Рекуррентные нейронные сети

Елизавета Лазарева

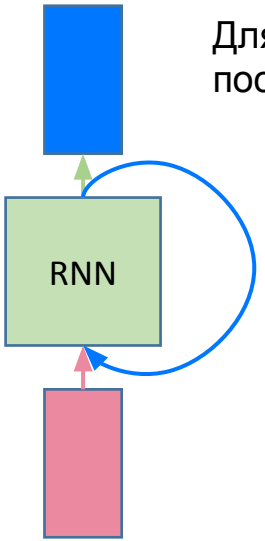
Мотивация

1. Как предсказать следующее слово?
2. Можно ли учитывать контекст, и как это делать?
3. Как обрабатывать длинные временные последовательности?
4. Как предсказывать изменения на карте (гео, метео) на основе карт за прошлые дни?

Сеть прямого распространения vs рекуррентная сеть

Один к одному

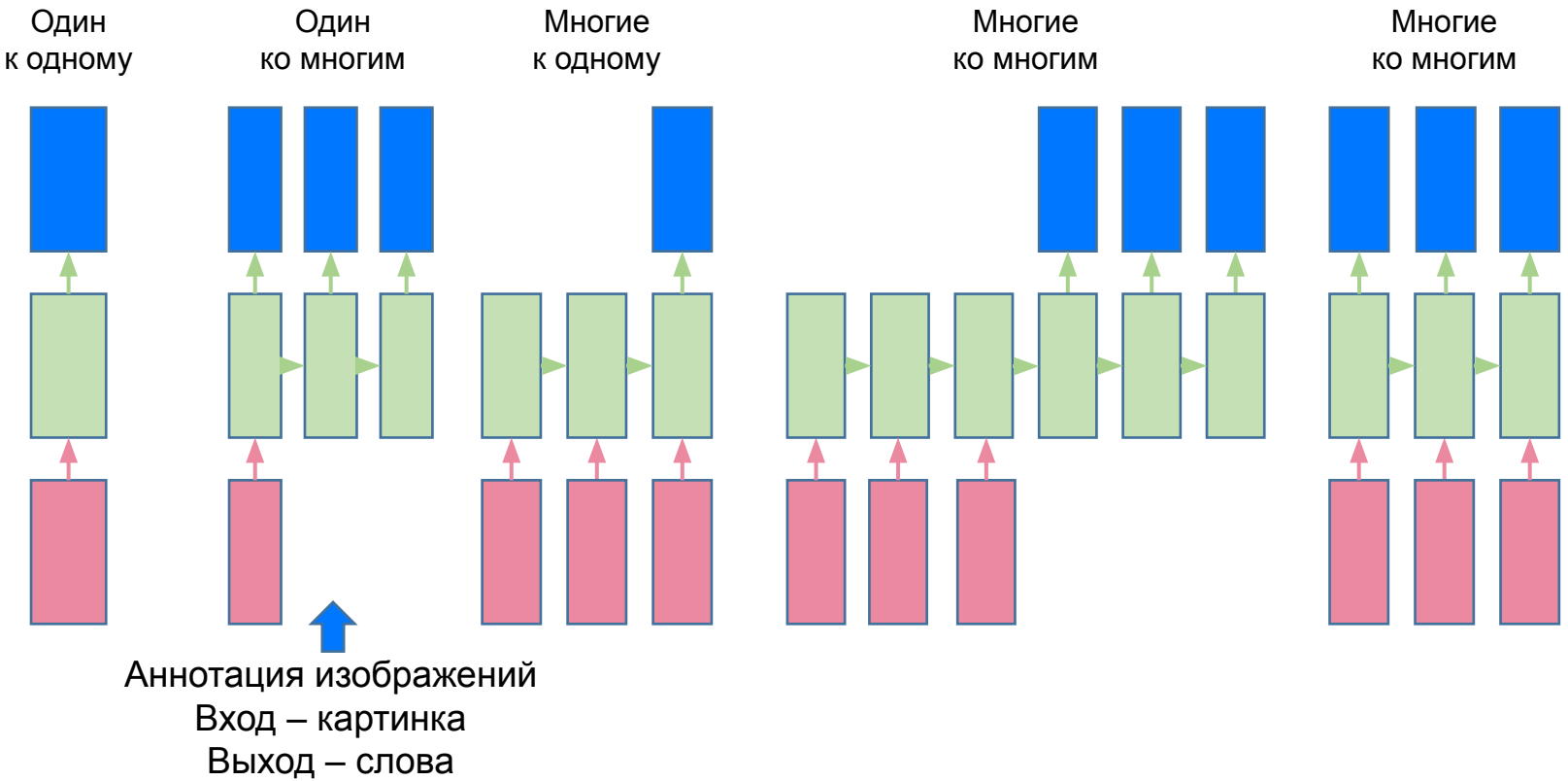
- MLP
- Сверточные сети



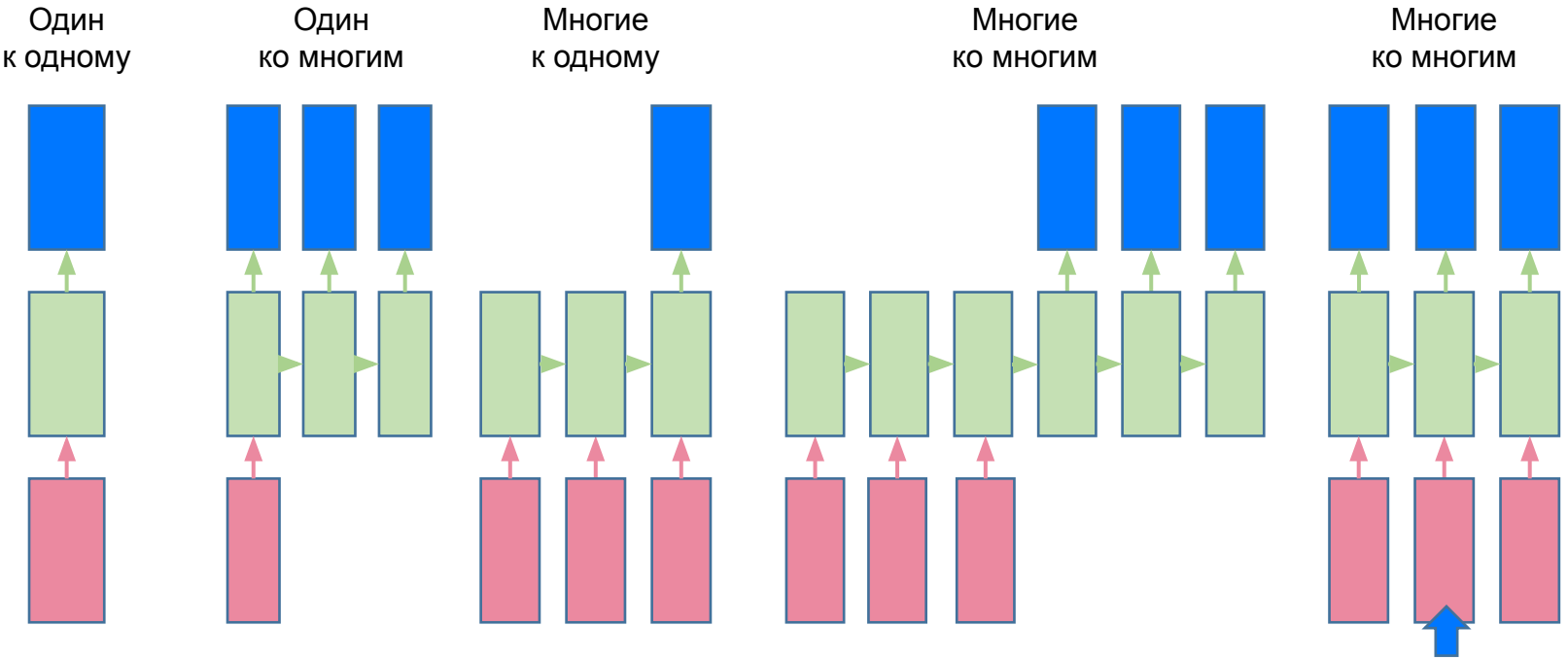
Для обработки последовательностей

Рекуррентная связь

Рекуррентная нейросеть



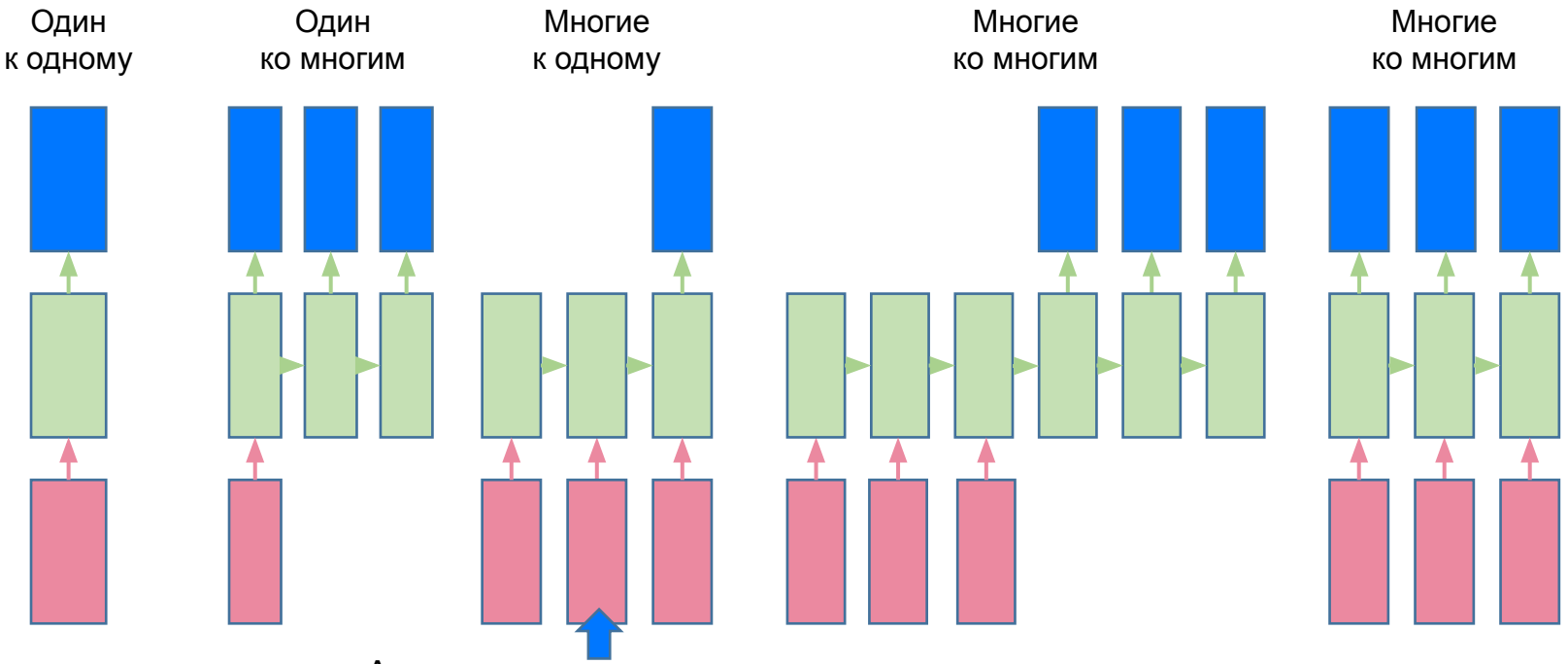
Рекуррентная нейросеть



Классификатор кадров видео

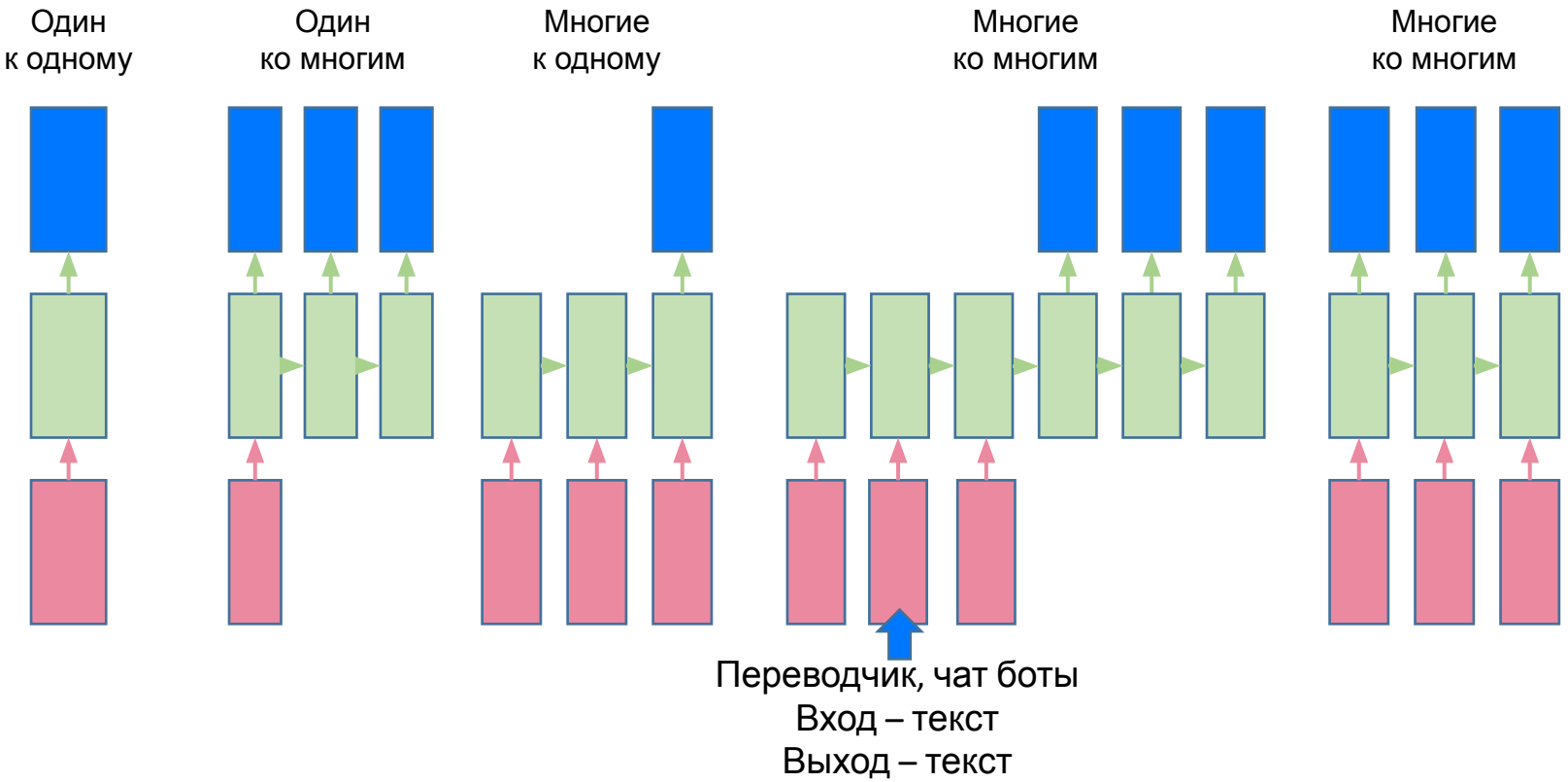
NAR, POS задачи

Рекуррентная нейросеть



Анализ тональности текста.
Вход – текст
Выход – класс тональности

Рекуррентная нейросеть



Рекуррентная формула

Сеть подает на вход в момент времени *t* выход *t-1*

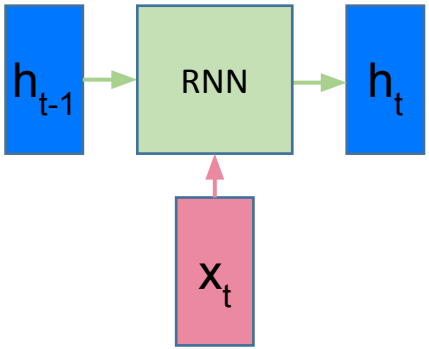
$$h_t = f_W(h_{t-1}, x_t)$$

Новое состояние

Функция с параметрами W

Старое состояние

Вход на шаге t



Рекуррентная формула

Сеть подает на вход в момент времени *t* выход *t-1*

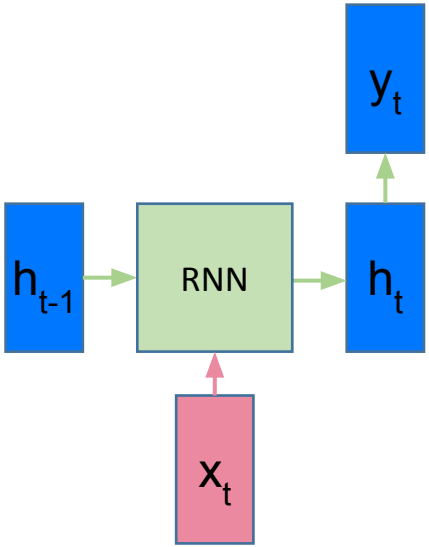
$$h_t = f_W (h_{t-1}, x_t)$$

$$y_t = g_{W'}(h_t)$$

Выход на шаге t

Функция с параметрами W’

