Deep Learning School

Рекуррентная нейронная сеть

План занятия

- Рекуррентный слой: идея. Рекуррентная нейросеть (RNN);
- Forward pass RNN;
- Обучение RNN (backward pass);
- Функции активации в RNN;
- Bidirectional RNN;
- GRU, LSTM

План занятия

- **Рекуррентный слой: идея. Рекуррентная нейросеть (RNN);**
- **Forward pass RNN;**
- Обучение RNN (backward pass);
- Функции активации в RNN;
- Bidirectional RNN;
- GRU, LSTM

Особенность текста и звука

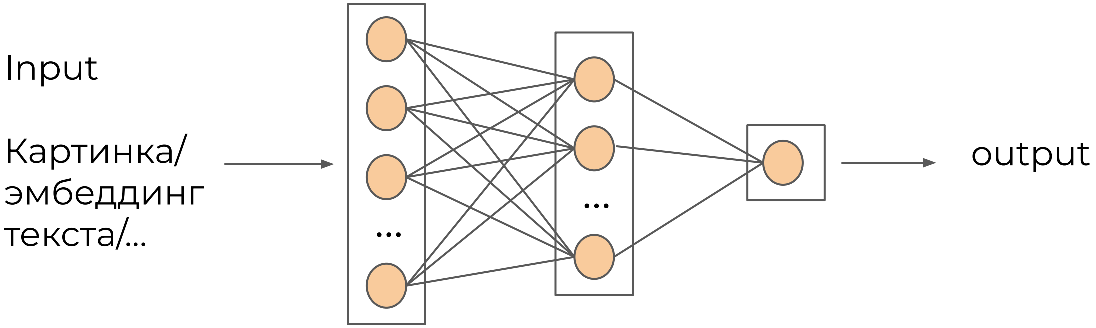
Отличие текста и звука от других типов данных (например, изображений) состоит в наличии временной компоненты.

Мы читаем текст не моментально, а слово за словом, в строго определенном порядке.

Возникает идея придумать идею нейросети, которая учитывала бы эту особенность этих типов данных.

Рекуррентный слой

Как работает обычная нейросеть:

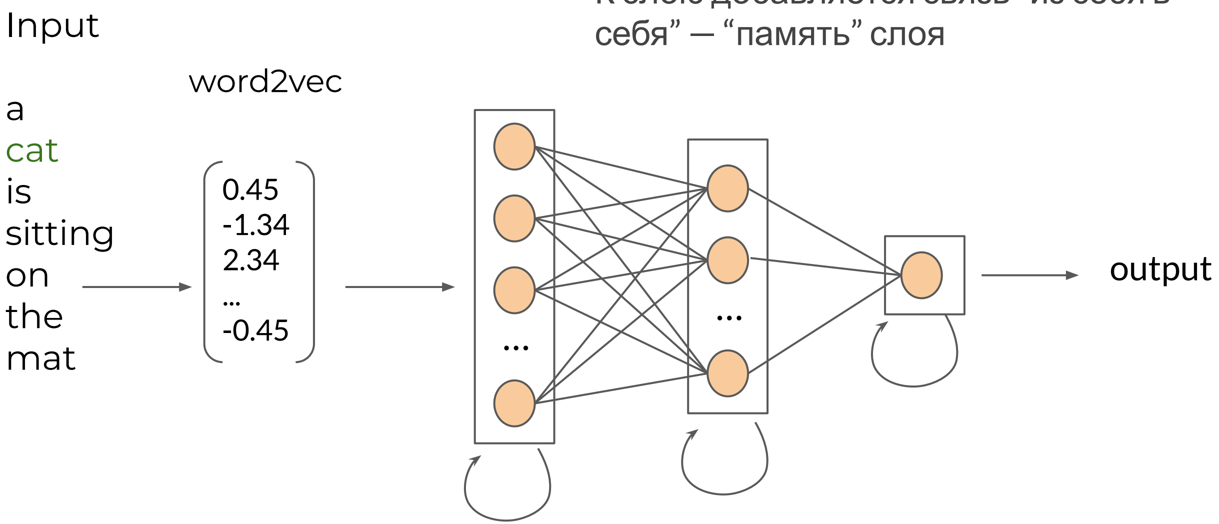


Рекуррентный слой

Как работает рекуррентная нейросеть:

Рекуррентная нейросеть обрабатывает один токен текста за один момент времени.

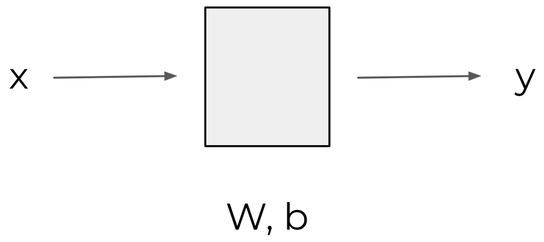
К слою добавляется связь “из себя в себя” — “память” слоя



Рекуррентный слой

Полносвязный слой

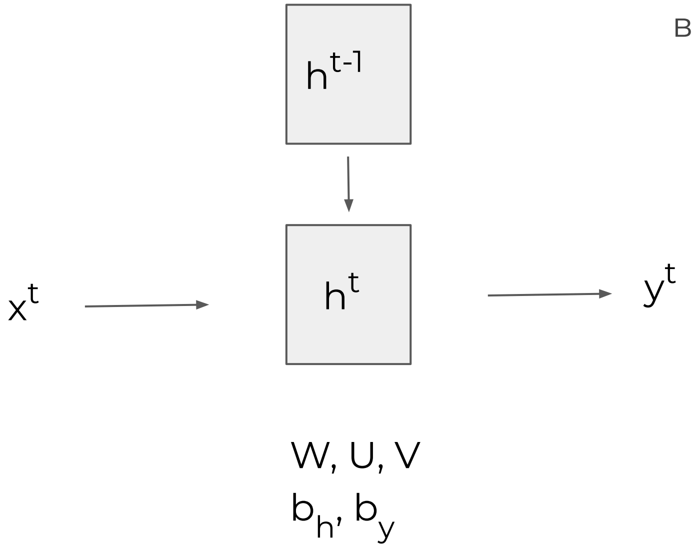
Вычисление выхода слоя:



$$y = \sigma(WX + b)$$

Рекуррентный слой

Рекуррентный слой



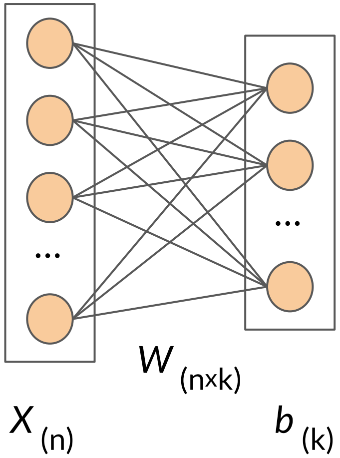
Обновление вектора скрытого состояния и вычисление выхода слоя:

$$\boldsymbol{h}^t = \sigma\left(\boldsymbol{W}\boldsymbol{X}^t + \boldsymbol{U}\boldsymbol{h}^{t-1} + \boldsymbol{b}_h\right)$$

$$\boldsymbol{y} = \sigma\left(\boldsymbol{V}\boldsymbol{h}^t + \boldsymbol{b}_y\right)$$

Рекуррентный слой

Слой полносвязной нейросети



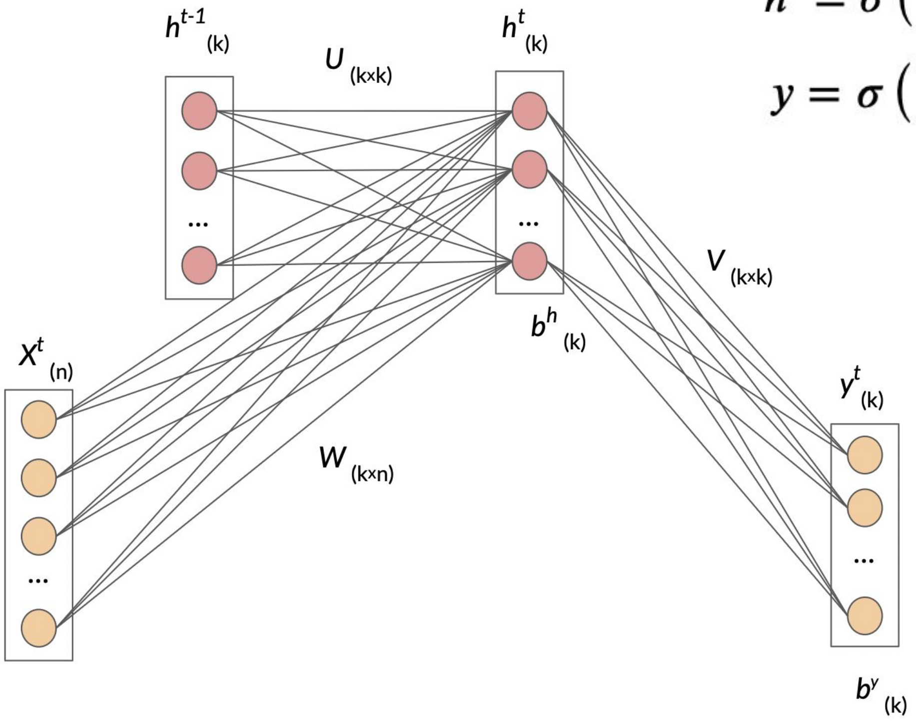
$$y = \sigma(WX + b)$$

Слой рекуррентной нейросети

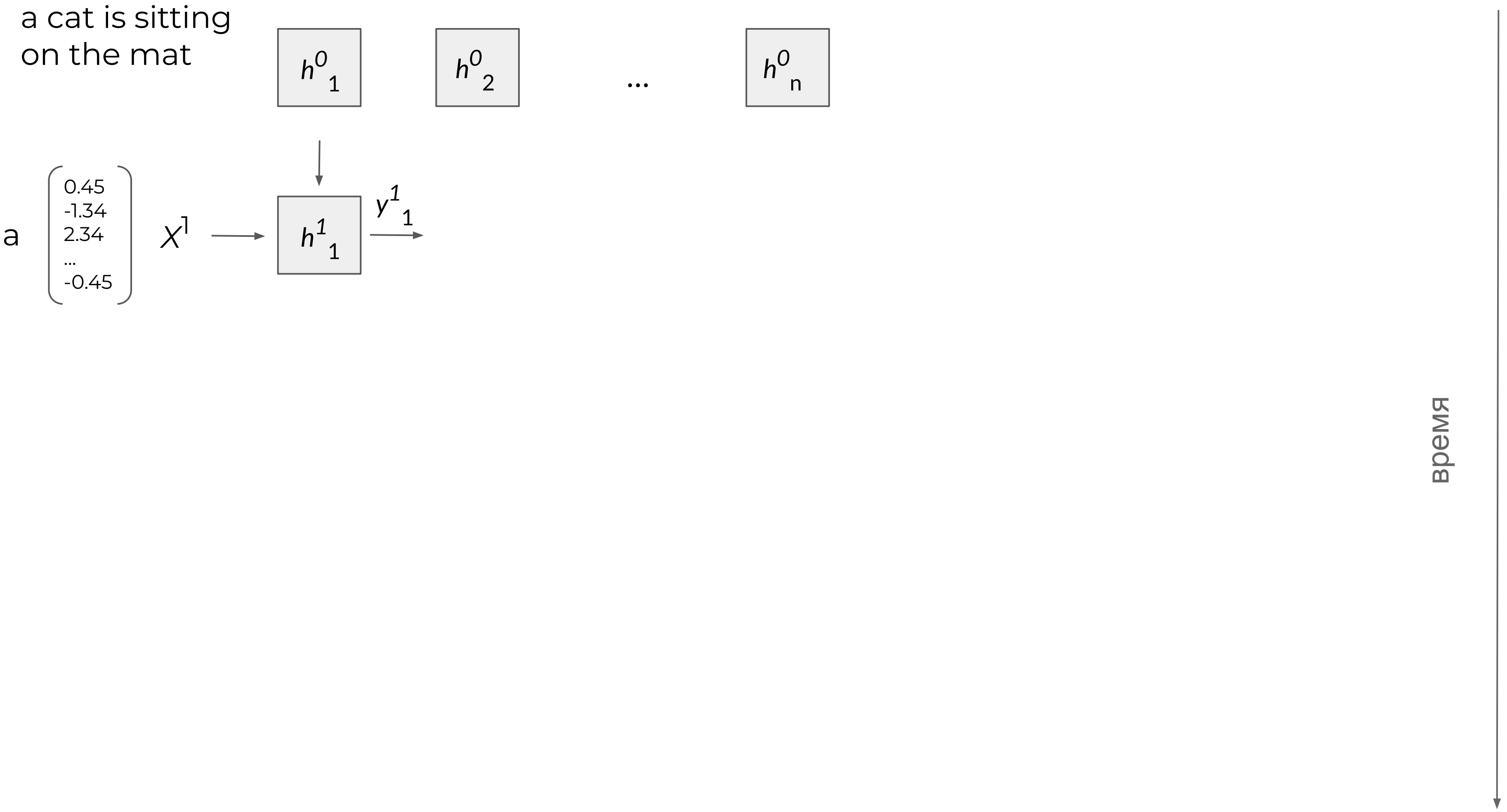
Обновление вектора скрытого состояния и вычисление выхода слоя:

