



Data Mining in Action

Сверточные и рекуррентные блоки



Блоки в нейронных сетях

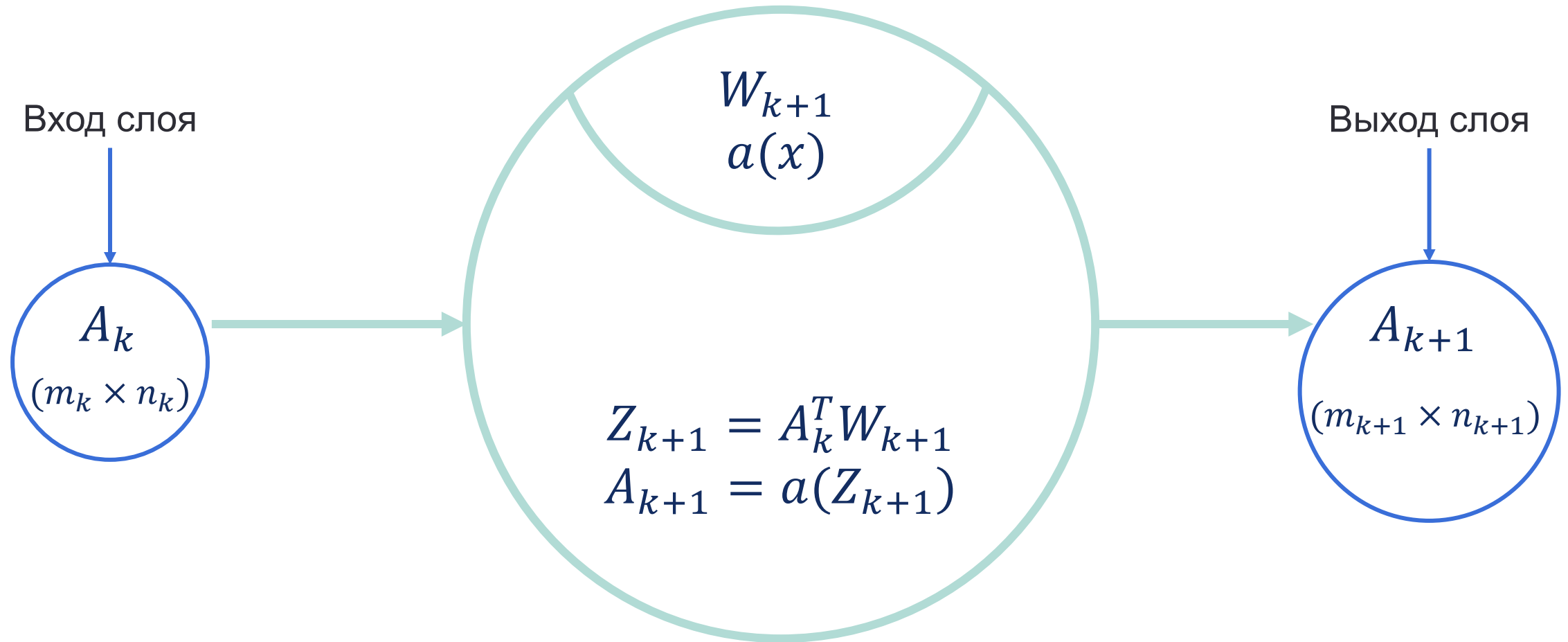
Нейронные сети – это графы операций, которые можно собрать из различных дифференцируемых блоков.

Сегодня на лекции уже появлялись блоки:

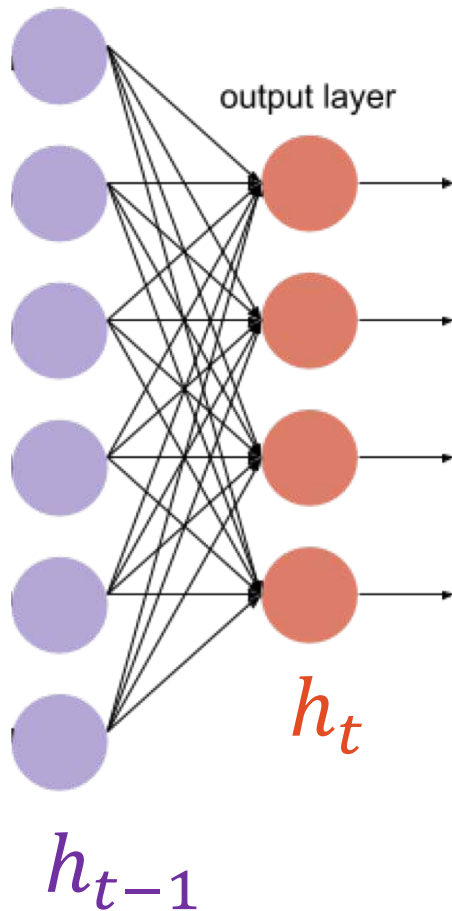
- Dropout
- BatchNorm/LayerNorm

Но они только *помогают* обучению и генерализации. Тогда какие операции/блоки *позволяют* учить сеть на разных форматах данных?

Полносвязный слой (Fully connected / Dense)



Более традиционный взгляд на dense layer



$$h_t = f(W h_{t-1} + b)$$

Dense Layer

На практике чаще всего используется для:

- получения выхода сети в классификации/регрессии;
- изменения размерности выхода предыдущего слоя;
- построения Multi-Layer Perceptron'a

План

1. Сверточные сети

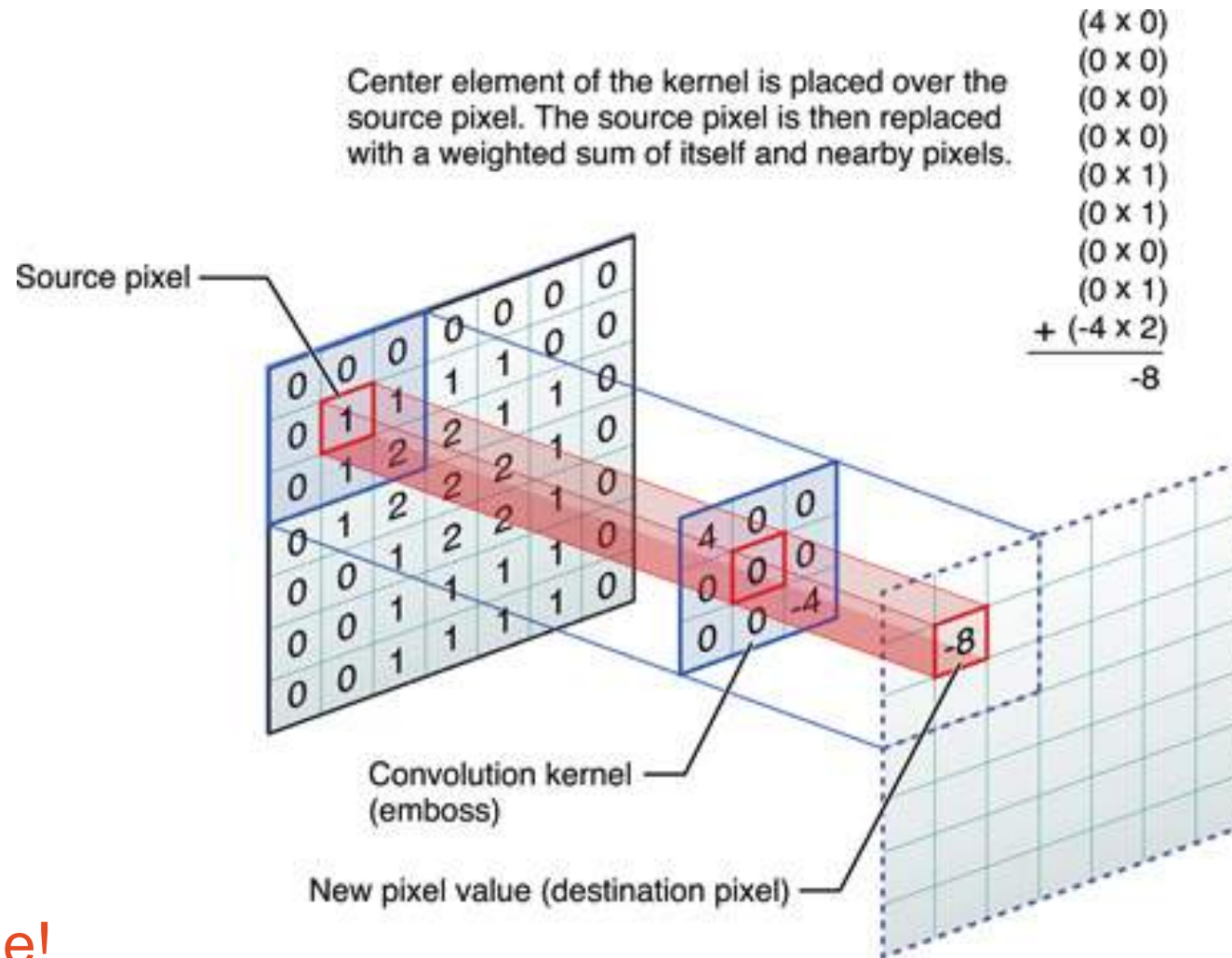
2. Рекуррентные сети

3. Затухание градиента и LSTM

4. Применение блоков

1. Сверточные сети

Операция свертки

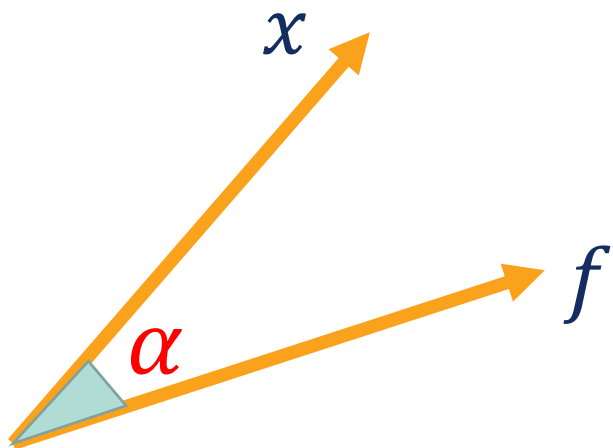


Это же
скалярное
произведение!

Напоминание: скалярное произведение

Пусть оба вектора нормированы (имеют единичную длину).

Когда их скалярное произведение максимально?



$$\langle f, x \rangle = \|f\| \|x\| \cos \alpha = \cos \alpha$$

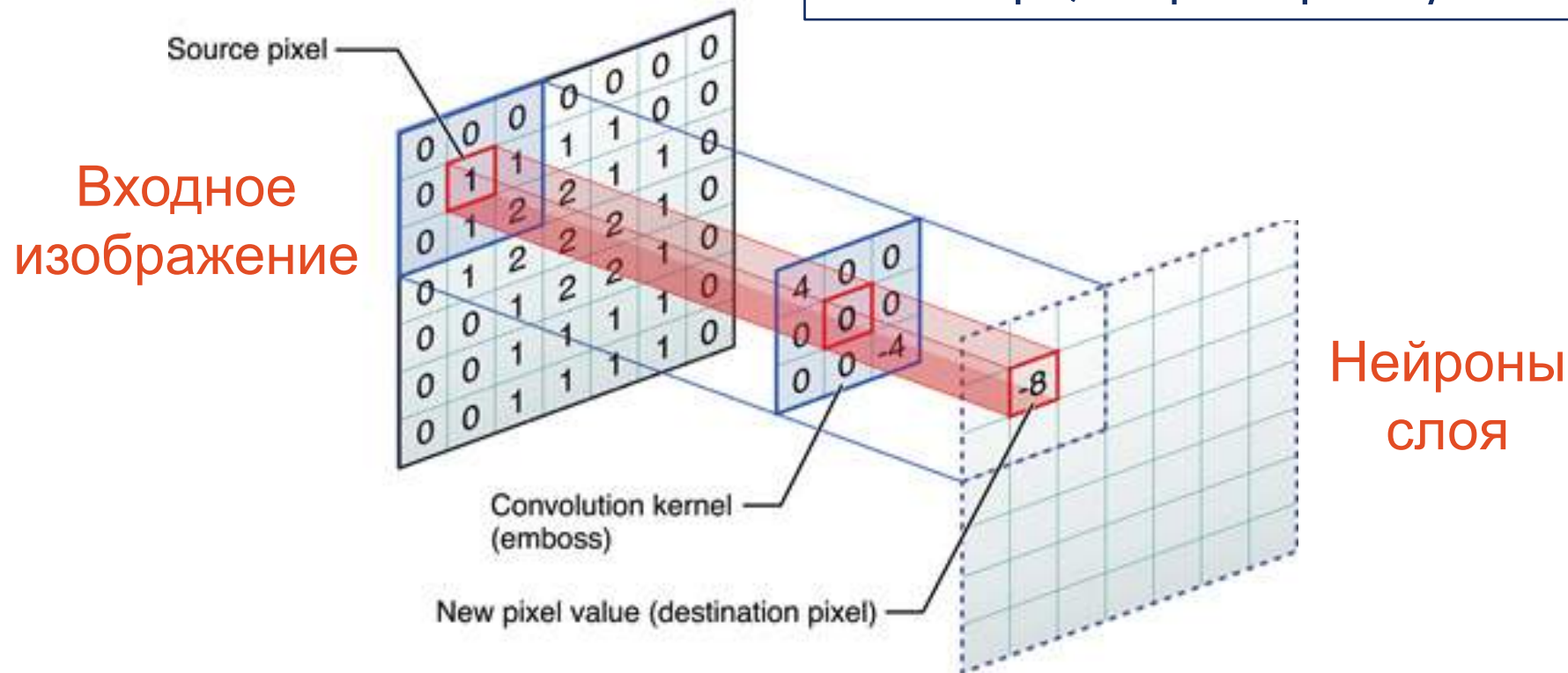
$$\cos \alpha = 1$$

$$\alpha = 0$$

Поэтому свертка с фильтром просто
вычисляет похожесть фрагмента
изображения на фильтр

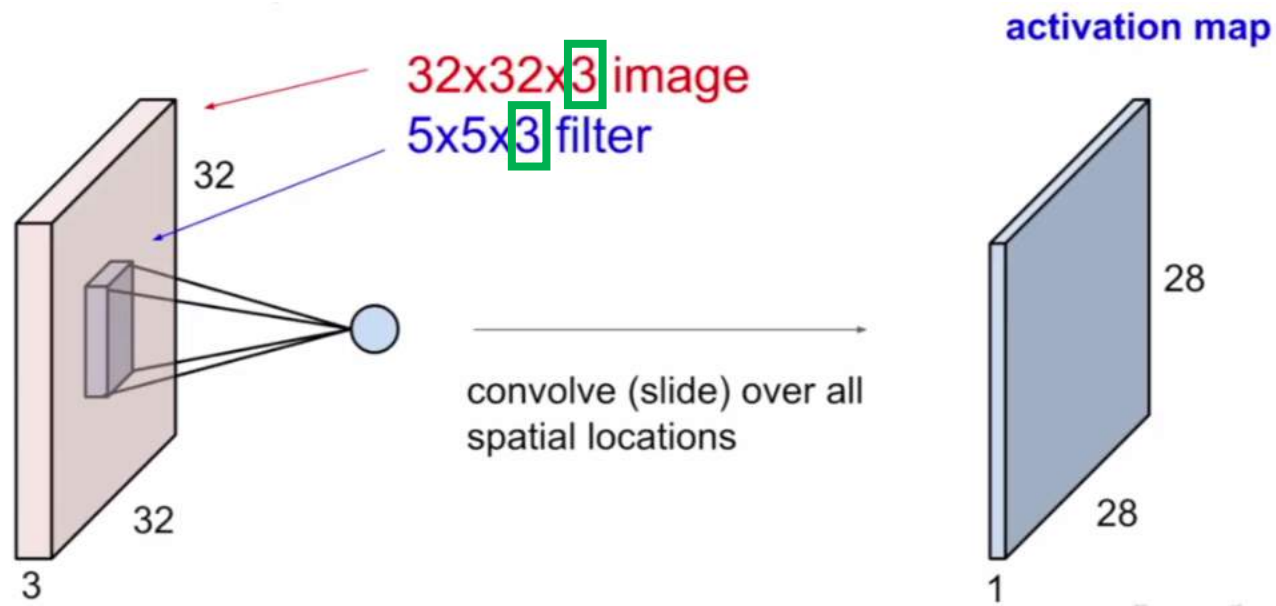
Сверточный слой (Convolutional layer)

1. Т.к. фильтр – это настраиваемые backprop веса, то **сеть сама «подберет» фильтры**
2. Но т.к. фильтр «замечает» только один паттерн, то фильтров нужно **больше**