Состояние на текущем шаге t

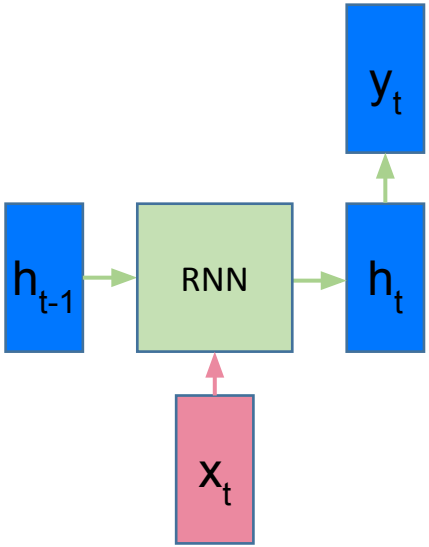


Рекуррентная формула

Сеть подает на вход в момент времени *t* выход *t-1*

$$h_t = f_W(h_{t-1}, x_t)$$

$$y_t = g_W(h_t)$$



Для каждого момента времени используются одни и те же функции и матрицы весов!

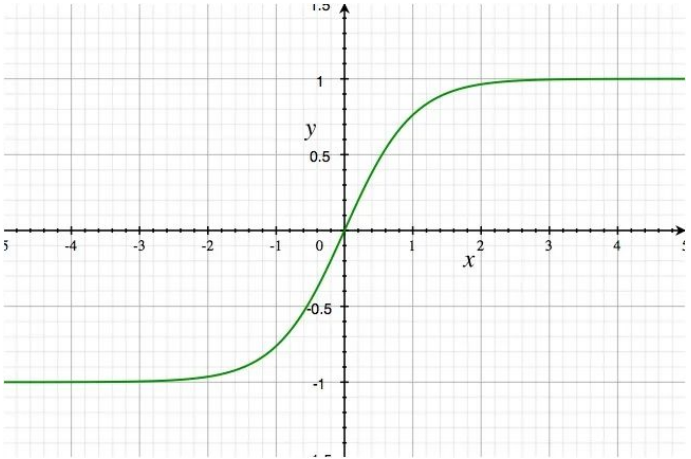
Базовая рекуррентная сеть (Vanilla RNN)

Функция состояния на момент t:

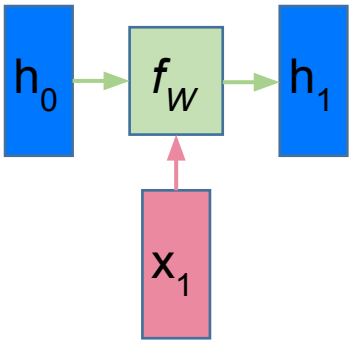
$$h_t = \tanh\left(W_{hh}h_{t-1} + W_{hx}x_t\right)$$

Выход на момент t:

$$y_t = W_{hy}h_t$$

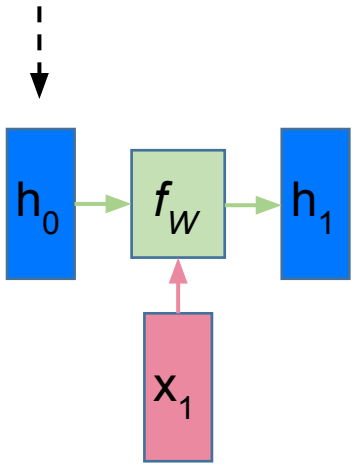


Вычислительный граф

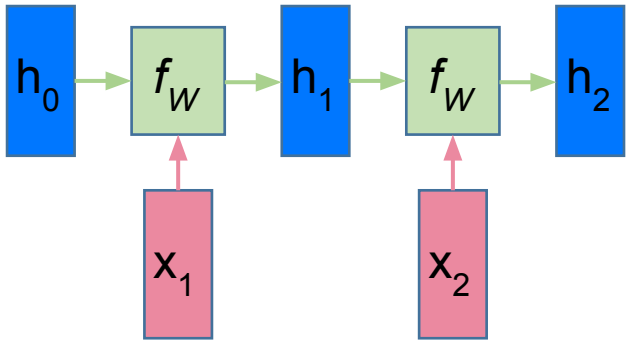


Вычислительный граф

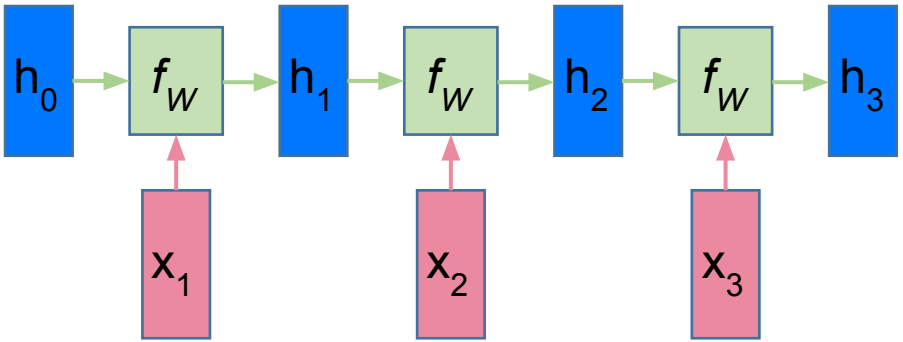
Чаще всего $h_0 = 0$.
Но можно сделать обучаемым или зависимым от контекстного параметра.