$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

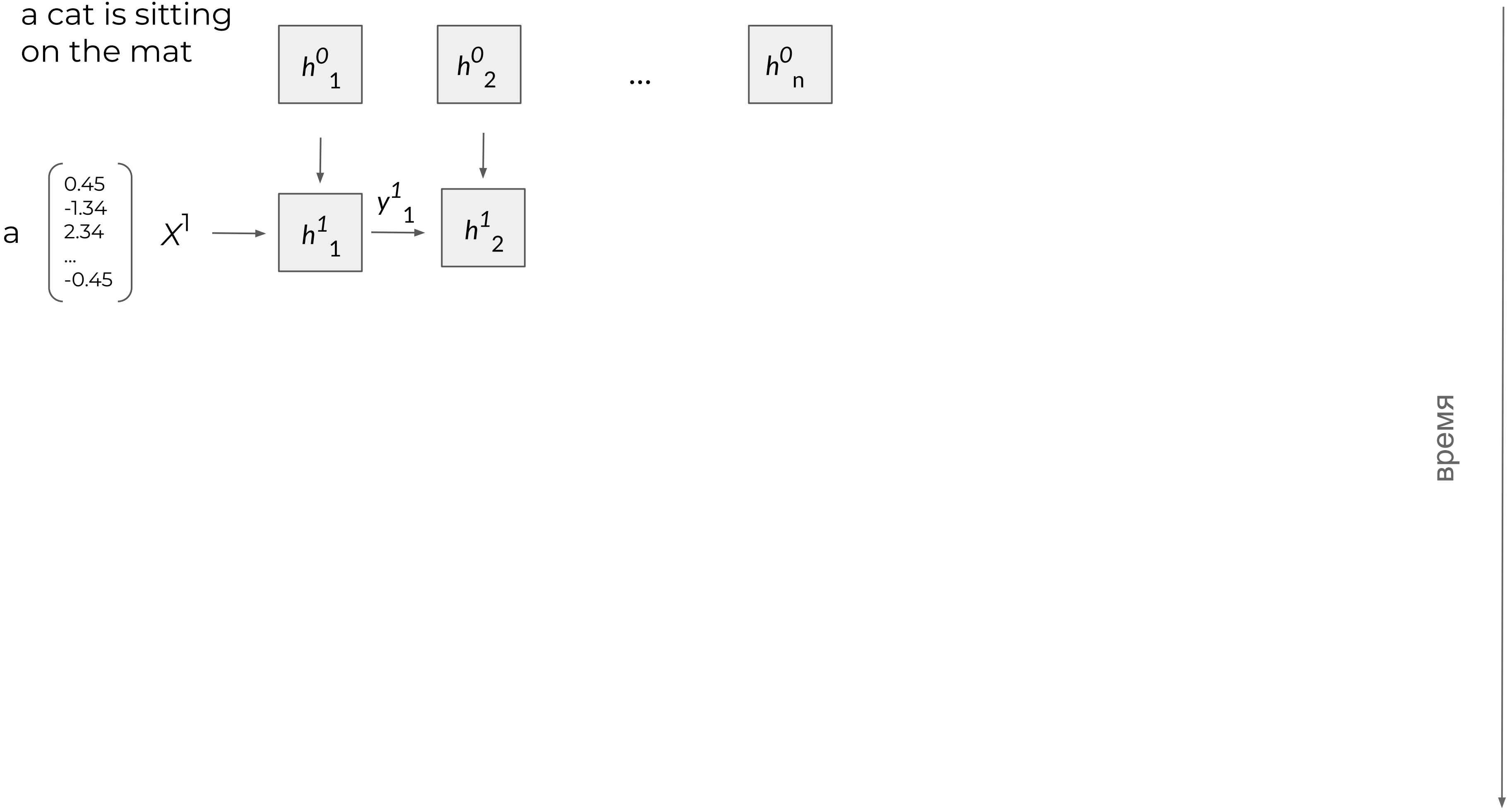
$$y = \sigma\left(Vh^t + b_y\right)$$

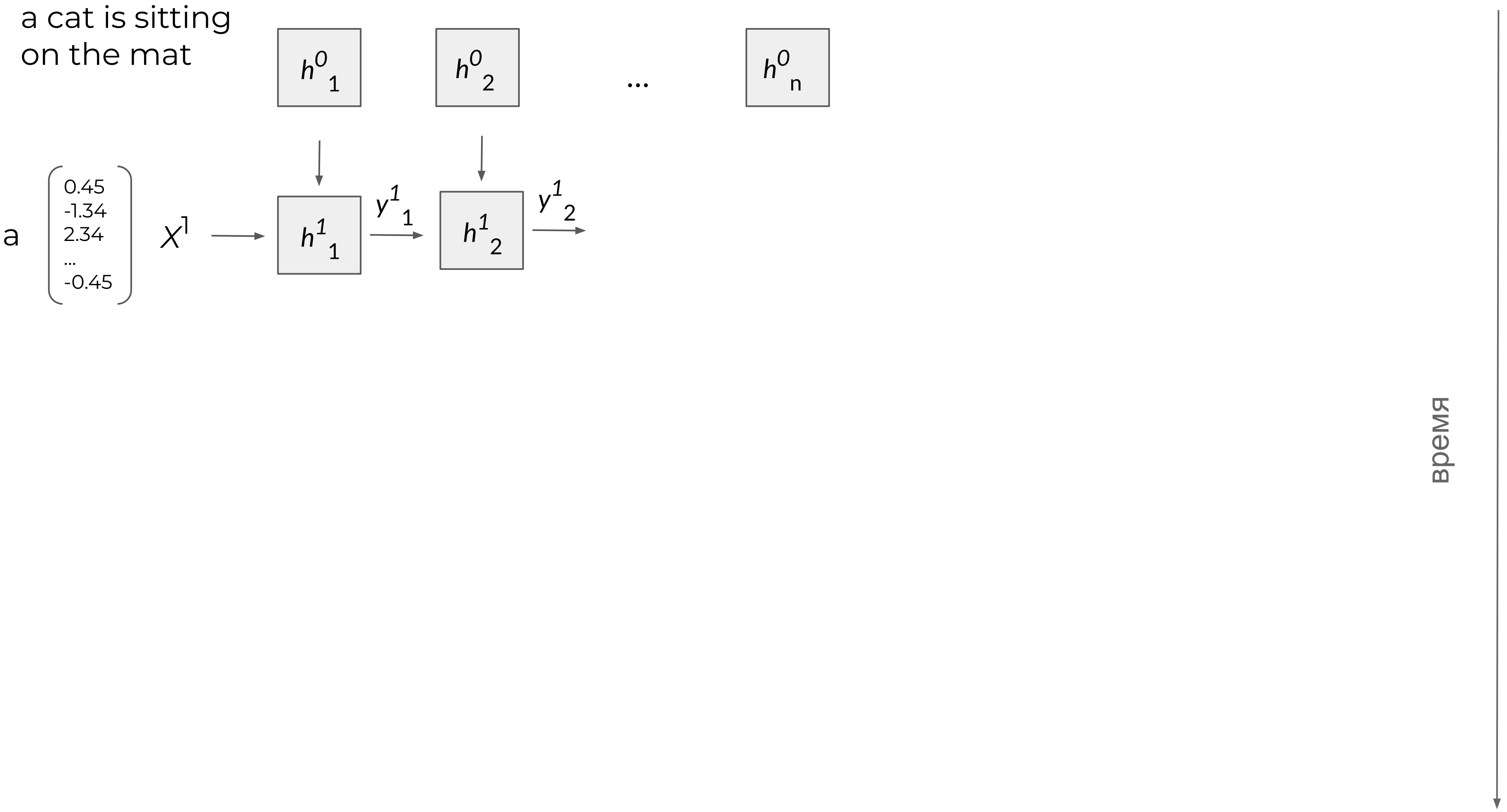


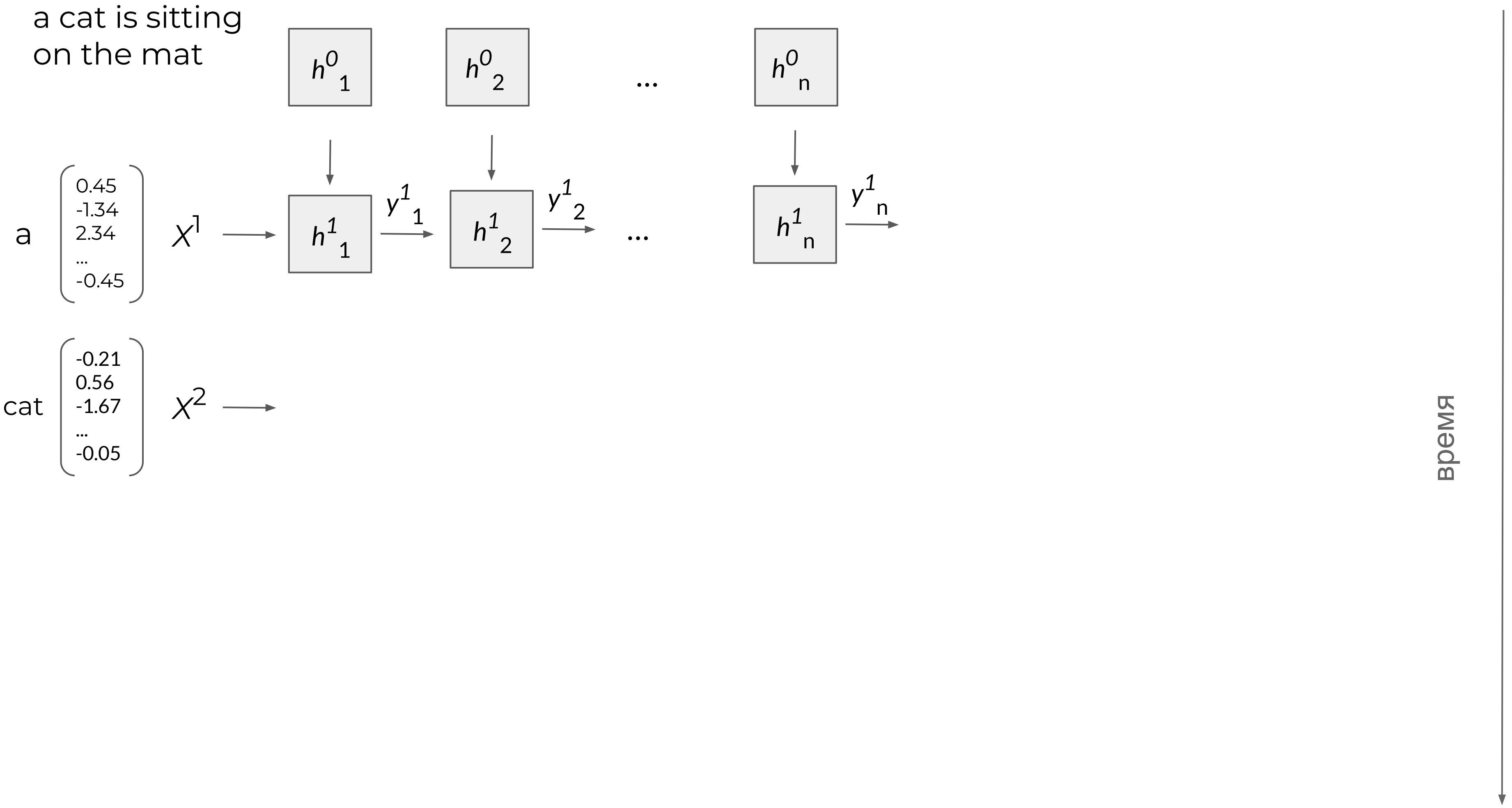

a cat is sitting
on the mat
h^0_1
h^0_2
...
h^0_n
a
0.45
-1.34
2.34
...
-0.45
X^1
h^1_1
время


a cat is sitting
on the mat
h^0_1
h^0_2
...
h^0_n
a
0.45
-1.34
2.34
...
-0.45
X^1
h^1_1
y^1_1
время

a cat is sitting
on the mat

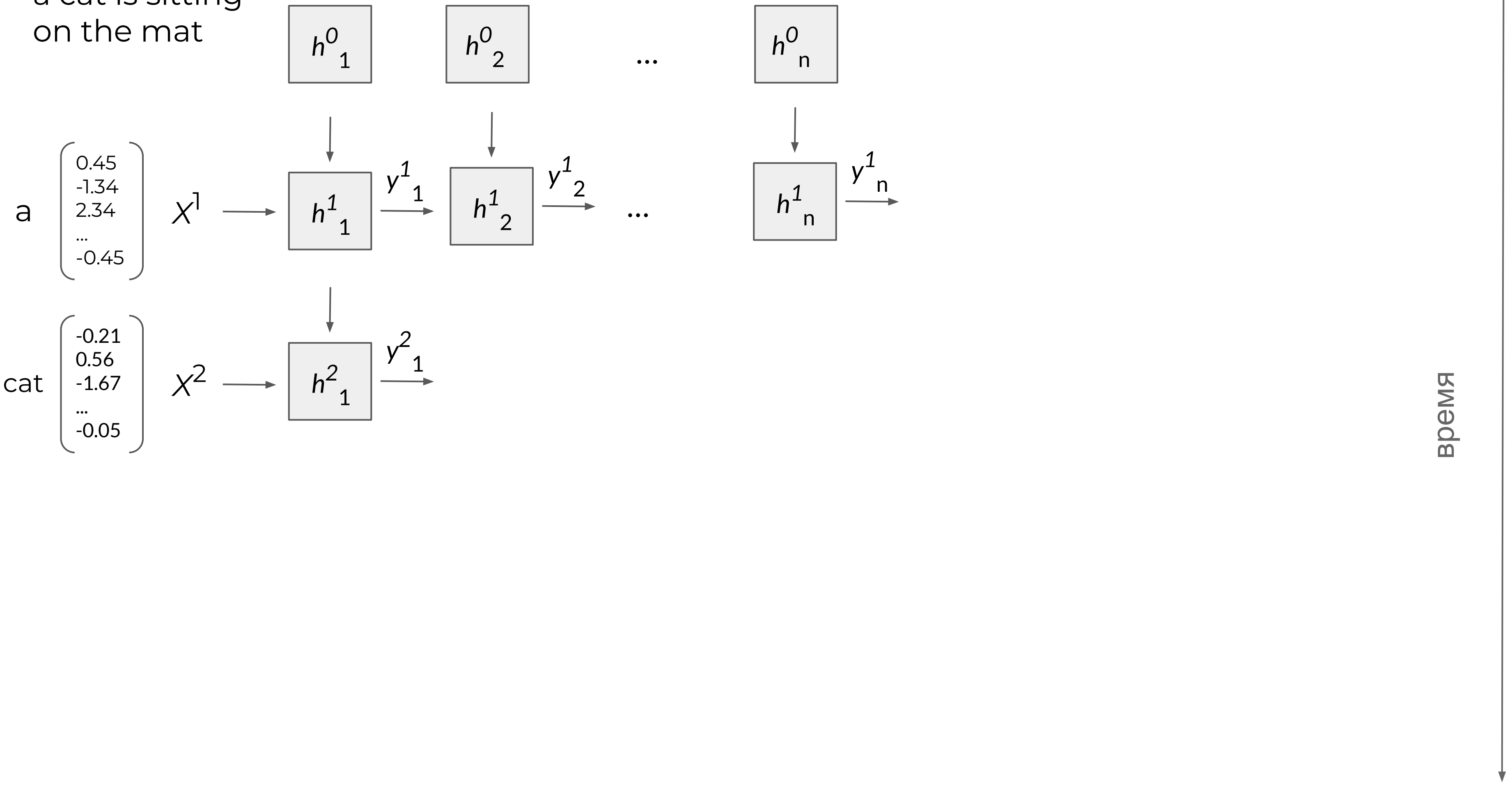


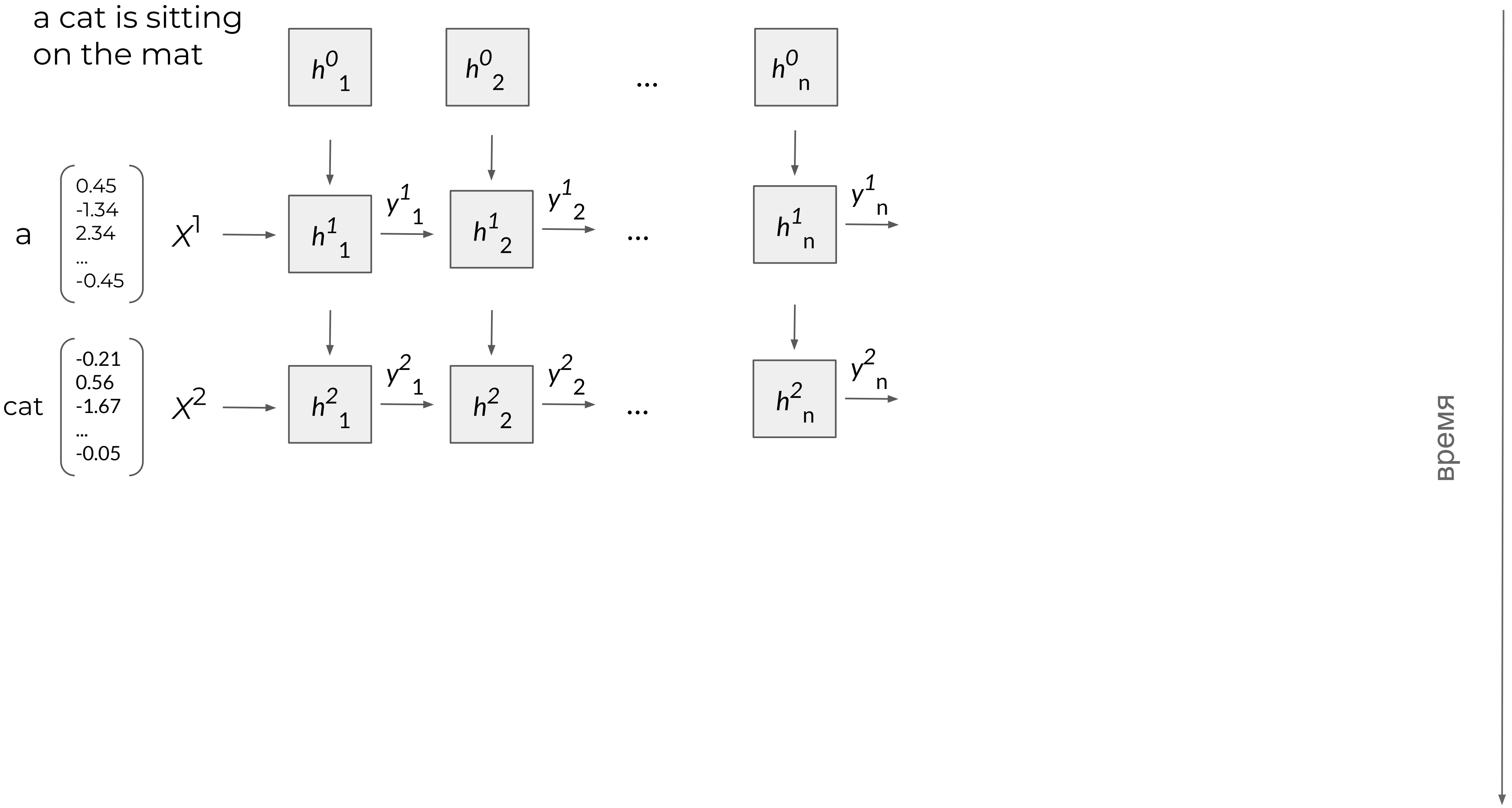

a cat is sitting
on the mat
h^0_1
h^0_2
...
h^0_n
a
0.45
-1.34
2.34
...
-0.45
X^1
h^1_1
y^1_1
h^1_2
y^1_2
время



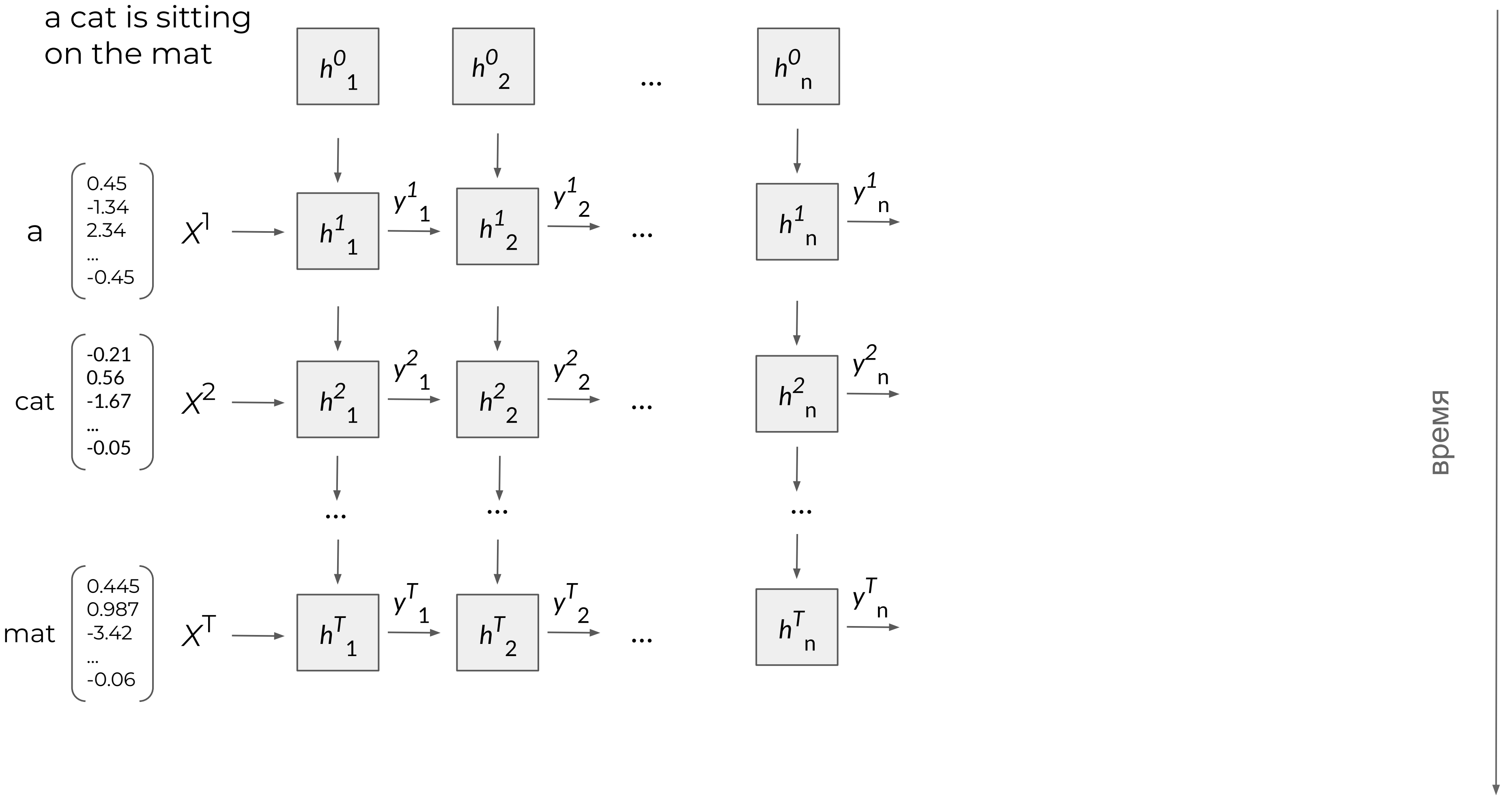
a cat is sitting
on the mat
h^0_1
h^0_2
...
h^0_n
a
0.45
-1.34
2.34
...
-0.45
X^1
h^1_1
y^1_1
h^1_2
y^1_2
...
h^1_n
y^1_n
cat
-0.21
0.56
-1.67
...
-0.05
X^2
время