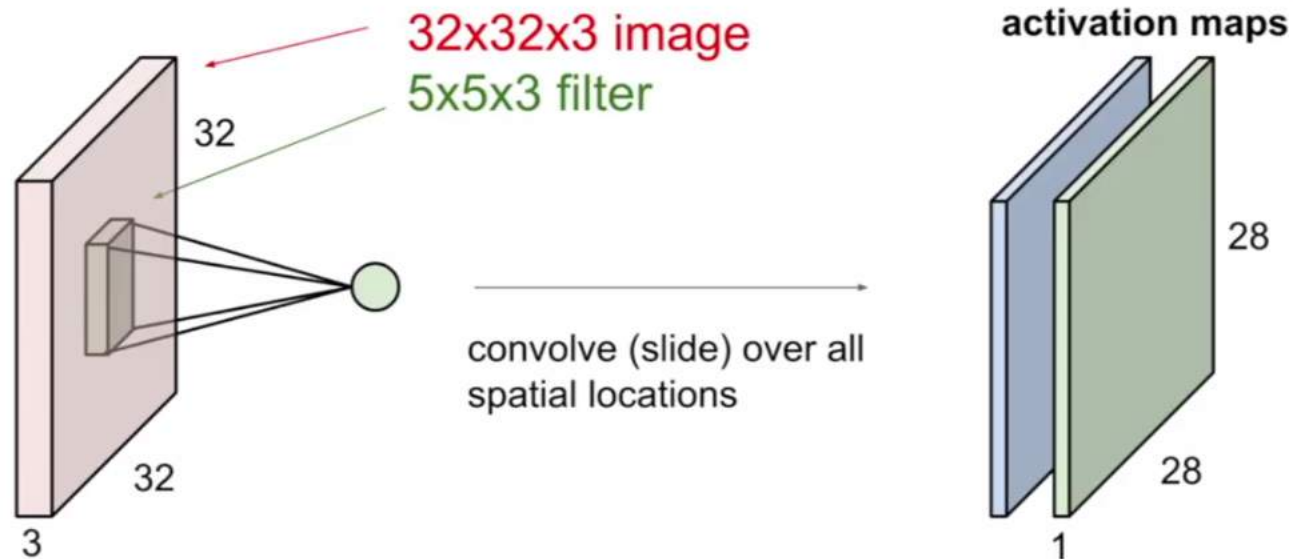
Сверточный слой

Свертка не плоская, а такой же «толщины» как исходное изображение



Сверточный слой

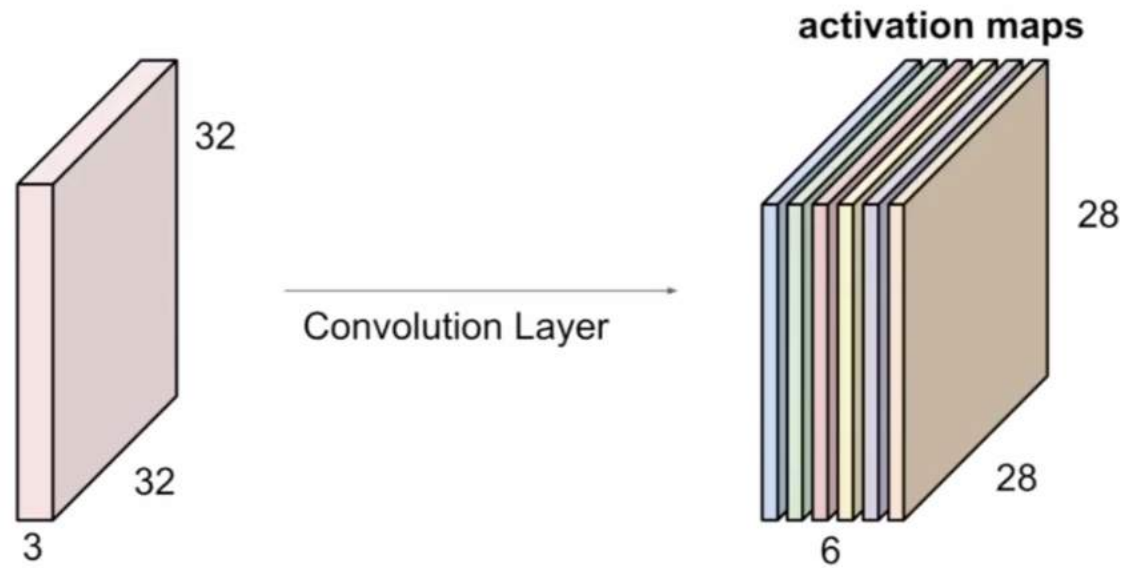
Свертка не плоская, а такой же «толщины» как исходное изображение



Каждая свертка порождает еще одну карту активации (карту признаков)

Сверточный слой

Свертка не плоская, а такой же «толщины» как исходное изображение



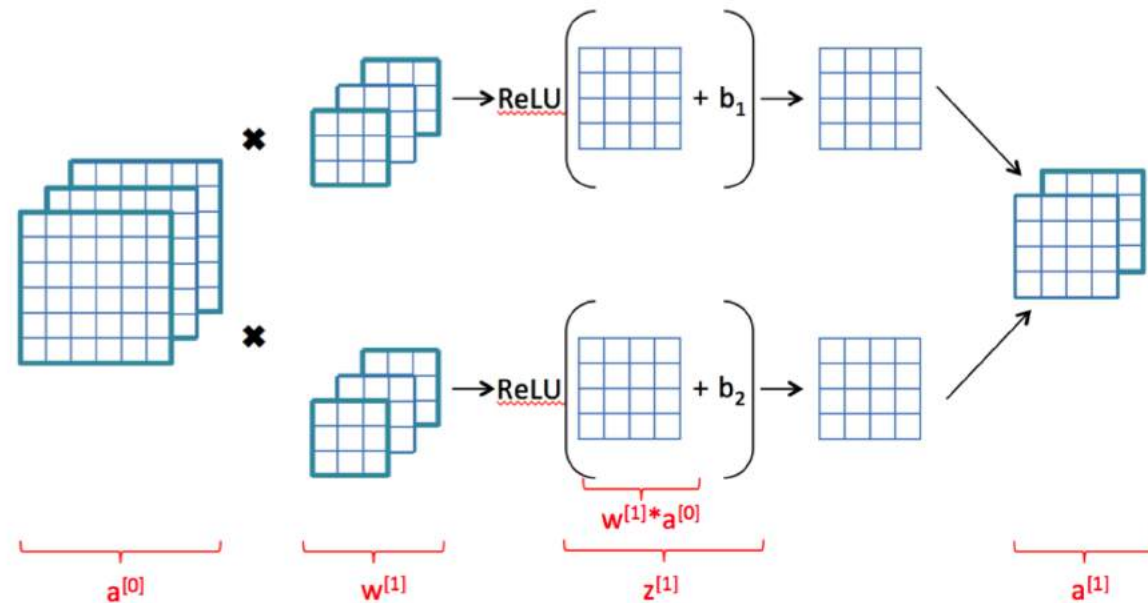
Каждая свертка порождает еще одну карту активации (карту признаков)

В одном слое делают много фильтров (а, значит, много карт активации)

Можно считать, что теперь в изображении столько каналов (такая толщина) – значит такой будет толщина фильтров в следующем слое

Сверточный слой: сдвиг и активация

К результатам свертки также, как это было в полносвязных сетях, добавляется сдвиг (порог) b и результат также подается на вход нелинейности (например ReLU)



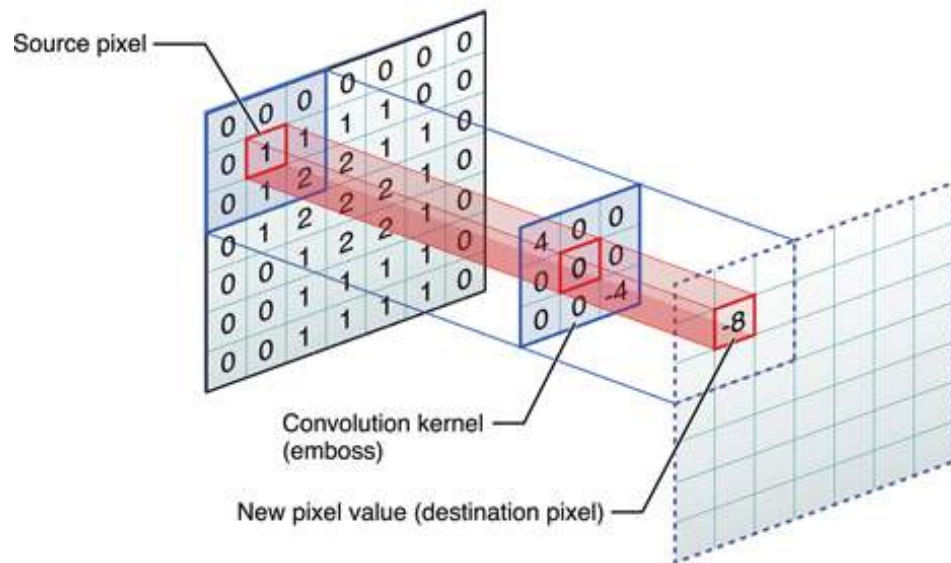
Сверточный слой: применение

На практике чаще всего используется для:

- анализа изображений
- анализа текстов
- анализа аудио

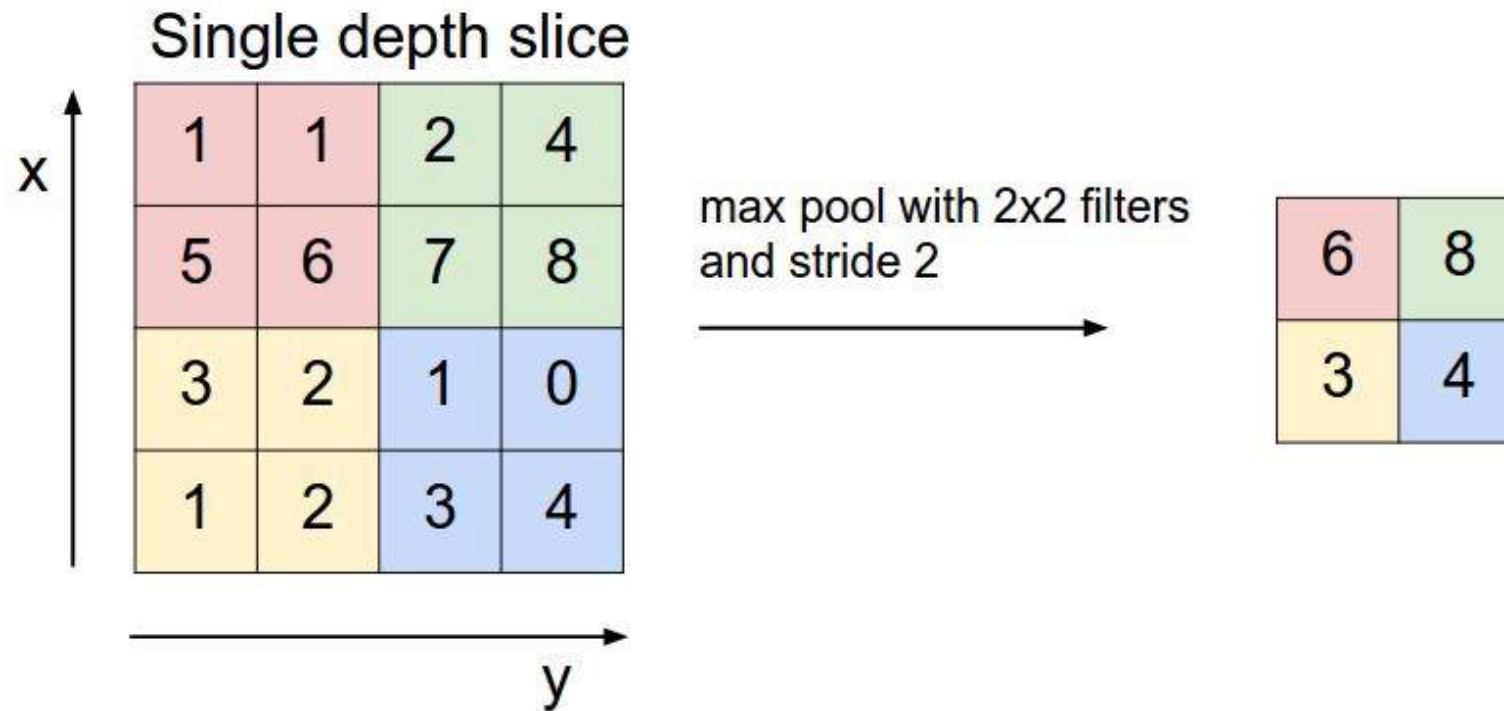
Все благодаря «нахождению паттернов»

Предпосылки для слоев пулинга

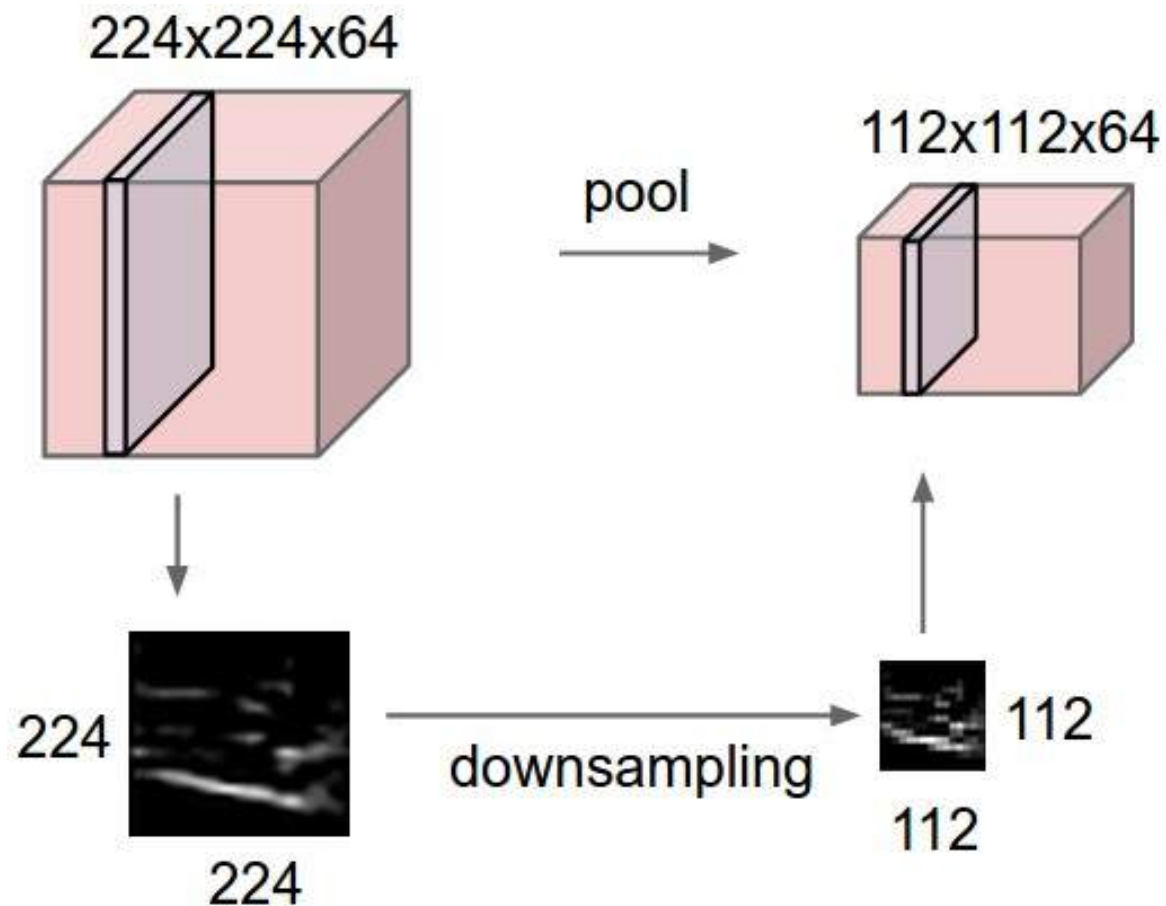


- Размер изображения после свертки такой же или почти такой же
- Размерность не уменьшается, а еще и фильтров будет много – значит только увеличится
- Получается очень много параметров – есть риск не получить никакой обобщающей способности у алгоритма

Слой пулинга (Pooling layer)



Слой пулинга (Pooling layer)



2. Рекуррентные нейросети

Рекуррентная нейросеть как формула

x_1, \dots, x_l — векторные представления последовательно идущих слов из текста

Рекуррентная нейросеть как формула

x_1, \dots, x_l — векторные представления последовательно идущих слов из текста

Скрытое представление текста к i -тому слову:

$$h_i = f_h(W_{xh}x_i + W_{hh}h_{i-1} + b_h)$$

Рекуррентная нейросеть как формула

x_1, \dots, x_l — векторные представления последовательно идущих слов из текста

Скрытое представление текста к i -тому слову:

$$h_i = f_h(W_{xh}x_i + W_{hh}h_{i-1} + b_h)$$

Прогноз ответа по тексту от 1 до i -того слова:

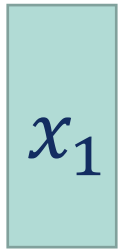
$$\hat{y}_i = f_y(W_{hy}h_i + b_y)$$

Рекуррентная нейросеть как формула

Значения скрытого слоя на каждом слове:

$$\begin{aligned}h_1 &= f_h(W_{xh}x_1 + 0 + b_h) \\h_2 &= f_h(W_{xh}x_2 + W_{hh}h_1 + b_h) \\h_3 &= f_h(W_{xh}x_3 + W_{hh}h_2 + b_h) \\&\dots\end{aligned}$$

Схема RNN



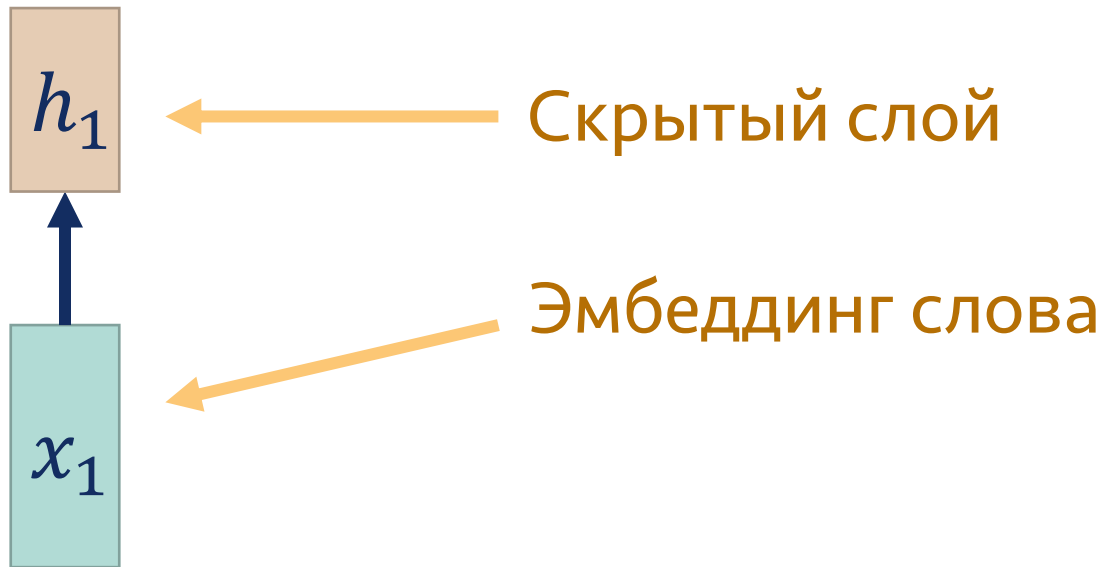
A light blue rectangular box containing the mathematical symbol x_1 .