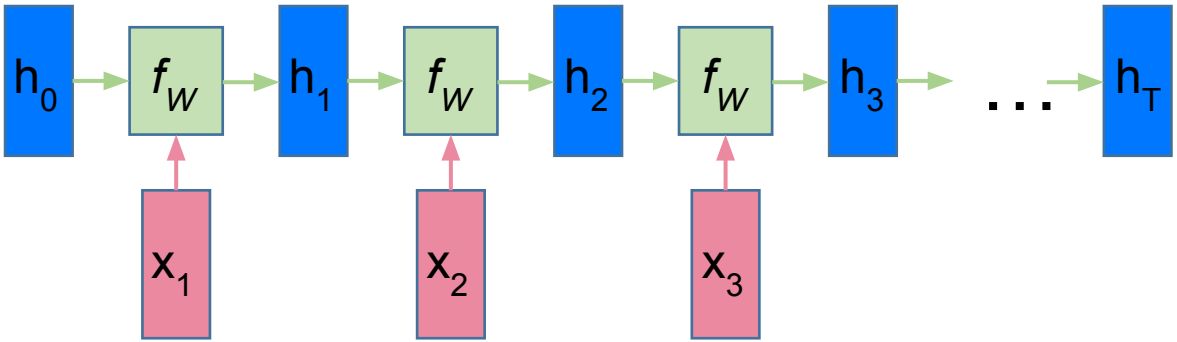
Вычислительный граф



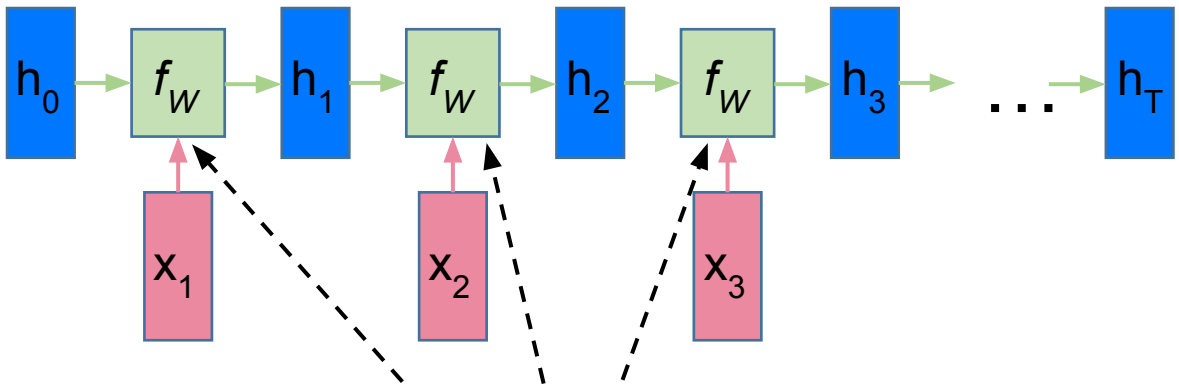
Вычислительный граф



Вычислительный граф

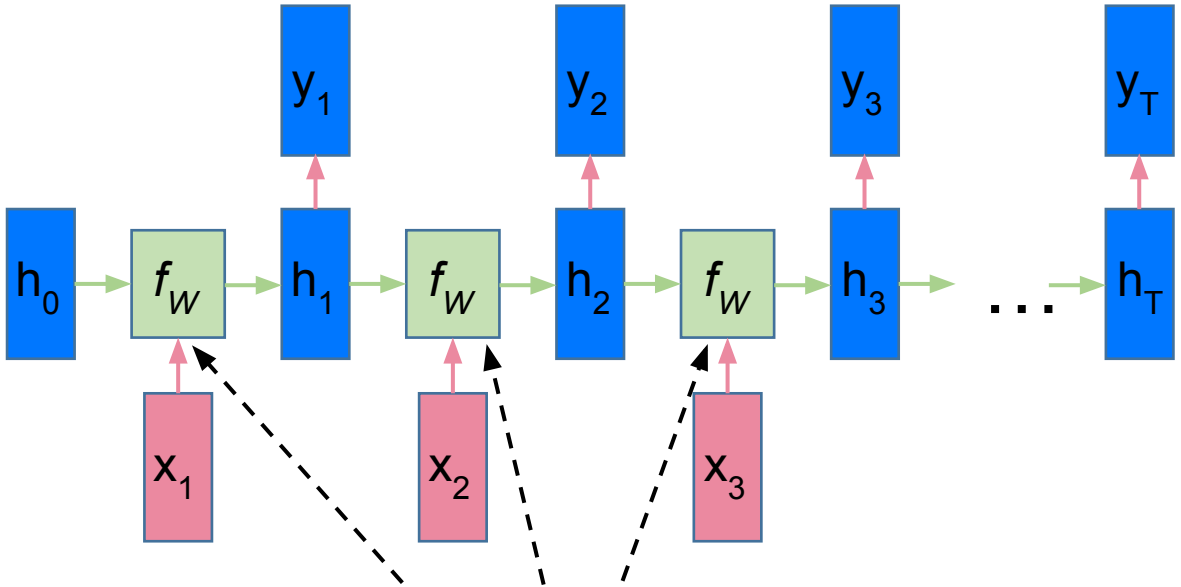


Вычислительный граф



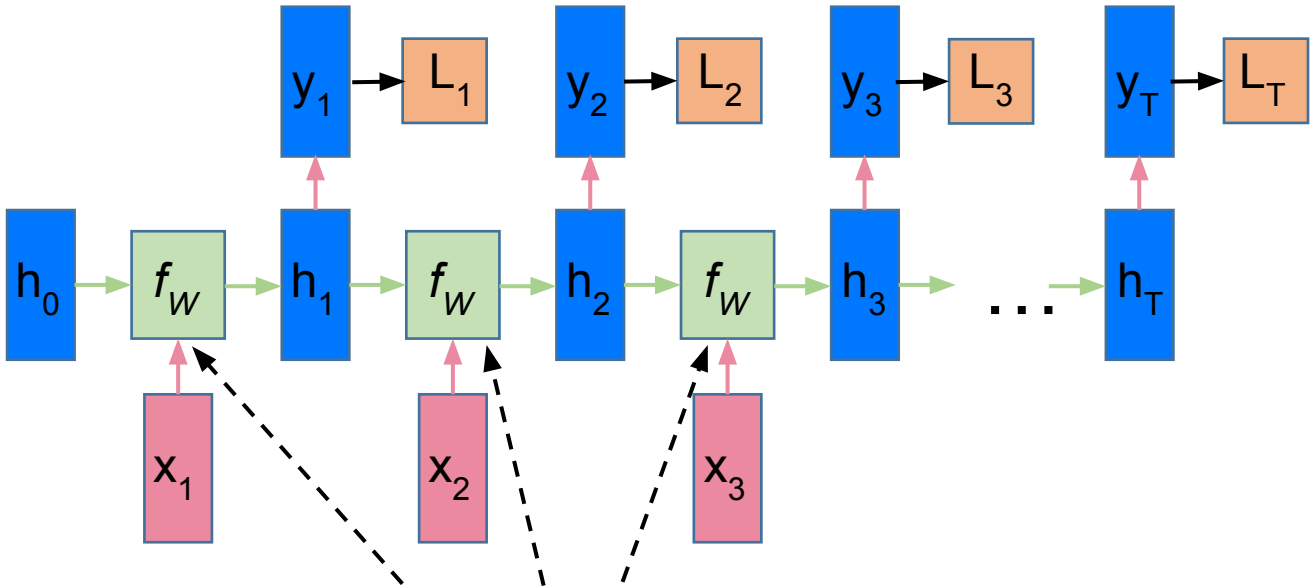
Одна матрица W на каждом шаге

Вычислительный граф. Многие ко многим



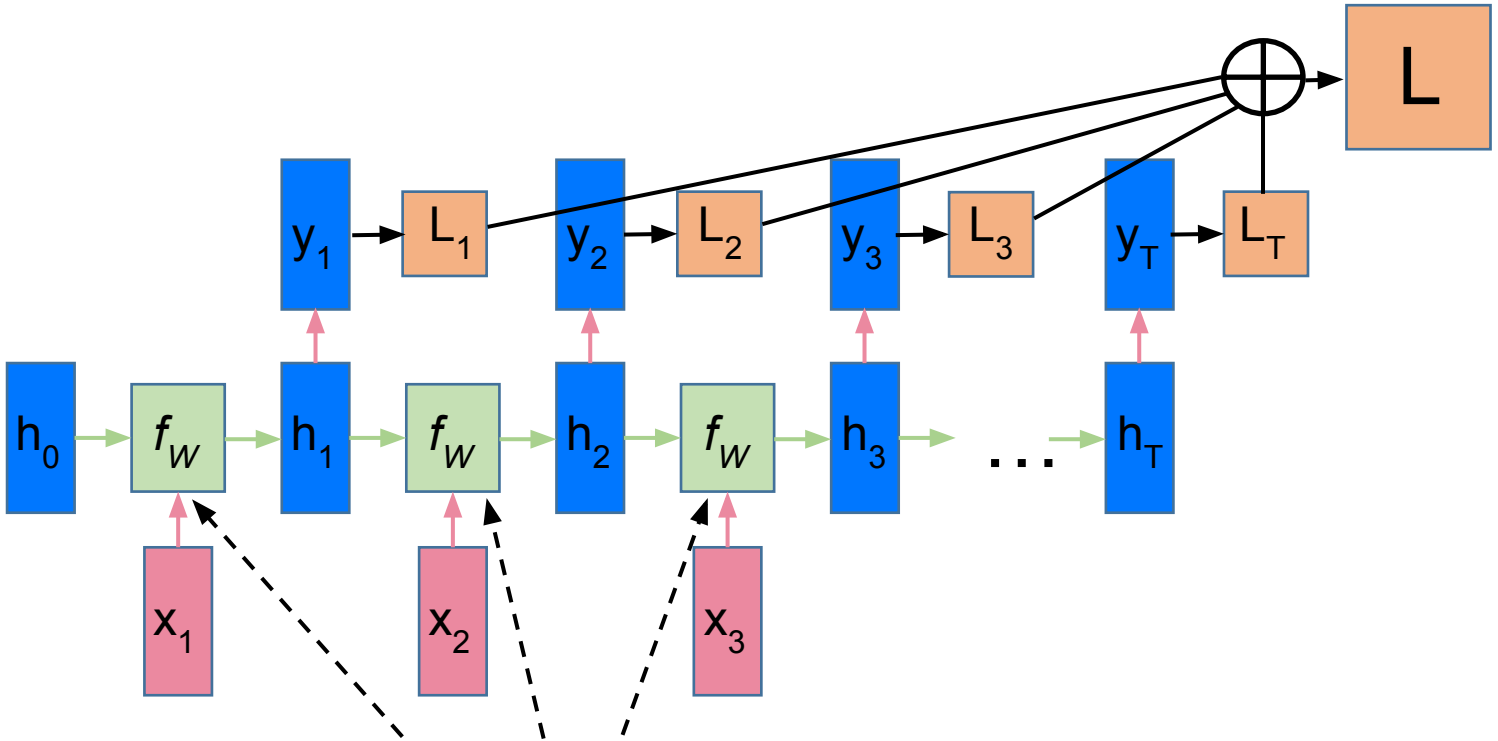
Одна матрица W на каждом шаге

Вычислительный граф. Многие ко многим



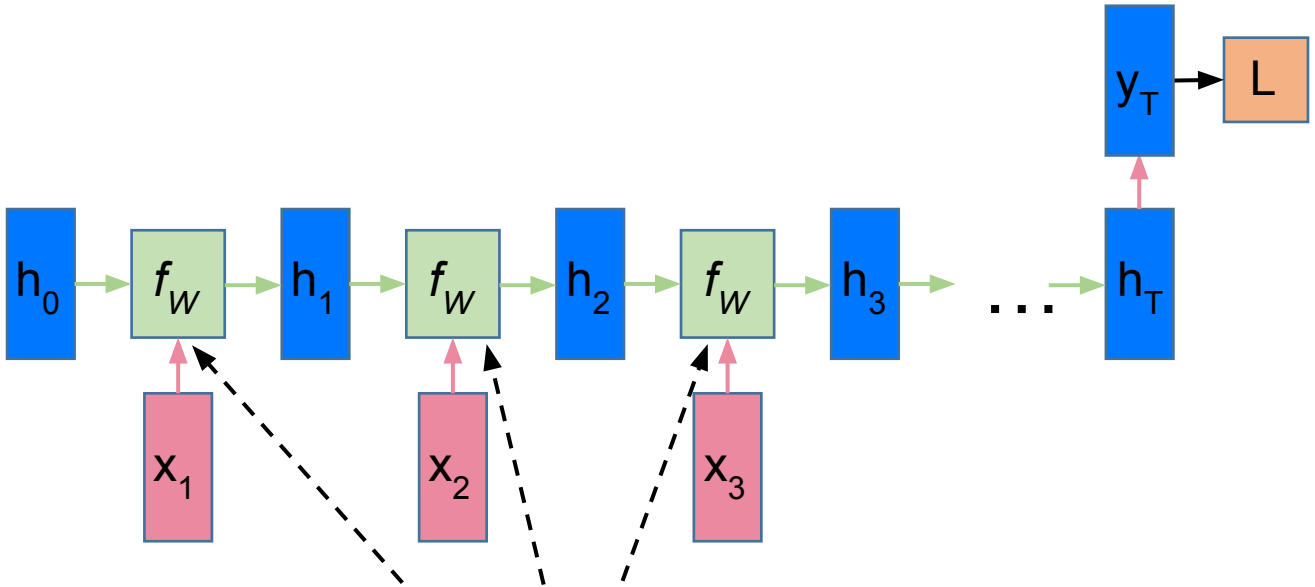
Одна матрица W на каждом шаге

Вычислительный граф. Многие ко многим

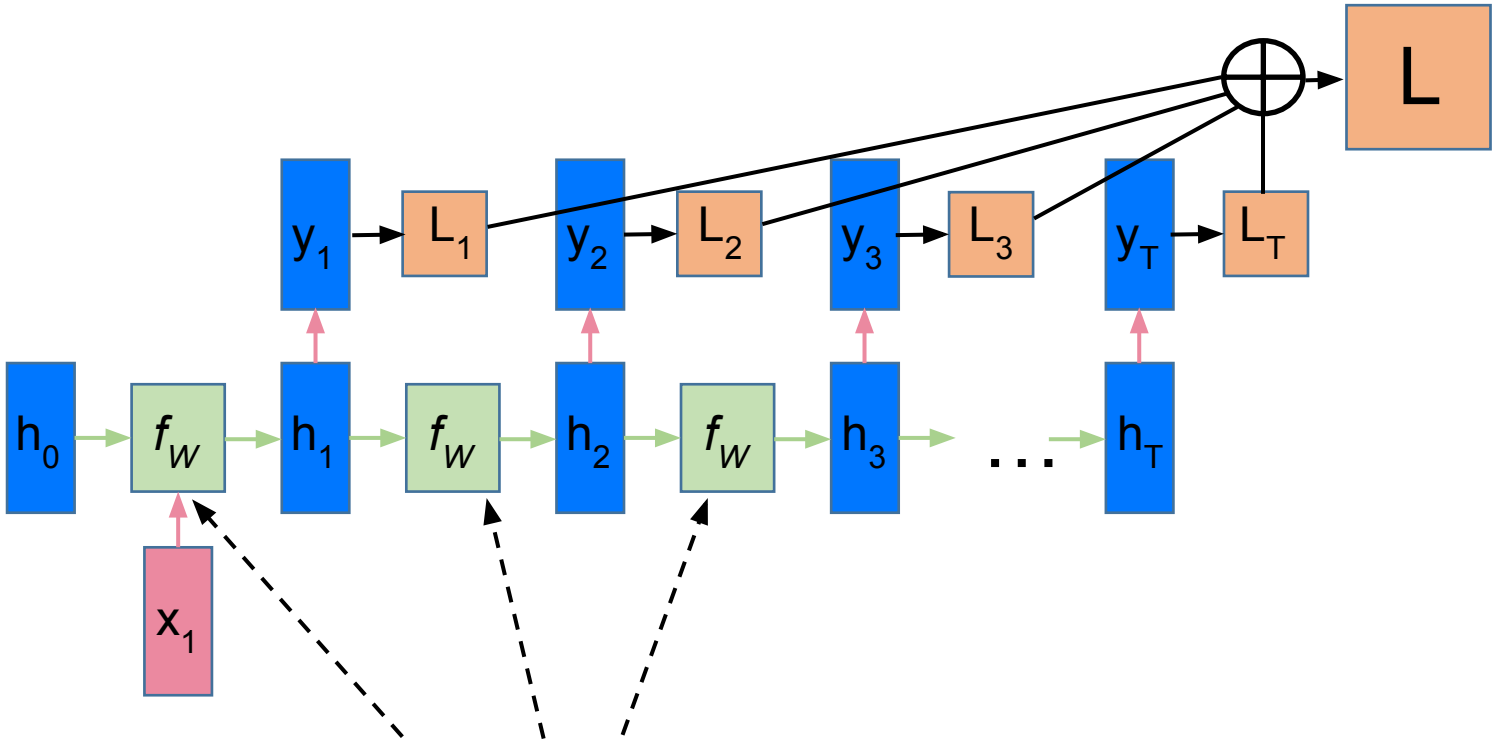


Одна матрица W на каждом шаге

Вычислительный граф. Многие к одному



Вычислительный граф. Один ко многим



RNN: особенности

Градиенты RNN

$$h_1 = \tanh(W_{hh} h_0 + W_{hx} x_1)$$
$$y_1 = \text{softmax}(W_{hy} h_1)$$

…

$$h_t = \tanh(W_{hh} h_{t-1} + W_{hx} x_t)$$
$$y_t = \text{softmax}(W_{hy} h_t)$$

…

$$h_T = \tanh(W_{hh} h_{T-1} + W_{hx} x_T)$$
$$y_T = \text{softmax}(W_{hy} h_T)$$

$$L = \sum_{t=1}^{T} \left(\ell(y_t, \hat{y}_t) \right)$$

Вычисление градиента по W_{hh}

Градиенты RNN

$$h_1 = \tanh(W_{hh} h_0 + W_{hx} x_1)$$
$$y_1 = \text{softmax}(W_{hy} h_1)$$

…

$$h_t = \tanh(W_{hh} h_{t-1} + W_{hx} x_t)$$
$$y_t = \text{softmax}(W_{hy} h_t)$$

…

$$h_T = \tanh(W_{hh} h_{T-1} + W_{hx} x_T)$$
$$y_T = \text{softmax}(W_{hy} h_T)$$

$$L = \sum_{t=1}^{T} \left(\ell(y_t, \hat{y}_t) \right)$$

Вычисление градиента по W_{hh}

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial \ell_t}{\partial W_{hh}}$$

Градиенты RNN

$$h_1 = \tanh(W_{hh} h_0 + W_{hx} x_1)$$
$$y_1 = \text{softmax}(W_{hy} h_1)$$

…

$$h_t = \tanh(W_{hh} h_{t-1} + W_{hx} x_t)$$
$$y_t = \text{softmax}(W_{hy} h_t)$$

…

$$h_T = \tanh(W_{hh} h_{T-1} + W_{hx} x_T)$$
$$y_T = \text{softmax}(W_{hy} h_T)$$

$$L = \sum_{t=1}^{T} \left(\ell(y_t, \hat{y}_t) \right)$$

Вычисление градиента по W_{hh}

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial \ell_t}{\partial W_{hh}}$$

$$\frac{\partial \ell_t}{\partial W_{hh}} = \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial W_{hh}} +$$

Градиенты RNN

$$h_1 = \tanh(W_{hh} h_0 + W_{hx} x_1)$$
$$y_1 = \text{softmax}(W_{hy} h_1)$$

…

$$h_t = \tanh(W_{hh} h_{t-1} + W_{hx} x_t)$$
$$y_t = \text{softmax}(W_{hy} h_t)$$

…

$$h_T = \tanh(W_{hh} h_{T-1} + W_{hx} x_T)$$
$$y_T = \text{softmax}(W_{hy} h_T)$$

$$L = \sum_{t=1}^{T} \left(\ell(y_t, \hat{y}_t) \right)$$

Вычисление градиента по W_{hh}

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial \ell_t}{\partial W_{hh}}$$

$$\frac{\partial \ell_t}{\partial W_{hh}} = \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial W_{hh}} + \boxed{\frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial W_{hh}}} +$$

Градиенты RNN

$$h_1 = \tanh(W_{hh} h_0 + W_{hx} x_1)$$
$$y_1 = \text{softmax}(W_{hy} h_1)$$

…

$$h_t = \tanh(W_{hh} h_{t-1} + W_{hx} x_t)$$
$$y_t = \text{softmax}(W_{hy} h_t)$$

…

$$h_T = \tanh(W_{hh} h_{T-1} + W_{hx} x_T)$$
$$y_T = \text{softmax}(W_{hy} h_T)$$

$$L = \sum_{t=1}^{T} \left(\ell(y_t, \hat{y}_t) \right)$$

Вычисление градиента по W_{hh}

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial \ell_t}{\partial W_{hh}}$$

$$\frac{\partial \ell_t}{\partial W_{hh}} = \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial W_{hh}} + \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial W_{hh}} +$$

$$\dots + \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \dots \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W_{hh}}$$

Градиенты RNN

$$h_1 = \tanh(W_{hh} h_0 + W_{hx} x_1)$$
$$y_1 = \text{softmax}(W_{hy} h_1)$$

…

$$h_t = \tanh(W_{hh} h_{t-1} + W_{hx} x_t)$$
$$y_t = \text{softmax}(W_{hy} h_t)$$

…

$$h_T = \tanh(W_{hh} h_{T-1} + W_{hx} x_T)$$
$$y_T = \text{softmax}(W_{hy} h_T)$$

$$L = \sum_{t=1}^{T} \left(\ell(y_t, \hat{y}_t) \right)$$

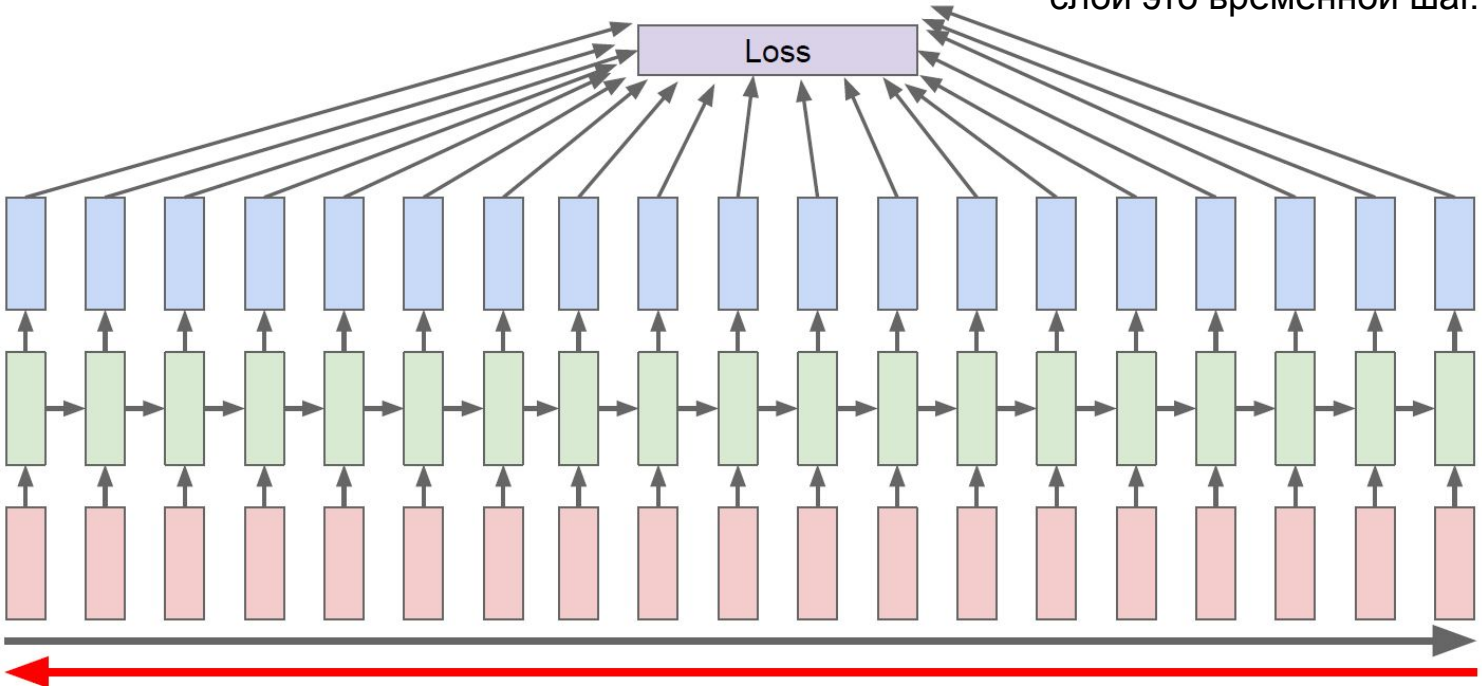
Вычисление градиента по W_{hh}

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial \ell_t}{\partial W_{hh}}$$

$$\frac{\partial \ell_t}{\partial W_{hh}} = \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial W_{hh}} + \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial W_{hh}} +$$
$$\dots + \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \dots \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W_{hh}}$$

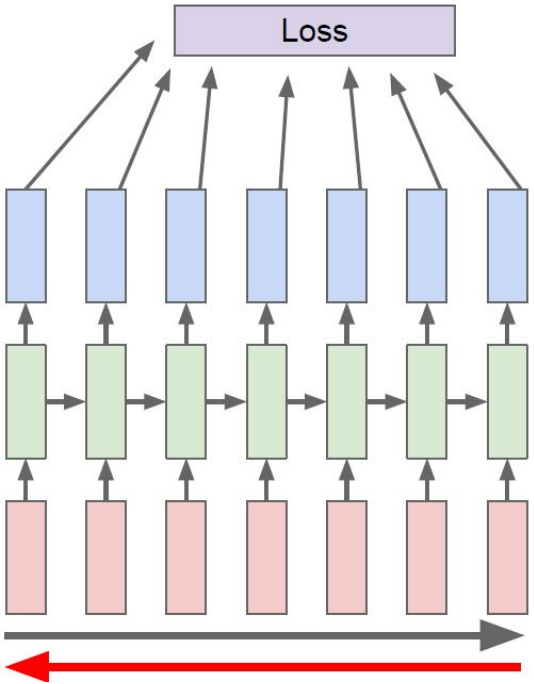
Backpropagation Through Time

Граф можно представить как **глубокую сеть с разделяемыми матрицами весов**, где каждый слой это временной шаг.