a cat is sitting
on the mat

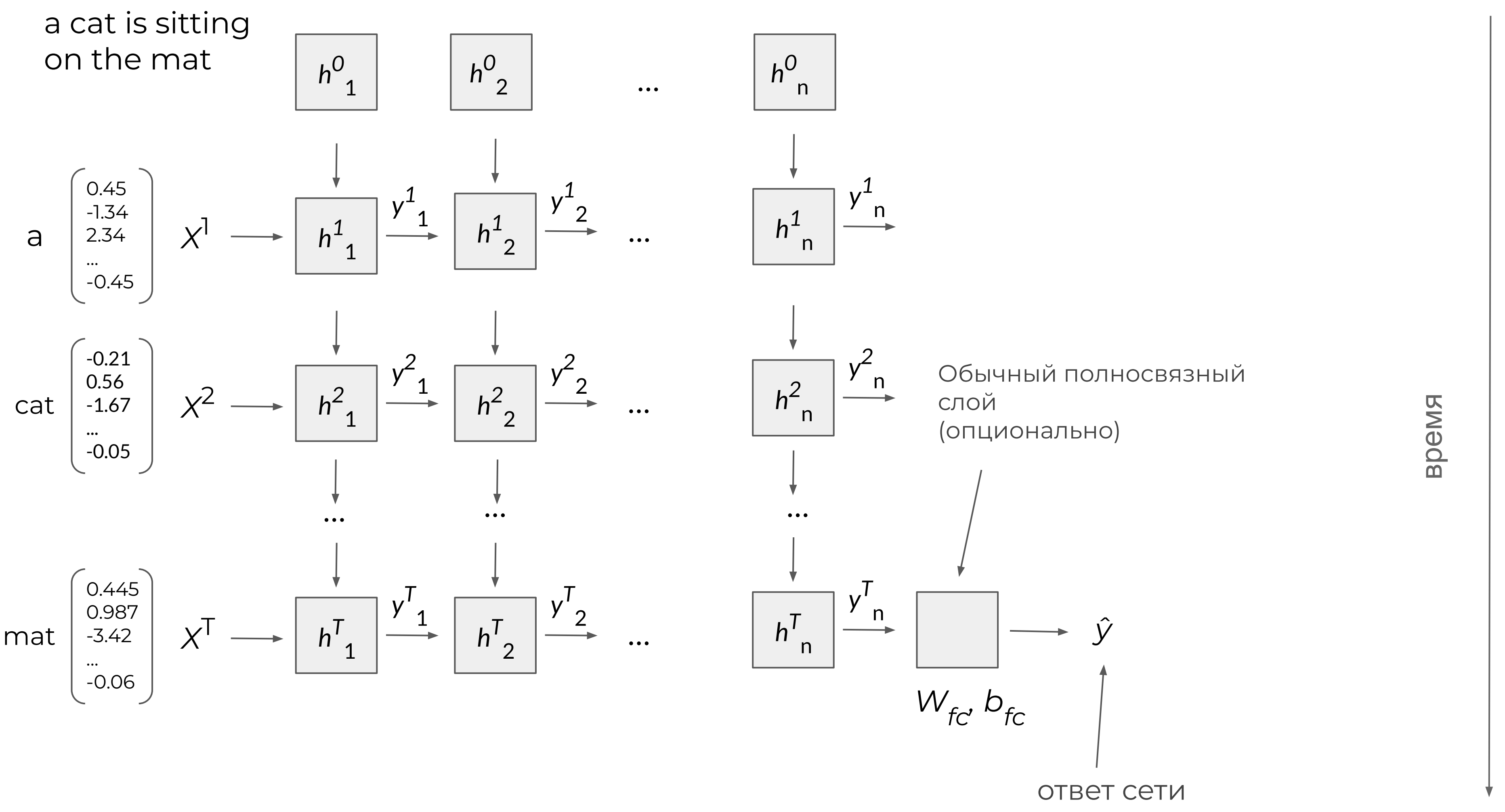




a cat is sitting
on the mat
a
0.45
-1.34
2.34
...
-0.45
cat
-0.21
0.56
-1.67
...
-0.05
время



a cat is sitting
on the mat
a
0.45
-1.34
2.34
...
-0.45
cat
-0.21
0.56
-1.67
...
-0.05
mat
0.445
0.987
-3.42
...
-0.06
X^1
X^2
X^T
h^0_1
h^0_2
h^0_n
h^1_1
h^1_2
h^1_n
h^2_1
h^2_2
h^2_n
h^T_1
h^T_2
h^T_n
y^1_1
y^1_2
y^1_n
y^2_1
y^2_2
y^2_n
y^T_1
y^T_2
y^T_n
время


a cat is sitting
on the mat
h^0_1
h^0_2
...
h^0_n
a
0.45
-1.34
2.34
...
-0.45
X^1
h^1_1
y^1_1
h^1_2
y^1_2
...
h^1_n
y^1_n
cat
-0.21
0.56
-1.67
...
-0.05
X^2
h^2_1
y^2_1
h^2_2
y^2_2
...
h^2_n
y^2_n
Обычный полносвязный
слой
(опционально)
...
mat
0.445
0.987
-3.42
...
-0.06
X^T
h^T_1
y^T_1
h^T_2
y^T_2
...
h^T_n
y^T_n
$\hat{y}$
W_{fc}, b_{fc}
ответ сети
время

Итоги видео

В этом видео мы:

- Познакомились с идеей устройства рекуррентного слоя и рекуррентной нейросети;
- Разобрали forward pass рекуррентной сети.

В следующем видео мы узнаем, как RNN обучается.

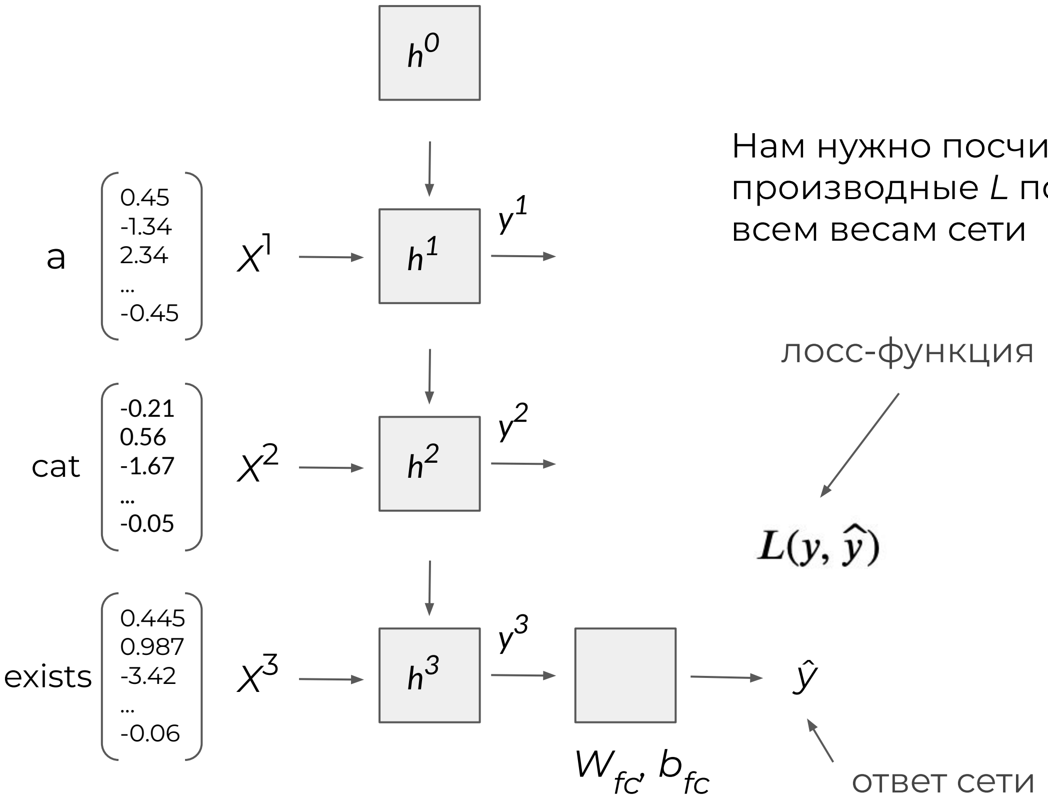
Deep Learning School

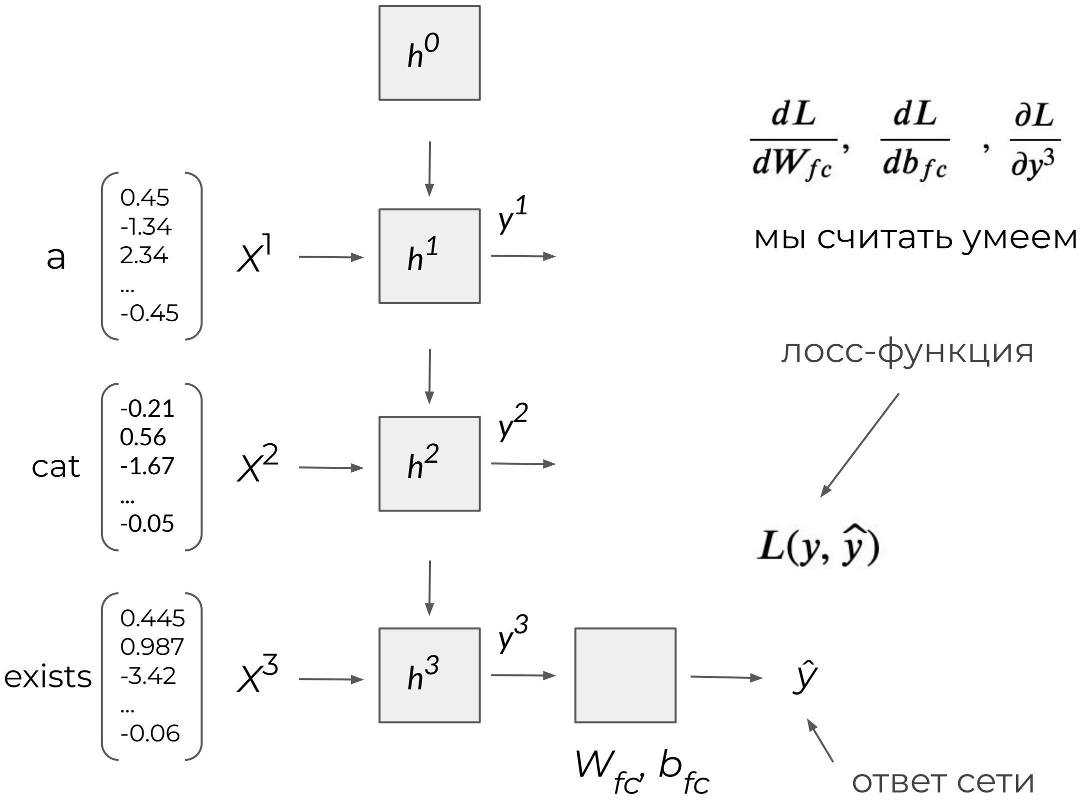
Обучение RNN

План занятия

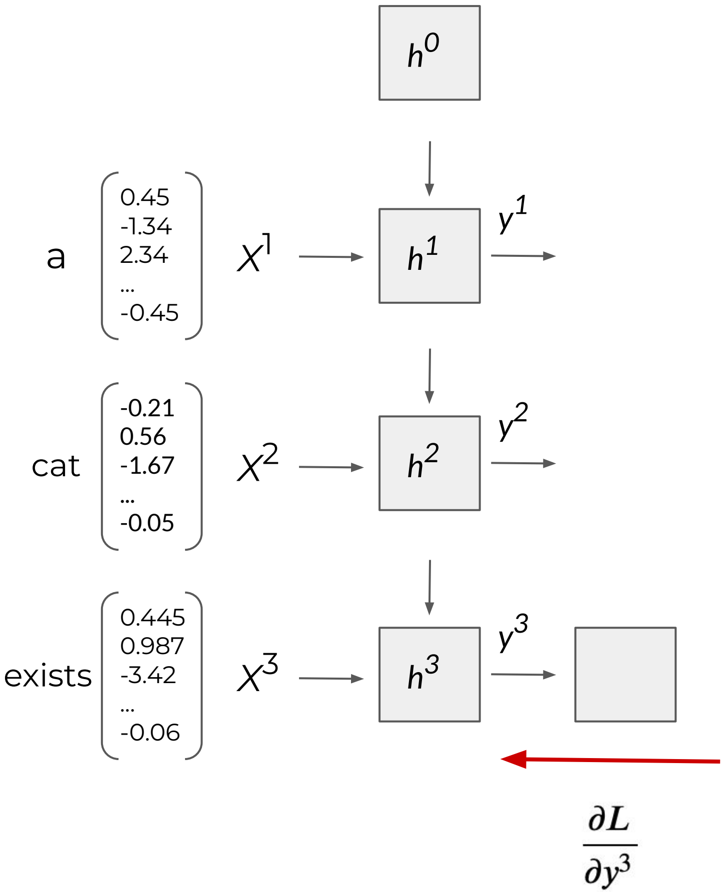
- Рекуррентный слой: идея.
  Рекуррентная нейросеть (RNN);
- Forward pass RNN;
- **Обучение RNN (backward pass);**
- Функции активации в RNN;
- Bidirectional RNN;
- GRU, LSTM

Нам нужно посчитать производные L по всем весам сети

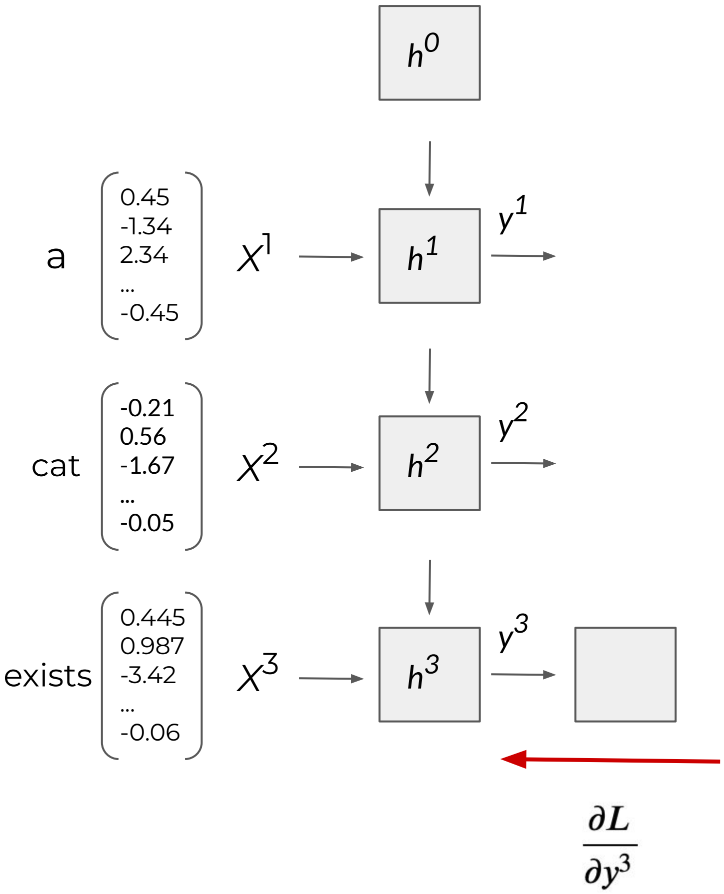




h^0
a
0.45
-1.34
2.34
...
-0.45
X^1
h^1
y^1
cat
-0.21
0.56
-1.67
...
-0.05
X^2
h^2
y^2
exists
0.445
0.987
-3.42
...
-0.06
X^3
h^3
y^3
W_{fc}, b_{fc}
$\hat{y}$
ответ сети
$\frac{dL}{dW_{fc}}, \frac{dL}{db_{fc}}, \frac{\partial L}{\partial y^3}$
мы считать умеем
лосс-функция
$L(y, \hat{y})$



h^0
a
0.45
-1.34
2.34
...
-0.45
X^1
h^1
y^1
cat
-0.21
0.56
-1.67
...
-0.05
X^2
h^2
y^2
exists
0.445
0.987
-3.42
...
-0.06
X^3
h^3
y^3
$\frac{\partial L}{\partial y^3}$

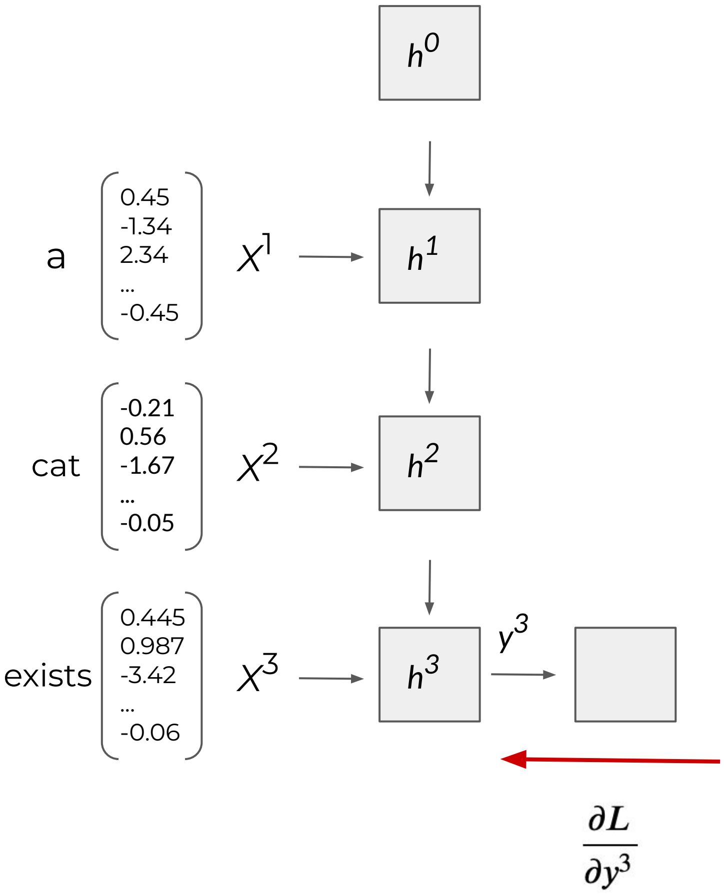


Обучаемые параметры слоя:

W, U, V, b_h, b_y

$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$y^t = \sigma\left(Vh^t + b_y\right)$$

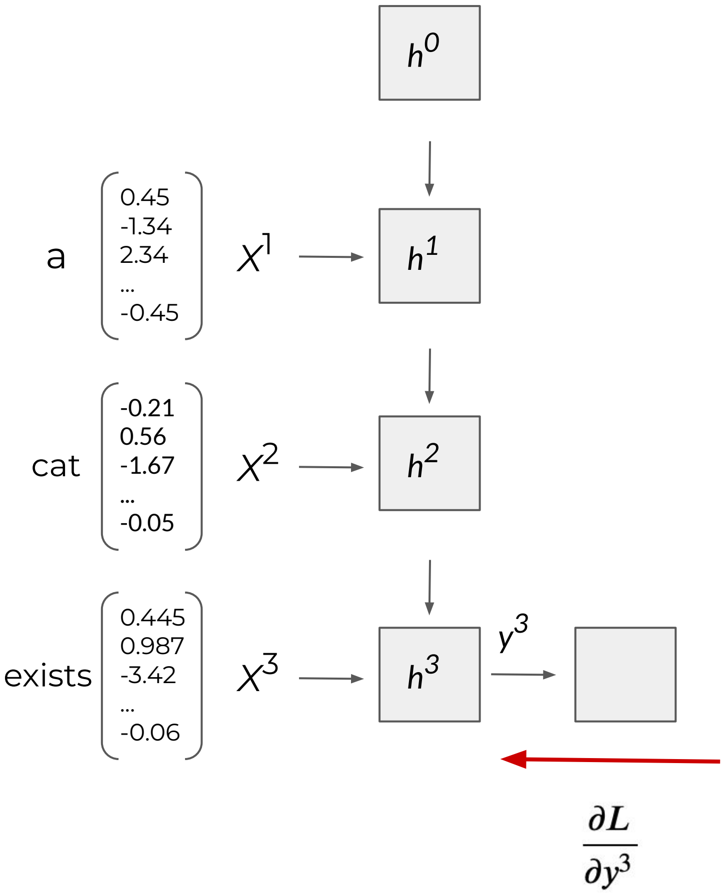


Обучаемые параметры слоя:

W, U, V, b_h, b_y

$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$y^t = \sigma\left(Vh^t + b_y\right)$$



Обучаемые параметры слоя:

$\boldsymbol{W, U, V, b_h, b_y}$

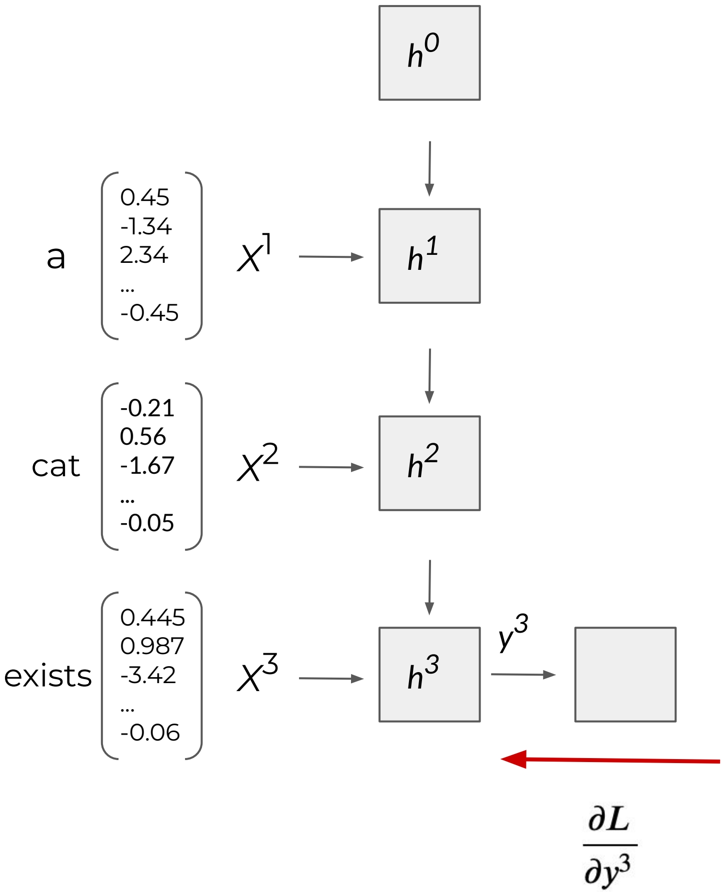
Для $t = 1, 2$:

$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

Для $t = 3$:

$$h^3 = \sigma\left(WX^3 + Uh^2 + b_h\right)$$

$$y^3 = \sigma\left(Vh^3 + b_y\right)$$



Обучаемые параметры слоя:

W, U, V, b_h, b_y

Для $t = 1, 2$:

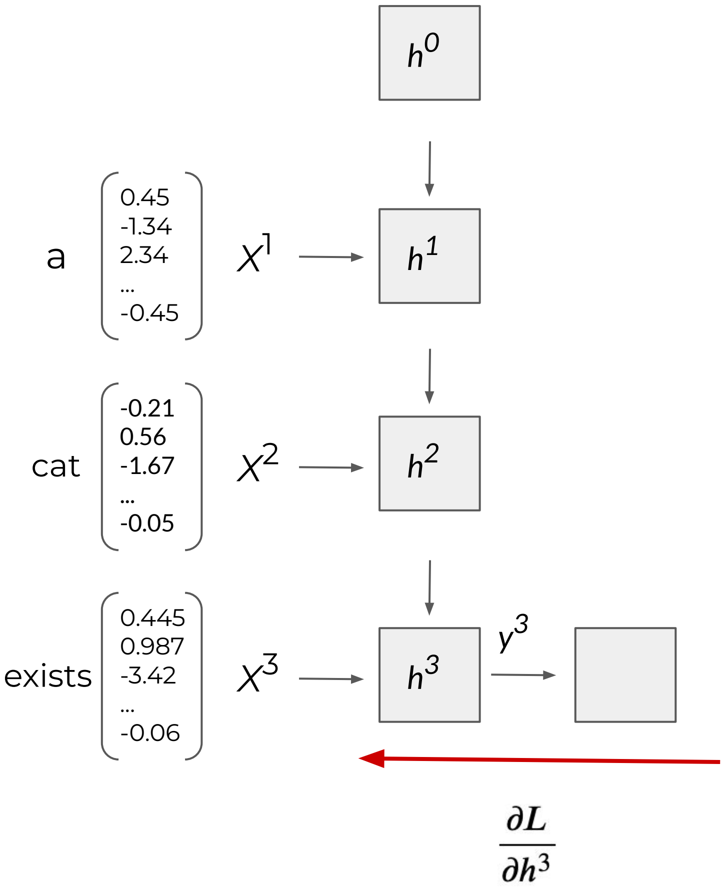
$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

Для $t = 3$:

$$h^3 = \sigma\left(WX^3 + Uh^2 + b_h\right)$$

$$y^3 = \sigma\left(Vh^3 + b_y\right)$$

$$\boxed{\frac{dL}{dV} = \frac{\partial L}{dy^3}\frac{\partial y^3}{dV}, \quad \frac{dL}{db_y} = \frac{\partial L}{\partial y^3}\frac{dy^3}{db_y}}$$



Обучаемые параметры слоя:

$$\boldsymbol{W, U, V, b_h, b_y}$$

Для $t = 1, 2$:

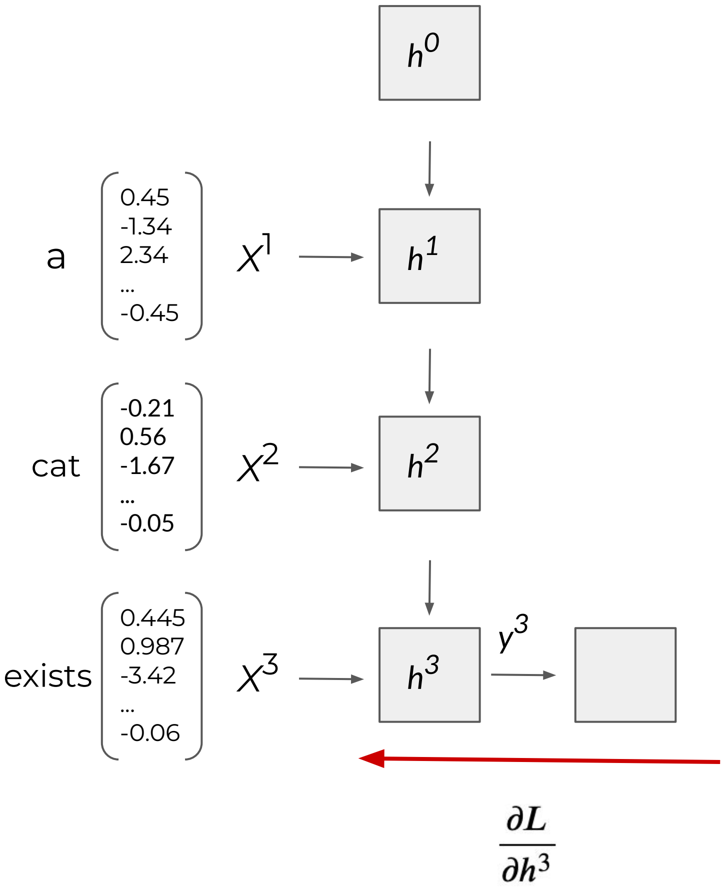
$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

Для $t = 3$:

$$h^3 = \sigma\left(WX^3 + Uh^2 + b_h\right)$$

$$y^3 = \sigma\left(Vh^3 + b_y\right)$$

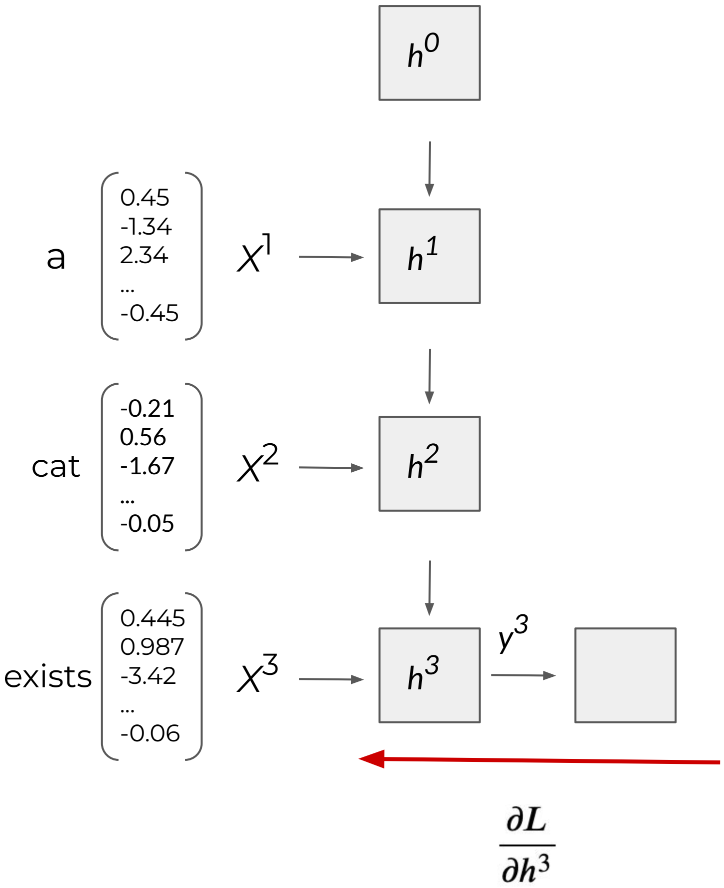
$$\boxed{\frac{dL}{dW} = \frac{\partial L}{\partial y^3}\frac{dy^3}{dW} = \frac{\partial L}{\partial y^3}\frac{\partial y^3}{\partial h^3}\frac{dh^3}{dW} = \frac{\partial L}{\partial h^3}\frac{dh^3}{dW}}$$



$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^3}\boxed{\frac{dh^3}{dW}}$$

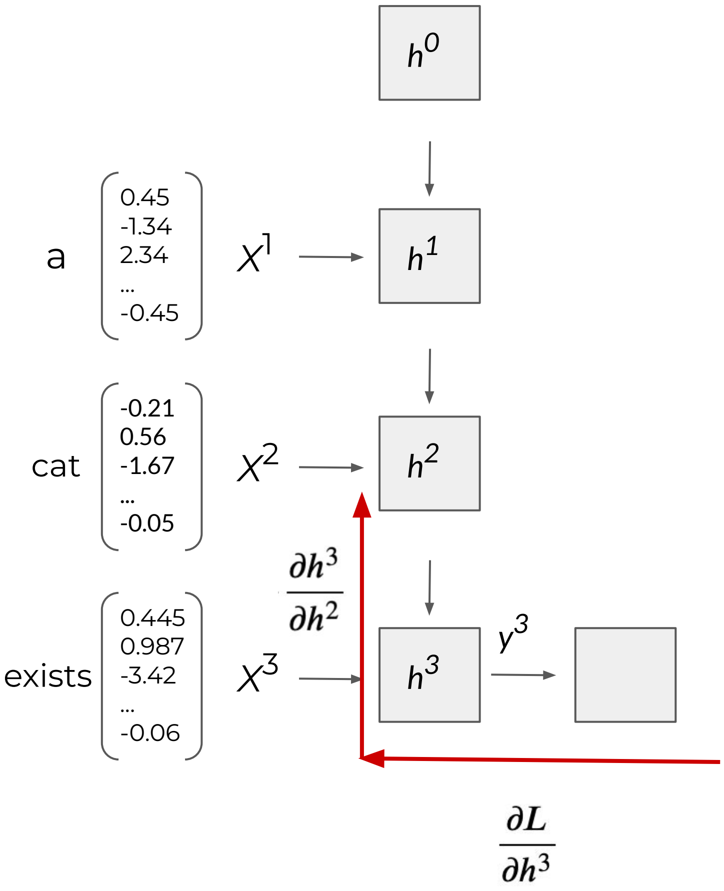
$$\frac{dh^3}{dW} = \frac{\partial h^3}{\partial W} + \frac{\partial h^3}{\partial h^2}\frac{dh^2}{dW}$$



$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^3}\frac{dh^3}{dW} \qquad =$$

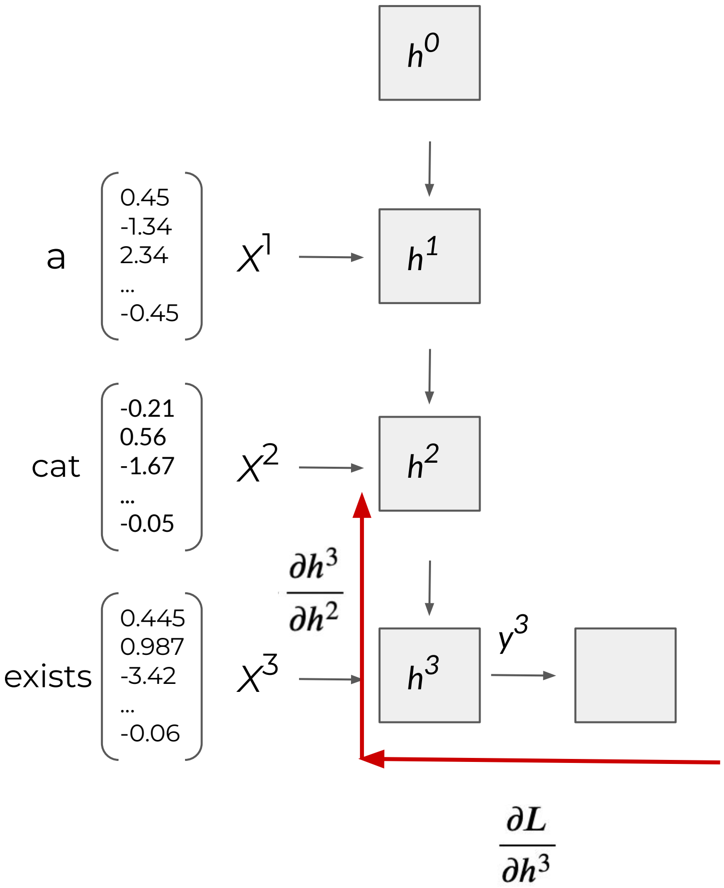
$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{dh^2}{dW}$$



$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^3}\frac{dh^3}{dW} \quad =$$

$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\boxed{\frac{\partial h^3}{\partial h^2}}\frac{dh^2}{dW}$$



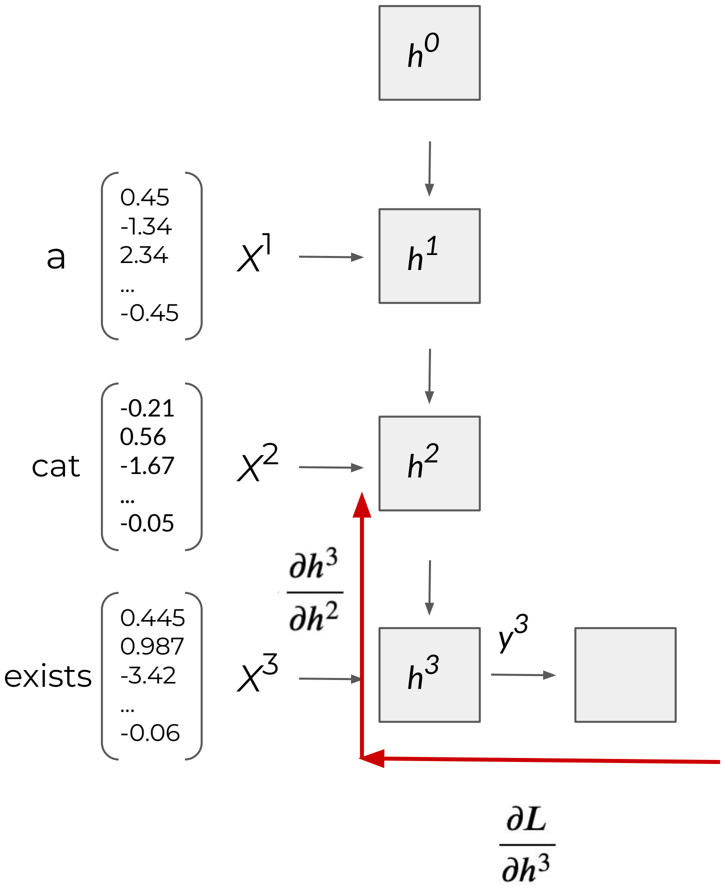
$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^3}\frac{dh^3}{dW} \qquad =$$

$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\boxed{\frac{dh^2}{dW}}$$

$$\|$$

$$\frac{\partial h^2}{\partial W} + \frac{\partial h^2}{\partial h^1}\frac{dh^1}{dW}$$



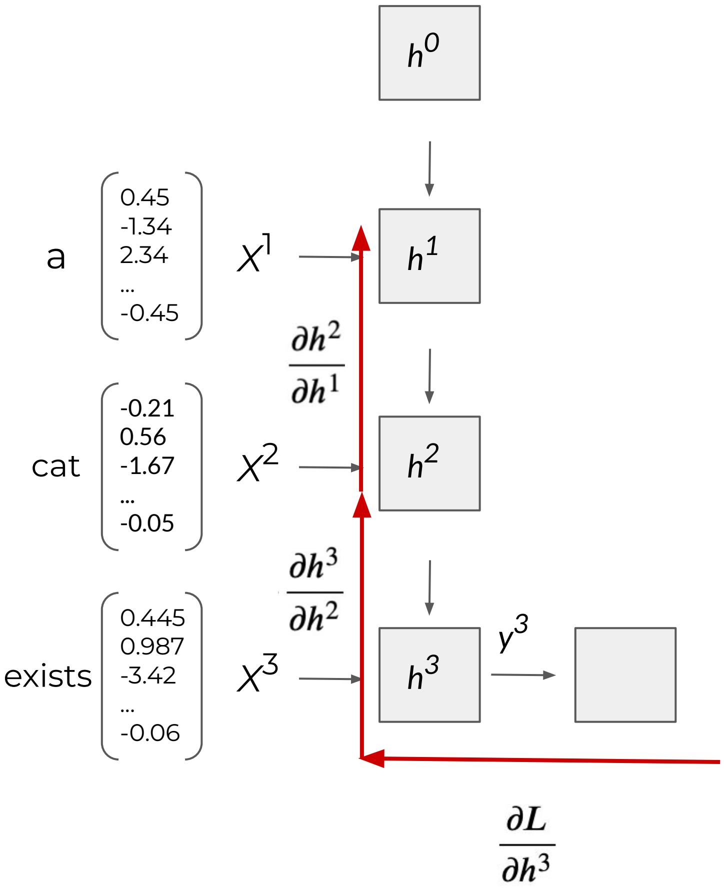
$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^3}\frac{dh^3}{dW} \quad =$$

$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{dh^2}{dW} \quad =$$

$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{\partial h^2}{\partial W} +$$

$$+ \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{\partial h^2}{\partial h^1}\frac{dh^1}{dW}$$