x_1

Эмбе́динг слова

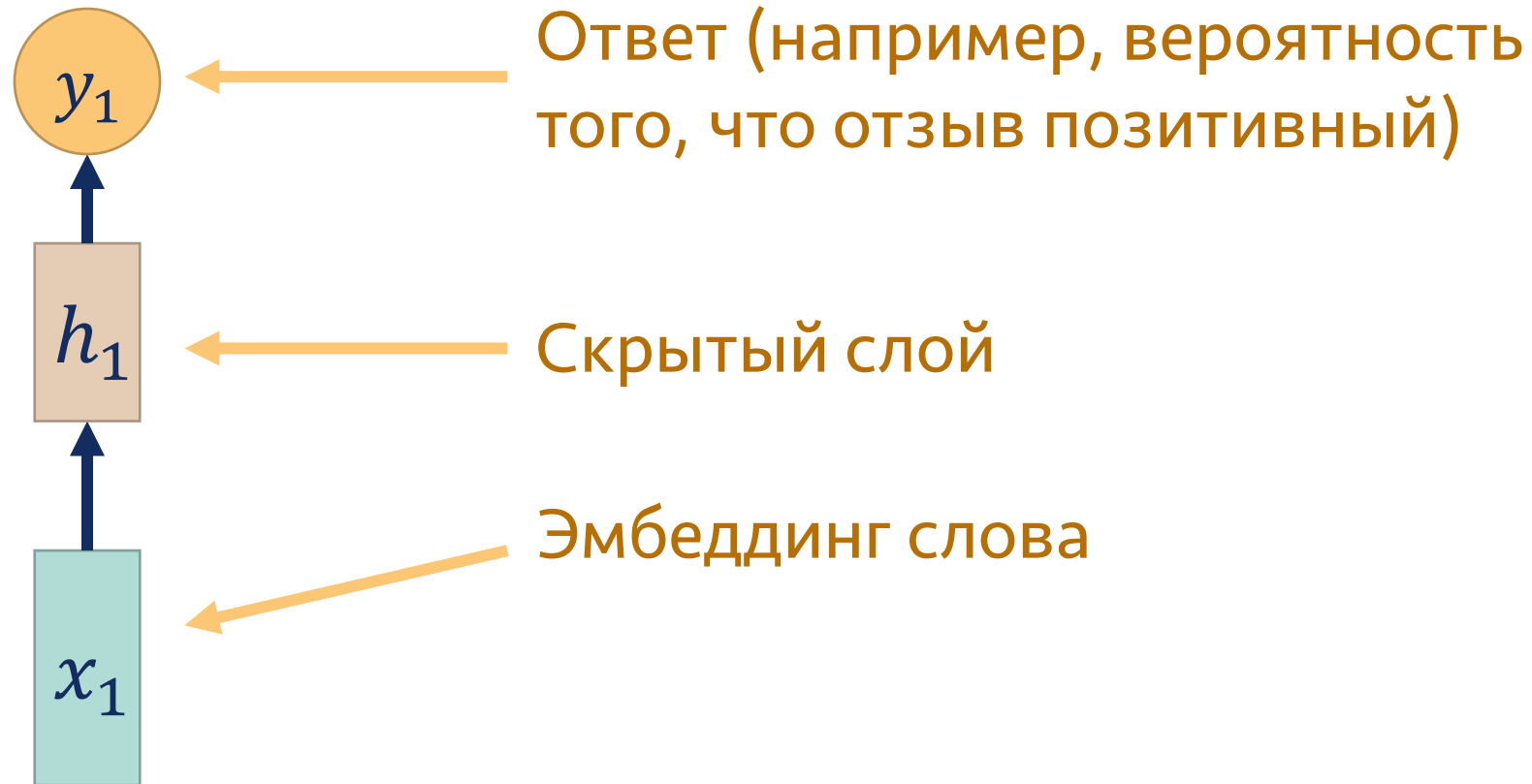
Вчера телефон перестал работать

Схема RNN



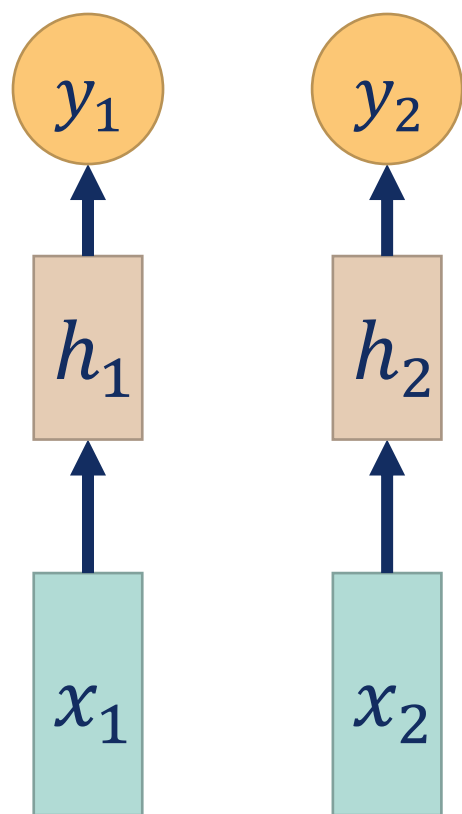
Вчера телефон перестал работать

Схема RNN



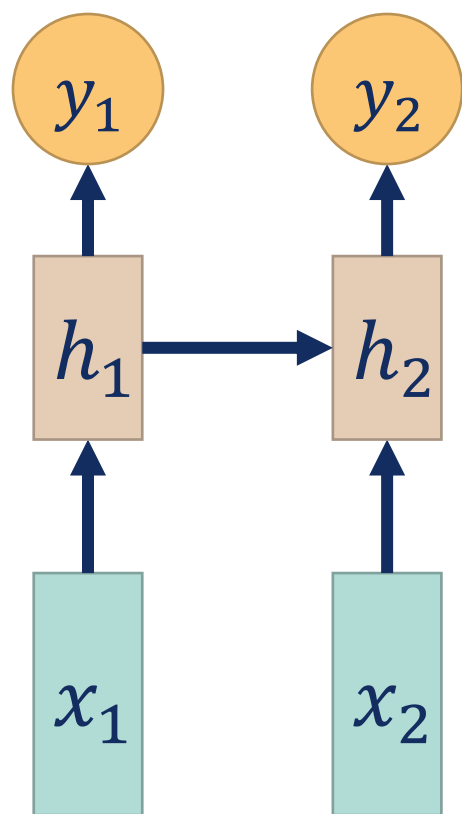
Вчера телефон перестал работать

Схема RNN



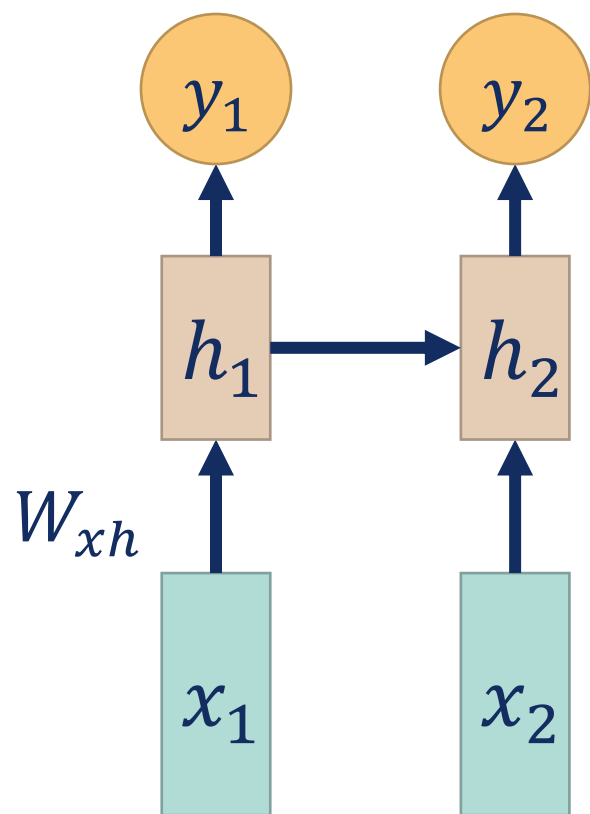
Вчера телефон перестал работать

Схема RNN



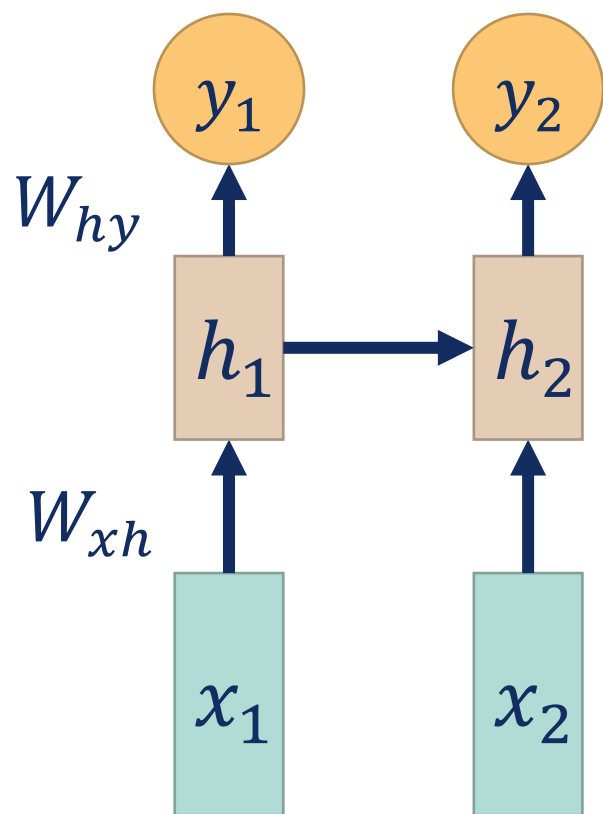
Вчера телефон перестал работать

Схема RNN



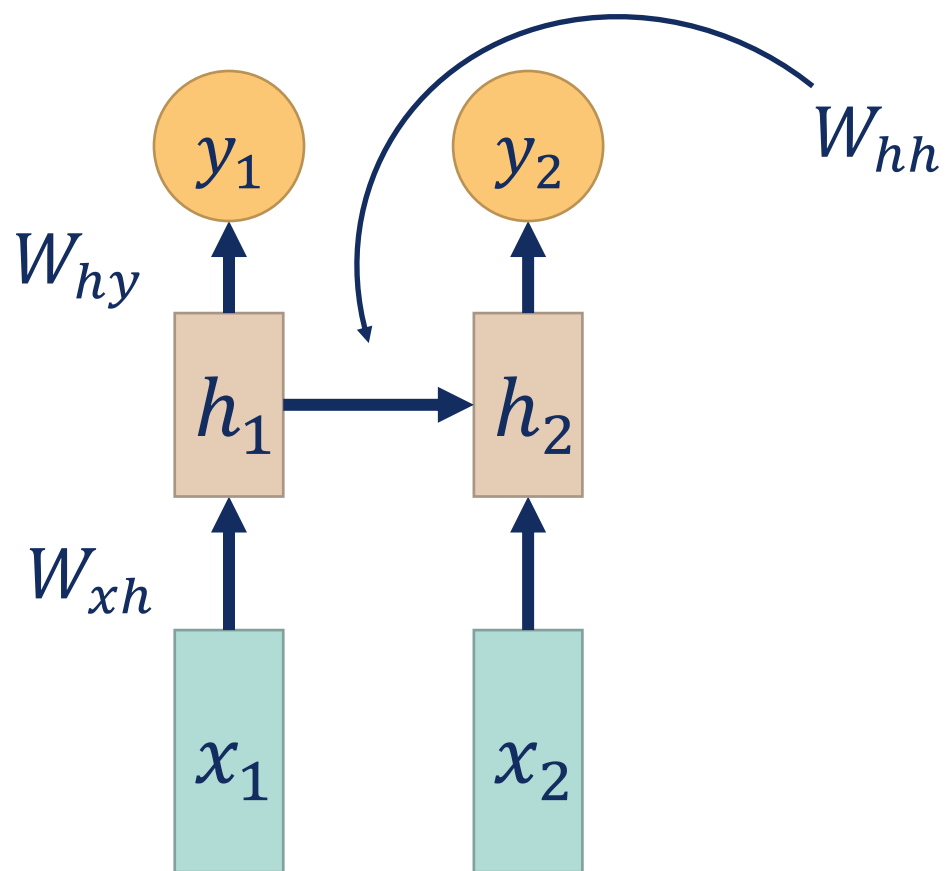
Вчера телефон перестал работать

Схема RNN



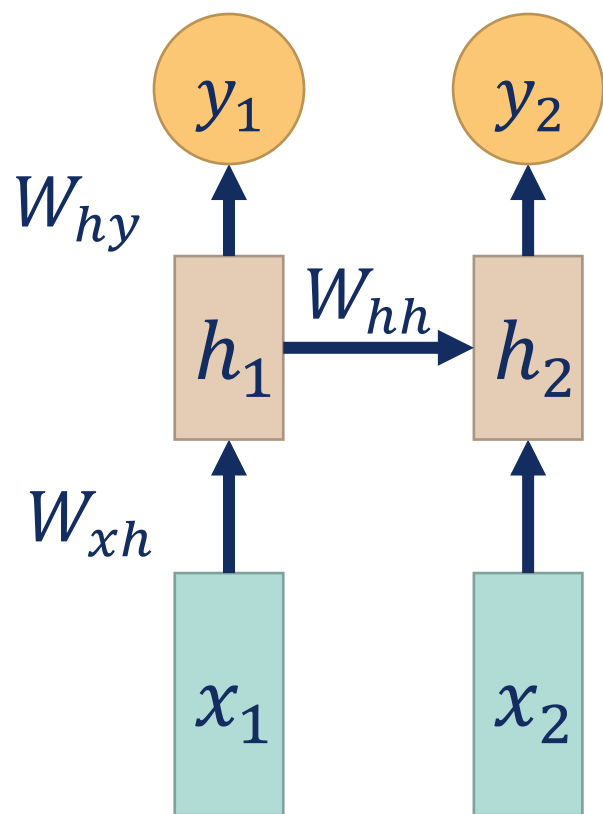
Вчера телефон перестал работать

Схема RNN



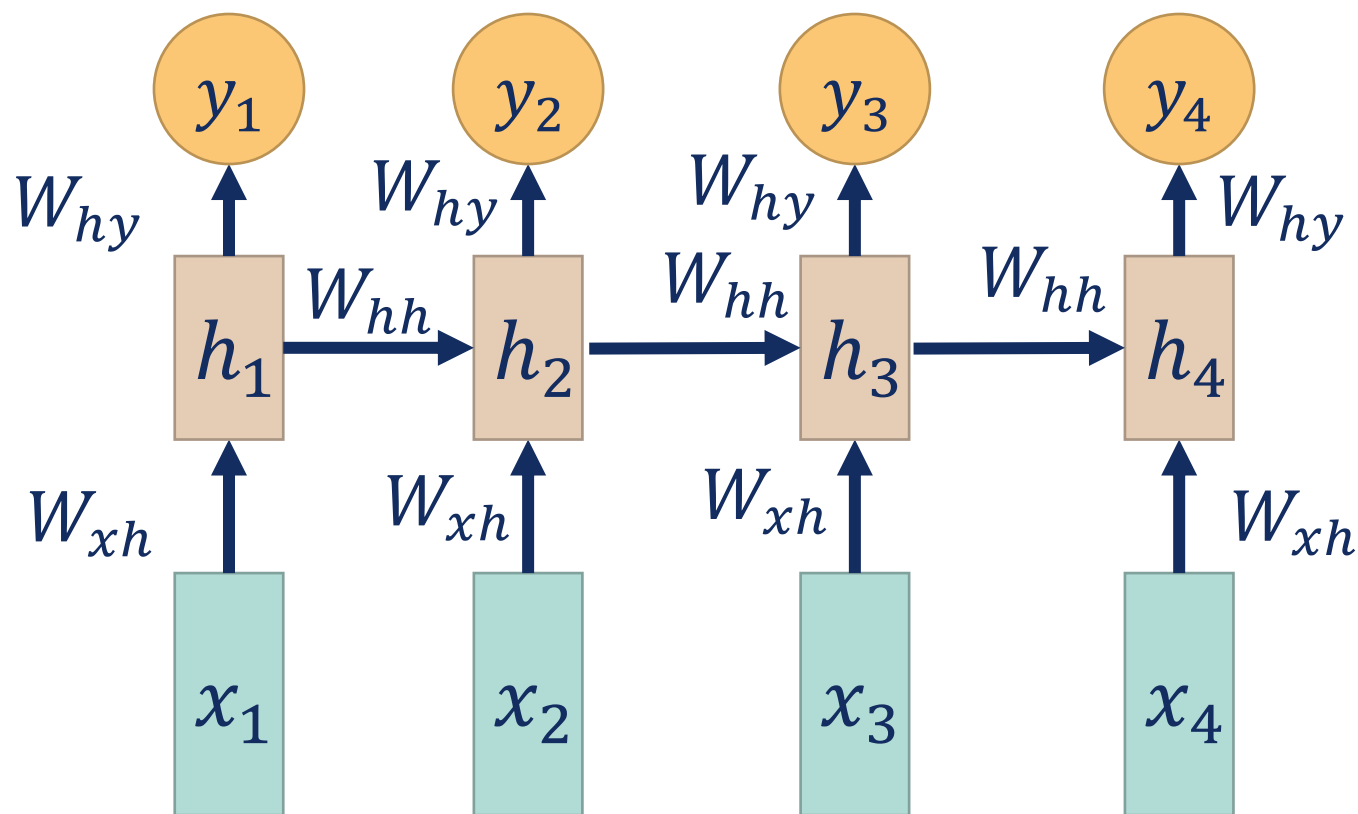
Вчера телефон перестал работать