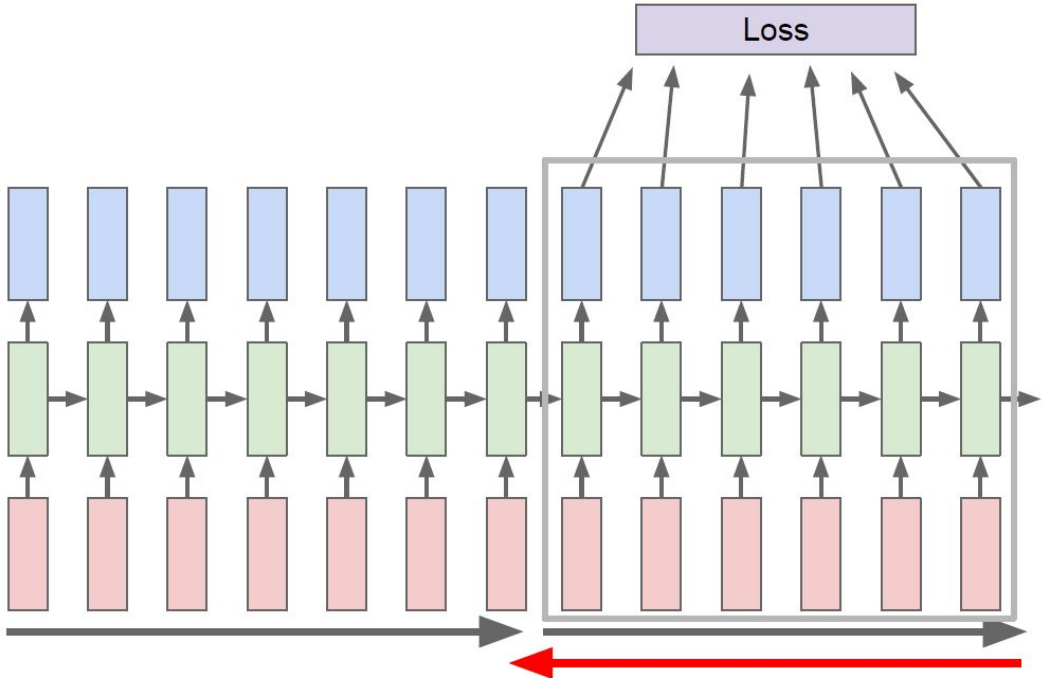
Прямой и обратный проход по всей последовательности

Truncated Backpropagation Through Time



Прямой и обратный проход только по кусочку всей последовательности

Truncated Backpropagation Through Time



Сохраняем состояния предыдущего кусочка и делаем прямой и обратный проход по следующему

Градиенты RNN

$$h_t = \tanh(W_{hh} h_{t-1} + W_{hx} x_t)$$
$$y_t = \text{softmax}(W_{hy} h_t)$$

$$L = \sum_{t=1}^{T} \left(\ell(y_t, \hat{y}_t) \right)$$

Градиента по W_{hh}:

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial \ell_t}{\partial W_{hh}}$$

$$\frac{\partial \ell_t}{\partial W_{hh}} = \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial W_{hh}} + \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial W_{hh}} + \cdots + \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \cdots \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W_{hh}}$$

Градиенты RNN

$$h_t = \tanh(W_{hh} h_{t-1} + W_{hx} x_t)$$
$$y_t = \text{softmax}(W_{hy} h_t)$$

$$L = \sum_{t=1}^{T} \left(\ell(y_t, \hat{y}_t) \right)$$

Содержит большое кол-во множителей вида:

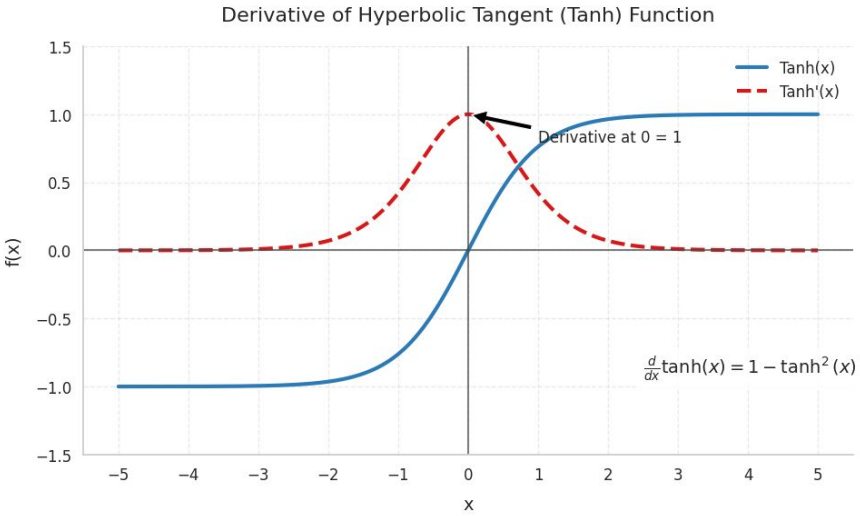
$$\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial \tanh(x)}{\partial x} \cdot W_{hh}$$

Градиента по W_{hh}:

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial \ell_t}{\partial W_{hh}}$$

$$\frac{\partial \ell_t}{\partial W_{hh}} = \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial W_{hh}} + \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial W_{hh}} + \cdots + \frac{\partial \ell_t}{\partial h_t} \cdot \boxed{\frac{\partial h_t}{\partial h_{t-1}} \cdot \cdots \cdot \frac{\partial h_2}{\partial h_1}} \cdot \frac{\partial h_1}{\partial W_{hh}}$$

Градиенты RNN



В диапазоне от 0 до 1

$$\frac{\partial h_t}{\partial h_{t-1}} = \boxed{\frac{\partial \tanh(x)}{\partial x}} \cdot W_{hh}$$

$$\frac{\partial \mathcal{L}_t}{\partial W_{hh}} = \frac{\partial \mathcal{L}_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial W_{hh}} + \frac{\partial \mathcal{L}_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial W_{hh}} + \cdots + \frac{\partial \mathcal{L}_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \cdots \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W_{hh}}$$

Градиенты RNN

$$h_t = \tanh(W_{hh} h_{t-1} + W_{hx} x_t)$$
$$y_t = \text{softmax}(W_{hy} h_t)$$

Отвечает за сохранение памяти о прошлом состоянии.
→ Может быть большой

$$L = \sum_{t=1}^{T} \left(\ell(y_t, \hat{y}_t)\right)$$

$$\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial \tanh(x)}{\partial x} \cdot \boxed{W_{hh}}$$

Градиента по W_{hh}:

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial \ell_t}{\partial W_{hh}}$$

$$\frac{\partial \ell_t}{\partial W_{hh}} = \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial W_{hh}} + \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial W_{hh}} + \cdots + \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \cdots \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W_{hh}}$$

Exploding Gradient Problem

$$h_t = \tanh(W_{hh} h_{t-1} + W_{hx} x_t)$$
$$y_t = \text{softmax}(W_{hy} h_t)$$

$$L = \sum_{t=1}^{T} \left(\ell(y_t, \hat{y}_t)\right)$$

Градиента по W_{hh}:

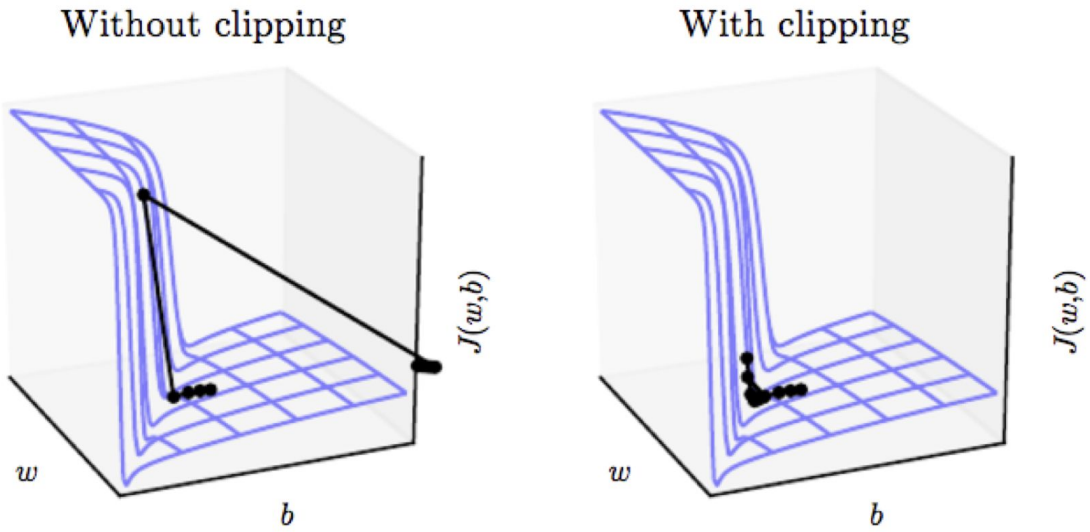
$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial \ell_t}{\partial W_{hh}}$$

$$\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial \tanh(x)}{\partial x} \cdot W_{hh}$$

Если этот множитель **> 1**, то слагаемые с большим кол-вом множителей $\rightarrow \infty$

$$\frac{\partial \ell_t}{\partial W_{hh}} = \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial W_{hh}} + \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial W_{hh}} + \cdots + \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \cdots \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W_{hh}}$$

Exploding Gradient Problem



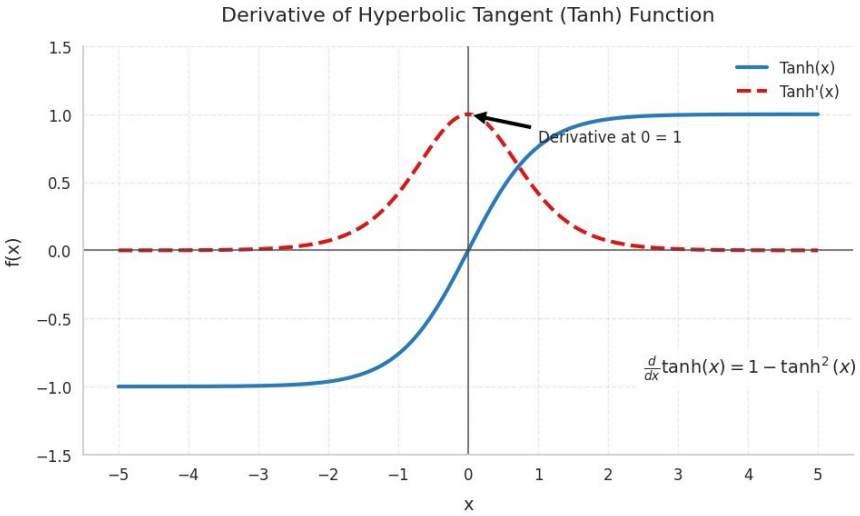
Следствия:

- Нестабильное обучение
- Числовое переполнение

Решение: отсечение градиента

```
torch.nn.utils.clip_grad_norm_(parameters, max_norm)
```

Vanishing Gradient Problem



$$\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial \tanh(x)}{\partial x} \cdot W_{hh}$$

Если этот множитель **< 1**, то слагаемые с большим кол-вом множителей $\rightarrow$**0**

$$\frac{\partial \mathcal{L}_t}{\partial W_{hh}} = \frac{\partial \mathcal{L}_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial W_{hh}} + \frac{\partial \mathcal{L}_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial W_{hh}} + \cdots + \frac{\partial \mathcal{L}_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \cdots \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W_{hh}}$$

Vanishing Gradient Problem

$$h_t = \tanh(W_{hh} h_{t-1} + W_{hx} x_t)$$
$$y_t = \text{softmax}(W_{hy} h_t)$$

$$L = \sum_{t=1}^{T} \left(\ell(y_t, \hat{y}_t)\right)$$

Градиента по W_{hh}:

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial \ell_t}{\partial W_{hh}}$$

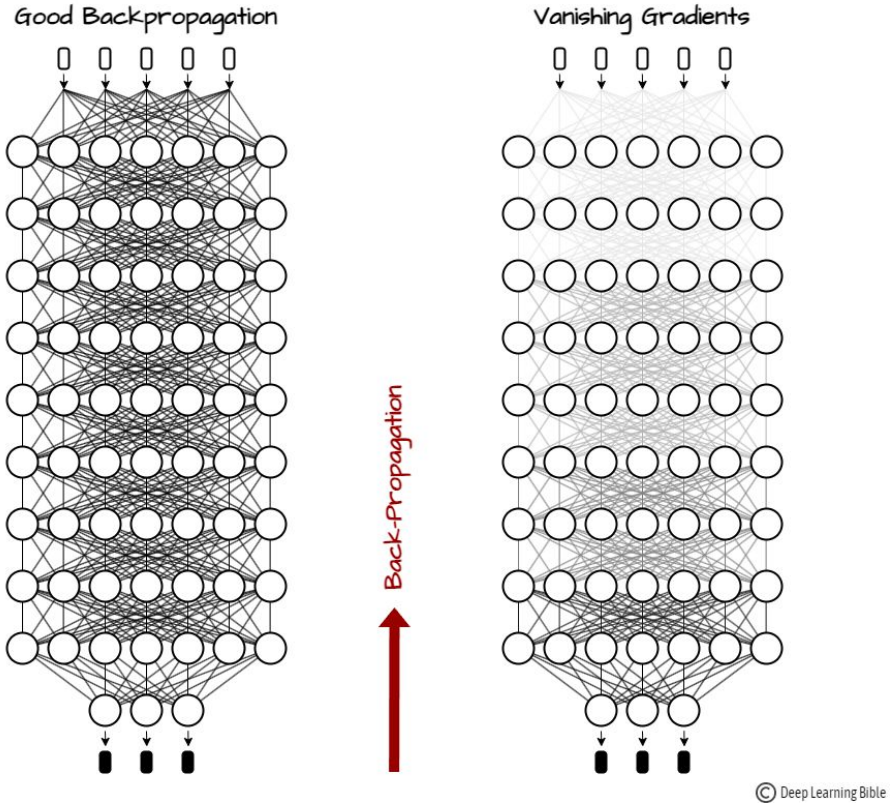
Матрица W_{hh} отвечает за сохранение “памяти” → Матрица должна быть “большой” → Производная tanh → 0

$$\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial \tanh(x)}{\partial x} \cdot W_{hh}$$