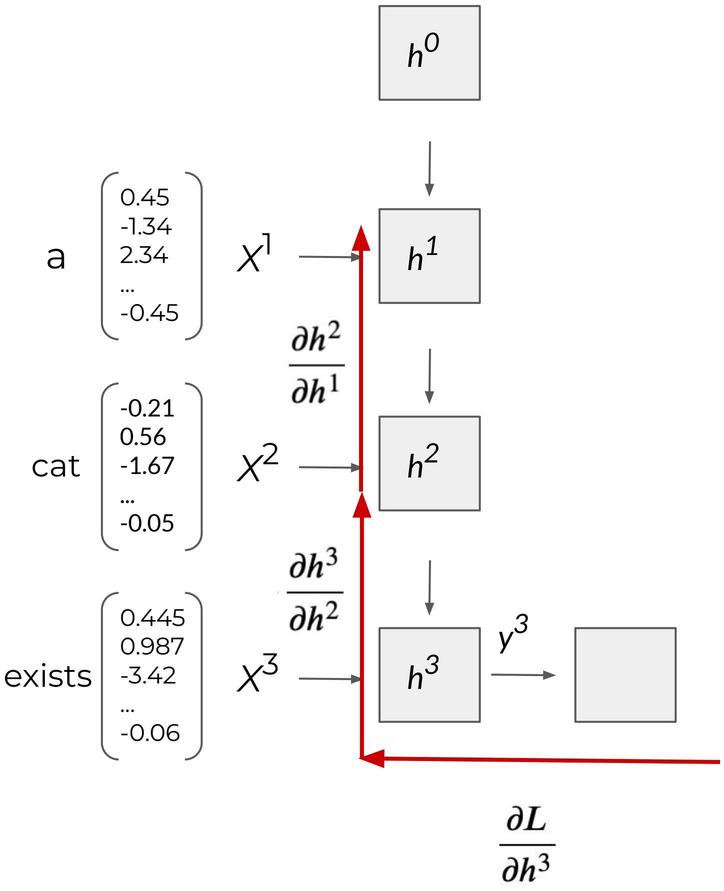


$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^3}\frac{dh^3}{dW} =$$

$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{dh^2}{dW} =$$

$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{\partial h^2}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\boxed{\frac{\partial h^2}{\partial h^1}}\frac{dh^1}{dW}$$



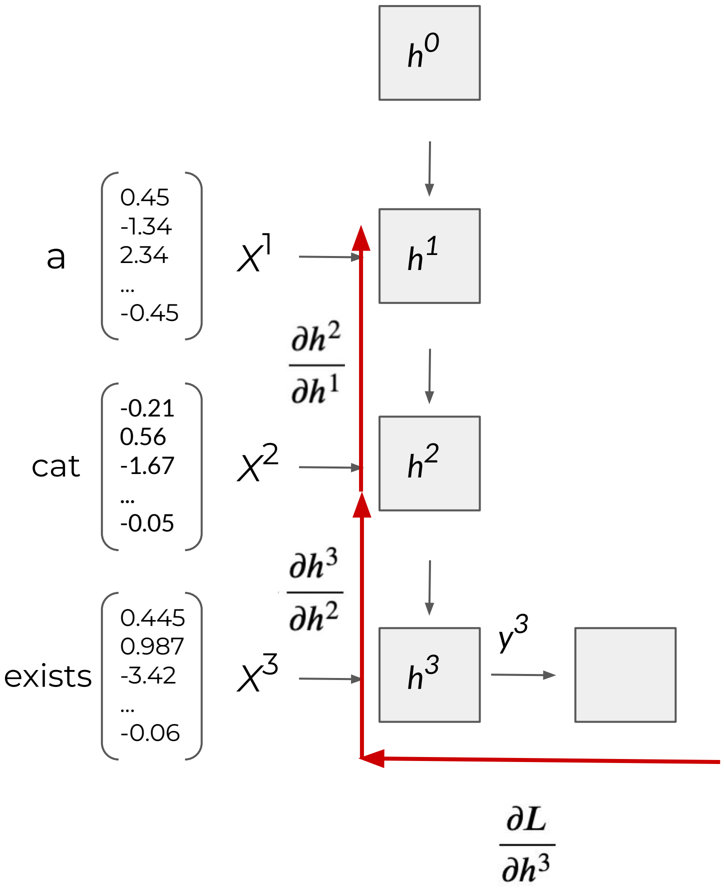
$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^3}\frac{dh^3}{dW} =$$

$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{dh^2}{dW} =$$

$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{\partial h^2}{\partial W} +$$

$$+ \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{\partial h^2}{\partial h^1}\boxed{\frac{dh^1}{dW}}$$



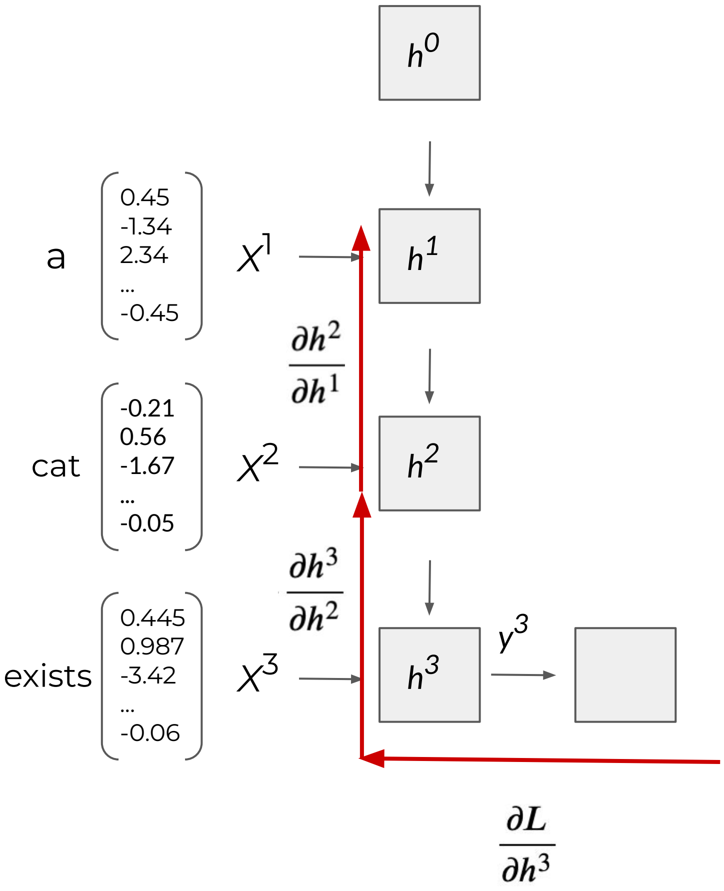
$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^3}\frac{dh^3}{dW} =$$

$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{dh^2}{dW} =$$

$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{\partial h^2}{\partial W} +$$

$$+ \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{\partial h^2}{\partial h^1}\underbrace{\boxed{\frac{dh^1}{dW}}}_{= \frac{\partial h^1}{\partial W}}$$



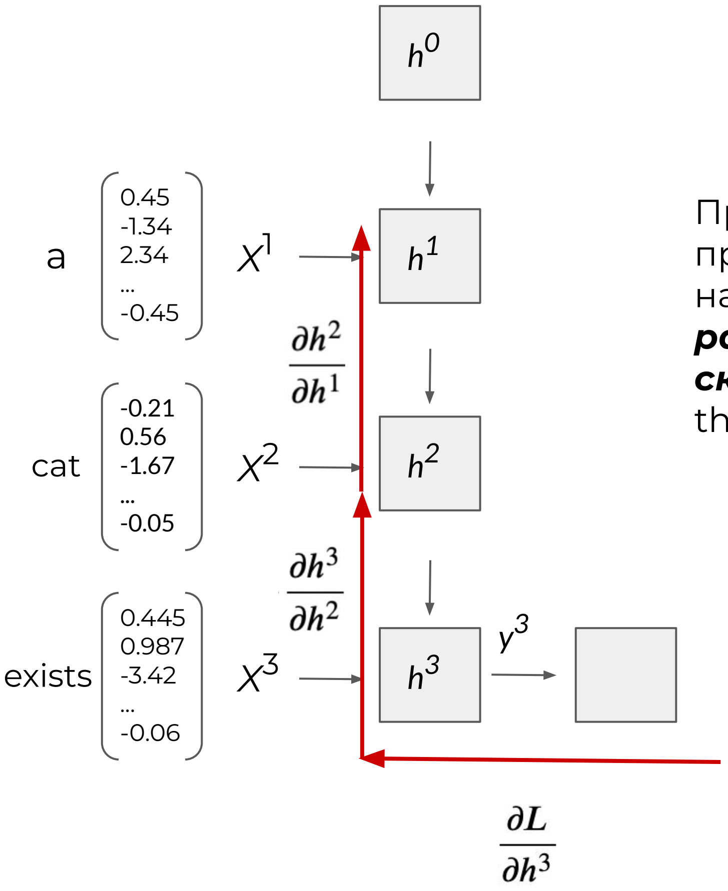
$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^3}\frac{dh^3}{dW} \quad =$$

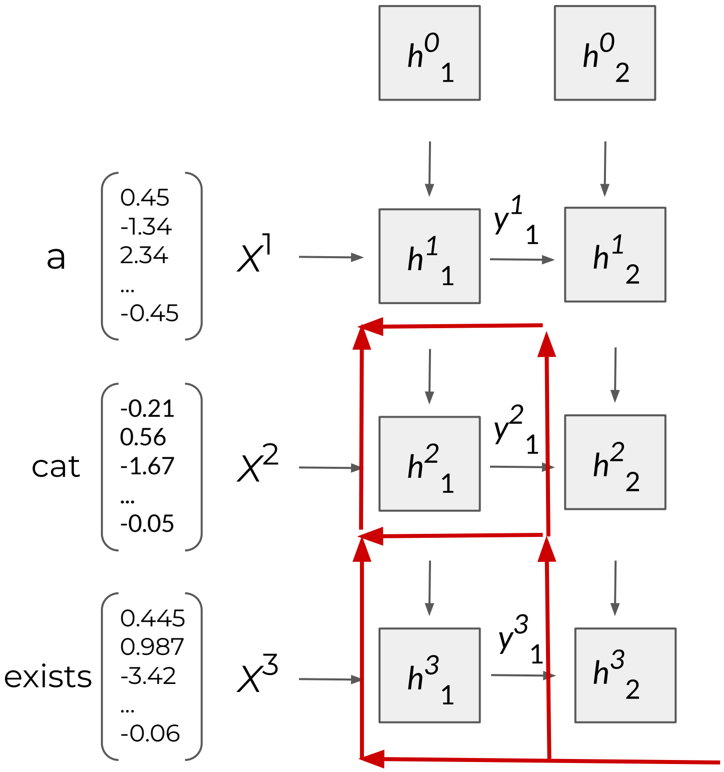
$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{dh^2}{dW} \quad =$$

$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{\partial h^2}{\partial W} +$$

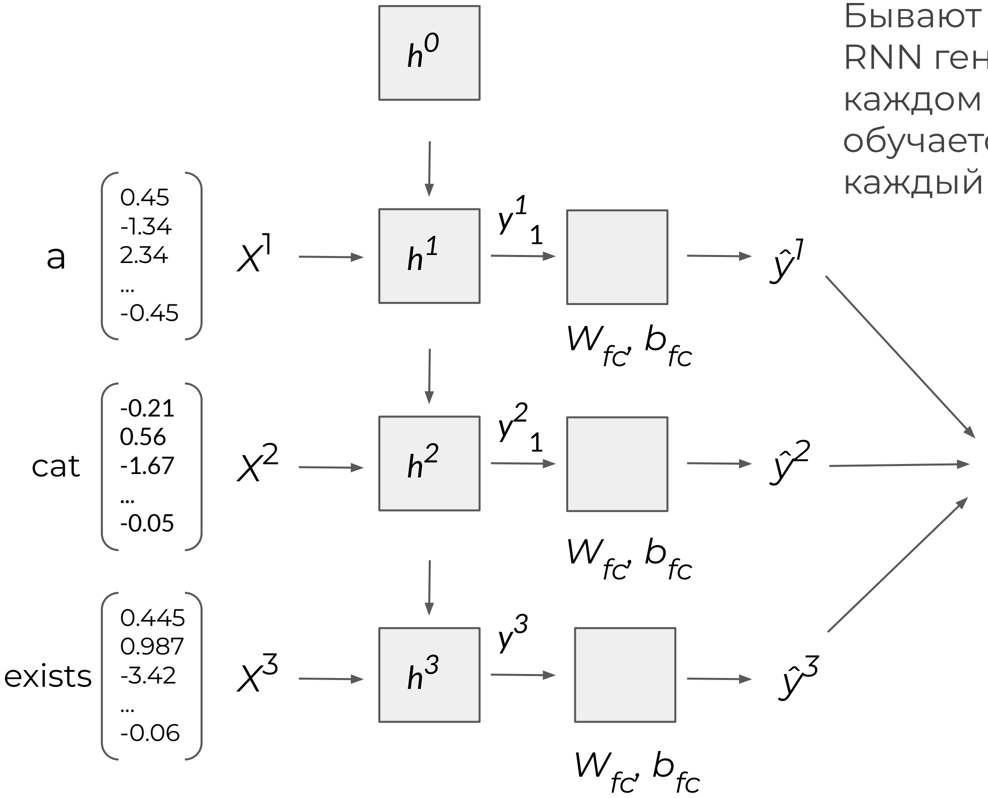
$$+ \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{\partial h^2}{\partial h^1}\frac{\partial h^1}{\partial W}$$



Процесс вычисления производных весов RNN называется ***обратное распространение ошибки сквозь время*** (backpropagation through time)



При добавлении слоев в RNN подсчет градиентов усложняется



Бывают задачи, в которых RNN генерирует ответ на каждом шаге. Тогда сеть обучается на сумме лоссов в каждый момент времени

$$\sum_{i=1}^{T} L(\hat{y}^t, y^t)$$

Итоги видео

В этом видео мы обсудили работу алгоритма обновления весов нейросети: обратное распространение ошибки сквозь время.

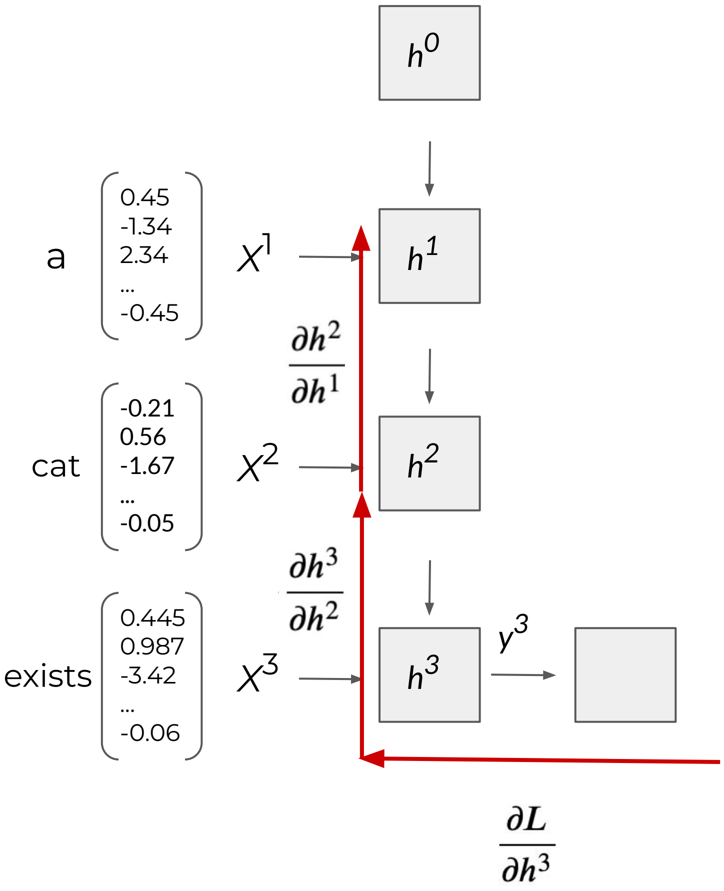
В следующем видео мы обсудим несколько нюансов RNN.

Deep Learning School

Функции активации в RNN. Bidirectional RNN

План занятия

- Рекуррентный слой: идея. Рекуррентная нейросеть (RNN);
- Forward pass RNN;
- Обучение RNN (backward pass);
- **Функции активации в RNN;**
- **Bidirectional RNN;**
- GRU, LSTM



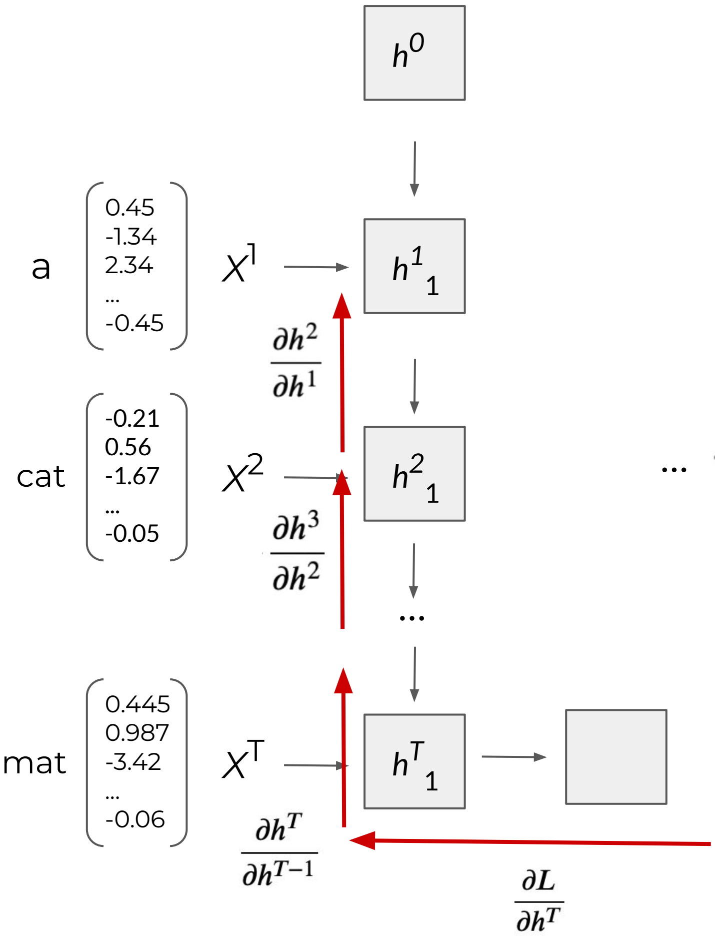
$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^3}\frac{dh^3}{dW} =$$

$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{dh^2}{dW} =$$

$$= \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial W} + \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{\partial h^2}{\partial W} +$$

$$+ \frac{\partial L}{\partial h^3}\frac{\partial h^3}{\partial h^2}\frac{\partial h^2}{\partial h^1}\frac{\partial h^1}{\partial W}$$