Схема RNN



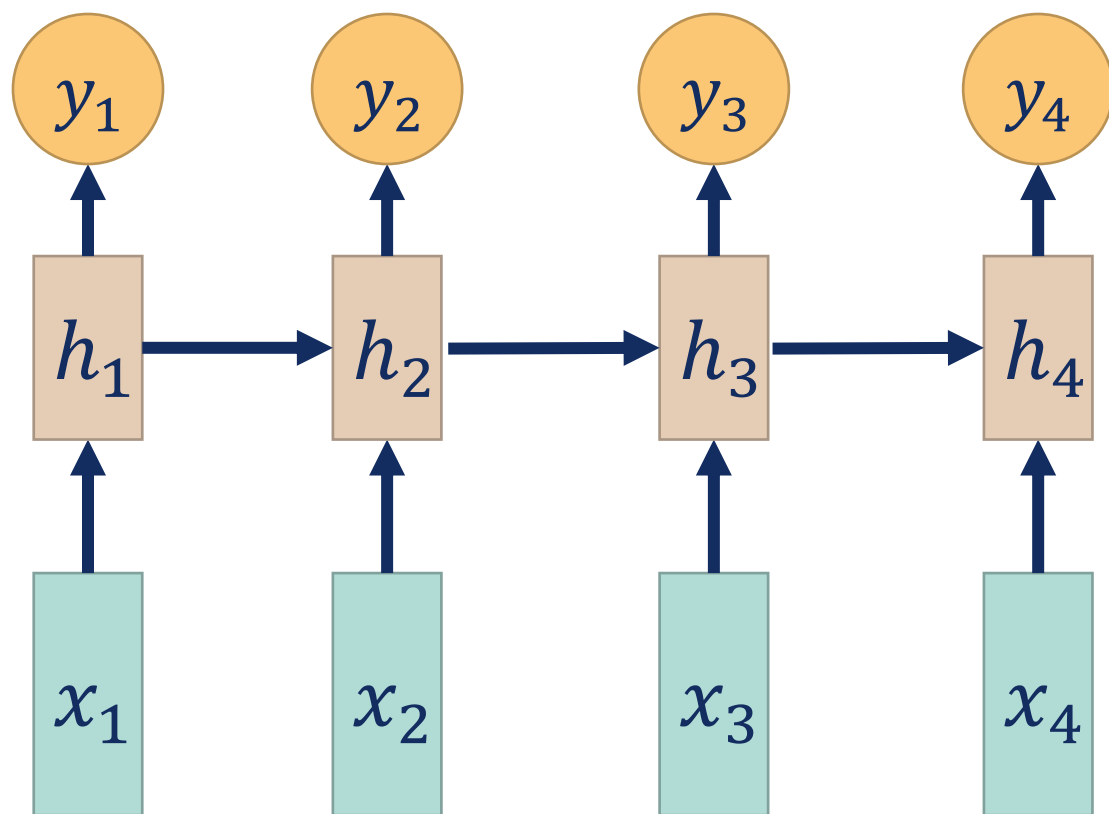
Вчера телефон перестал работать

Схема RNN



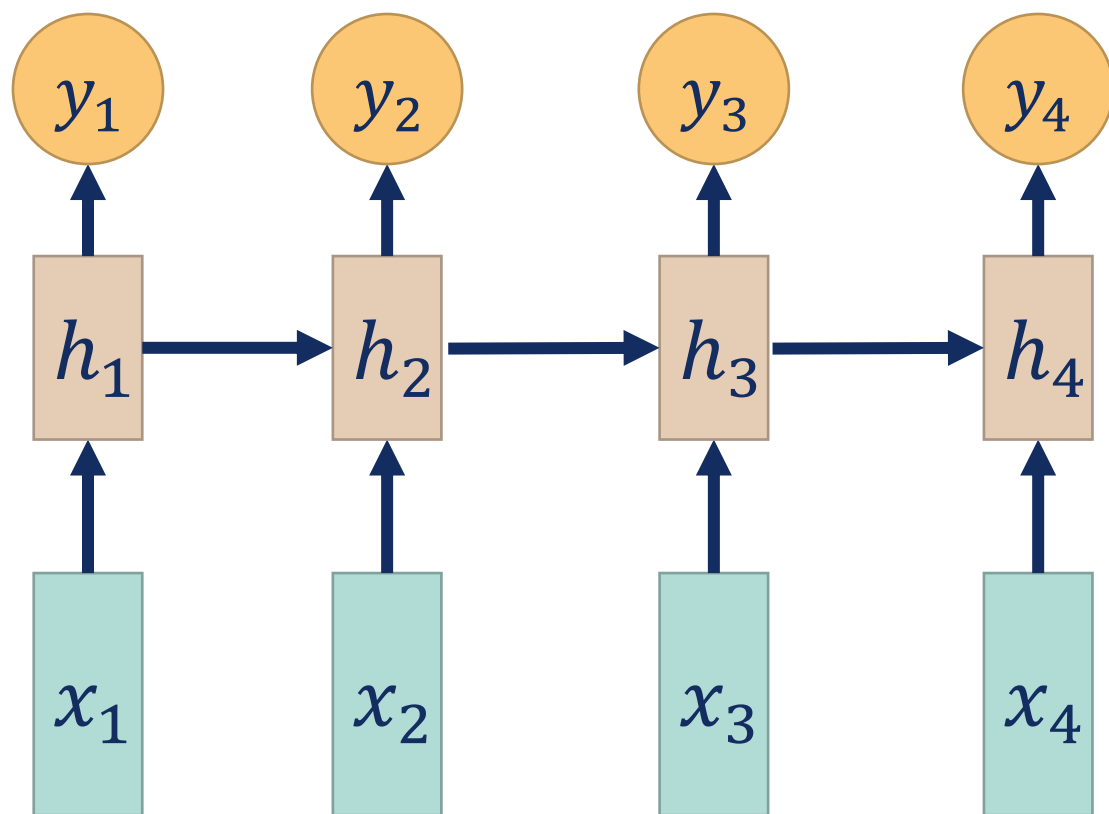
Вчера телефон перестал работать

Схема RNN

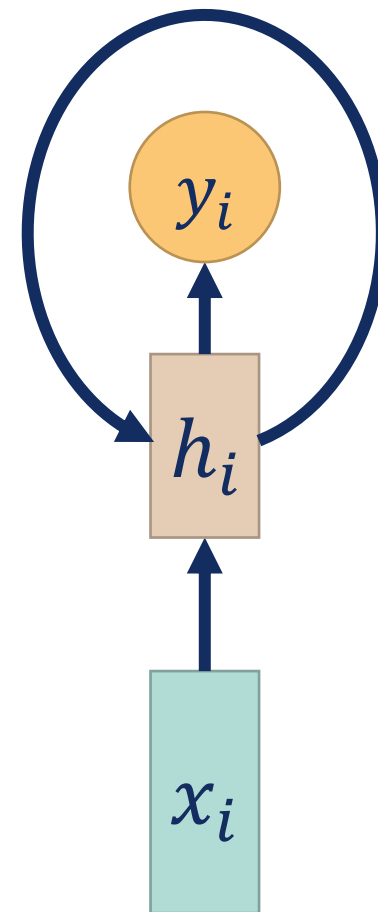


Вчера телефон перестал работать

Схема RNN

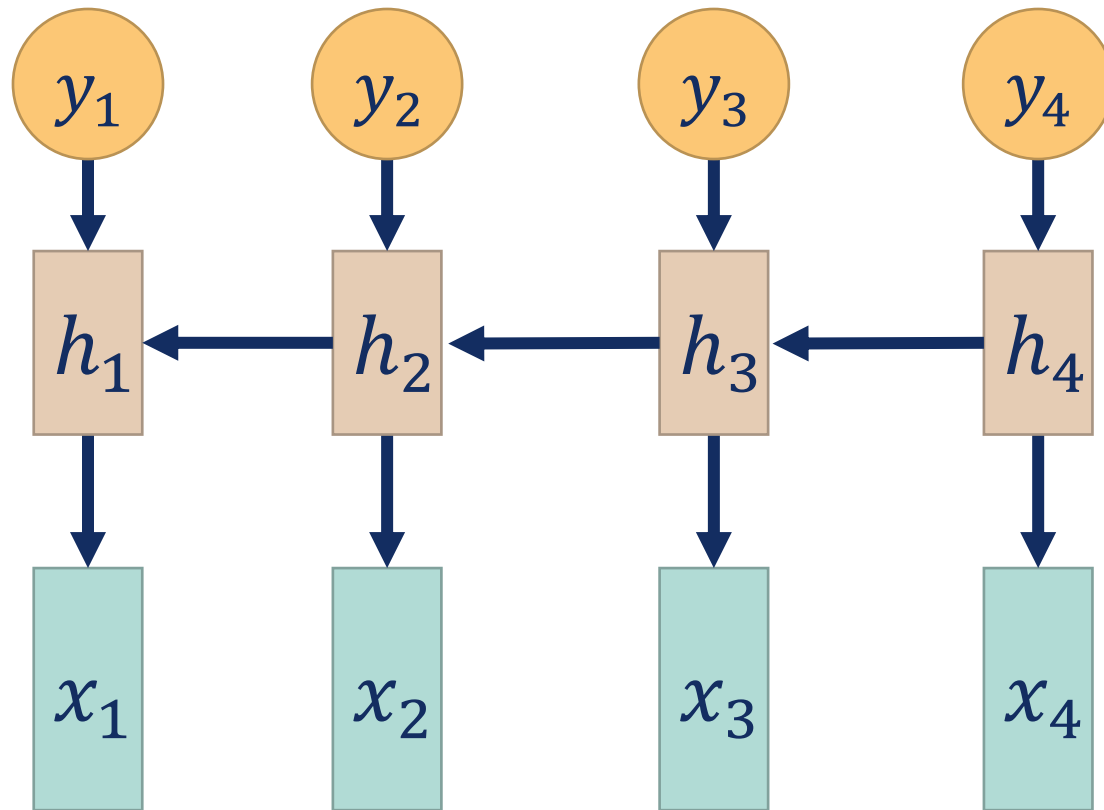


~



Вчера телефон перестал работать

Обучение RNN: backpropagation through time



Вчера телефон перестал работать

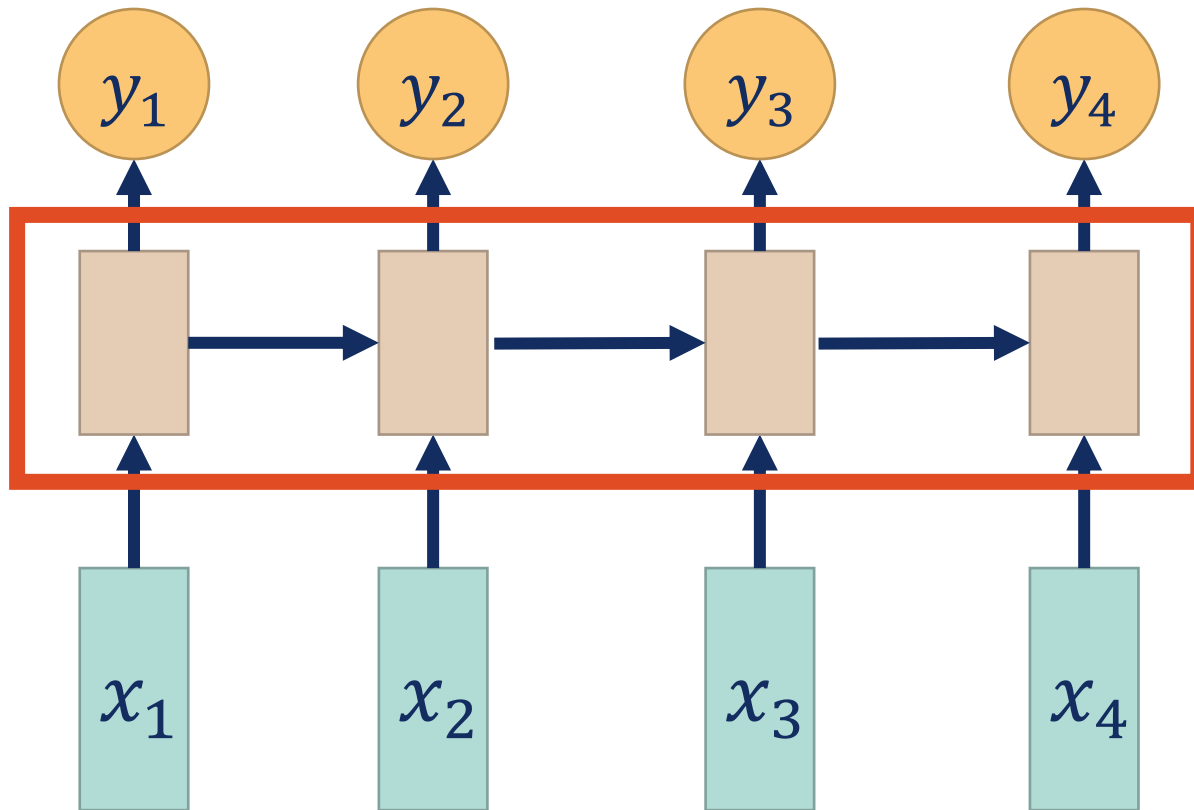
Основная проблема Vanilla RNN

Учитывать длинные последовательности слов сложно
из-за затухания градиентов

Вся следующая секция посвящена борьбе с этой проблемой

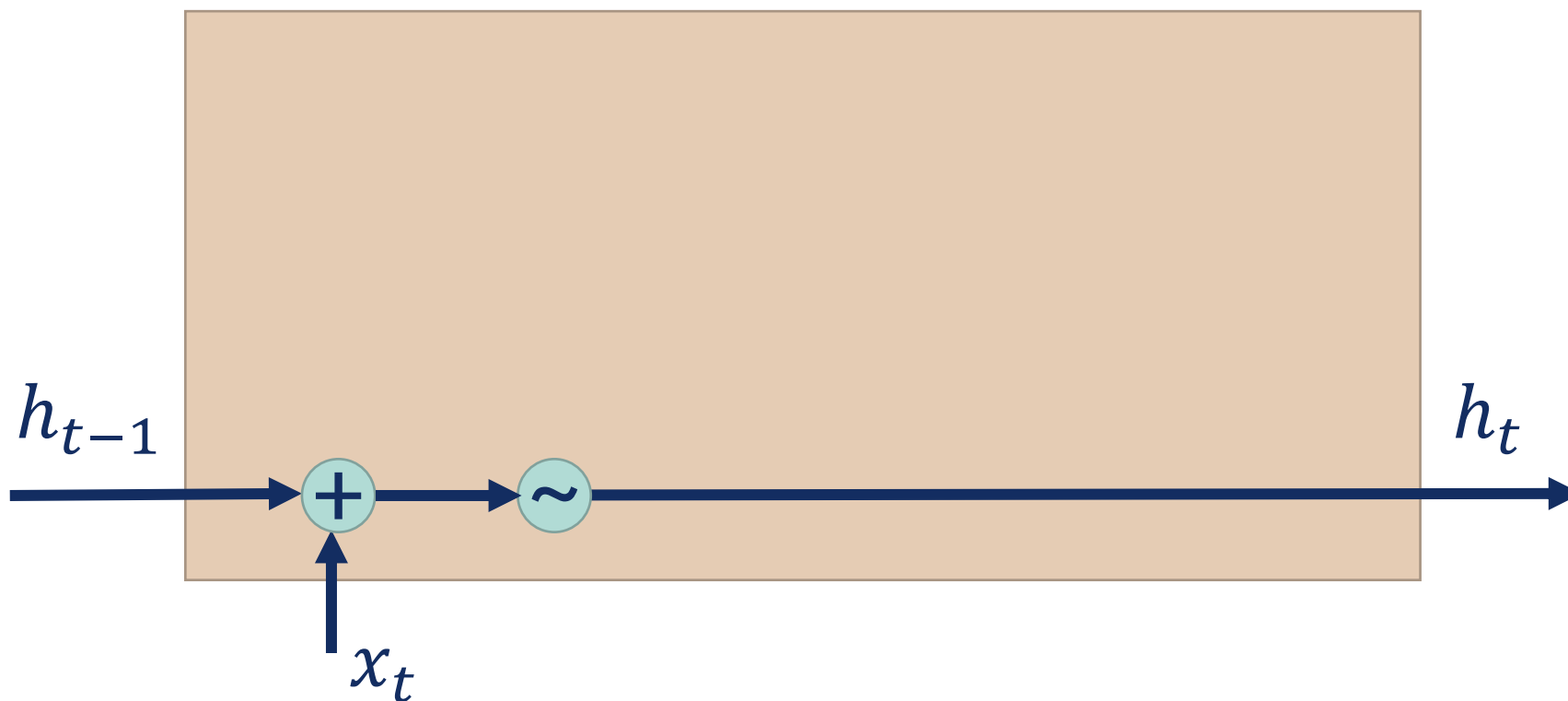
3. LSTM (Long-Short Term Memory)