Если этот множитель **< 1**, то слагаемые с большим кол-вом множителей →**0**

$$\frac{\partial \ell_t}{\partial W_{hh}} = \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial W_{hh}} + \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial W_{hh}} + \cdots + \frac{\partial \ell_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_{t-1}} \cdot \cdots \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W_{hh}}$$

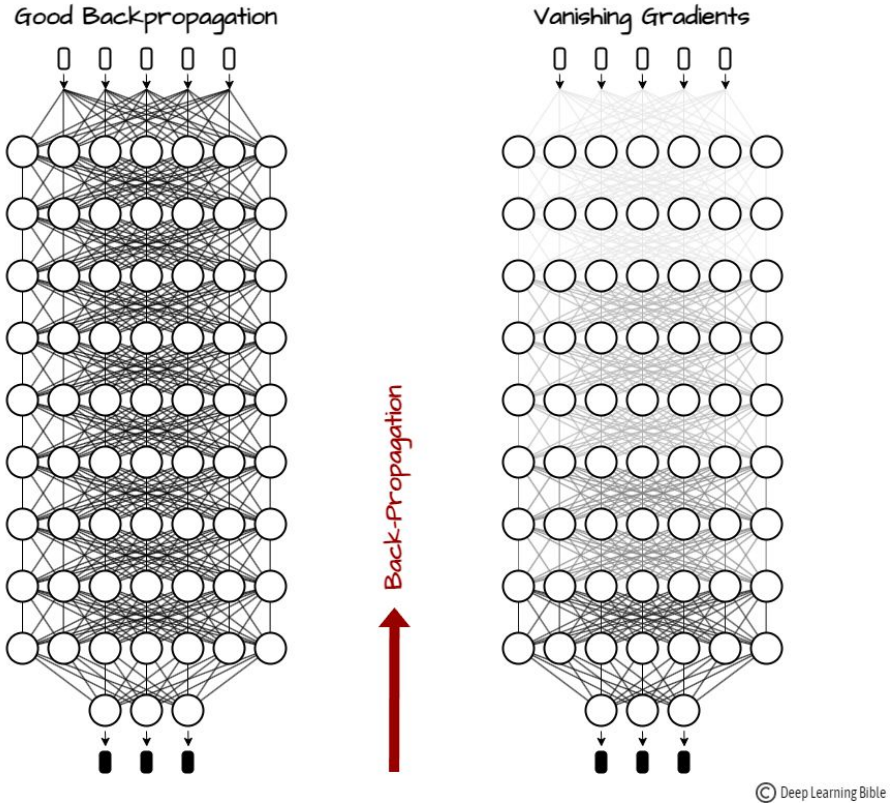
Vanishing Gradient Problem



Сеть **не может** научиться связывать информацию (например, подлежащее в предложении), которая появилась в самом начале последовательности, с необходимостью использовать эту информацию позже (например, глагол, с которым нужно согласовать подлежащее).

Причина: Информации о том, как должен измениться W_{hh}, чтобы исправить ошибку, связанную с ранним шагом, просто **не достигает** обновления весов, потому что градиент, несущий эту информацию, **исчез**.

Vanishing Gradient Problem



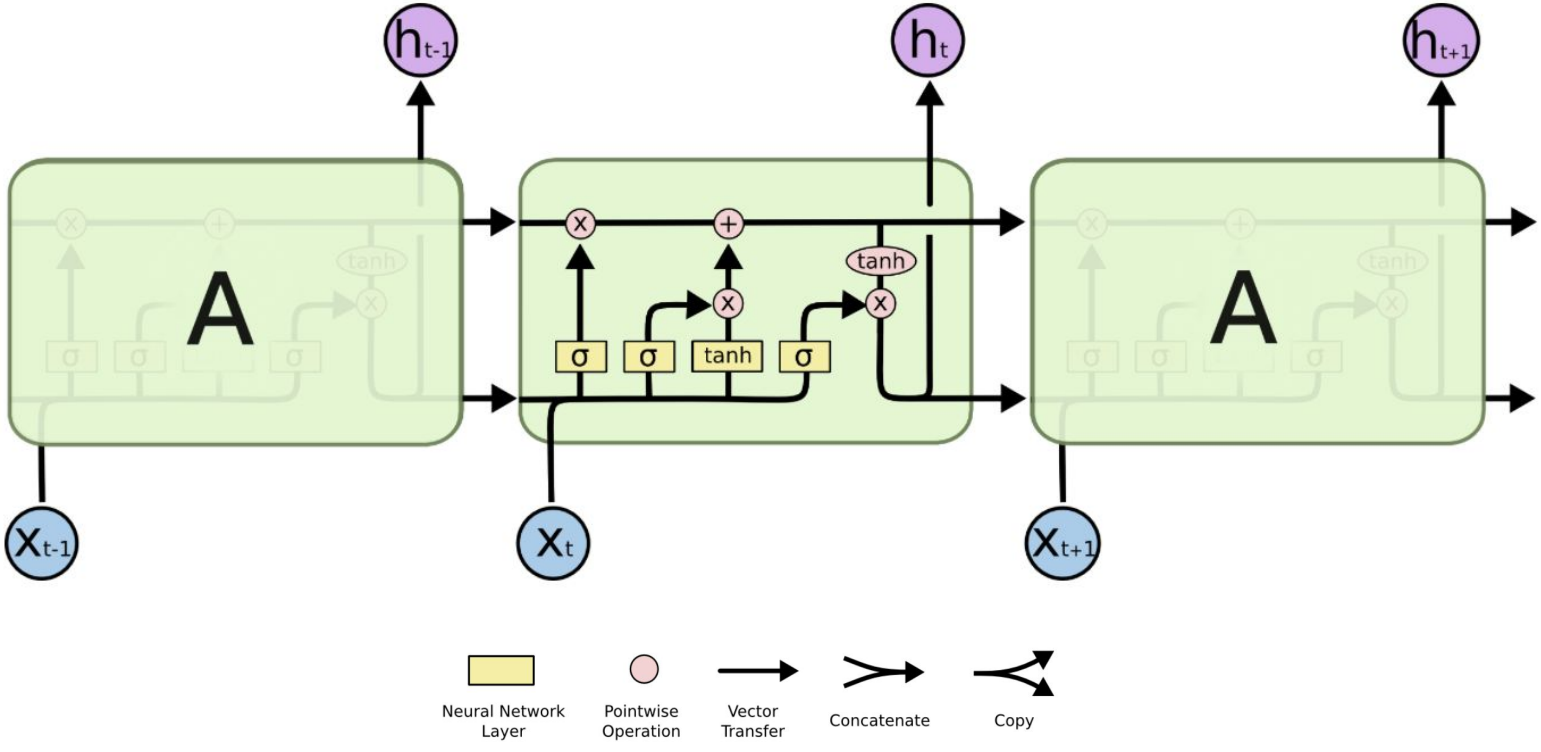
У Vanilla RNN нет возможности извлекать глобальный контекст.

Работает эффективно только с небольшими последовательностями.

→ **Нужно менять архитектуру**

LSTM

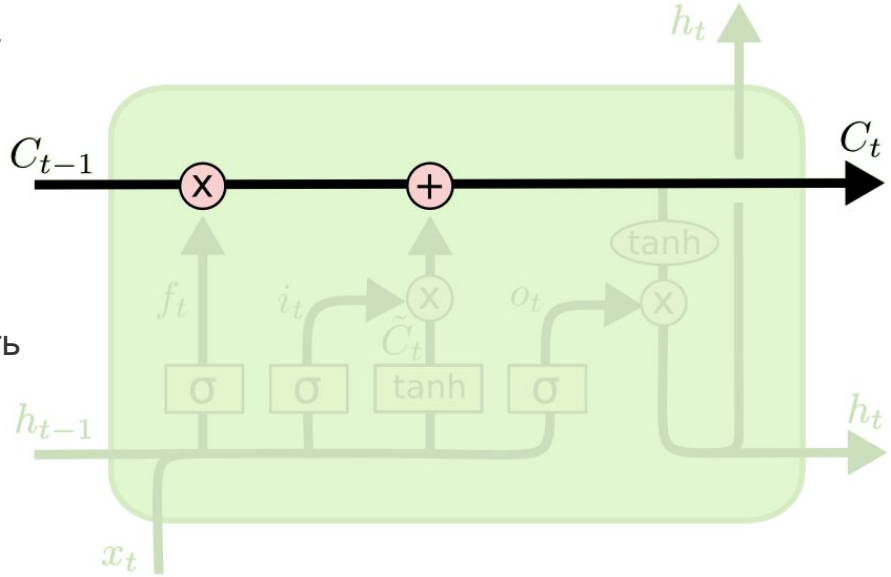
Long Short Term Memory

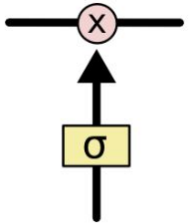


Long Short Term Memory

Cell state – проходит через всю цепочку с минимальными изменениями

LSTM может добавлять или убирать информацию из cell state с мощью **gates**



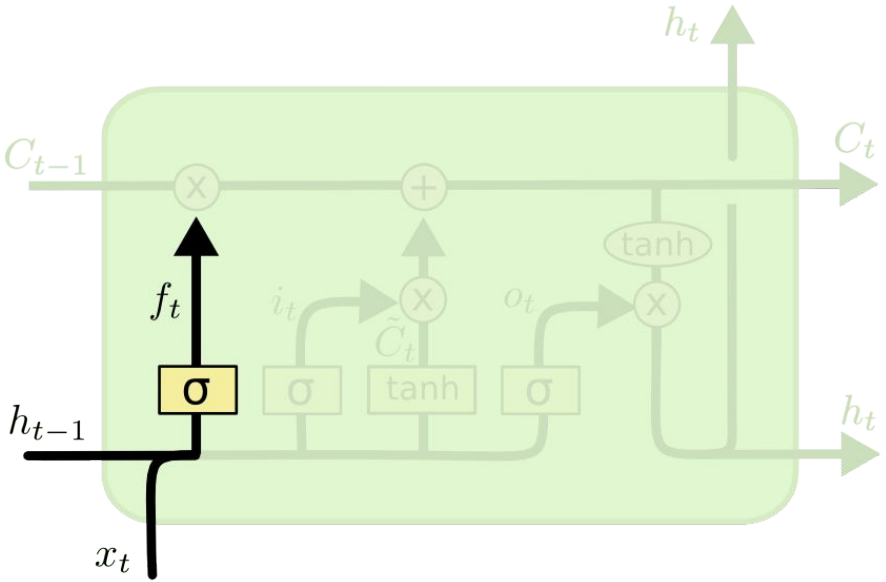


Gate регулирует сколько информации из текущего состояния нужно забыть или пропустить дальше

LSTM: Forget Gate Layer

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

Пример: в задаче предсказания следующего слова состояние ячейки может включать род текущего подлежащего.



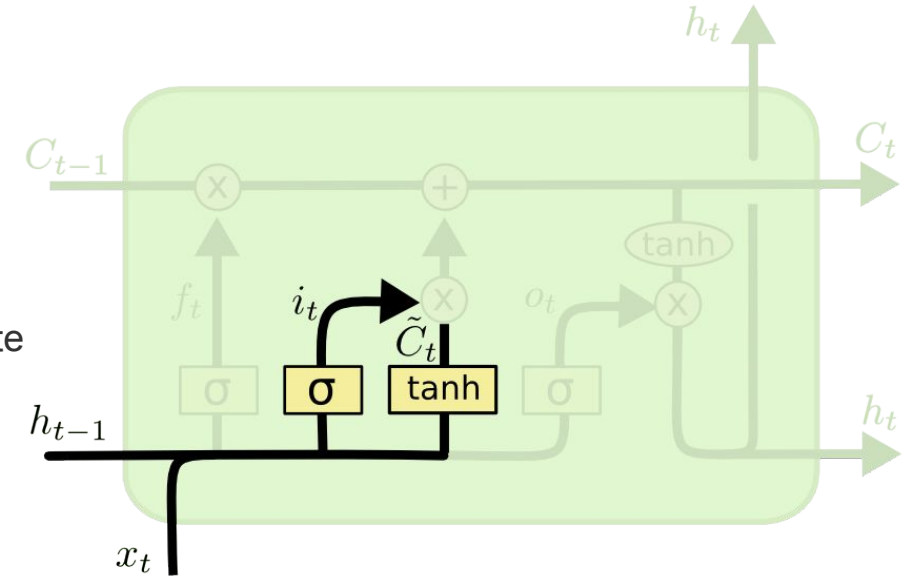
Когда мы встречаем новое подлежащее, мы хотим забыть род предыдущего.

LSTM: Input Gate Layer

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

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

Какую новую информацию мы хотим сохранить?