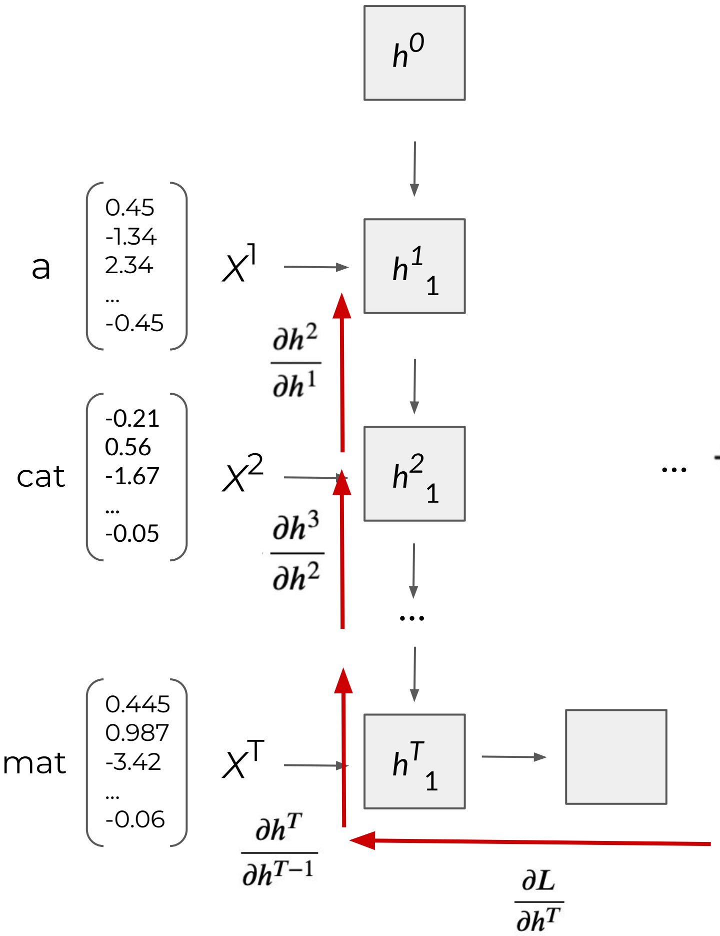


$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^T}\frac{dh^T}{dW} \quad =$$

$$= \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial W} + \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial W} + \ \ldots$$

$$\ldots + \frac{\partial L}{\partial h^T}\underbrace{\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial h^{T-2}}\cdots\frac{dh^2}{dh^1}}_{\text{T-1}}\frac{dh^1}{dW}$$



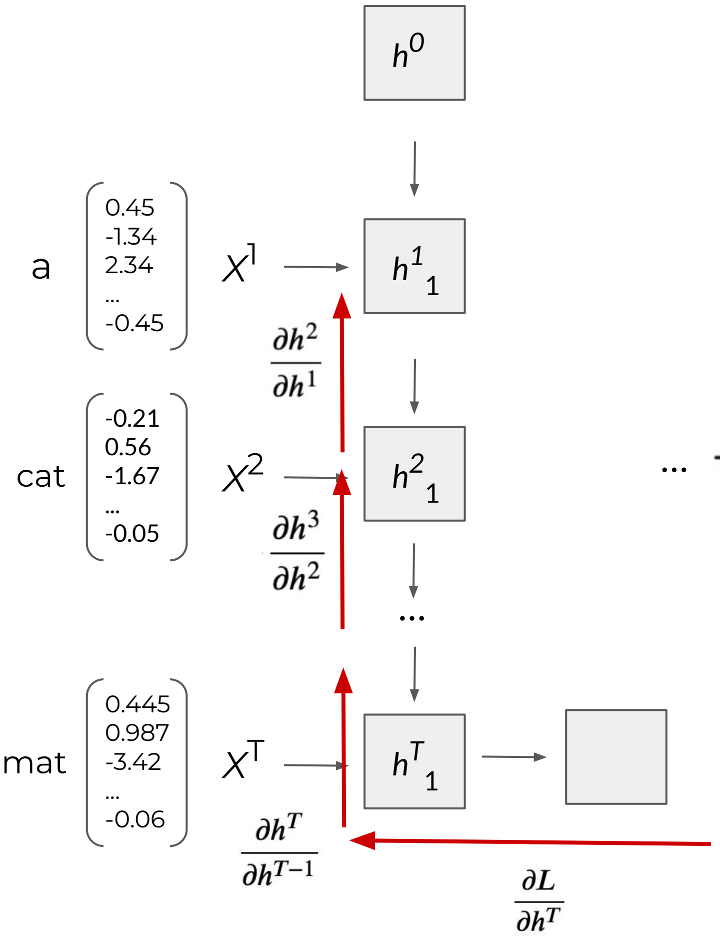
$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^T}\frac{dh^T}{dW} =$$

$$= \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial W} + \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial W} + \dots$$

$$\dots + \frac{\partial L}{\partial h^T}\underbrace{\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial h^{T-2}}\cdots\frac{dh^2}{dh^1}}_{\text{T-1}}\frac{dh^1}{dW}$$

Для длинных входных последовательностей характерна проблема ***затухания градиентов***



$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

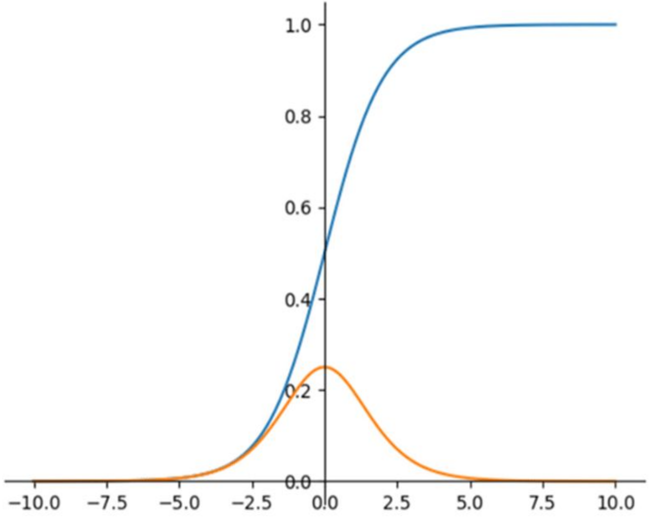
$$\frac{dL}{dW} = \frac{\partial L}{\partial h^T}\frac{dh^T}{dW} =$$

$$= \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial W} + \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial W} + \ldots$$

$$\ldots + \frac{\partial L}{\partial h^T}\underbrace{\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial h^{T-2}}\cdots\frac{dh^2}{dh^1}}\frac{dh^1}{dW}$$

В этих множителях
содержится производная
функции активации σ

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



Сигмоидная функция активации и ее производная

$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

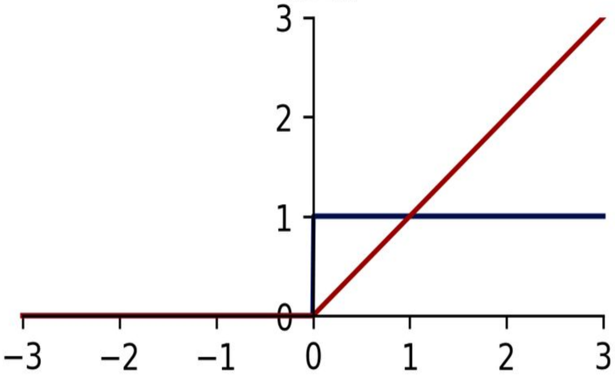
$$\frac{dL}{dW} = \frac{\partial L}{\partial h^T}\frac{dh^T}{dW} =$$

$$= \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial W} + \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial W} + \dots$$

$$. + \frac{\partial L}{\partial h^T}\underbrace{\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial h^{T-2}}\cdots\frac{dh^2}{dh^1}}_{}\frac{dh^1}{dW}$$

В этих множителях содержится производная функции активации σ

$$f'(x) = \begin{cases} 0 \text{ for } x < 0 \\ 1 \text{ for } x \geq 0 \end{cases}$$



Функция активации ReLU
и ее производная

$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

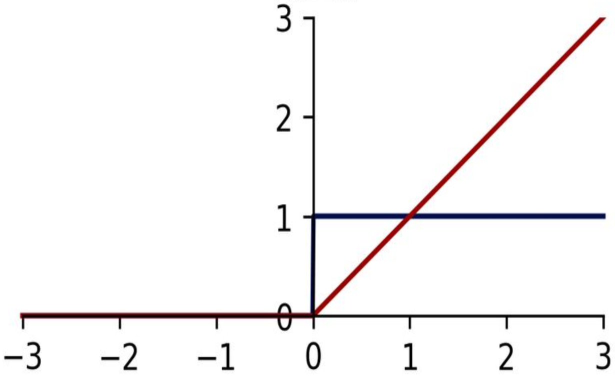
$$\frac{dL}{dW} = \frac{\partial L}{\partial h^T}\frac{dh^T}{dW} \quad =$$

$$= \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial W} + \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial W} + \ \dots$$

$$.. + \frac{\partial L}{\partial h^T}\underbrace{\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial h^{T-2}}\cdots\frac{dh^2}{dh^1}}_{U^{T-1}}\frac{dh^1}{dW}$$

Значение элементов
этой матрицы может
быть очень большим

$$f'(x) = \begin{cases} 0 \text{ for } x < 0 \\ 1 \text{ for } x \geq 0 \end{cases}$$



Функция активации ReLU
и ее производная

$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

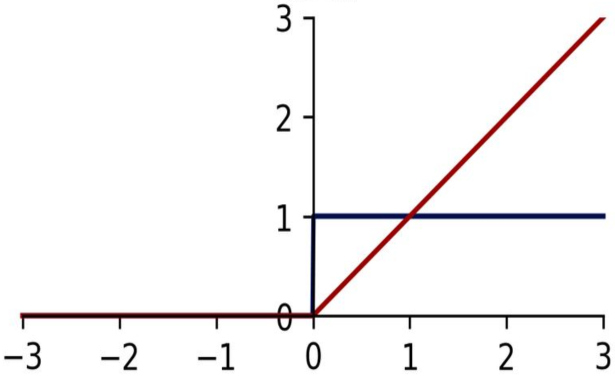
$$\frac{dL}{dW} = \frac{\partial L}{\partial h^T}\frac{dh^T}{dW} \quad =$$

$$= \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial W} + \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial W} + \;\dots$$

$$.. + \frac{\partial L}{\partial h^T}\underbrace{\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial h^{T-2}}\cdots\frac{dh^2}{dh^1}}_{U^{T-1}}\frac{dh^1}{dW}$$

Для длинных входных последовательностей с функцией активации ReLU характерна проблема ***взрыва градиентов***

$$f'(x) = \begin{cases} 0 \text{ for } x < 0 \\ 1 \text{ for } x \geq 0 \end{cases}$$



Функция активации ReLU и ее производная

$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^T}\frac{dh^T}{dW} =$$

$$= \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial W} + \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial W} + \ \dots$$

$$.. + \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial h^{T-2}} \cdots \frac{dh^2}{dh^1}\frac{dh^1}{dW}$$

Еще один минус RELU — он не ограничивает распределение выхода слоя.

Распределение вектора h^t может меняться в течение времени, что ухудшает работу нейросети.

$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^T}\frac{dh^T}{dW} \quad =$$

$$= \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial W} + \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial W} + \ \dots$$

$$\dots \ + \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial h^{T-2}}\cdots\frac{dh^2}{dh^1}\frac{dh^1}{dW}$$

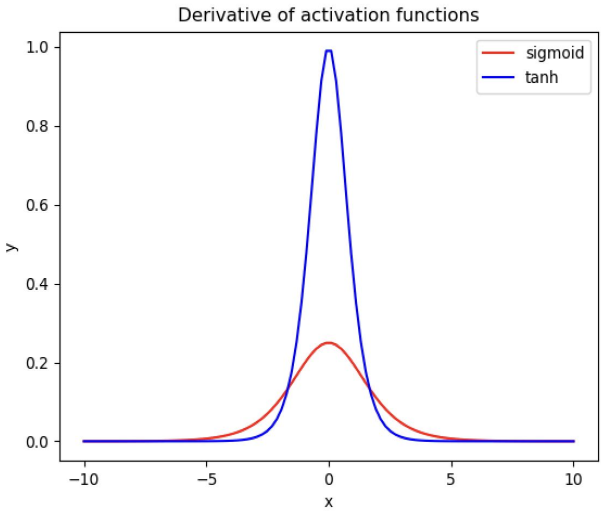
Как бороться?

Против взрыва градиентов:
- Gradient clipping;

Против затухания градиентов:
- Модели нейрона GRU, LSTM

В целом: использовать функцию активации Tanh

$$h^t = \sigma\left(WX^t + Uh^{t-1} + b_h\right)$$

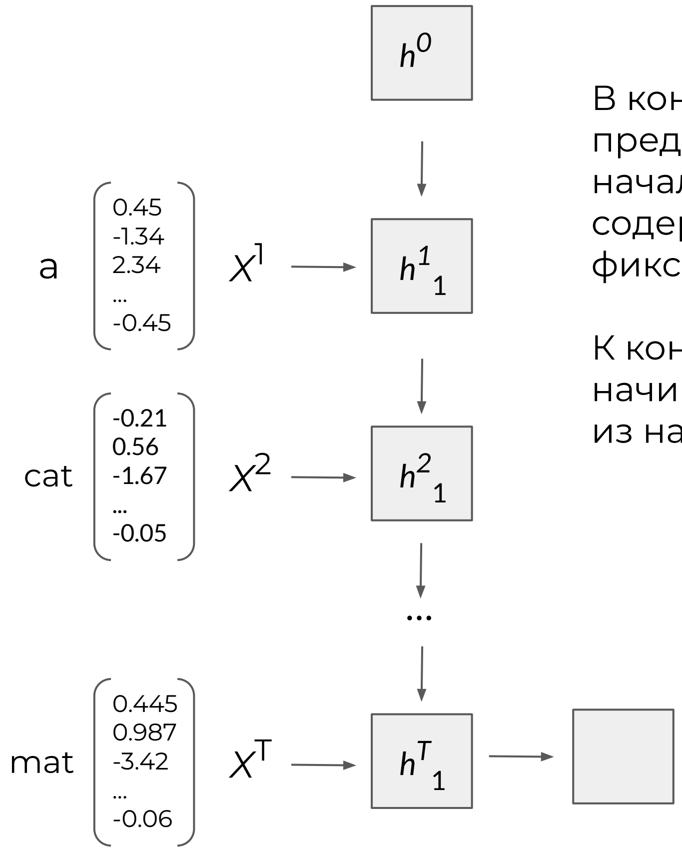


Производные сигмоиды и tanh

$$\frac{dL}{dW} = \frac{\partial L}{\partial h^T}\frac{dh^T}{dW} =$$

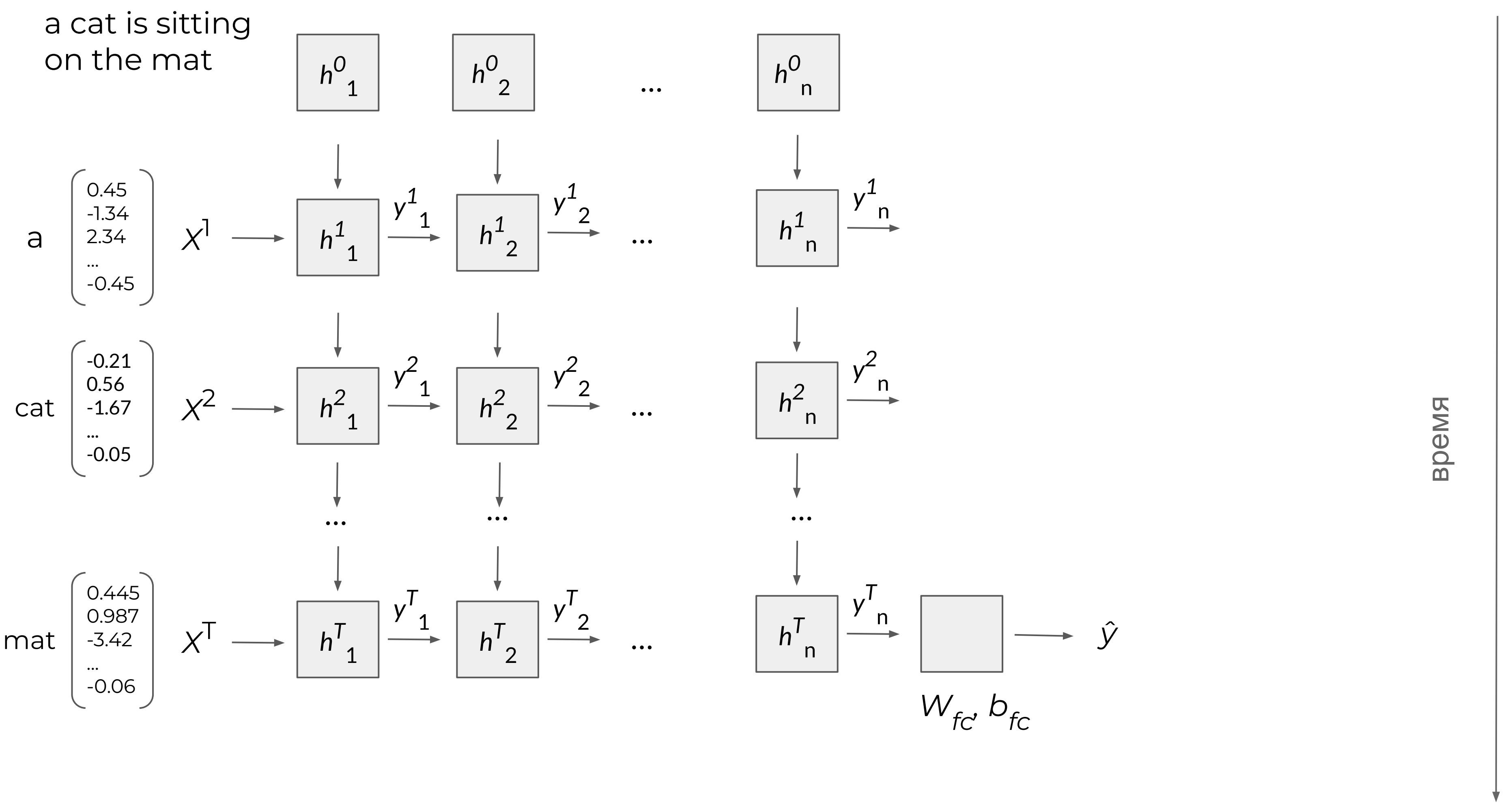
$$= \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial W} + \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial W} + \ldots$$

$$\ldots + \frac{\partial L}{\partial h^T}\frac{\partial h^T}{\partial h^{T-1}}\frac{\partial h^{T-1}}{\partial h^{T-2}}\cdots\frac{dh^2}{dh^1}\frac{dh^1}{dW}$$

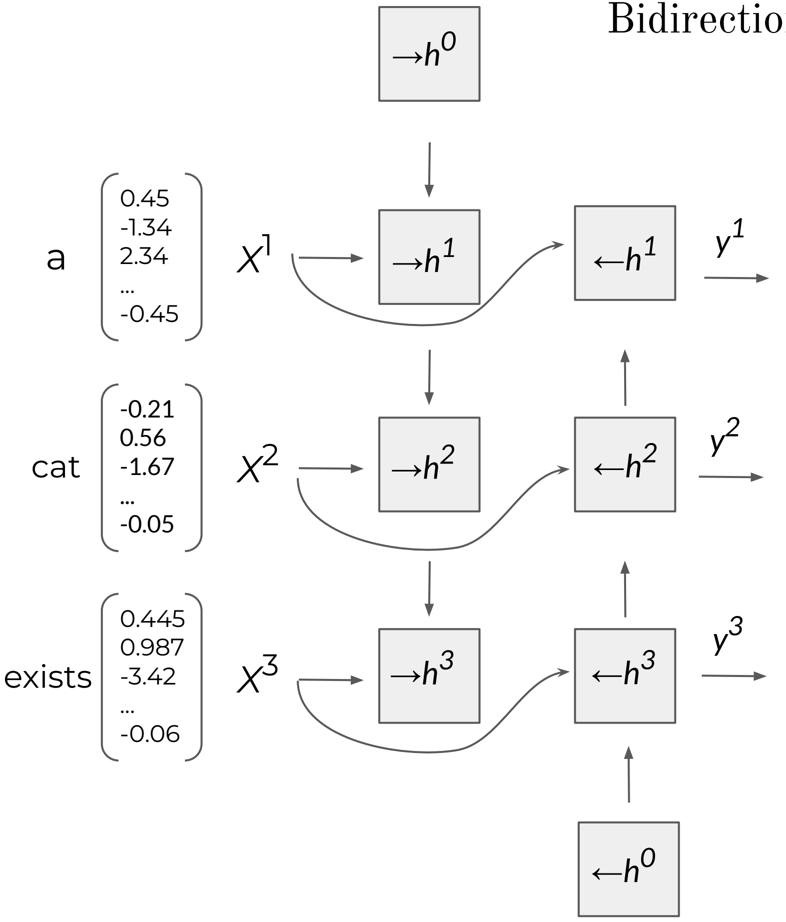


В конце обработки длинного предложения вся информация о начале предложения должна содержаться в одном векторе h фиксированного размера.