Что будем модифицировать в этой секции



Вчера телефон перестал работать

Как было в RNN



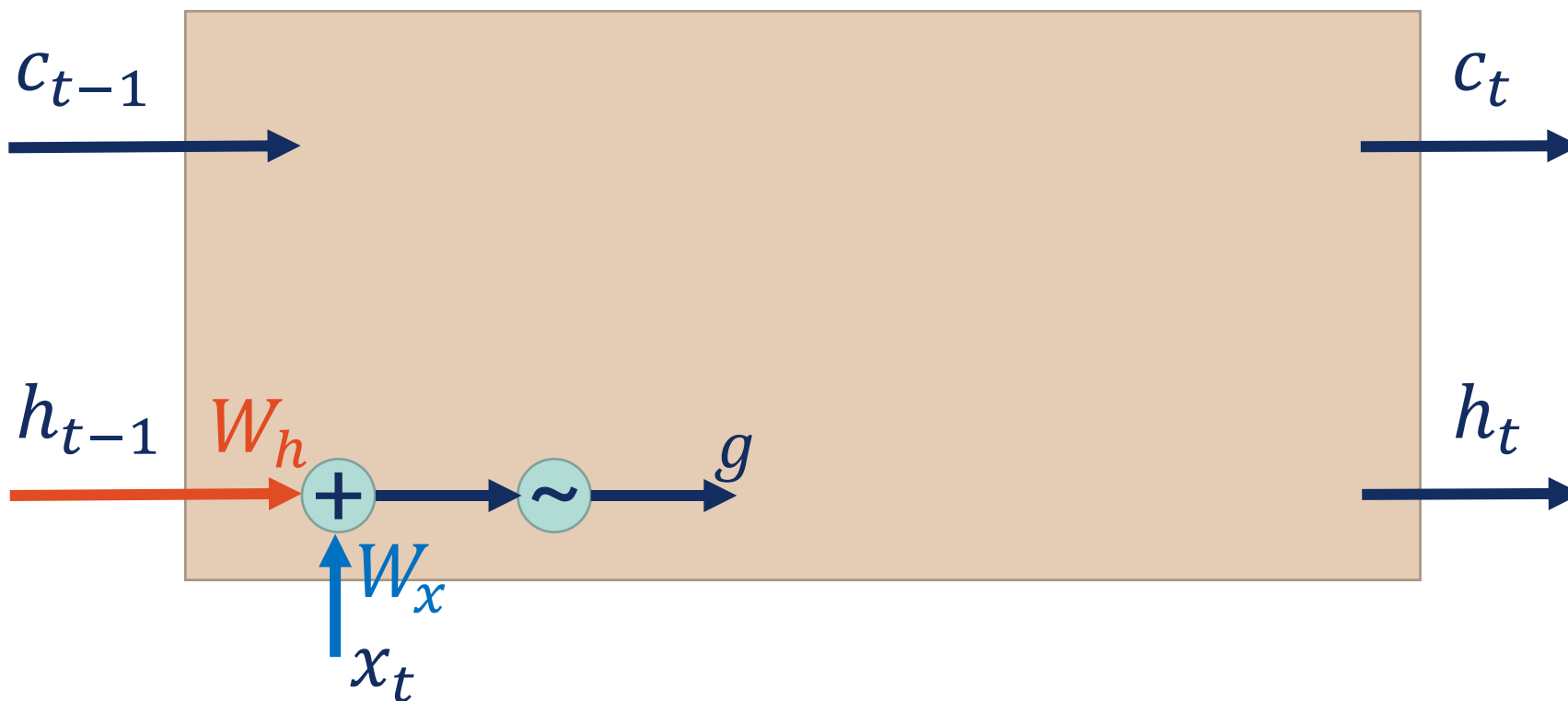
Добавляем память



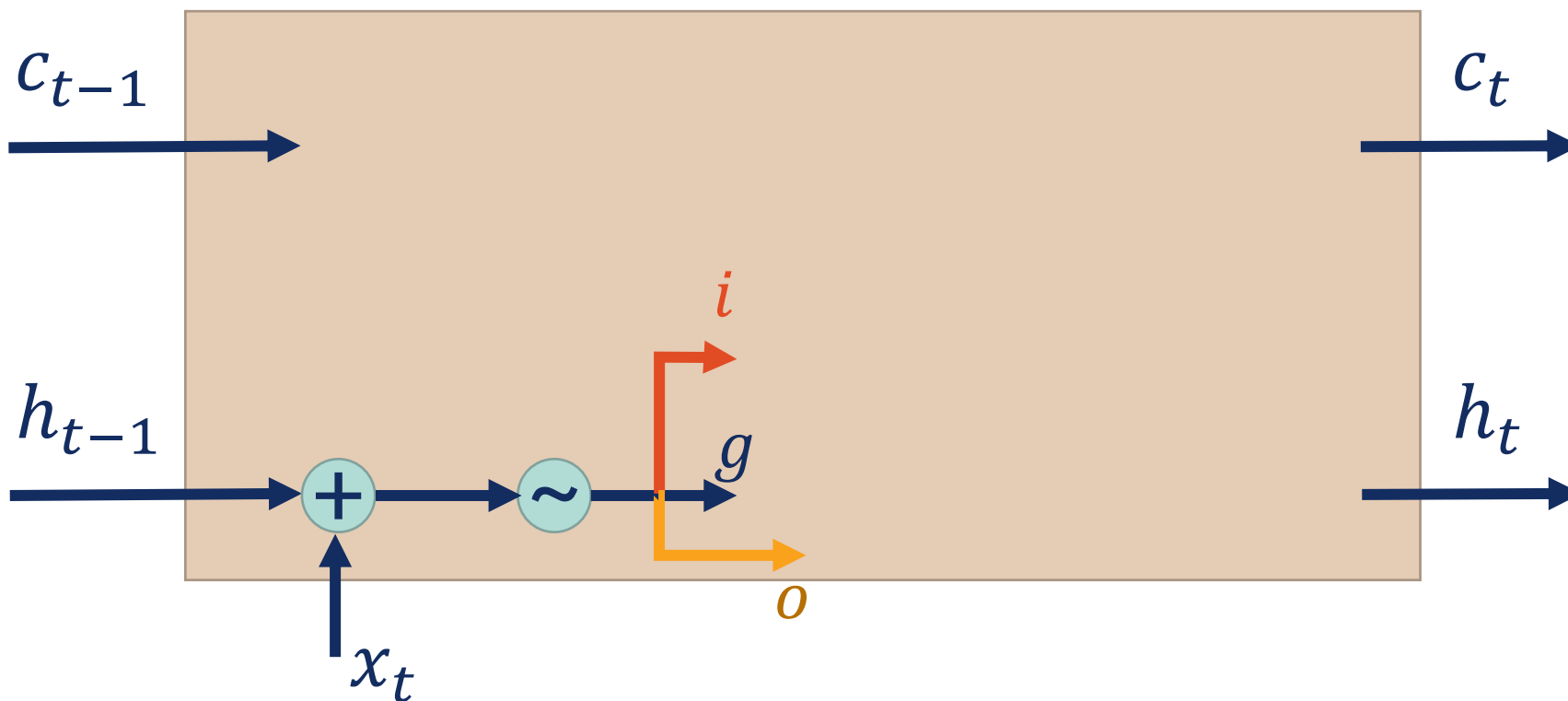
Добавляем память



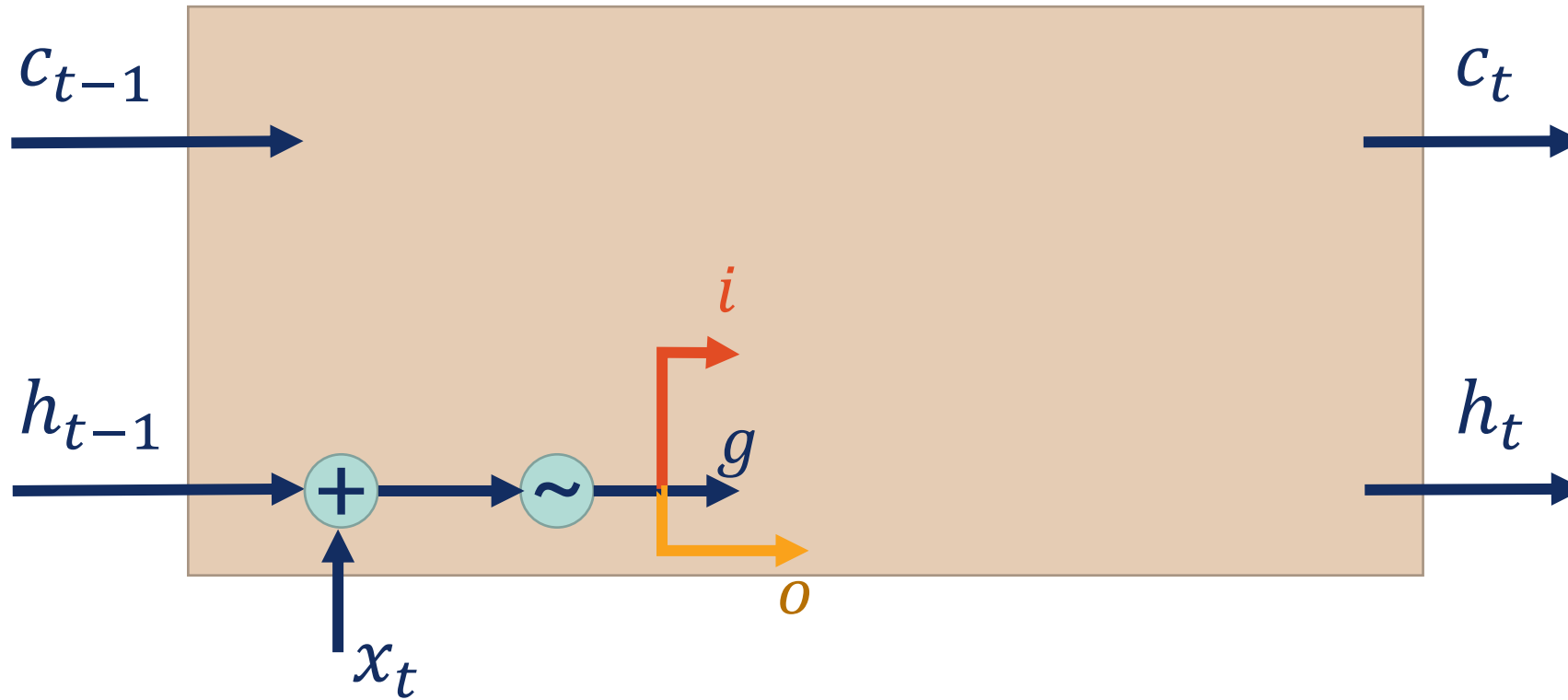
Добавляем память



Добавляем память



Добавляем память

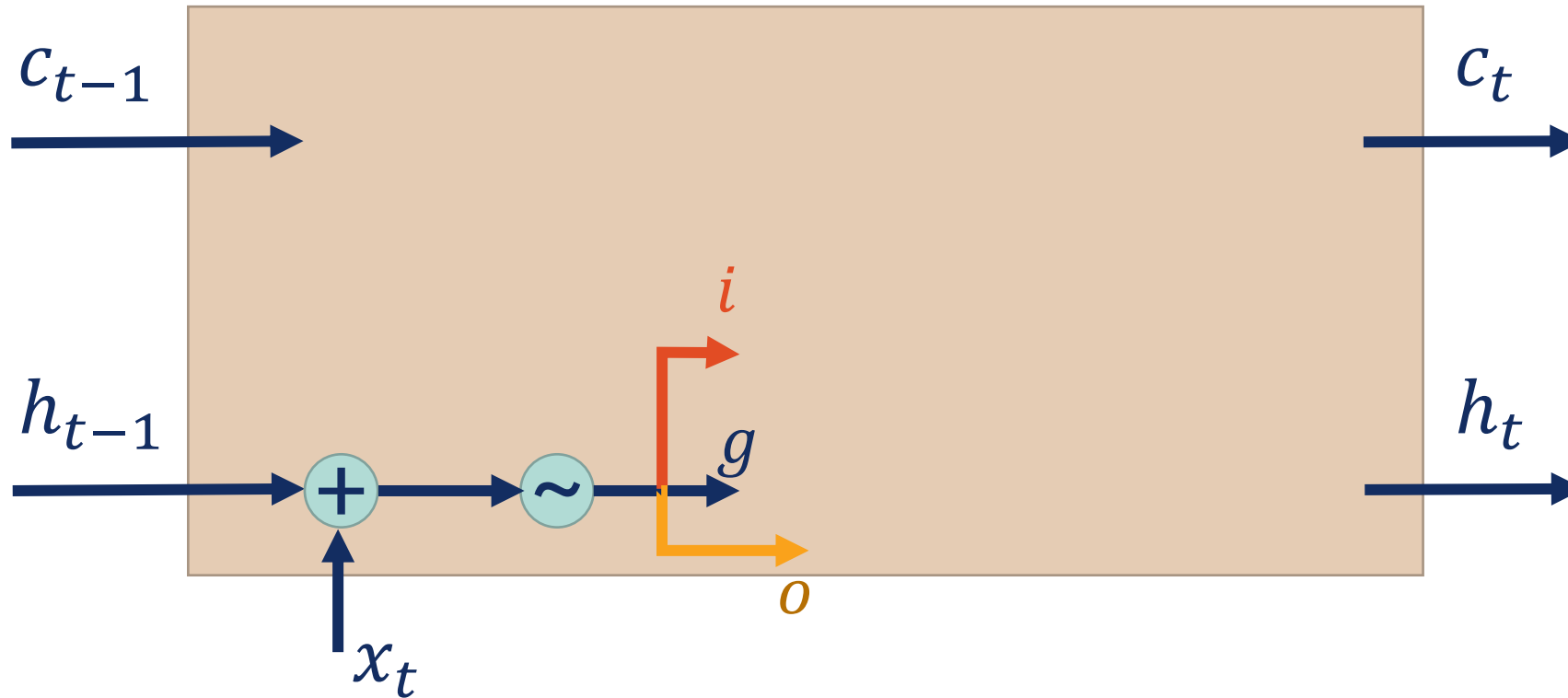


$$g_t = \varphi(W_{xg}x_t + W_{hg}h_{t-1} + b_g)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

Добавляем память



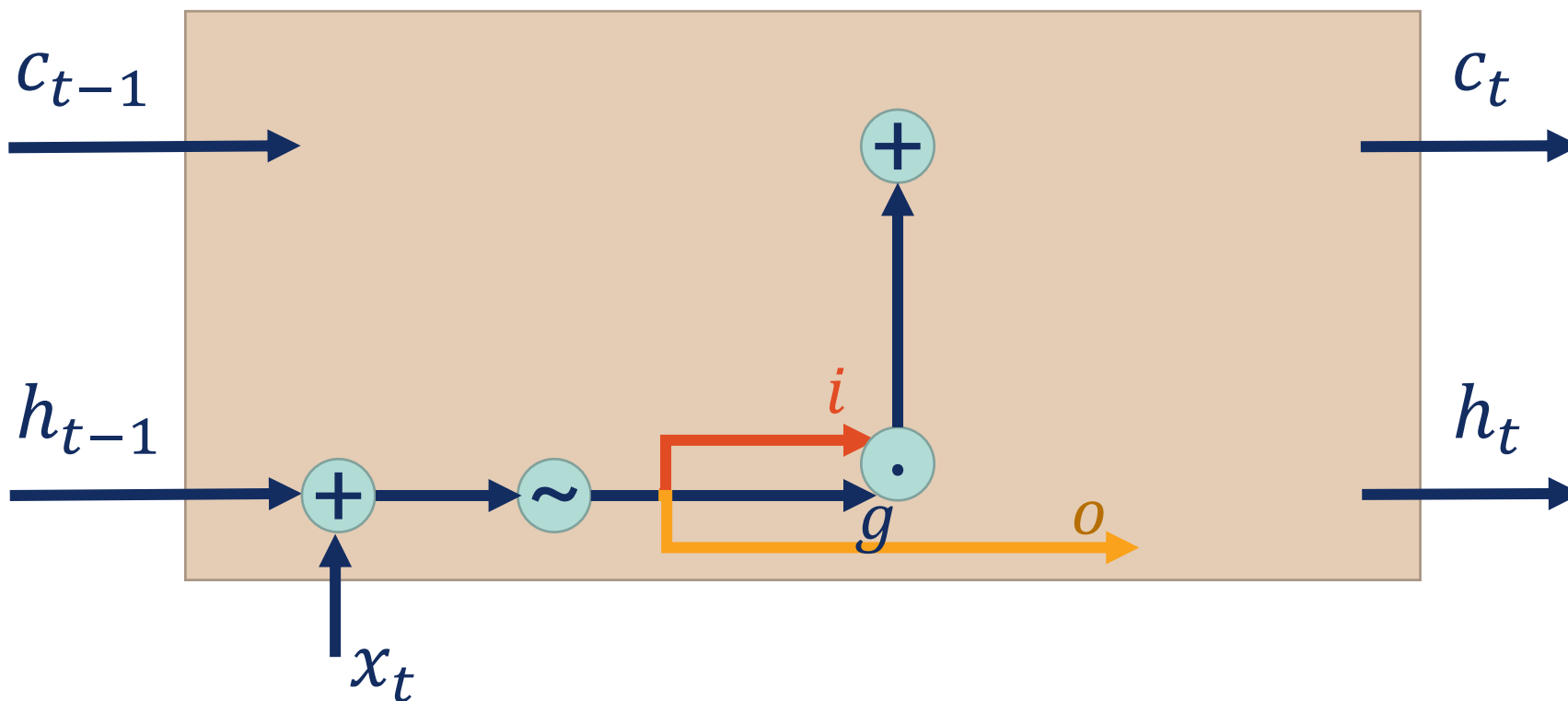
$$g_t = \varphi(W_{xg}x_t + W_{hg}h_{t-1} + b_g)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$

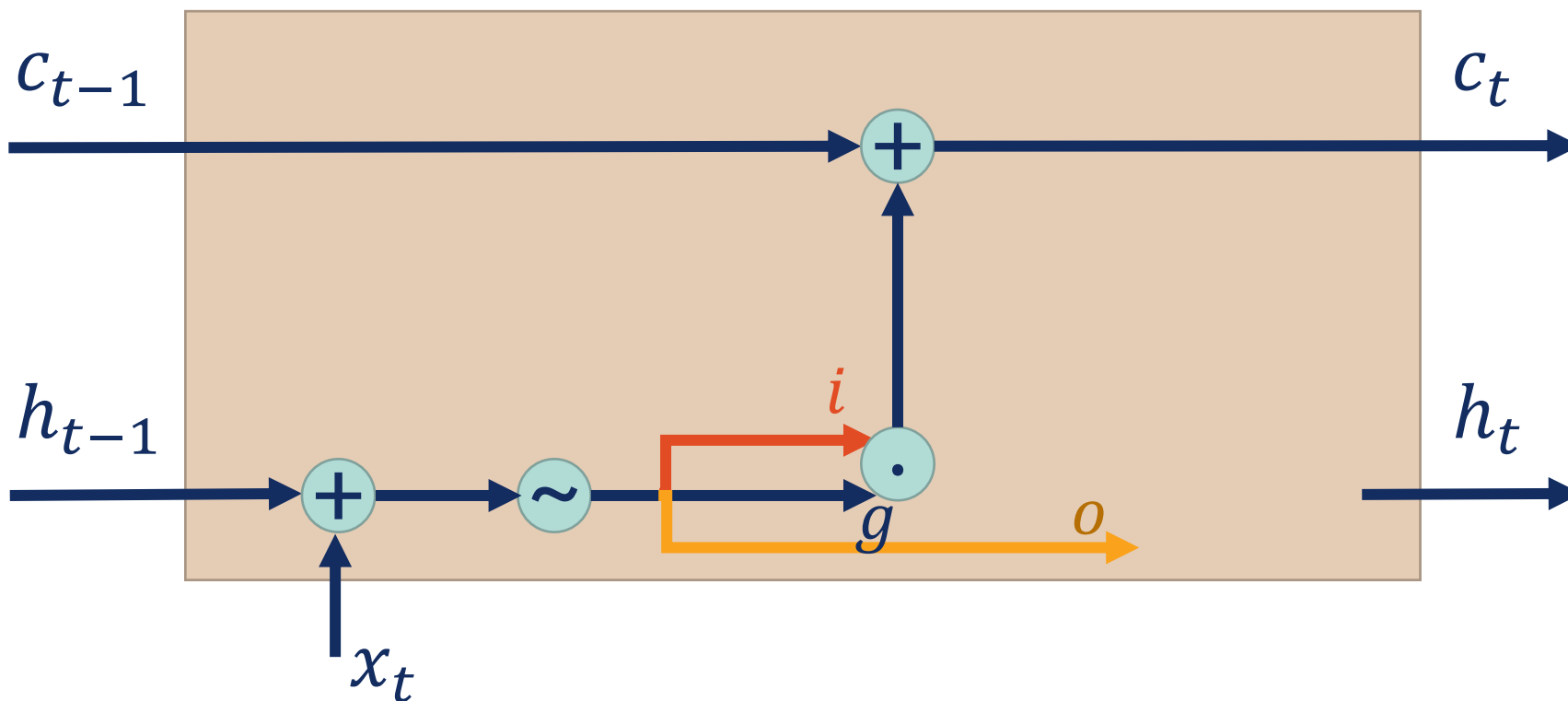
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

$$\begin{pmatrix} g_t \\ i_t \\ o_t \end{pmatrix} = \begin{pmatrix} \varphi \\ \sigma \\ \sigma \end{pmatrix} (W_x x_t + W_h h_{t-1} + b)$$