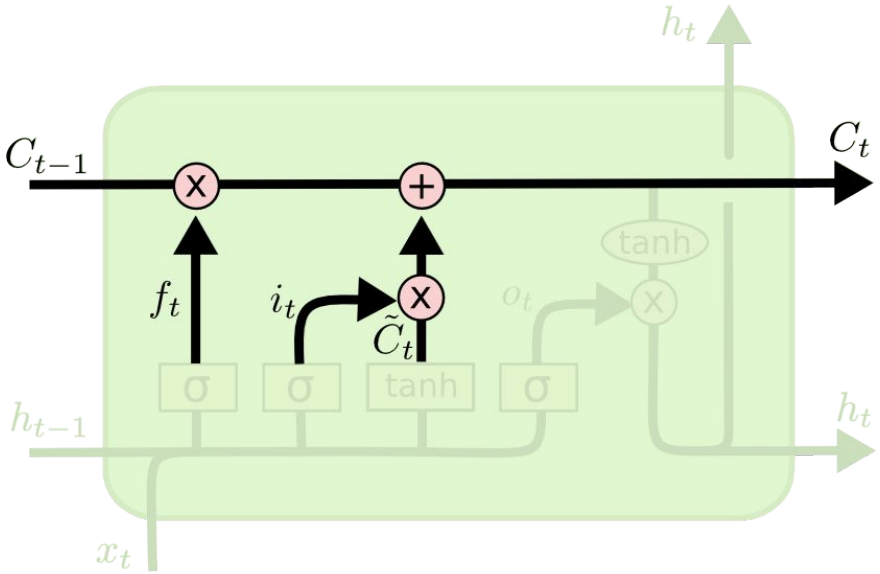
1. Сигмоида: определяет какие компоненты cell state вектора будут обновляться

2. Tanh: создает вектор значений-кандидатов, которые могут быть добавлены к cell state

LSTM: Update Cell State

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

На прошлых шагах уже получили все вектора. Осталось обновить cell state

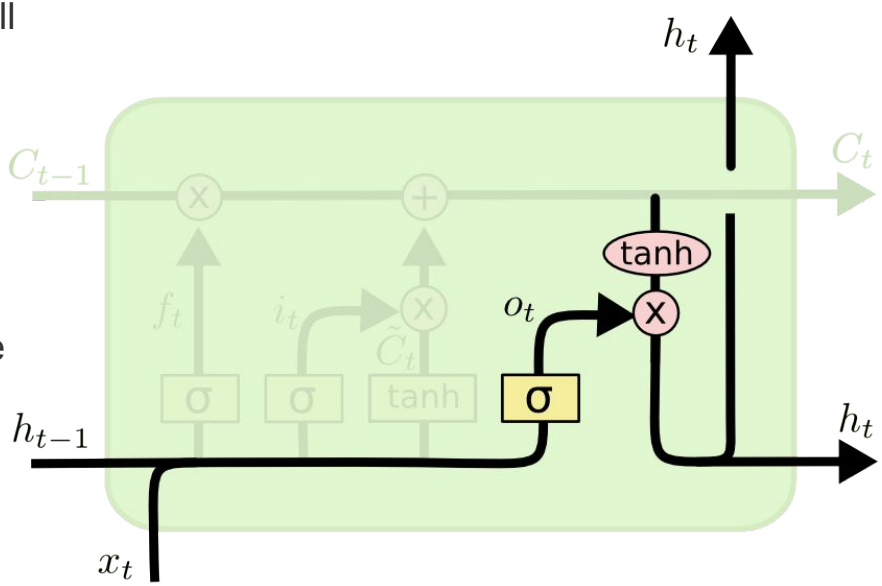


LSTM: Output

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

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

Отдаем на выход cell state, но модифицированный

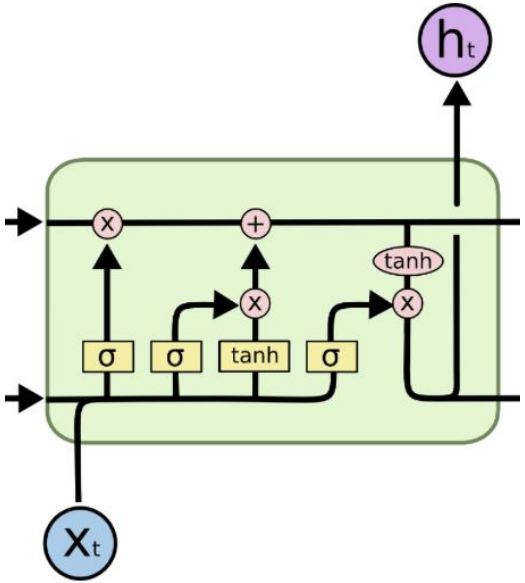


1. Сигмоида: определяет какие компоненты cell state вектора стоит отдавать на выход

2. Tanh: прогоняем cell state вектор через tanh чтобы получить значения в диапазоне (-1, 1)

3. Перемножаем с сигмоидой → отдаем только то, что решили отдавать

Long Short Term Memory



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

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

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

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

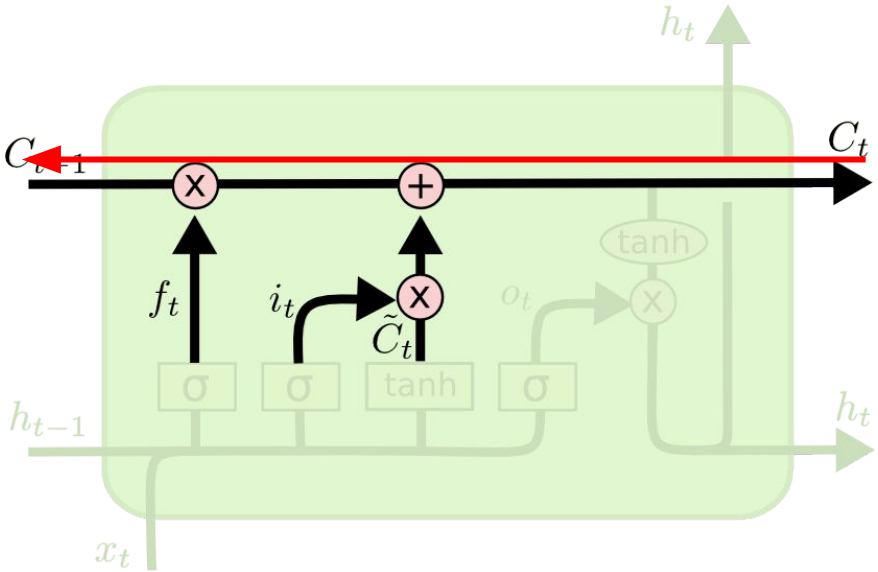
$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$

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

LSTM: Градиенты

$$\frac{\partial \ell_t}{\partial C_{t-1}} = \frac{\partial \ell_t}{\partial C_t} * \frac{\partial C_t}{\partial C_{t-1}} =$$

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

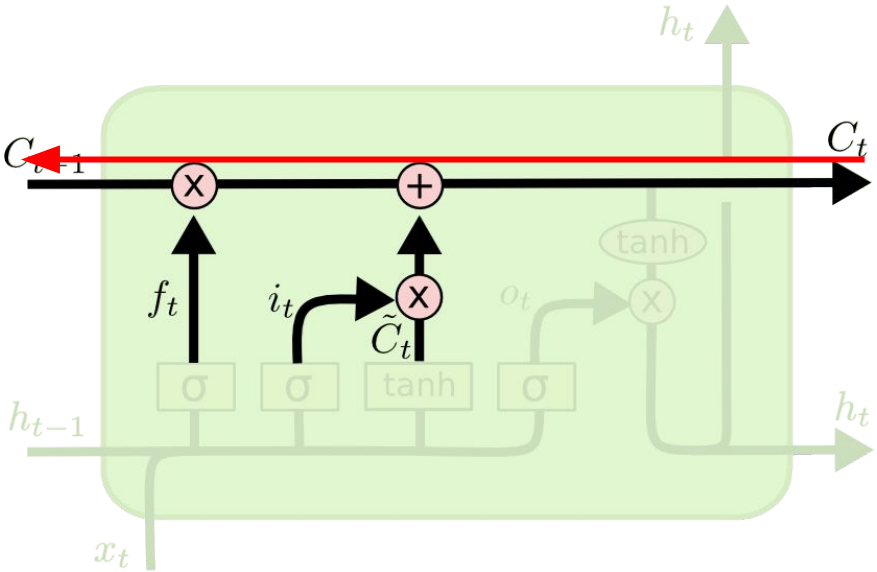


LSTM: Градиенты

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

$$\frac{\partial \ell_t}{\partial C_{t-1}} = \frac{\partial \ell_t}{\partial C_t} * \frac{\partial C_t}{\partial C_{t-1}} =$$

$$= \frac{\partial \ell_t}{\partial C_t} * \frac{\partial}{\partial C_{t-1}} \left(f_t * C_{t-1} + i_t * \tilde{C}_t \right) =$$



LSTM: Градиенты

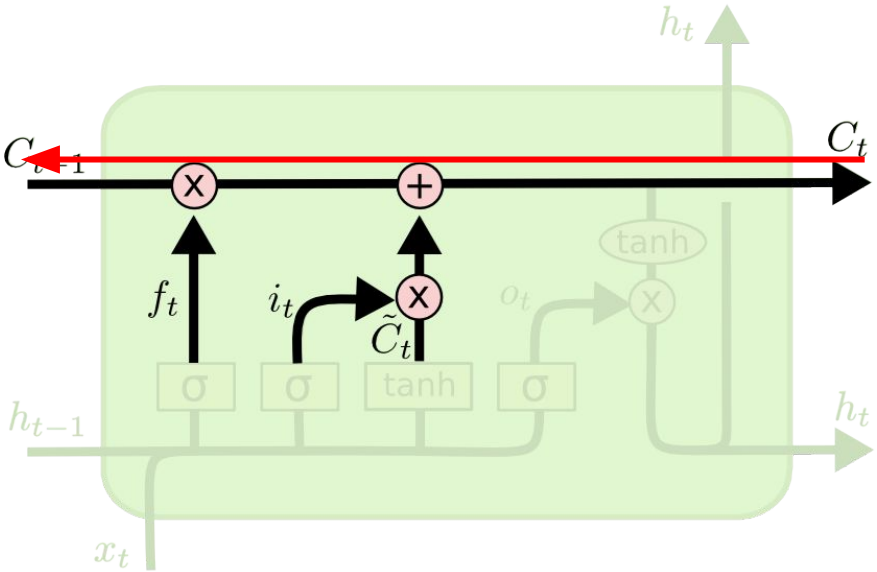
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$\frac{\partial \ell_t}{\partial C_{t-1}} = \frac{\partial \ell_t}{\partial C_t} * \frac{\partial C_t}{\partial C_{t-1}} =$$

$$= \frac{\partial \ell_t}{\partial C_t} * \frac{\partial}{\partial C_{t-1}} \left(f_t * C_{t-1} + i_t * \tilde{C}_t \right) =$$

$$= f_t$$

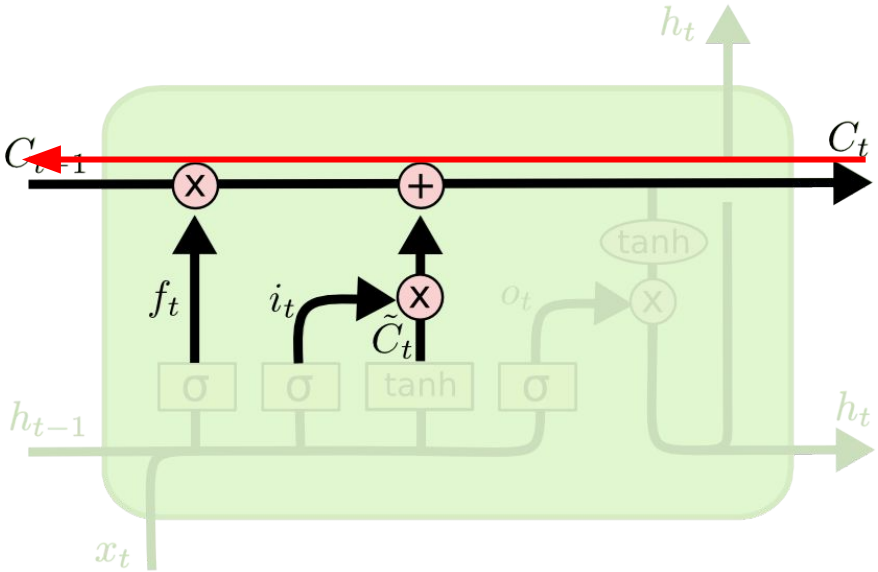
это сигмоида, опять в диапазоне от 0 до 1



LSTM: Градиенты

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

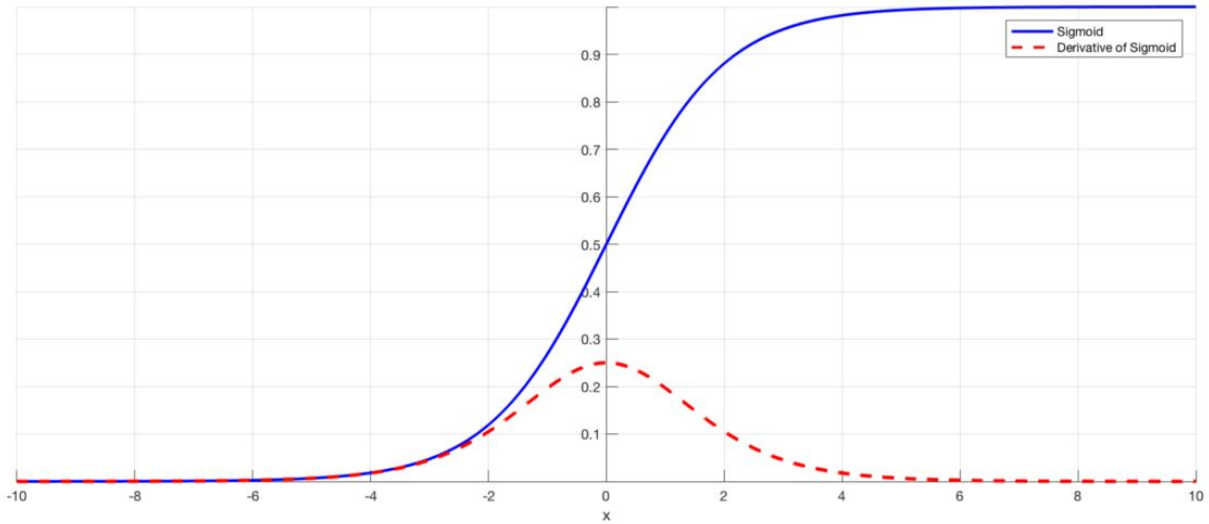
Если бы $f_t = 1 \rightarrow$ идеальное прохождение градиента, но тогда сеть ничего не забывает.



Идея: инициализировать bias так, чтобы на начальном этапе: $f_t \approx 1$, но не $f_t = 1$

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

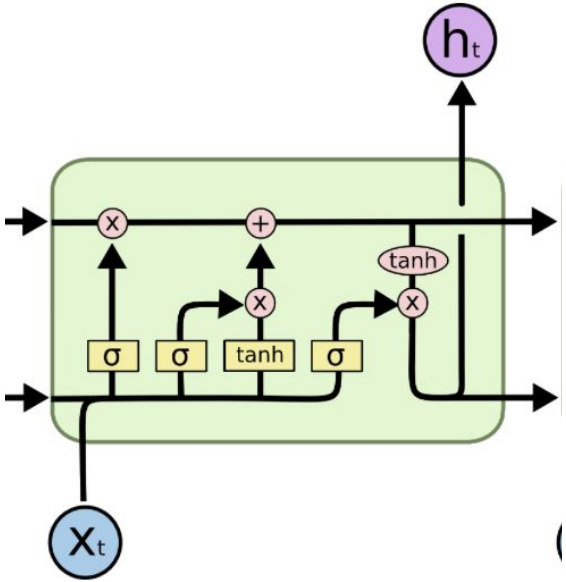
LSTM: Градиенты



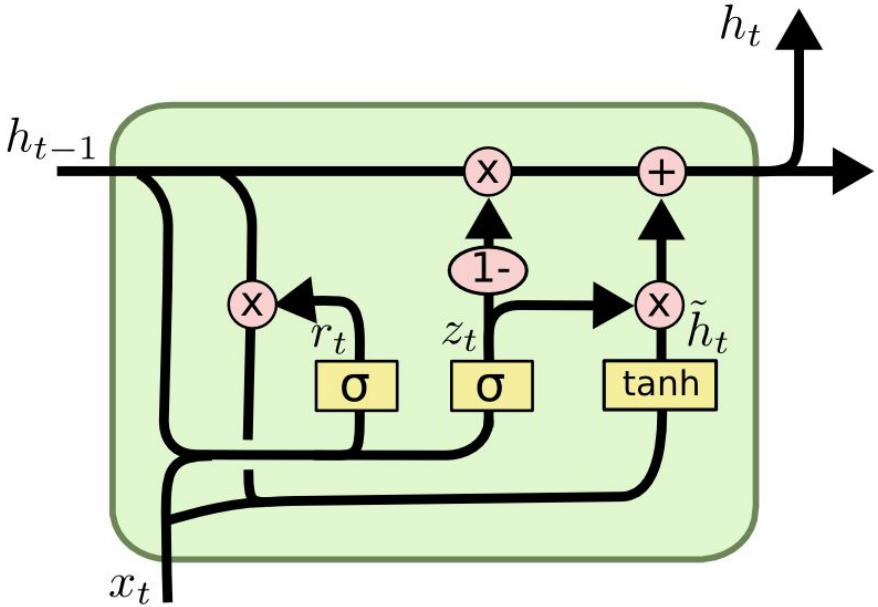
Перемножение сигмоид лучше, чем перемножение производных сигмоиды и нестабильно меняющихся весов.

LSTM: Выводы

1. Long Short Term Memory использует 2 памяти:
   - долгосрочную cell state C_t
   - скрытое состояние / рабочую память h_t
2. LSTM использует 3 gates, которые контролируют потоки информации
3. LSTM решает проблему Vanishing Gradient, но не решает проблему Exploding Gradient.