К концу длинного предложения h начинает “забывать” информацию из начала.


a cat is sitting
on the mat
a
0.45
-1.34
2.34
...
-0.45
cat
-0.21
0.56
-1.67
...
-0.05
mat
0.445
0.987
-3.42
...
-0.06
X^1
X^2
X^T
h^0_1
h^0_2
h^0_n
h^1_1
h^1_2
h^1_n
h^2_1
h^2_2
h^2_n
h^T_1
h^T_2
h^T_n
y^1_1
y^1_2
y^1_n
y^2_1
y^2_2
y^2_n
y^T_1
y^T_2
y^T_n
W_{fc}, b_{fc}
$\hat{y}$
время

Bidirectional RNN

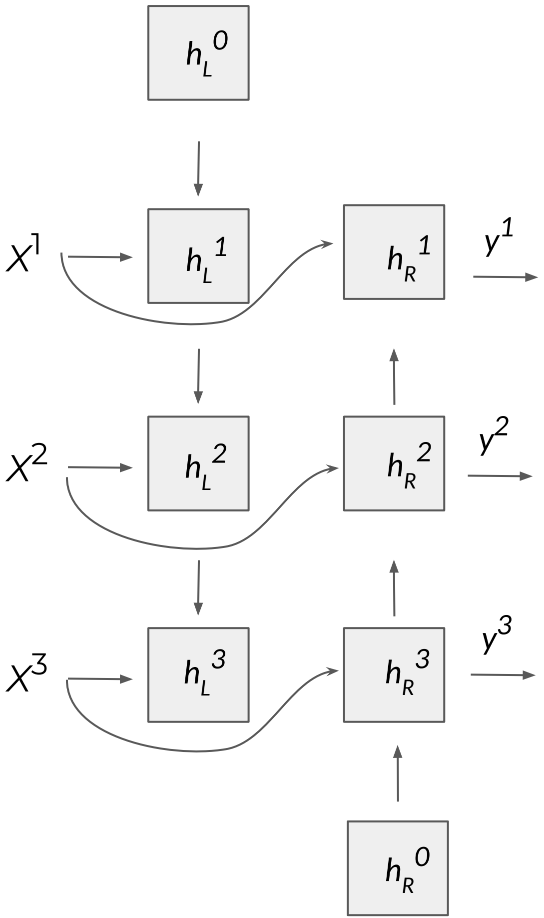


$$\overrightarrow{h^t} = \sigma\left(W^r X^t + U^r h^{t-1} + b_h^r\right)$$

$$\overleftarrow{h^t} = \sigma\left(W^l X^t + U^l h^{t-1} + b_h^l\right)$$

$$h^t = concatenate([\overrightarrow{h^t}, \overleftarrow{h^t}])$$

$$y^t = \sigma\left(V h^t + b_y\right)$$



$$h_L^t = \sigma(W_L X^t + U_L h_L^{t-1} + b_L)$$

$$h_R^t = \sigma(W_R X^t + U_R h_R^{t-1} + b_R)$$

$$h^t = [h_L^t, h_R^t]$$

$$y^t = \sigma\left(V h^t + b_y\right)$$

Итоги видео

В этом видео мы обсудили некоторые нюансы RNN:

- Проблемы затухания и взрыва градиентов;
- Выбор функции активации;
- “Забывание” сети.

А также идею борьбы с проблемой забывания: bidirectional RNN.

В следующем видео мы рассмотрим идеи устройства GRU и LSTM вариантов слоев рекуррентной сети.

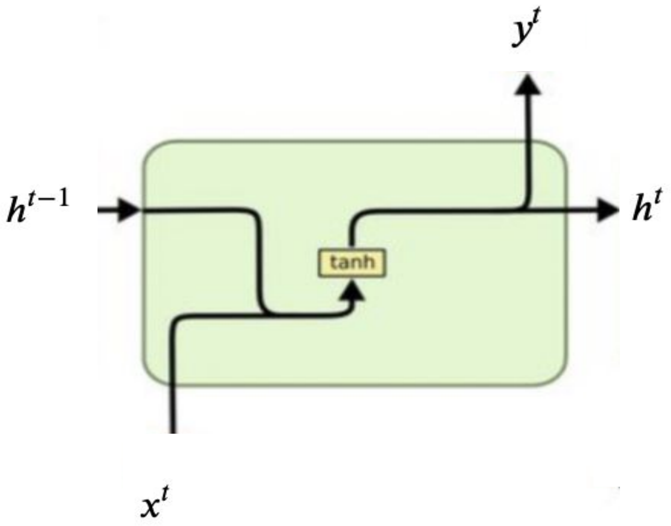
Deep Learning School

LSTM, GRU

План занятия

- Рекуррентный слой: идея.
  Рекуррентная нейросеть (RNN);
- Forward pass RNN;
- Обучение RNN (backward pass);
- Функции активации в RNN;
- Bidirectional RNN;
- **GRU, LSTM**

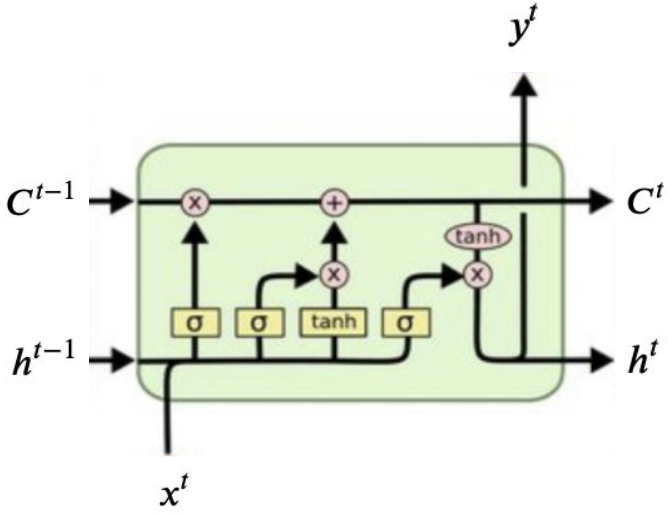
Vanilla RNN



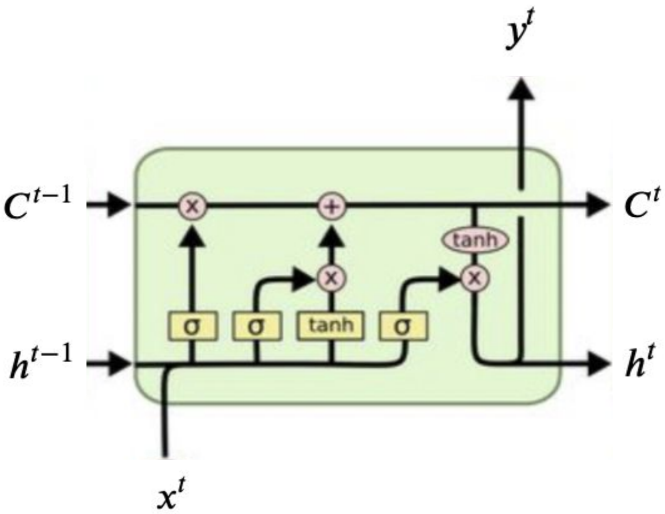
$$h^t = \tanh(WX^t + Uh^{t-1} + b_h)$$

$$y^t = \sigma(W_y h^t + b_y)$$

LSTM (Long Short Term Memory)



LSTM (Long Short Term Memory)

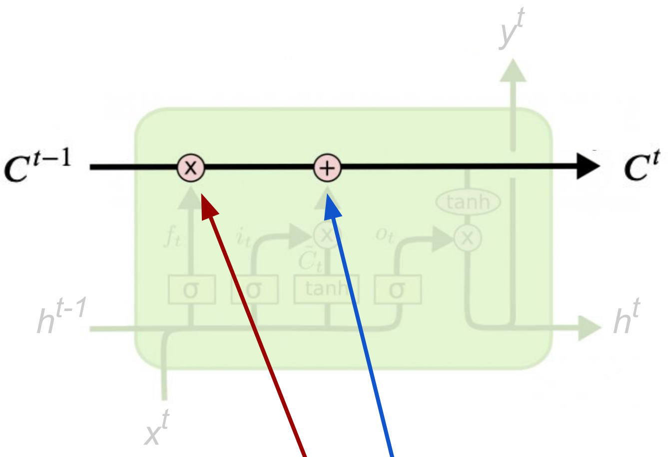


C^t (cell) — “(долгосрочная) память”

h^t — “краткосрочная память” или “текущее состояние слоя”

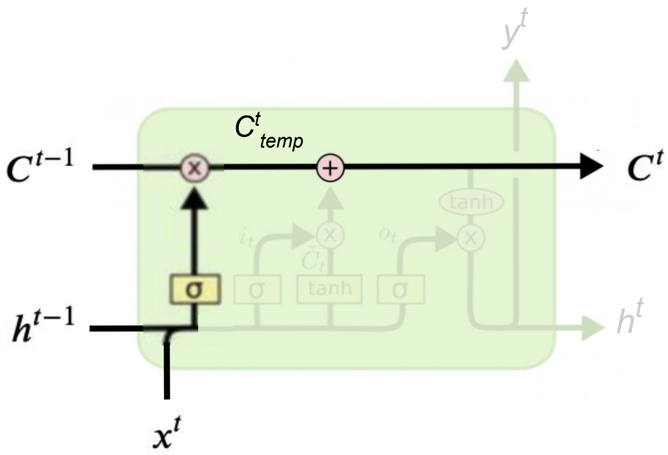
Оба вектора имеют тот же размер, что и x^t

LSTM



LSTM может удалять информацию из памяти C_t аписывать в нее новую информацию

LSTM



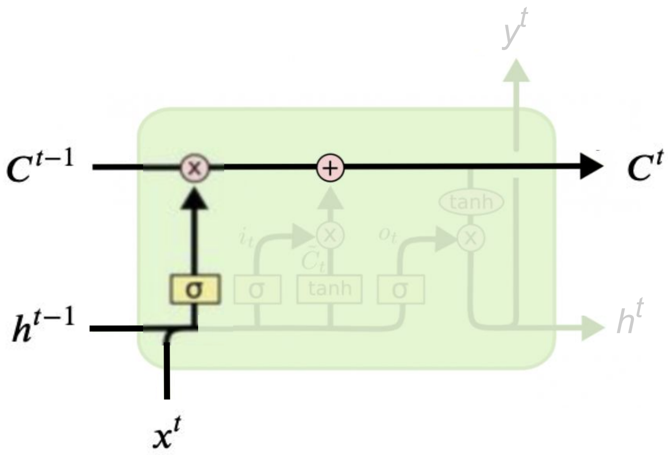
"Ворота забывания" ("forget gate"):

$$f^t = \sigma(W_f \cdot [h^{t-1}, x^t] + b_f)$$

$$C^t_{temp} = f^t * C^{t-1}$$

Берем текущее состояние краткосрочной памяти (h^{t-1}) и новую информацию, пришедшую на вход (x^t). На их основе понимаем, какую информацию из долгосрочной памяти (C^{t-1})уже можно выкинуть

LSTM



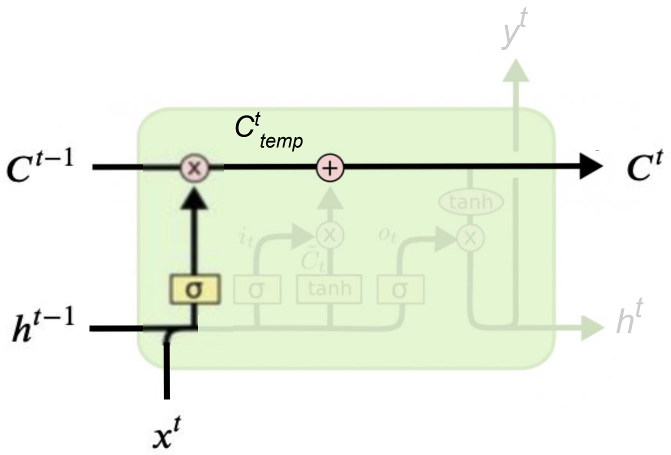
"Ворота забывания" ("forget gate"):

$$f^t = \sigma(W_f \cdot [h^{t-1}, x^t] + b_f)$$

$$C^t_{temp} = f^t * C^{t-1}$$

Берем текущее состояние краткосрочной памяти (h^{t-1}) и новую информацию, пришедшую на вход (x^t). На их основе понимаем, какую информацию из долгосрочной памяти (C^{t-1})уже можно выкинуть

LSTM



“Ворота забывания” (“forget gate”):

сигмоида

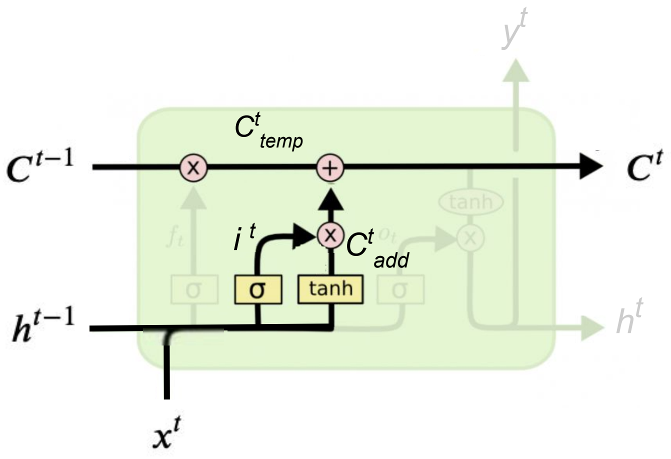
$$f^t = \sigma(W_f \cdot [h^{t-1}, x^t] + b_f)$$

Вектор значений от 0 до 1. Размер тот же, что у C^{t-1}

$$C^t_{temp} = \underbrace{f^t * C^{t-1}}$$

Каждый элемент вектора C^{t-1} умножается на значение от 0 до 1, т.е. часть информации из всех элементов исчезает

LSTM



"Ворота входа" ("input gate"):

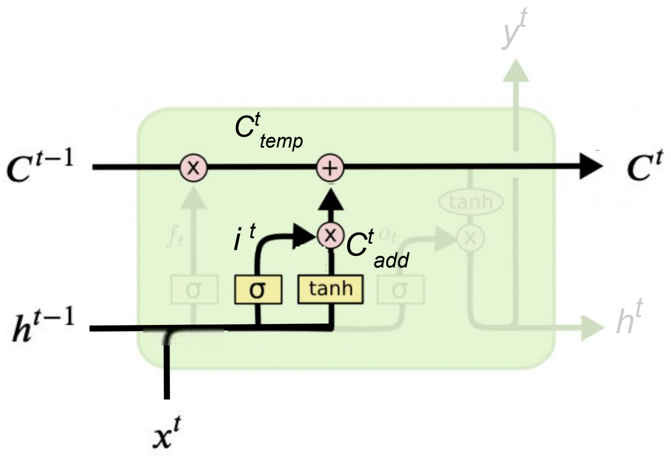
$$i^t = \sigma(W_i \cdot [h^{t-1}, x^t] + b_i)$$

$$C^t_{add} = tanh(W_C \cdot [h^{t-1}, x^t] + b_C)$$

$$C^t = C^t_{temp} + i^t * C^t_{add}$$

Берем текущее состояние краткосрочной памяти (h^{t-1}) и новую информацию, пришедшую на вход (x^t). Понимаем, какую информацию из них нам надо сохранить в долгосрочную память (C^{t-1})

LSTM



“Ворота входа” (“input gate”):

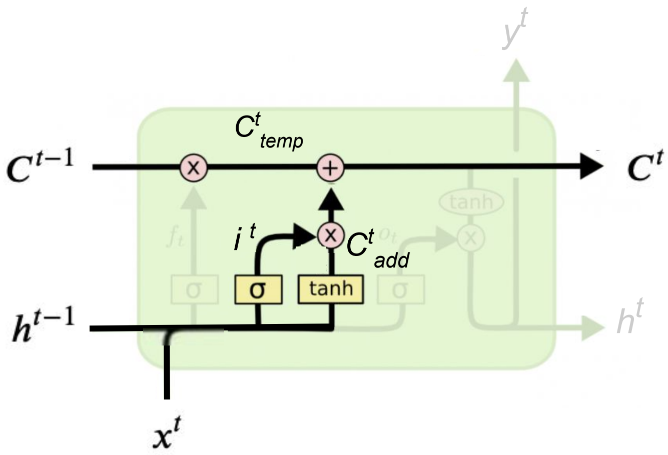
$$i^t = \sigma(W_i \cdot [h^{t-1}, x^t] + b_i)$$

$$\underbrace{C^t_{add} = tanh(W_C \cdot [h^{t-1}, x^t] + b_C)}_{}$$

$$C^t = C^t_{temp} + i^t * C^t_{add}$$

На основе h^{t-1} и x^t понимаем, какую новую информацию хотим добавить в C^t

LSTM



“Ворота входа” (“input gate”):

сигмоида

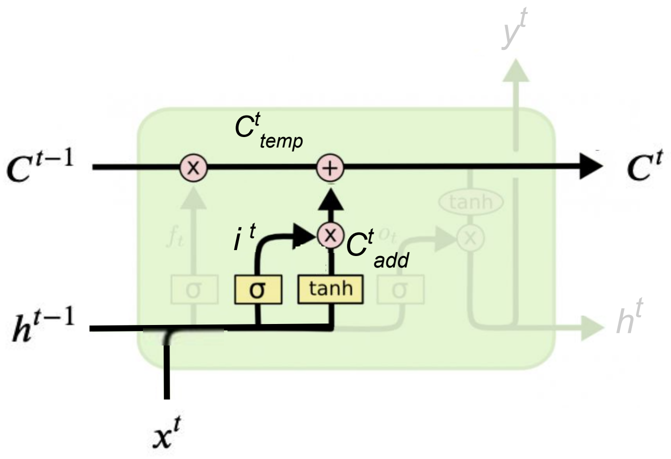
$$i^t = \sigma(W_i \cdot [h^{t-1}, x^t] + b_i)$$

Вектор значений от 0 до 1.