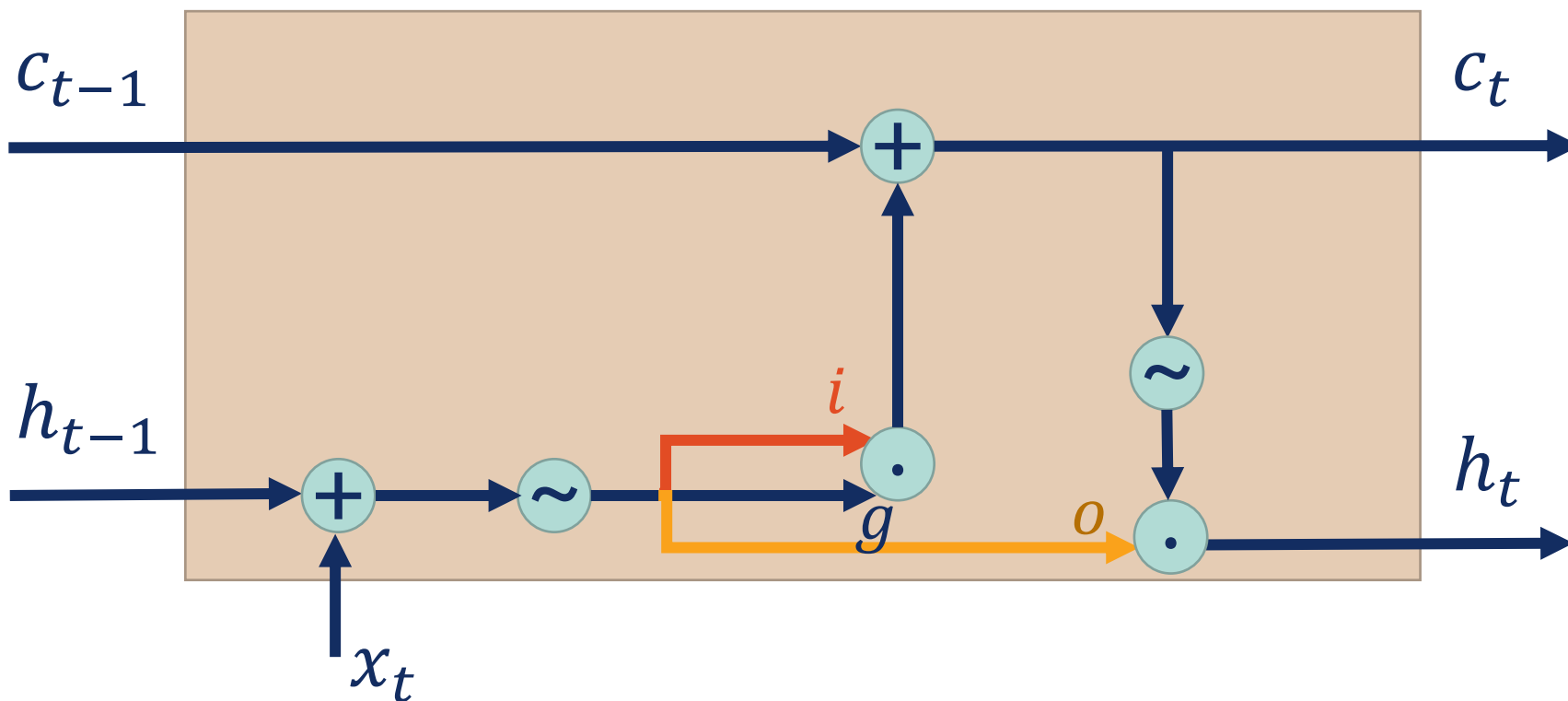
Добавляем память



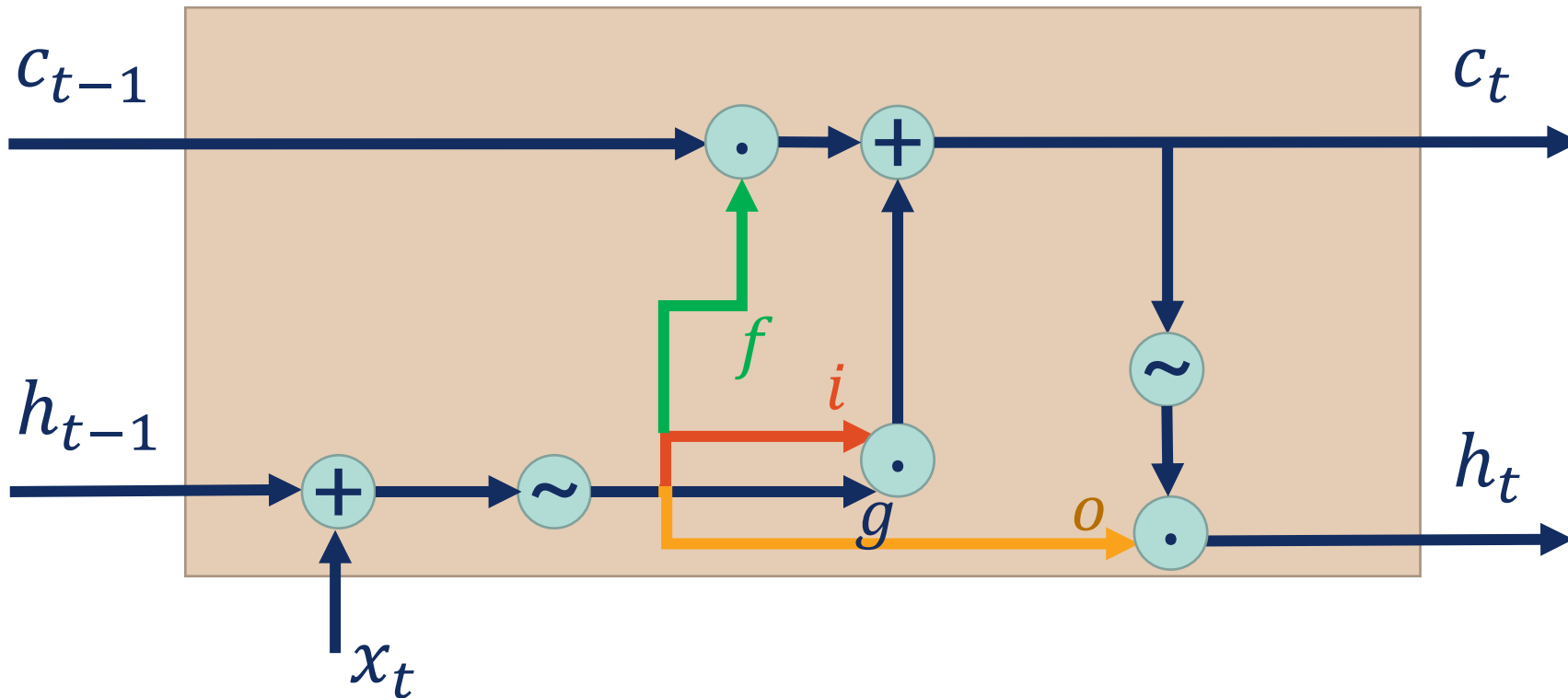
Добавляем память



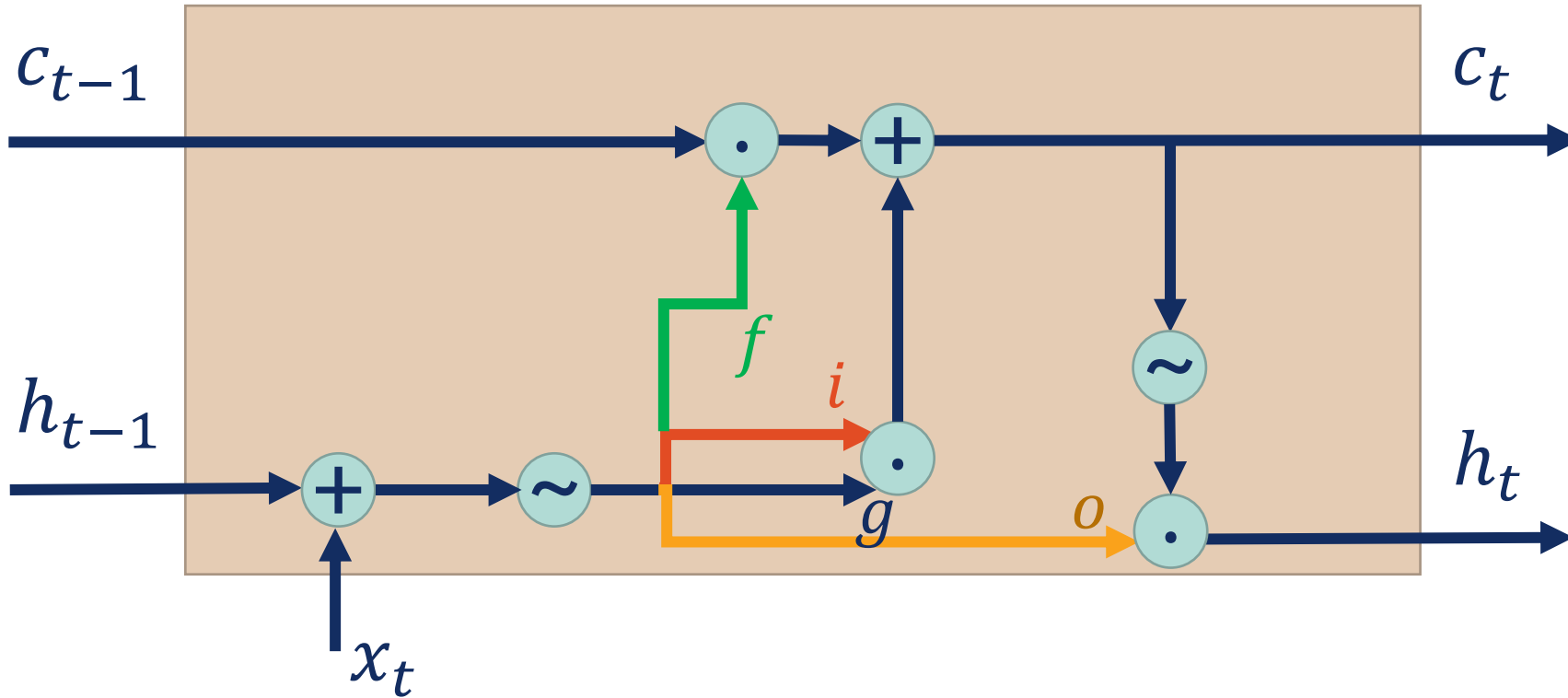
Почти LSTM: не хватает forget gate



Long-Short Term Memory (LSTM)



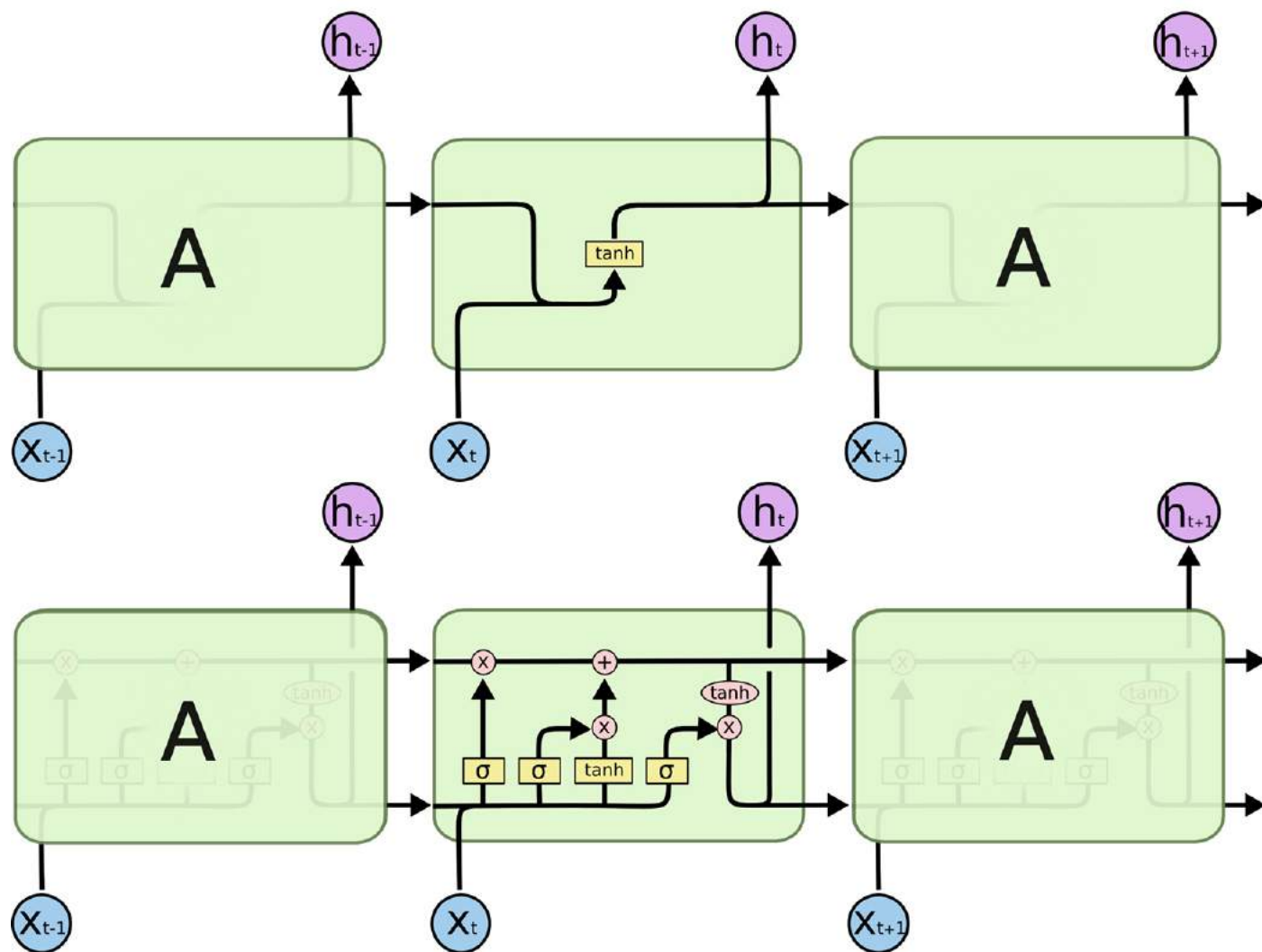
Long-Short Term Memory (LSTM)



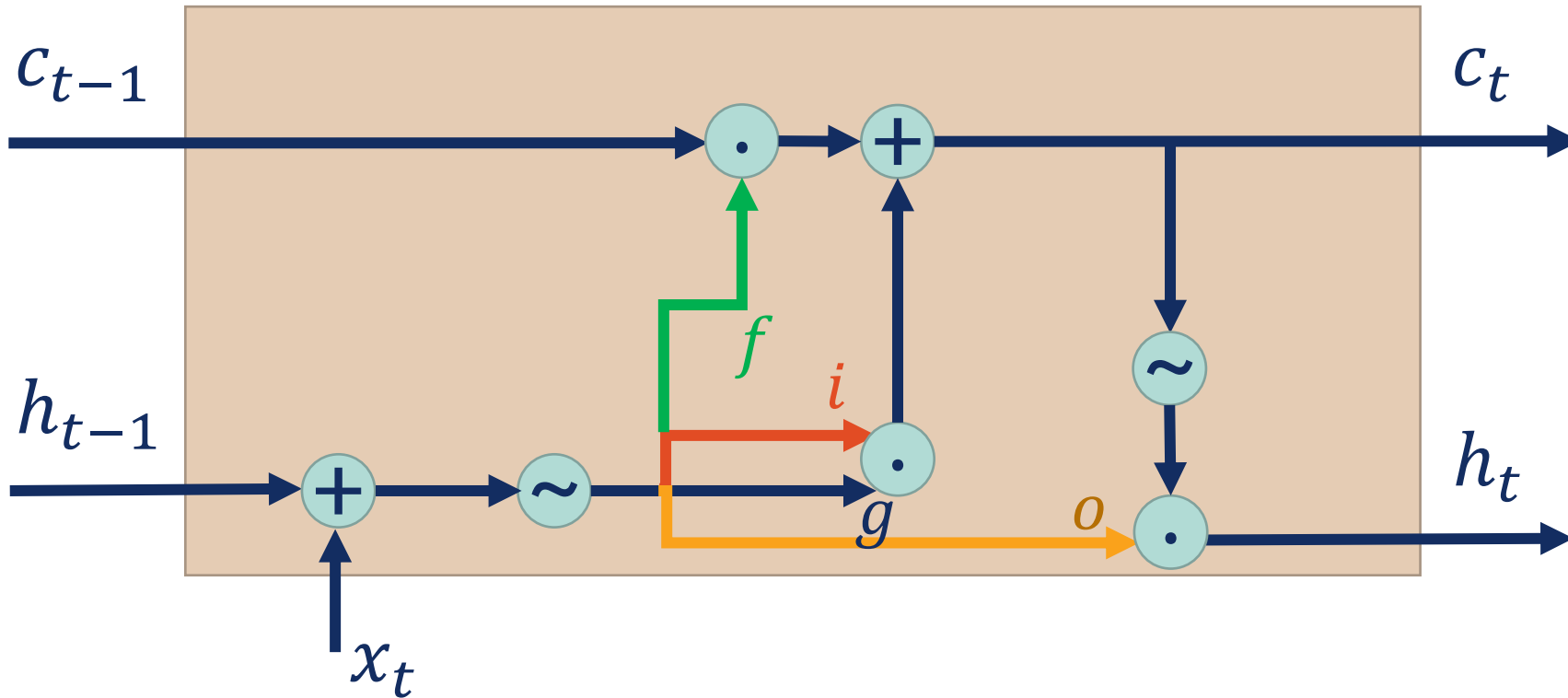
$$\begin{pmatrix} g_t \\ i_t \\ o_t \\ f_t \end{pmatrix} = \begin{pmatrix} \varphi \\ \sigma \\ \sigma \\ \sigma \end{pmatrix} (W_x x_t + W_h h_{t-1} + b)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t$$
$$h_t = o_t \cdot \phi(c_t)$$