GRU (Gated Recurrent Units)

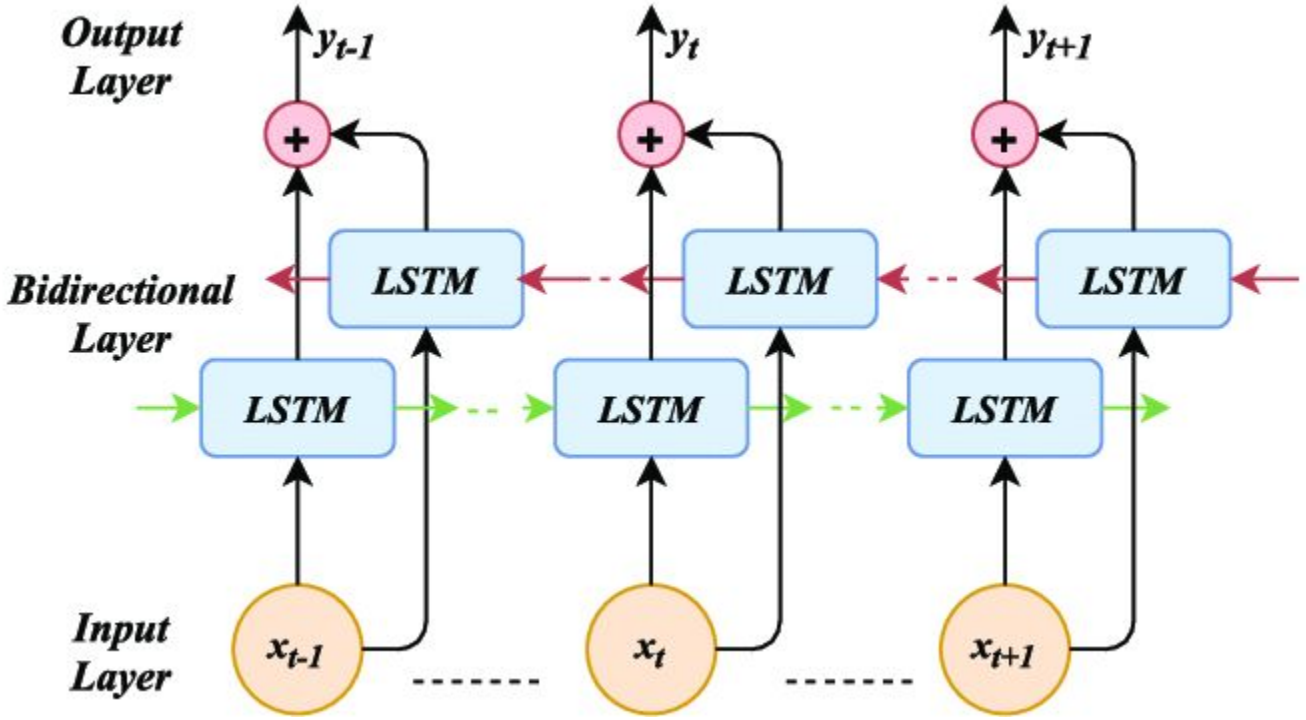


Разобрать самим!

$$\boldsymbol{u}_t = \sigma(W_{xu}\boldsymbol{x}_t + W_{hu}\boldsymbol{h}_{t-1} + \boldsymbol{b}_u),$$
$$\boldsymbol{r}_t = \sigma(W_{xr}\boldsymbol{x}_t + W_{hr}\boldsymbol{h}_{t-1} + \boldsymbol{b}_r),$$
$$\boldsymbol{h}'_t = \tanh(W_{xh'}\boldsymbol{x}_t + W_{hh'}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1})),$$
$$\boldsymbol{h}_t = (1 - \boldsymbol{u}_t) \odot \boldsymbol{h}'_t + \boldsymbol{u}_t \odot \boldsymbol{h}_{t-1}.$$

GRU [*Learning phrase representations using rnn encoder-decoder for statistical machine translation*, Cho et al. 2014]

Bidirectional LSTM

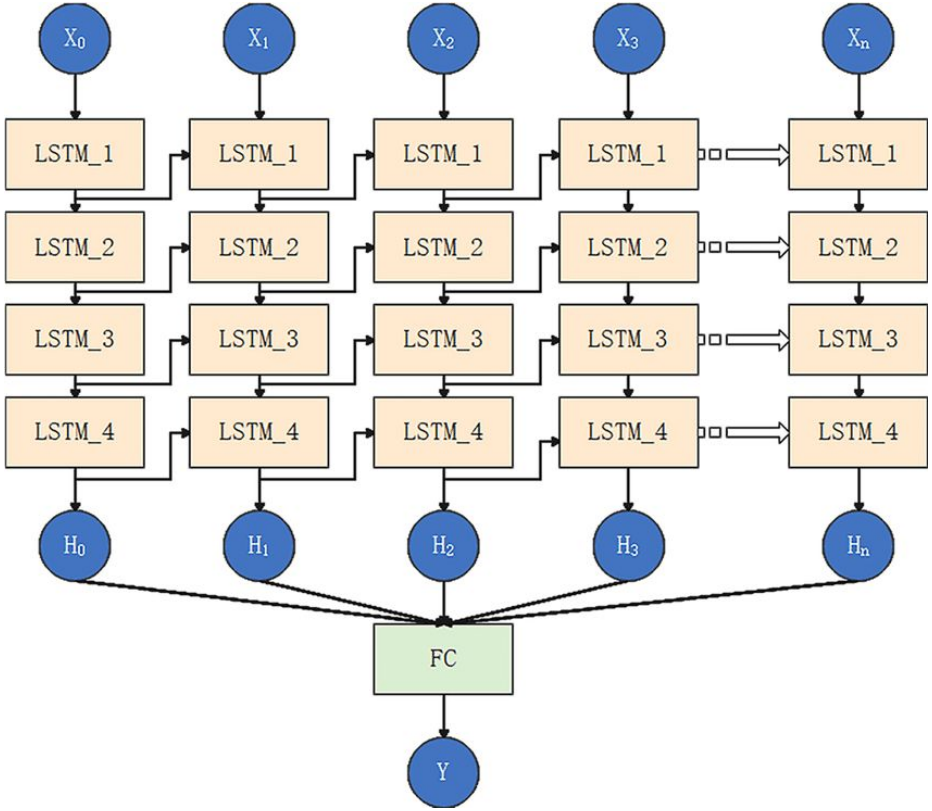


Stacked LSTM

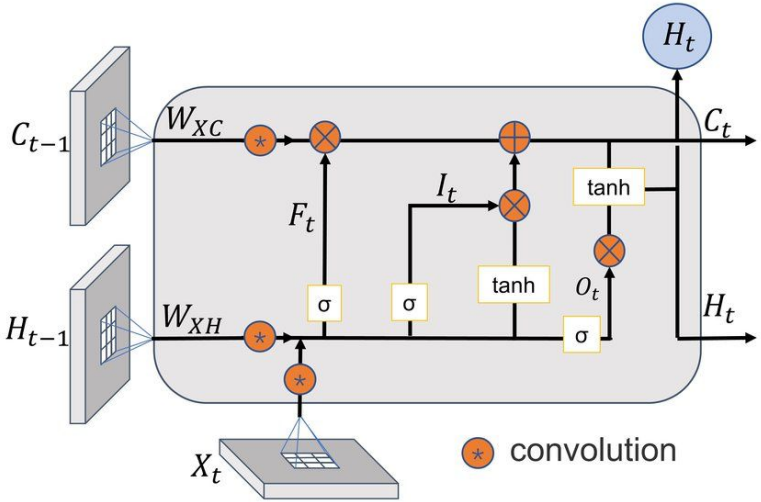
Можно использовать несколько LSTM. Выходы из LSTM №1 передаются на входы в LSTM №2.

Нижние слои – локальные признаки.

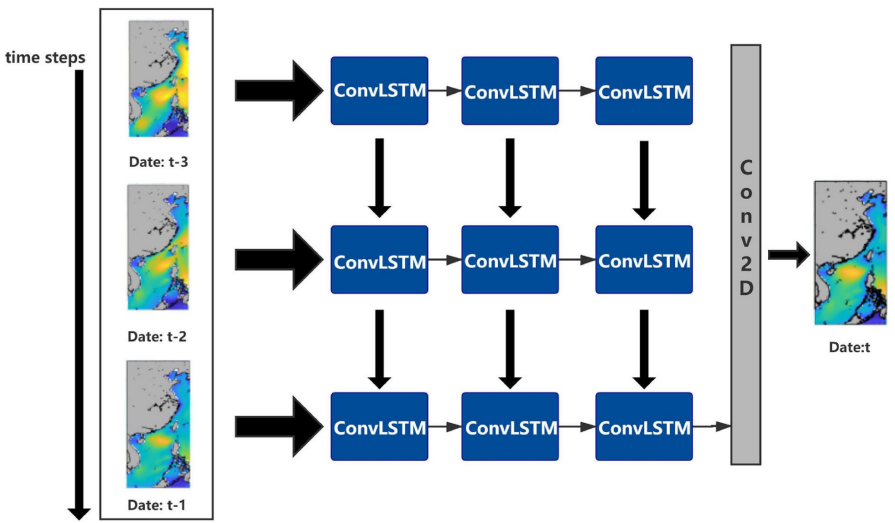
Верхние слои – общий семантический смысл



Conv LSTM



Conv LSTM



Предсказание карты на день t на основе предыдущих дней

Применение:
Image Captioning

Image Captioning

A cat sitting on a suitcase on the floor

A cat is sitting on a tree branch

A dog is running in the grass with a frisbee

A white teddy bear sitting in the grass

Two people walking on the beach with surfboards

A tennis player in action on the court

Two giraffes standing in a grassy field

A man riding a dirt bike on a dirt track

Image Captioning: таргет

таргет → A cat sitting on a suitcase on the floor

Image Captioning: таргет

Пусть есть словарь всех возможных слов.
Размер словаря - |V|.

таргет → A cat sitting on a suitcase on the floor

Image Captioning: таргет

Пусть есть словарь всех возможных слов.
Размер словаря - |V|.

таргет → A cat sitting on a suitcase on the floor

25 234 1566

индексы слов в словаре = номер класс

Image Captioning: таргет

Пусть есть словарь всех возможных слов.
Размер словаря - |V|.

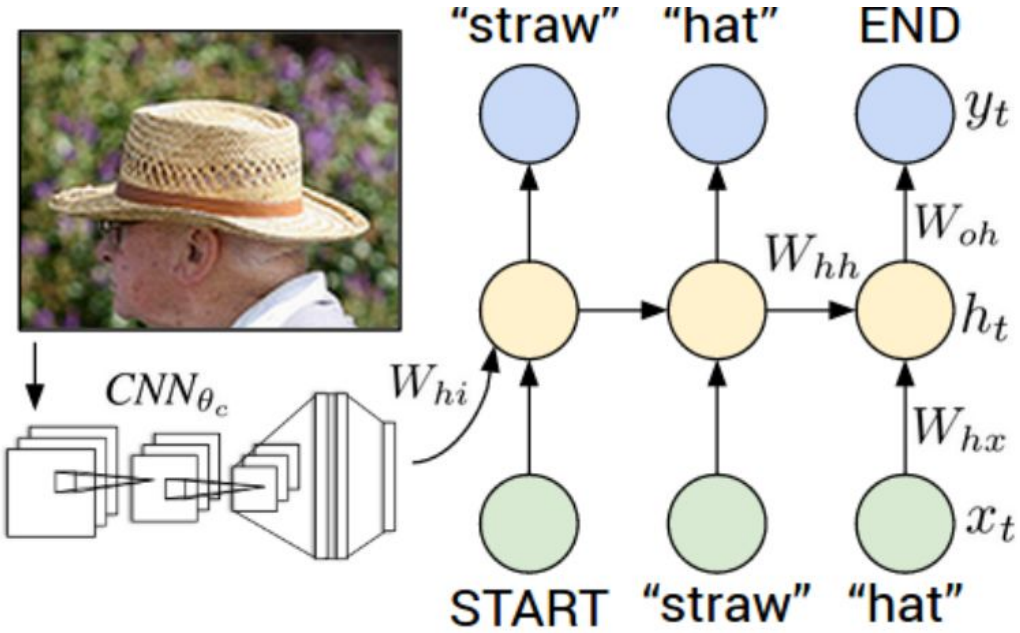
таргет → A cat sitting on a suitcase on the floor

25 234 1566

индексы слов в словаре = номер класс

Хотим, чтобы **модель предсказывала T векторов размера |V|**, где каждый i-ый элемент вектора t – это вероятность слова с индексом i на месте t.

Image Captioning: архитектура



Deep Visual-Semantic Alignments for Generating Image Descriptions, Karpathy and Fei-Fei

Image Captioning: возможная архитектура

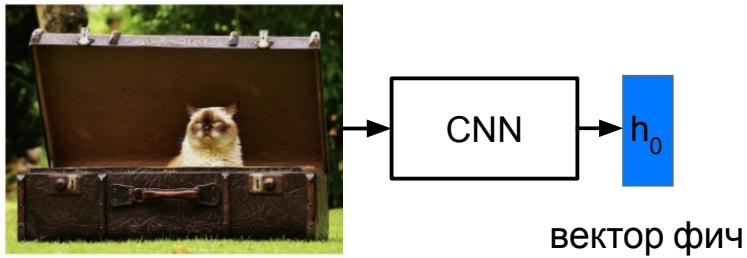
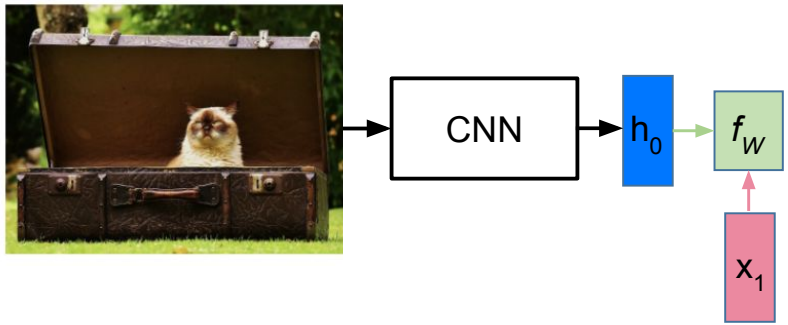


