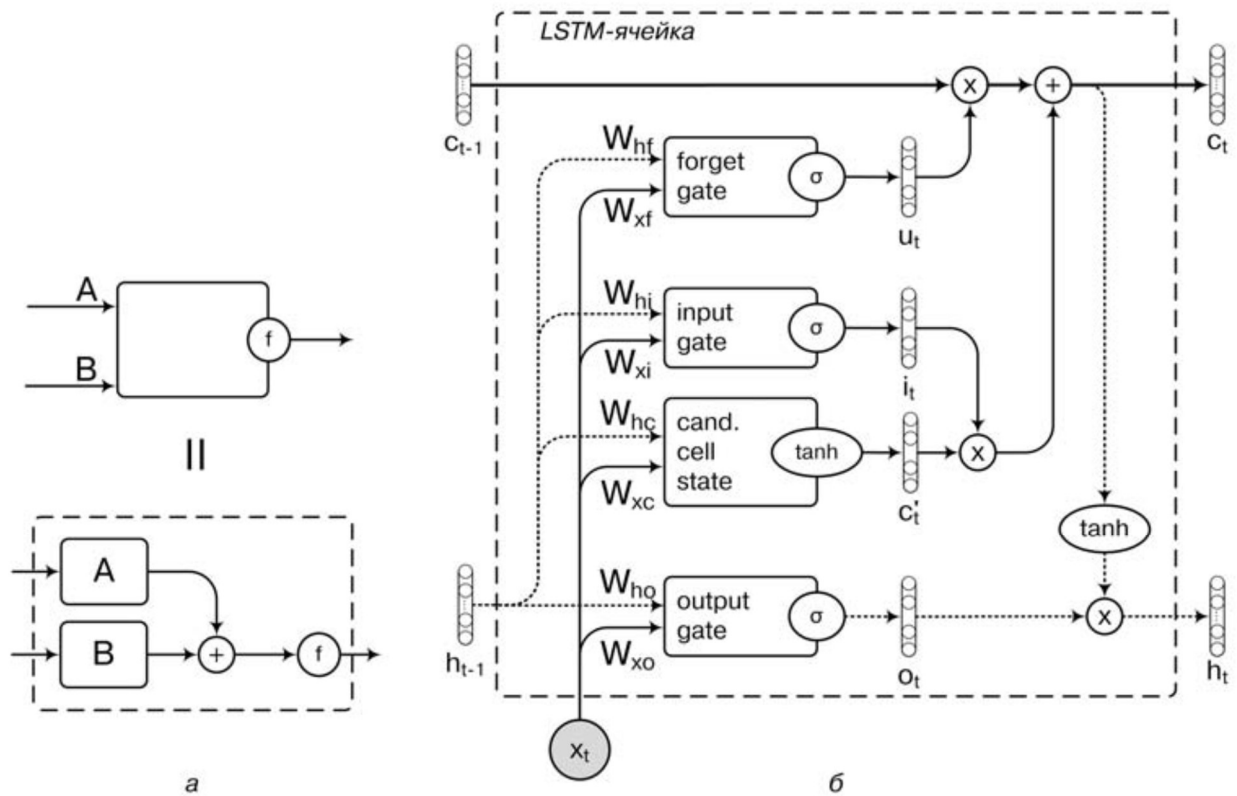
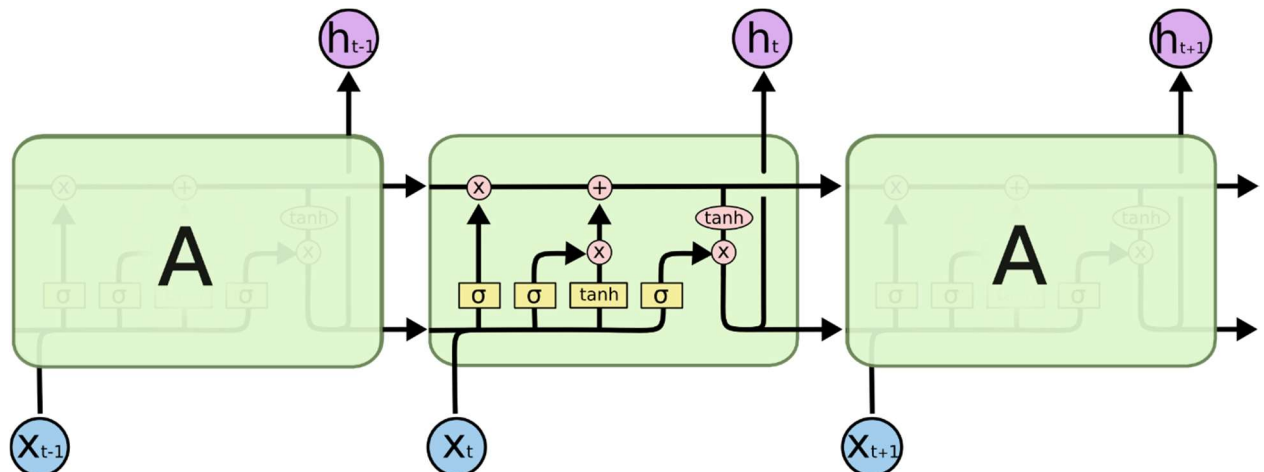