Другая иллюстрация к RNN и LSTM



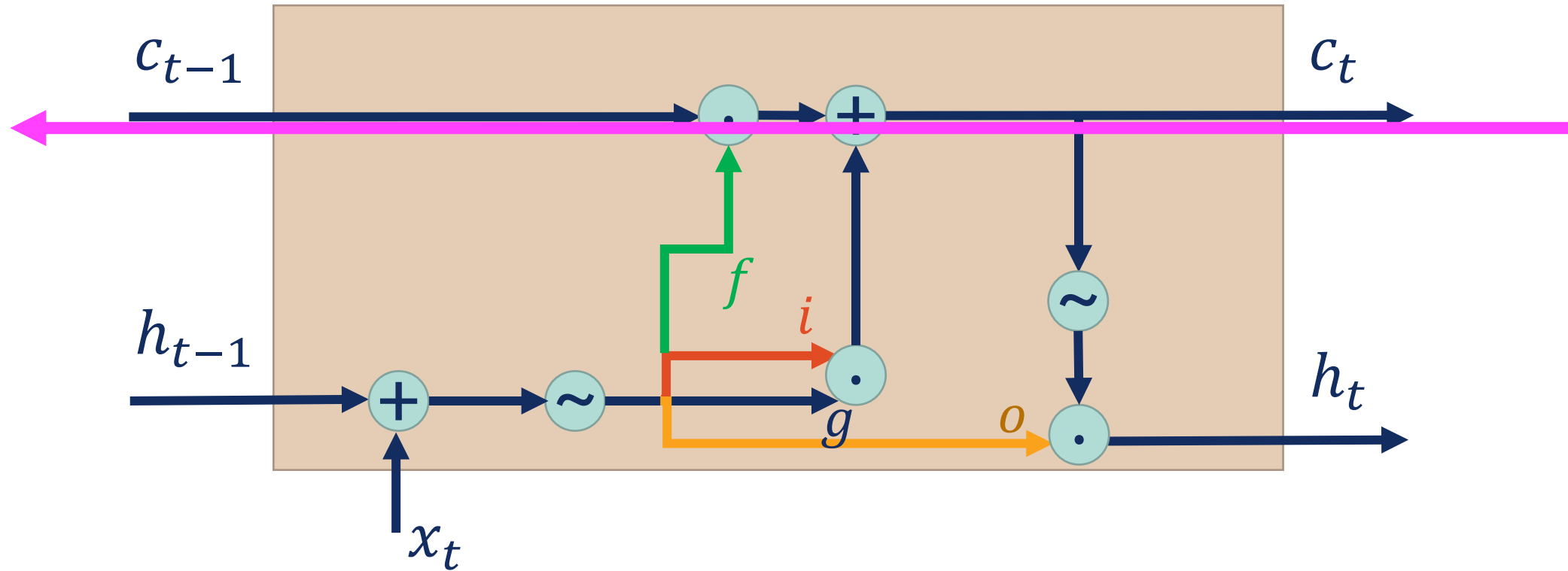
Как решили проблему затухания градиента?



$$\begin{pmatrix} g_t \\ i_t \\ o_t \\ f_t \end{pmatrix} = \begin{pmatrix} \varphi \\ \sigma \\ \sigma \\ \sigma \end{pmatrix} (W_x x_t + W_h h_{t-1} + b)$$

$$\begin{aligned} c_t &= f_t \cdot c_{t-1} + i_t \cdot g_t \\ h_t &= o_t \cdot \phi(c_t) \end{aligned}$$

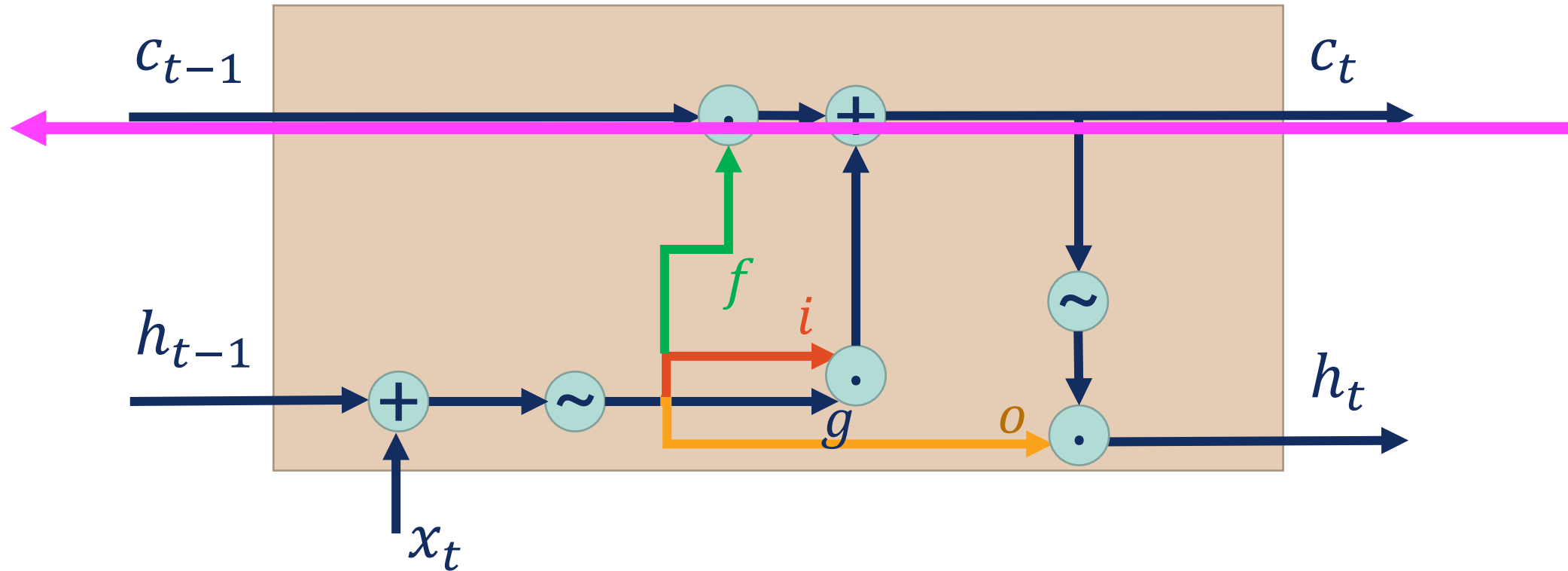
Как решили проблему затухания градиента?



$$\begin{pmatrix} g_t \\ i_t \\ o_t \\ f_t \end{pmatrix} = \begin{pmatrix} \varphi \\ \sigma \\ \sigma \\ \sigma \end{pmatrix} (W_x x_t + W_h h_{t-1} + b)$$

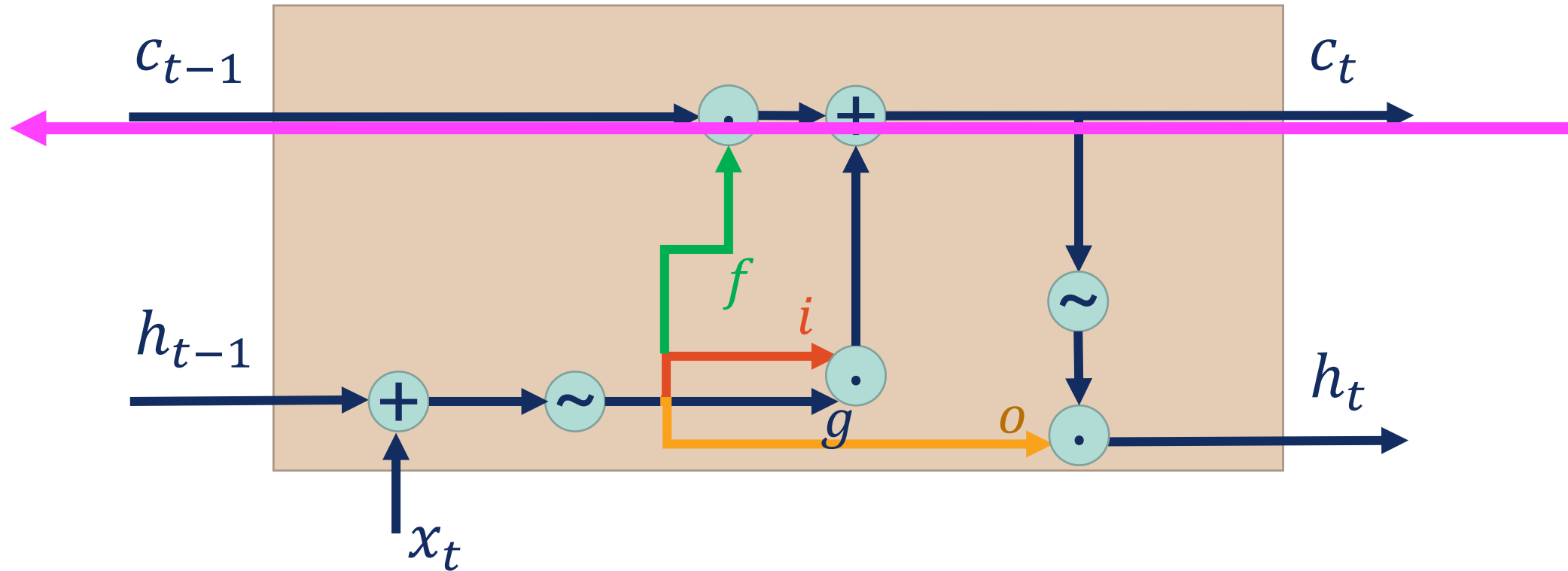
$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t$$
$$h_t = o_t \cdot \phi(c_t)$$

Как решили проблему затухания градиента?



Но нужно, чтобы хотя бы в начале forget gate был открыт.
Вопрос: как этого добиться?

Как решили проблему затухания градиента?



Но нужно, чтобы хотя бы в начале forget gate был открыт.
Вопрос: как этого добиться? (b_f)

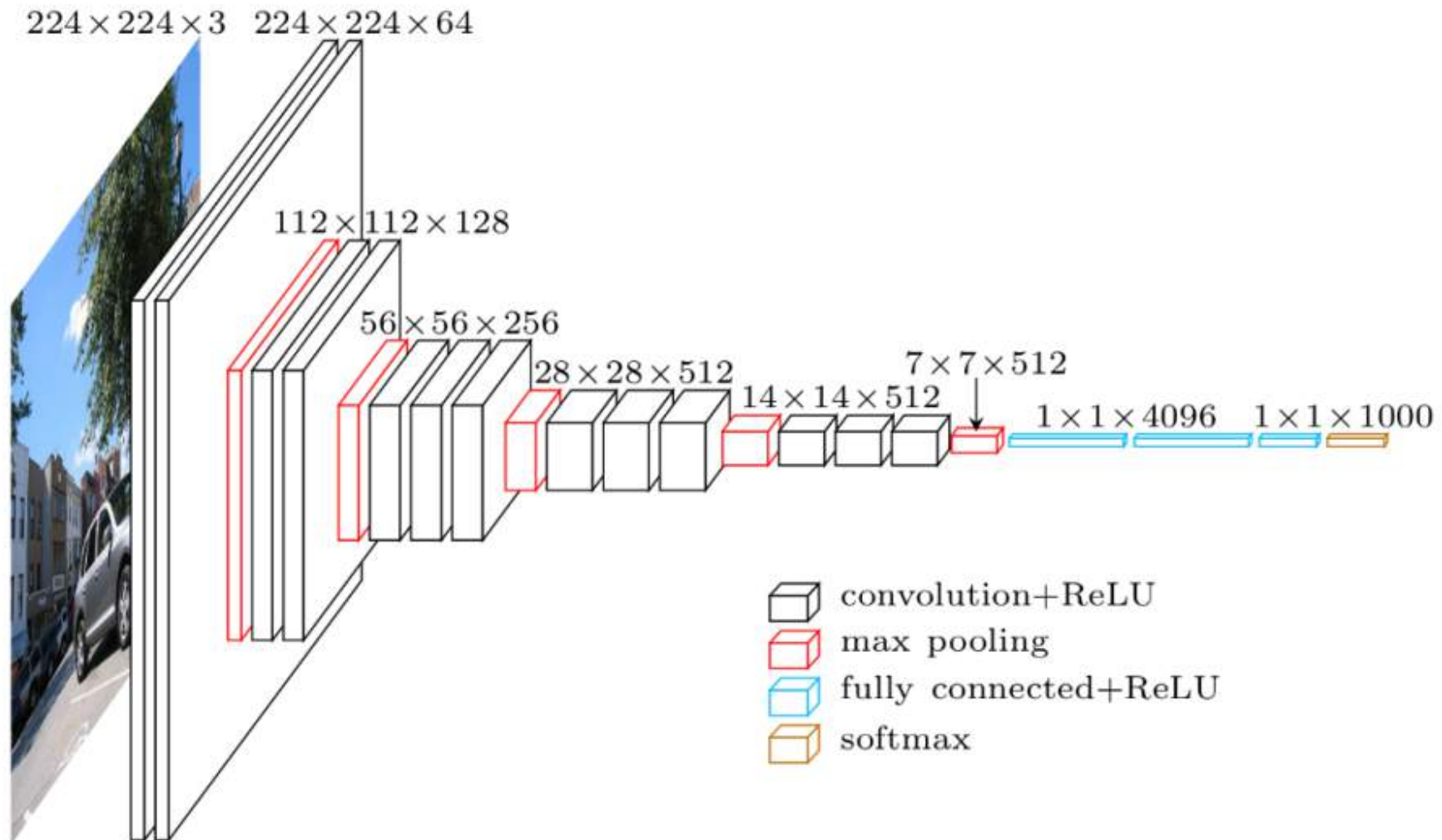
Рекуррентные слои

На практике чаще всего используется для:

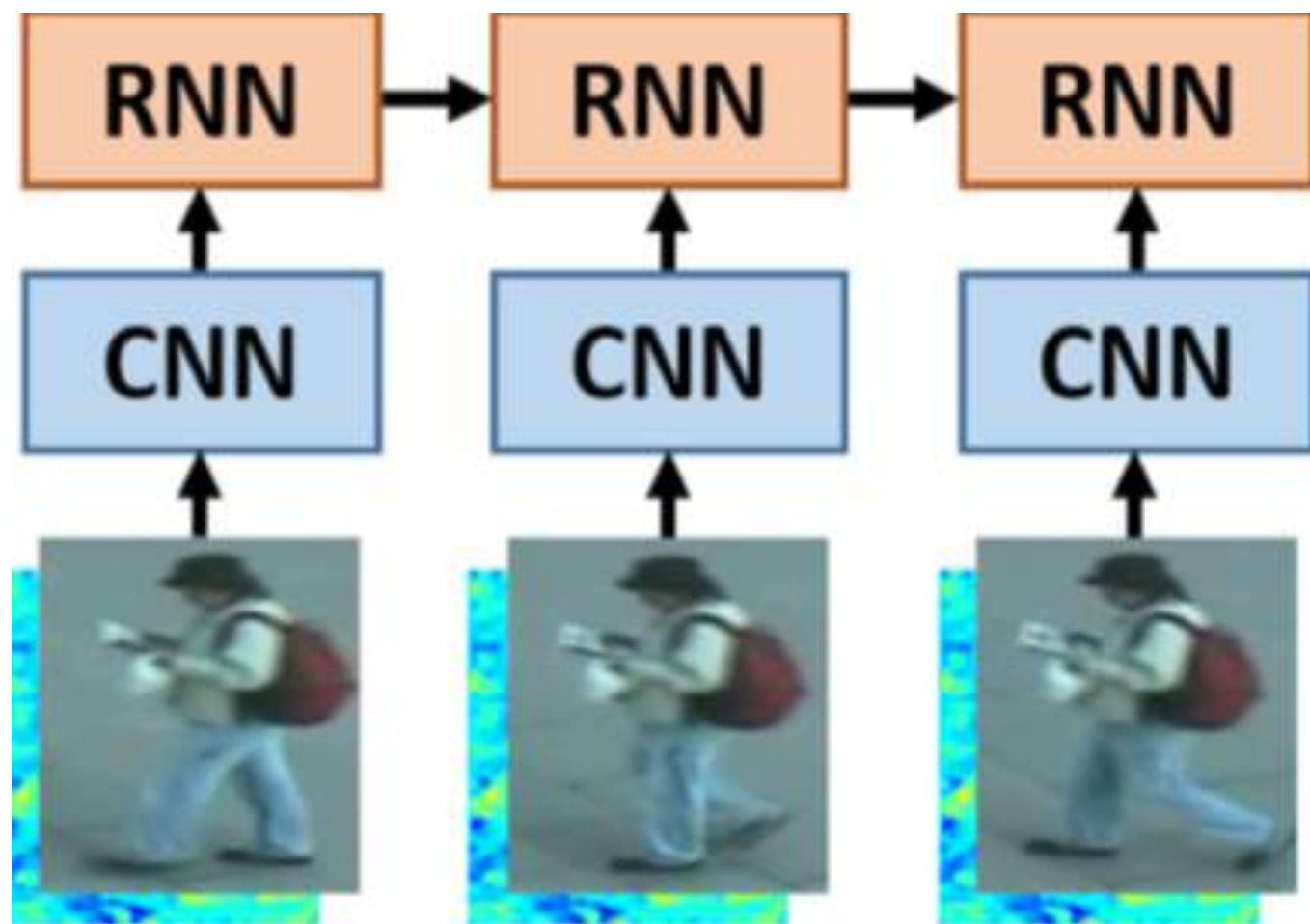
- анализа временных рядов
- анализа текстов
- анализа аудио
- все благодаря возможности работать с последовательностями не только в ключе нахождения паттернов внутри фиксированного окна