Image Captioning: возможная архитектура



Специальный токен, указывает
на начало предложения

Image Captioning: возможная архитектура

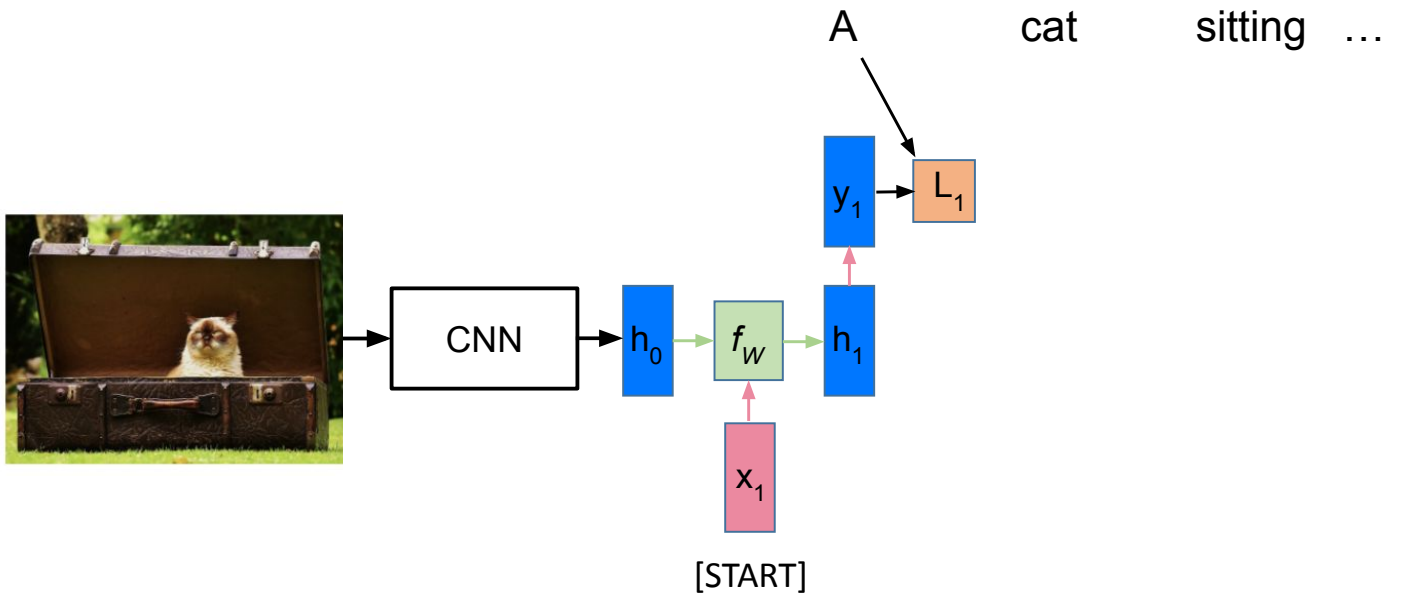
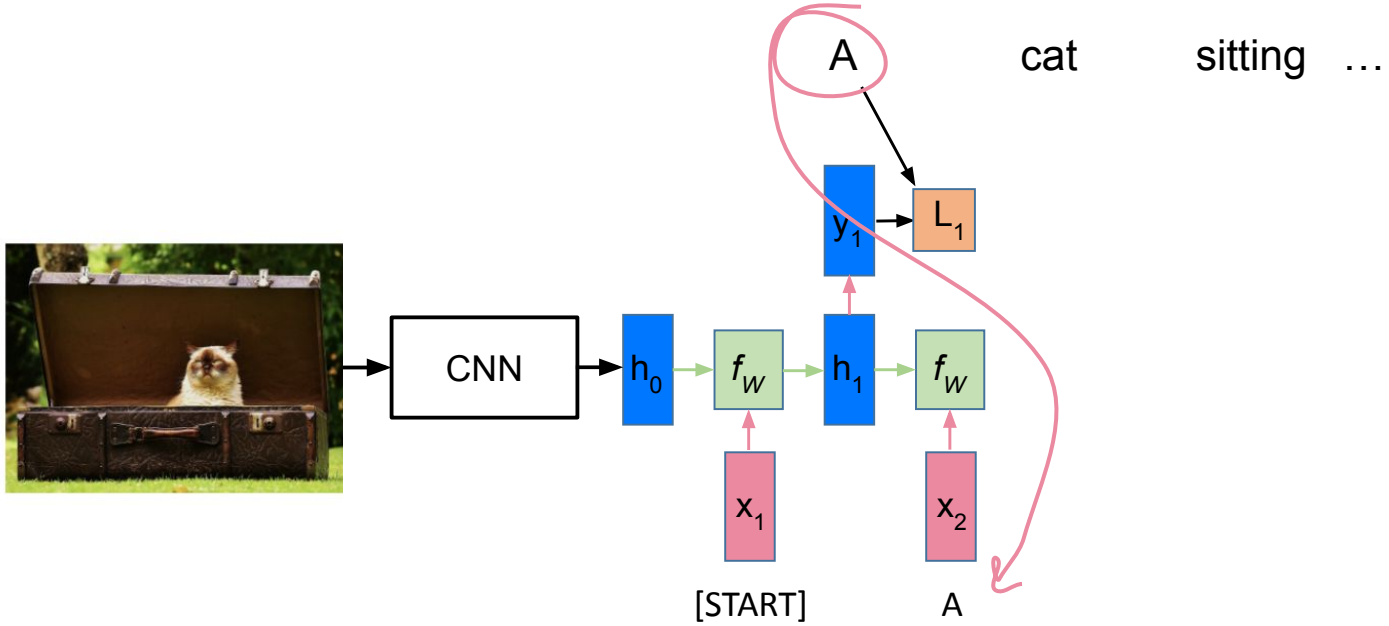


Image Captioning: возможная архитектура



На этапе обучения подаем на следующий шаг ground truth слово.

Image Captioning: возможная архитектура

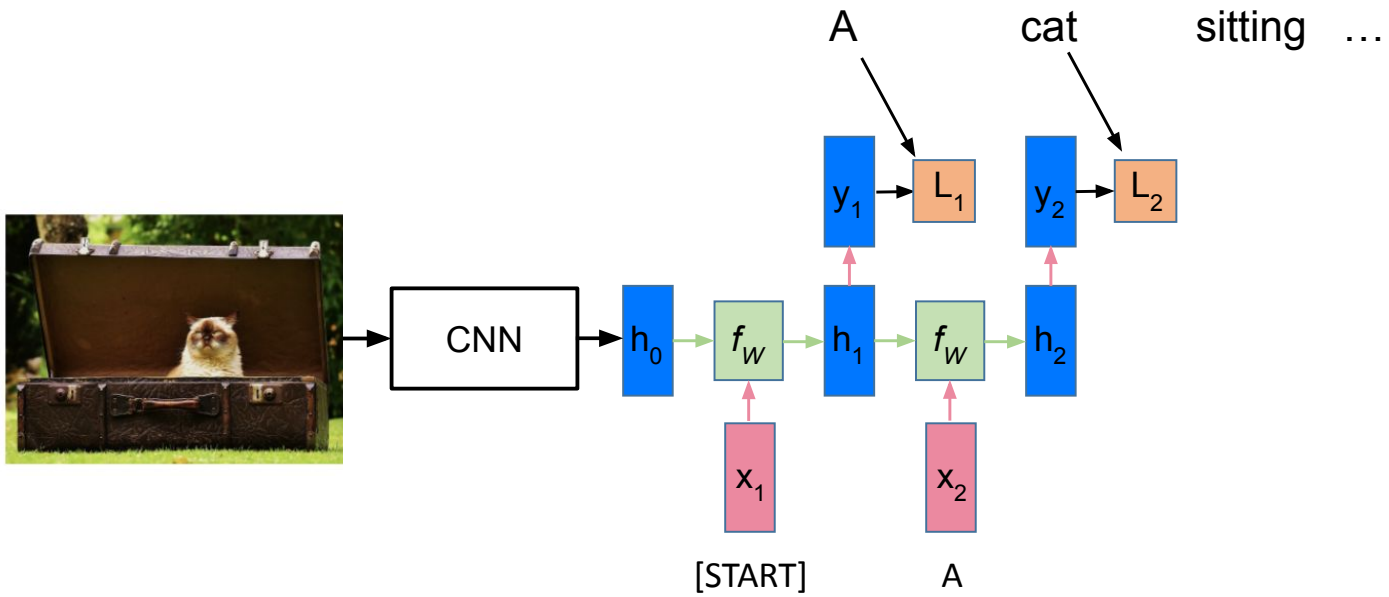


Image Captioning: возможная архитектура

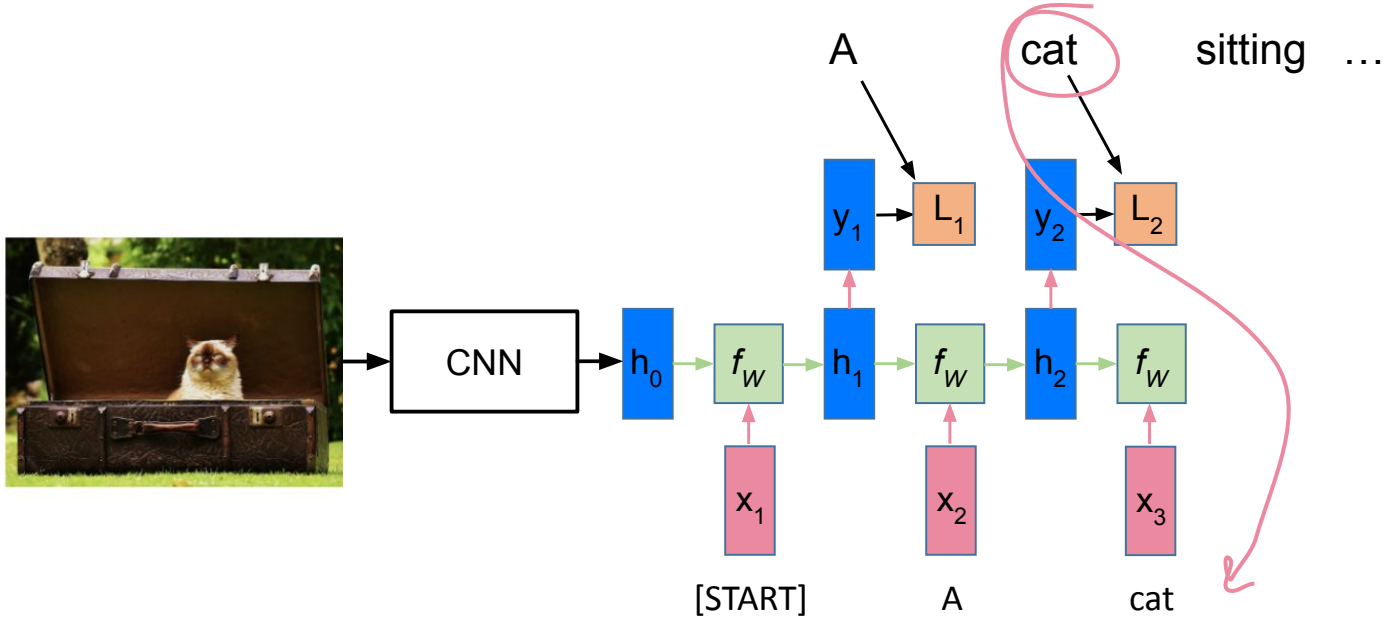


Image Captioning: возможная архитектура

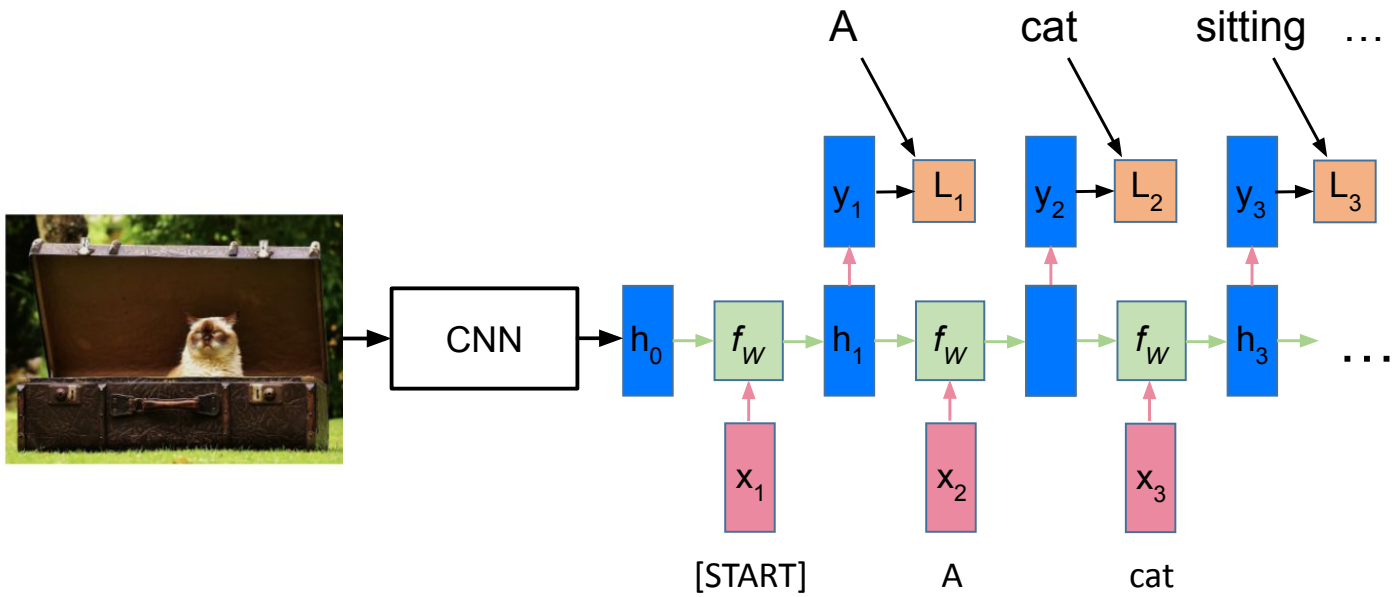
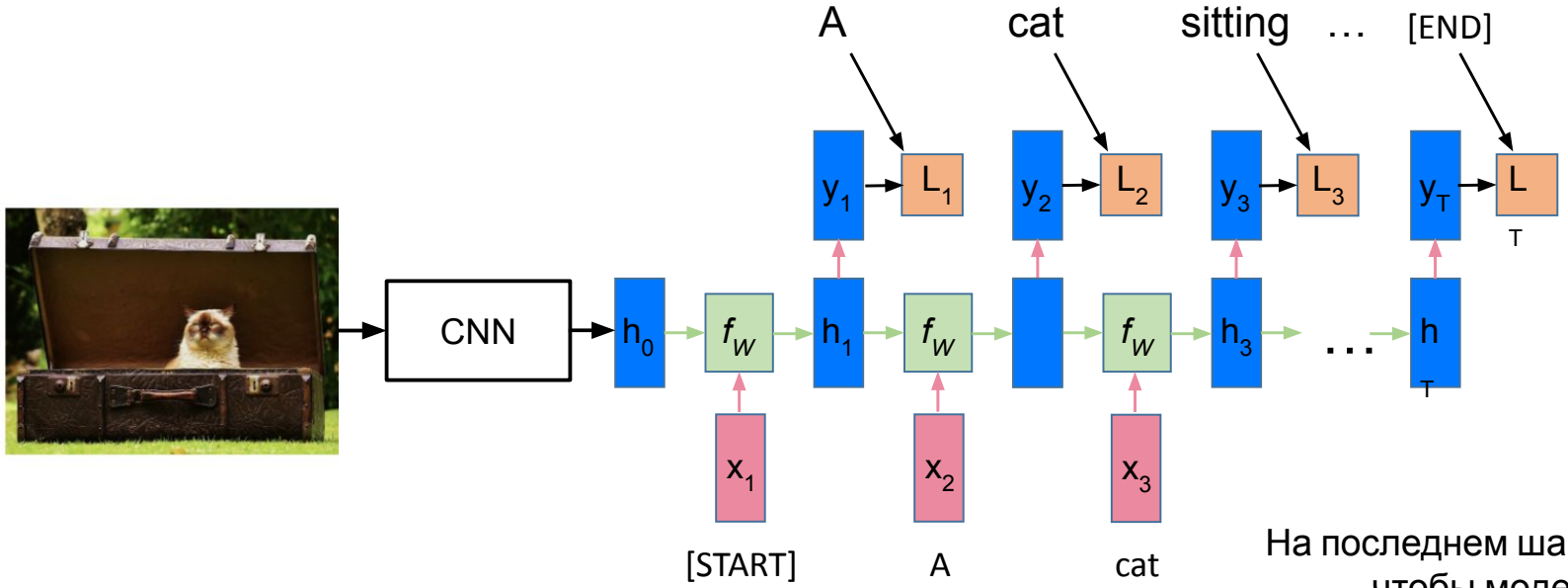


Image Captioning: возможная архитектура



На последнем шаге хотим, чтобы модель предсказывала специальный токен END.

Спасибо за внимание!