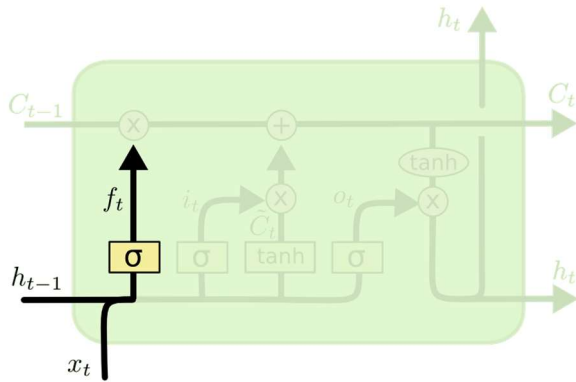
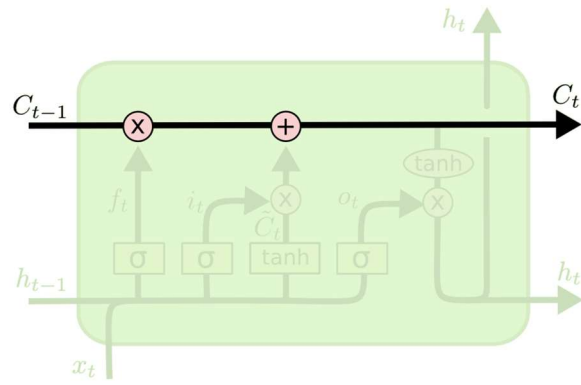
Рисунок 5. LSTM: *а* — обозначение вентиля (gate) с двумя входами; *б* — структура LSTM-ячейки

Вентиль (gate) LSTM управляет передачей сигнала внутри ячейки LSTM

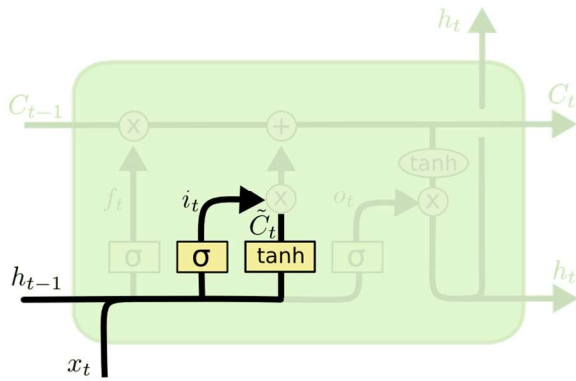
Типы вентиля в LSTM

- Вентиль забвения (forget gate)
- Входной вентиль (input gate)
- Выходной вентиль (output gate)



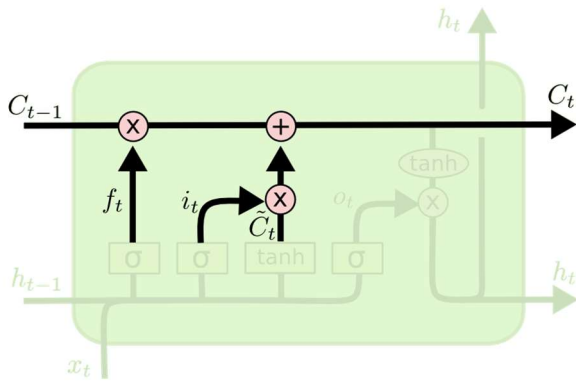


$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$

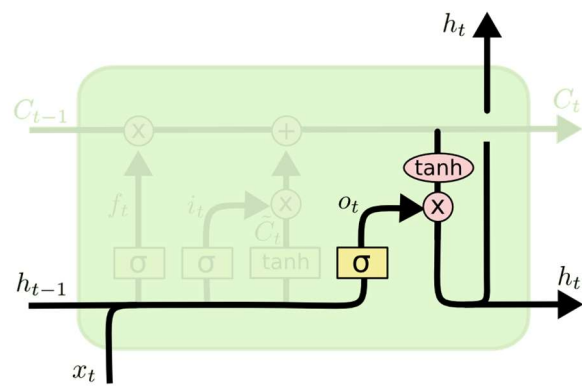


$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i)$$

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



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



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

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