4. Применение блоков

Conv – Pool – Repeat x n – Dense



Рекуррентный слой после сверточного

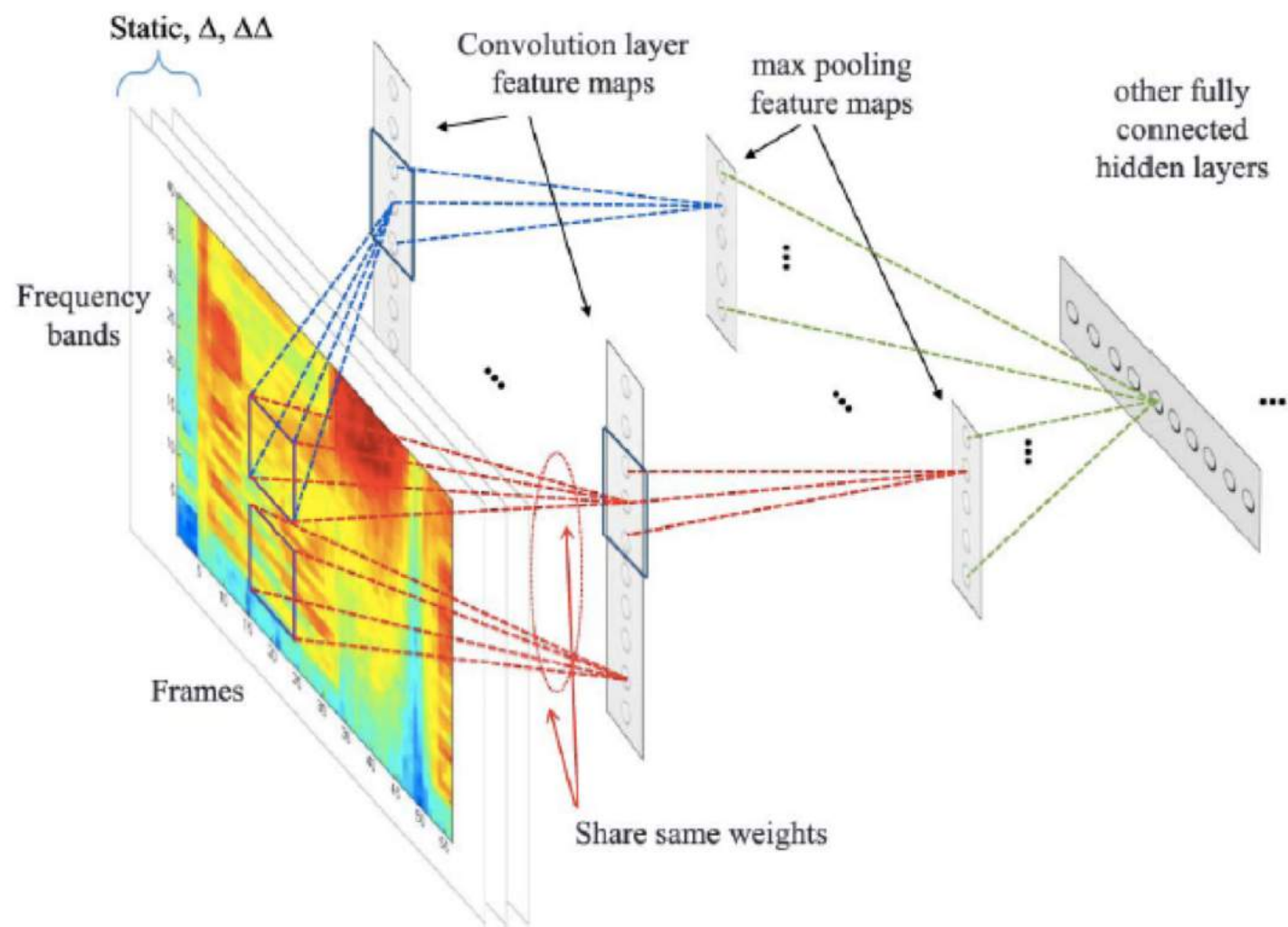


Распознавание речи

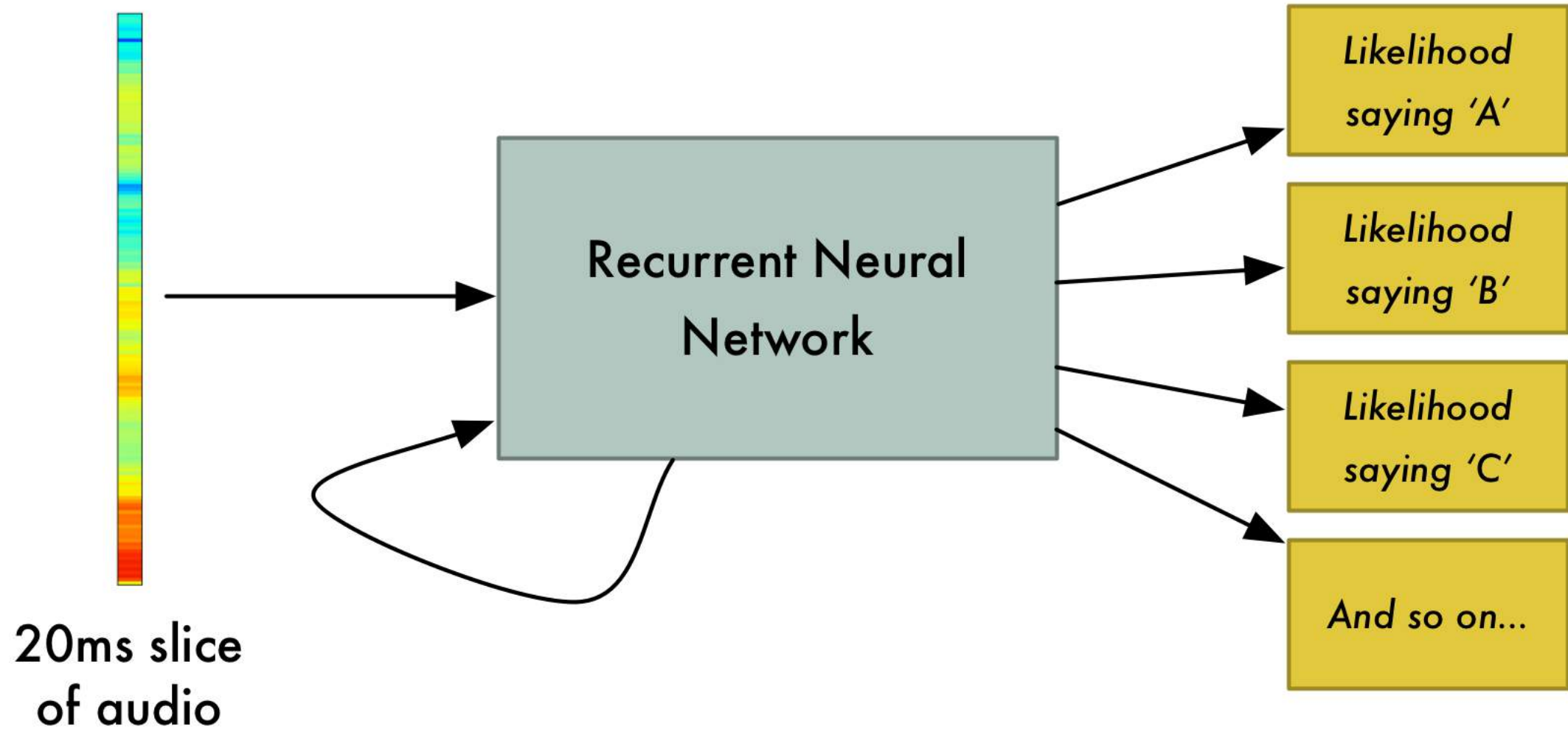
Сверточные или рекуррентные сети?

И то и другое!

Распознавание речи: сверточные сети



Распознавание речи: рекуррентные сети

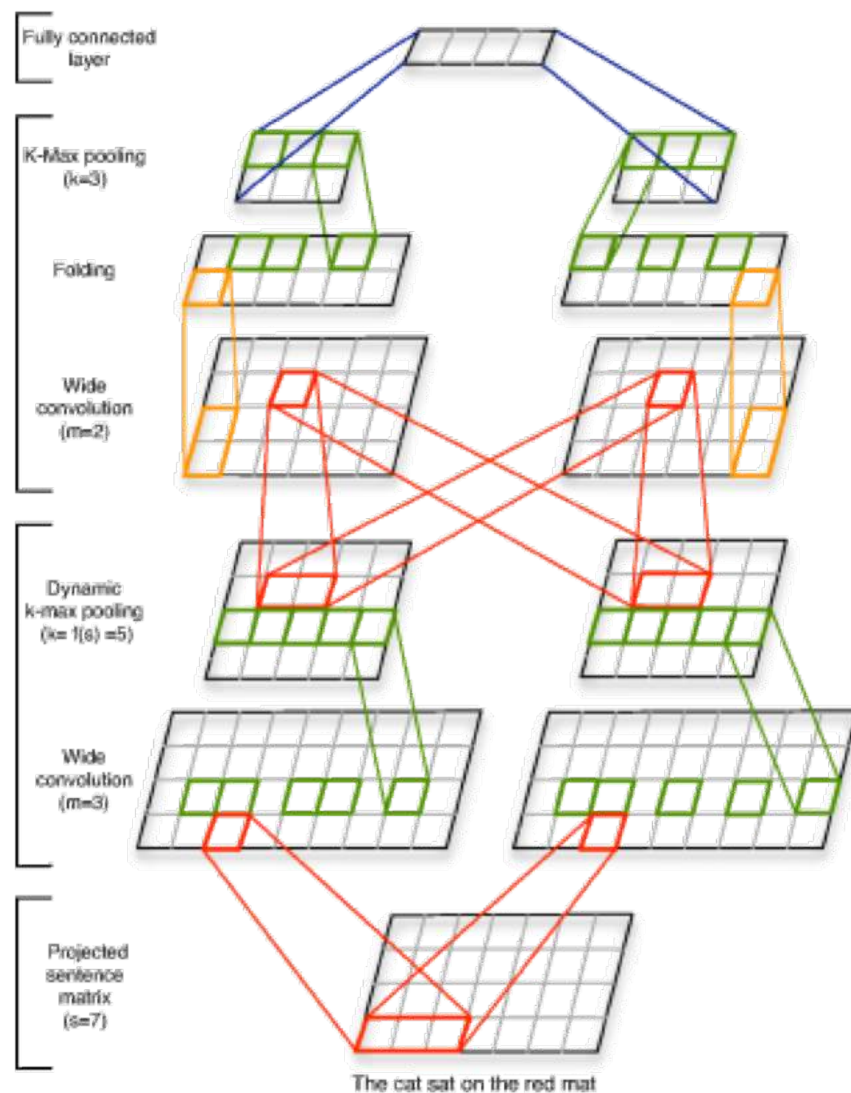


Классификация текста

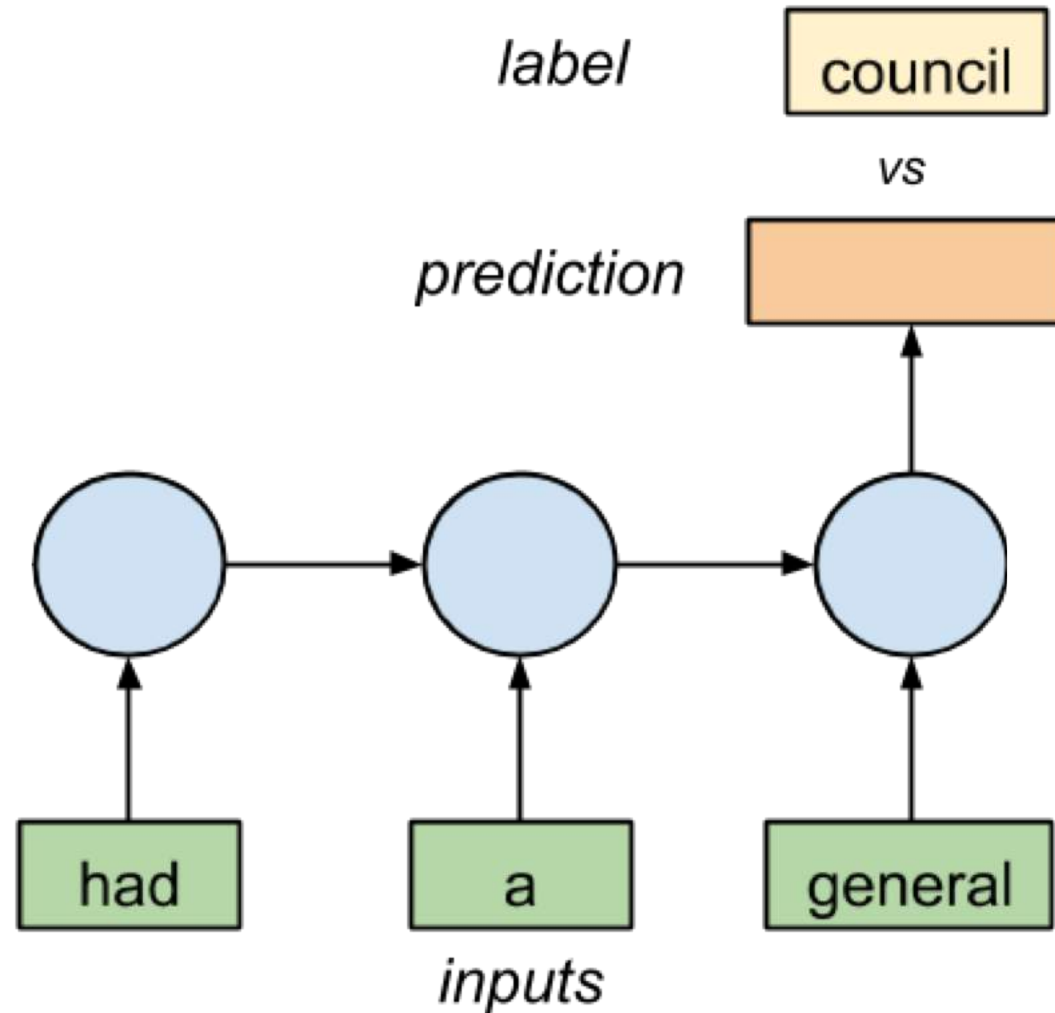
Сверточные или рекуррентные сети?

И то и другое!

Классификация текстов: сверточные сети



Классификация текстов: рекуррентные сети



Классификация изображений

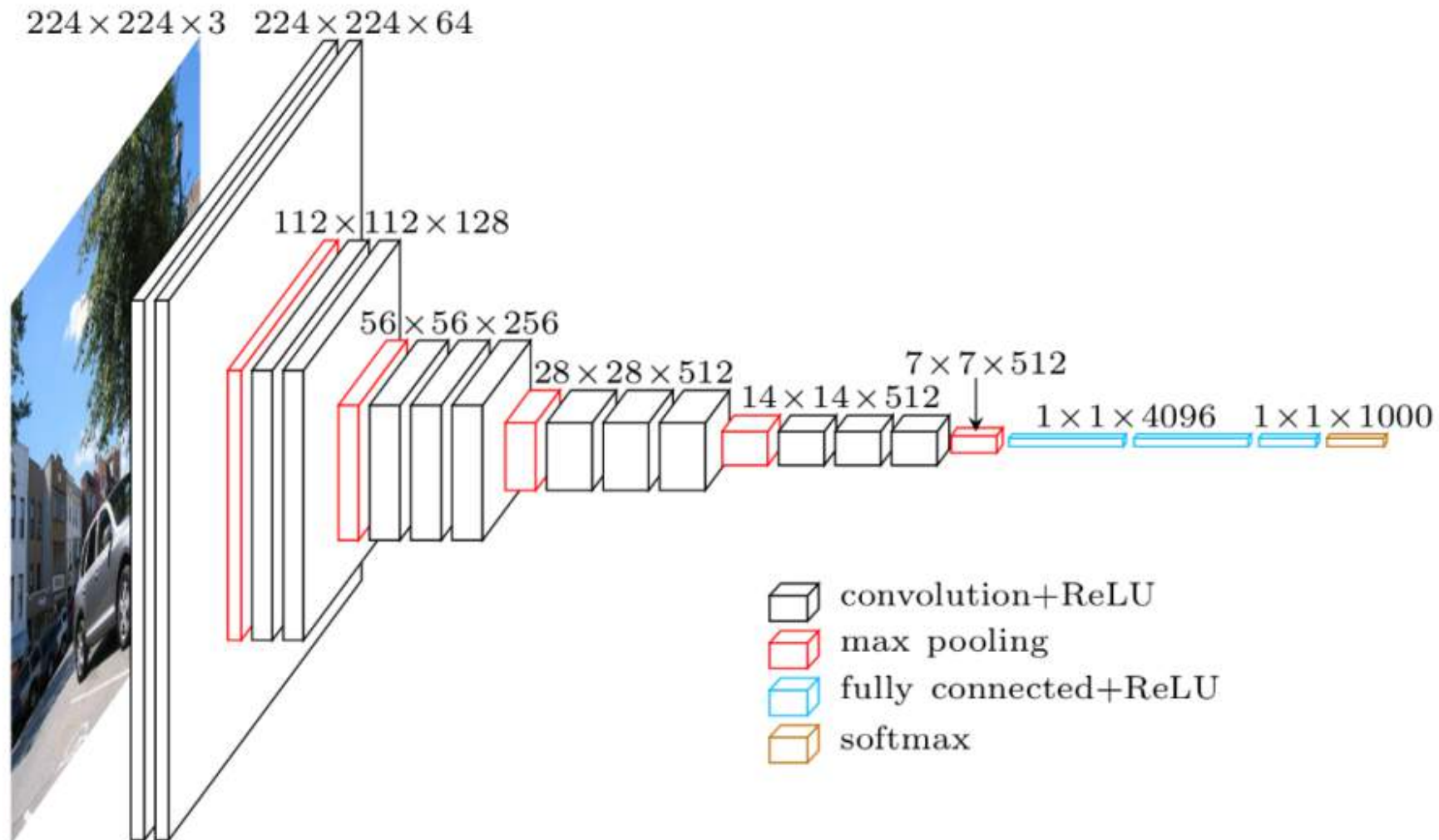
Сверточные или рекуррентные?

Классификация изображений

Сверточные или рекуррентные?

В основном, сверточные

Классификация изображений: сверточные сети



План

1. Свёрточные сети

2. Рекуррентные сети

3. Затухание градиента и LSTM

4. Применение блоков

Data Mining in Action

Группа направления «Глубокое обучение» в Telegram:



<https://t.me/joinchat/B1OlTkodHlbbT6QEmlz5Xw>