$$C^t_{add} = tanh(W_C \cdot [h^{t-1}, x^t] + b_C)$$

$$C^t = C^t_{temp} + i^t * C^t_{add}$$

LSTM



"Ворота входа" ("input gate"):

$$i^t = \sigma(W_i \cdot [h^{t-1}, x^t] + b_i)$$

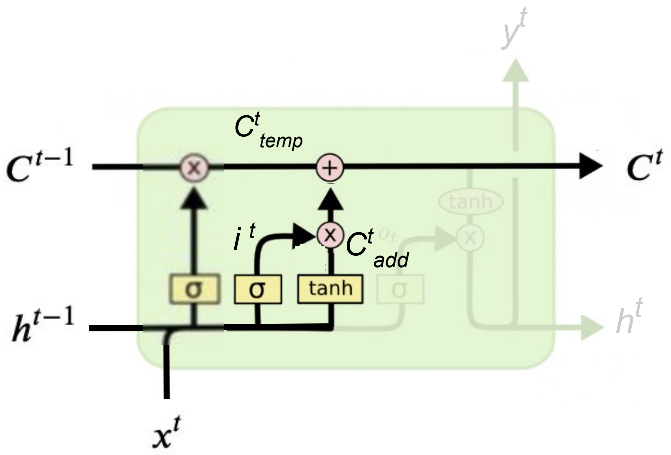
$$C^t_{add} = tanh(W_C \cdot [h^{t-1}, x^t] + b_C)$$

$$\underbrace{C^t = C^t_{temp} + i^t * C^t_{add}}$$

Добавляем в C^t новую информацию. Каждый элемент вектора умножается на число от 0 до 1: так регулируется, сколько именно этой информации поступит в вектор C^t

LSTM

Полное обновление вектора C^t:



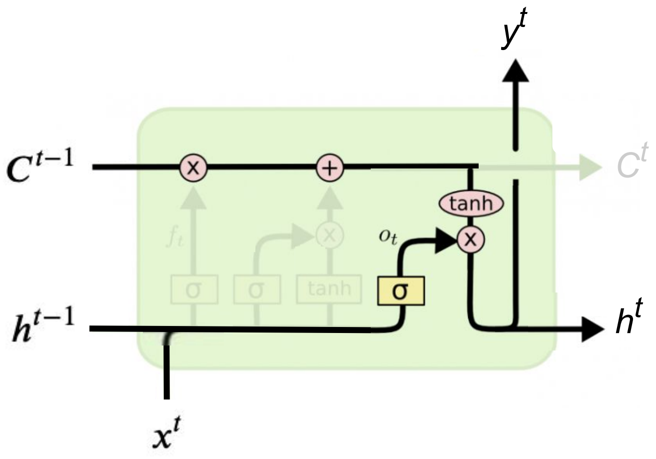
$$f^t = \sigma(W_f \cdot [h^{t-1}, x^t] + b_f)$$

$$i^t = \sigma(W_i \cdot [h^{t-1}, x^t] + b_i)$$

$$C^t_{add} = tanh(W_C \cdot [h^{t-1}, x^t] + b_C)$$

$$C^t = f^t * C^{t-1} + i^t * C^t_{add}$$

LSTM



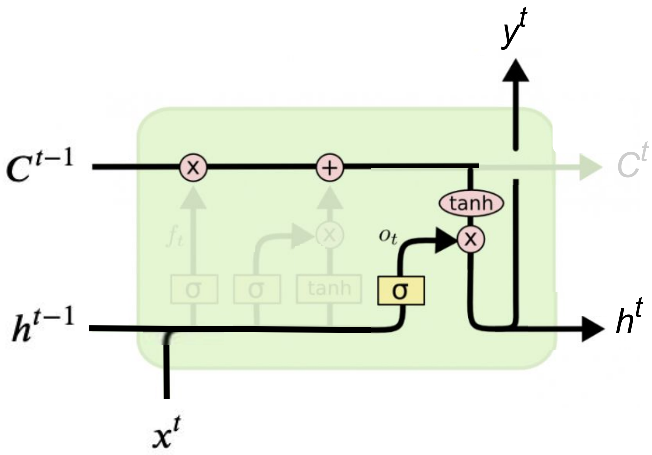
Обновление вектора h^t и получение выхода ячейки y^t :

$$o^t = \sigma(W_o \cdot [h^{t-1}, x^t] + b_o)$$

$$h^t = o^t * tanh(C^t)$$

$$y^t = \sigma(W_y h^t + b_y)$$

LSTM



Обновление вектора h^t и получение выхода ячейки y^t :

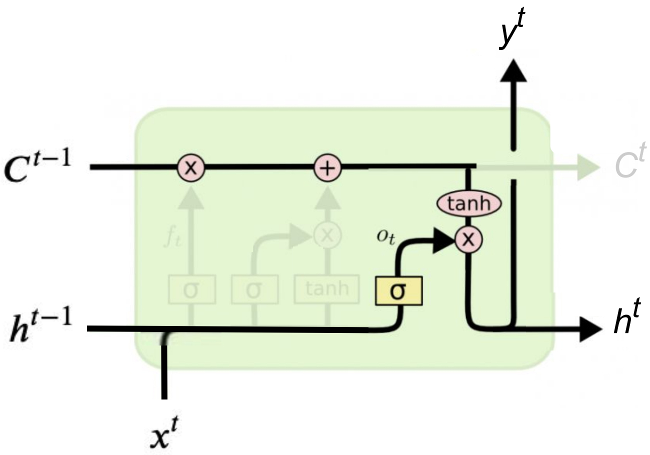
$$o^t = \sigma(W_o \cdot [h^{t-1}, x^t] + b_o)$$

$$h^t = o^t * tanh(C^t)$$

$$y^t = \sigma(W_y h^t + b_y)$$

Переносим часть информации из долгосрочной памяти (C^t) в краткосрочную (h^t) и получаем ответ

LSTM



Обновление вектора h^t и получение выхода ячейки y^t :

сигмоида

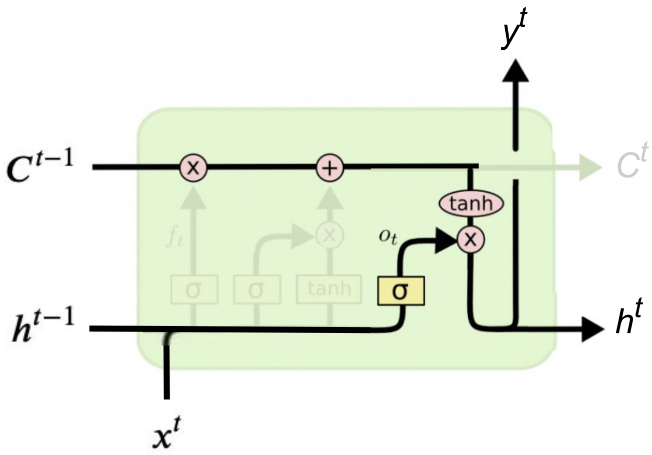
$$o^t = \sigma(W_o \cdot [h^{t-1}, x^t] + b_o)$$

Вектор из 0 и 1

$$h^t = o^t * tanh(C^t)$$

$$y^t = \sigma(W_y h^t + b_y)$$

LSTM



Обновление вектора h^t и получение выхода ячейки y^t :

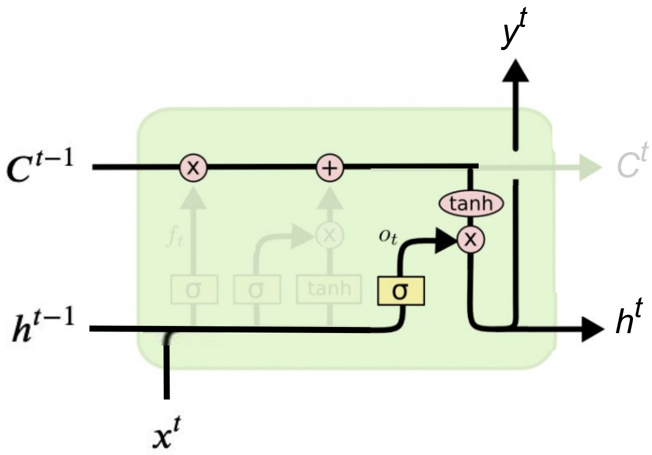
$$o^t = \sigma(W_o \cdot [h^{t-1}, x^t] + b_o)$$

$$\underbrace{h^t = o^t * tanh(C^t)}$$

$$y^t = \sigma(W_y h^t + b_y)$$

Часть информации из C^t переносится в краткосрочную память

LSTM



Обновление вектора h^t и получение выхода ячейки y^t :

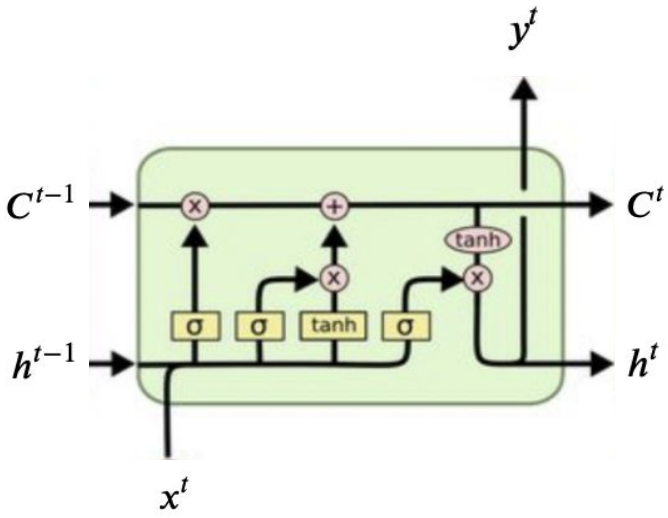
$$o^t = \sigma(W_o \cdot [h^{t-1}, x^t] + b_o)$$

$$h^t = o^t * tanh(C^t)$$

$$\underbrace{y^t = \sigma(W_y h^t + b_y)}$$

Считаем выход ячейки

LSTM



$$f^t = \sigma(W_f \cdot [h^{t-1}, x^t] + b_f)$$

$$i^t = \sigma(W_i \cdot [h^{t-1}, x^t] + b_i)$$

$$C^t_{add} = tanh(W_C \cdot [h^{t-1}, x^t] + b_C)$$

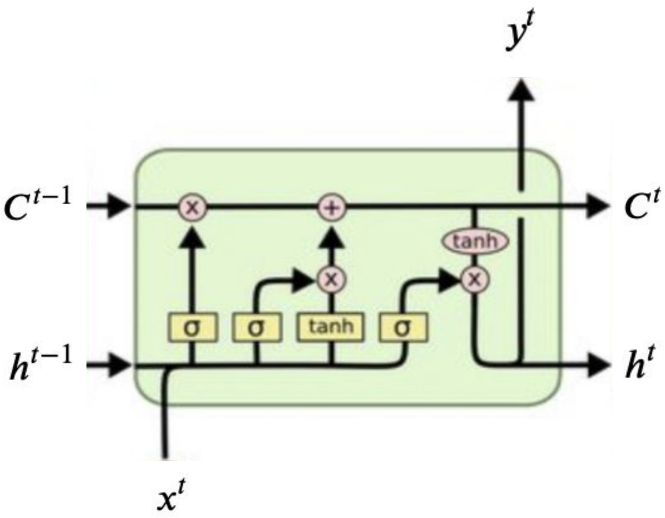
$$C^t = f^t * C^{t-1} + i^t * C^t_{add}$$

$$o^t = \sigma(W_o \cdot [h^{t-1}, x^t] + b_o)$$

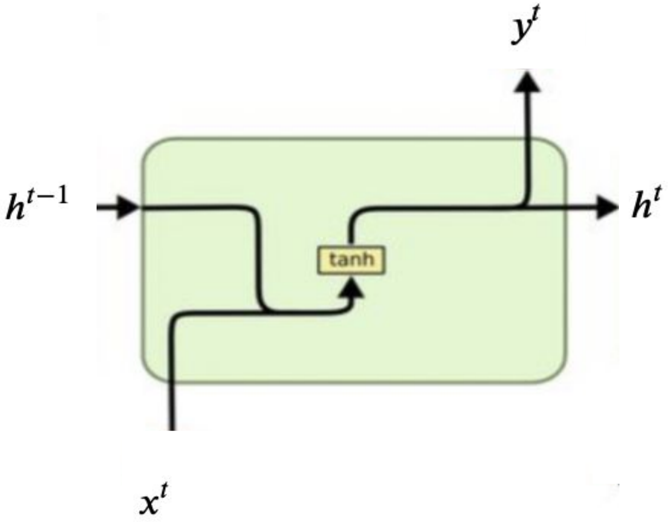
$$h^t = o^t * tanh(C^t)$$

$$y^t = \sigma(W_y h^t + b_y)$$

LSTM

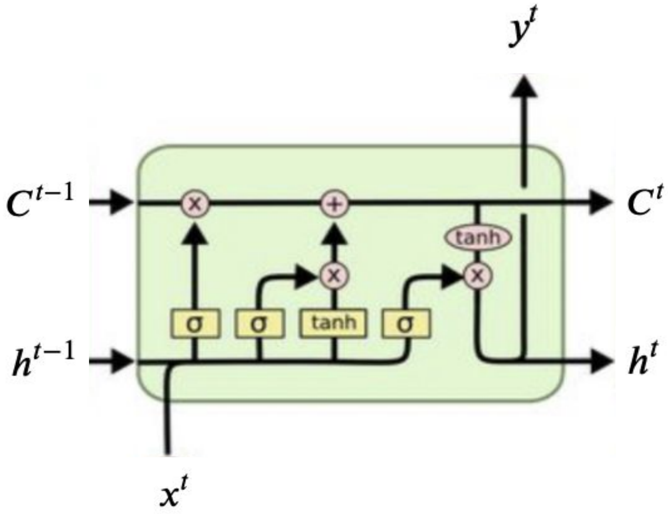


Vanilla RNN



$$h^t = \tanh(WX^t + Uh^{t-1} + b_h)$$

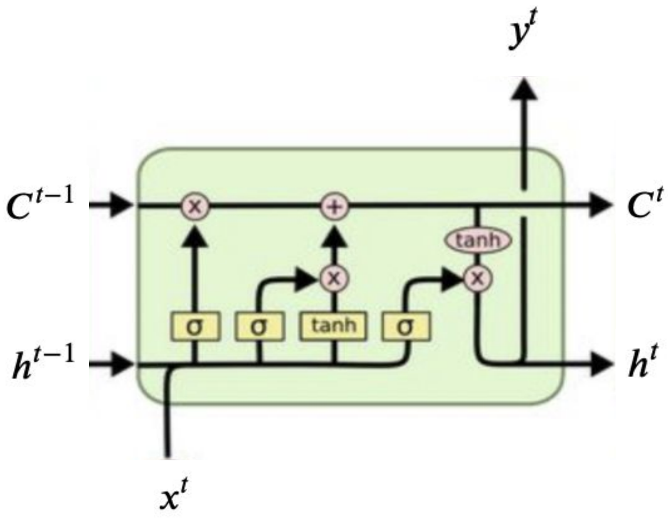
LSTM



LSTM реализует механизм, похожий на skip connection в ResNet.

Это помогает в борьбе с затуханием градиентов.

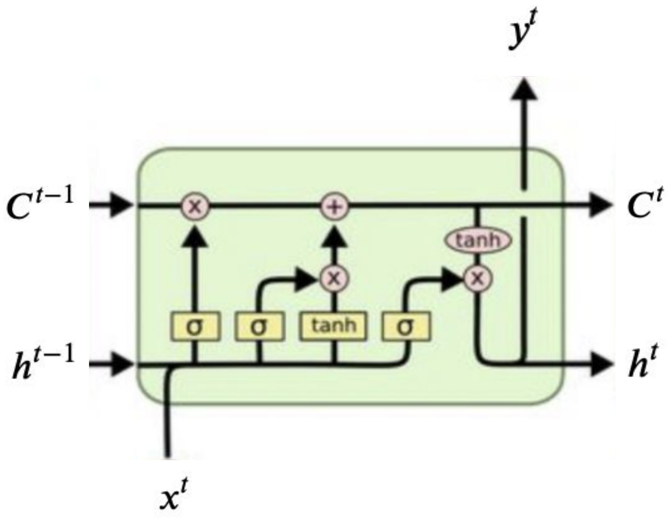
LSTM



Пример: пусть мы решаем задачу классификации текста по грамматической правильности (правильный/неправильный текст с точки зрения грамматики)

“Человек в шляпе с красивым попугаем на плече, которого вчера видели на пересечении Косого переулка и улицы Роз, зашел в бар”

LSTM

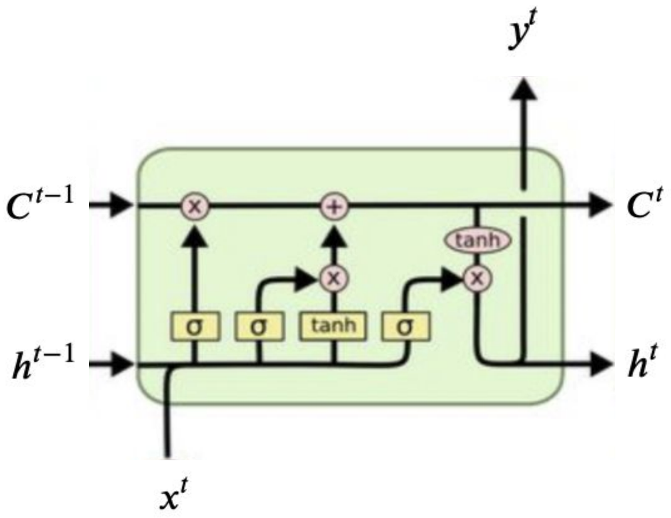


Пример: пусть мы решаем задачу классификации текста по грамматической правильности (правильный/неправильный текст с точки зрения грамматики)

“Человек в шляпе с красивым попугаем на плече, которого вчера видели на пересечении Косого переулка и улицы Роз, зашел в бар”

Когда мы стоим на слове “человек”, мы должны добавить в память C^t информацию об этом слове. Например, что оно мужского рода.

LSTM

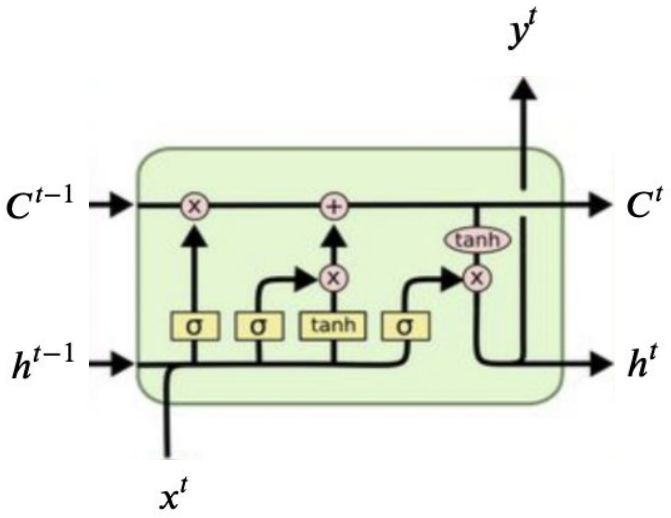


Пример: пусть мы решаем задачу классификации текста по грамматической правильности (правильный/неправильный текст с точки зрения грамматики)

“Человек в шляпе с красивым попугаем на плече, которого вчера видели на пересечении Косого переулка и улицы Роз, зашел в бар”

Когда мы стоим на слове “красивым”, мы также должны добавить в память C^t информацию об этом слове. Что оно мужского рода.

LSTM

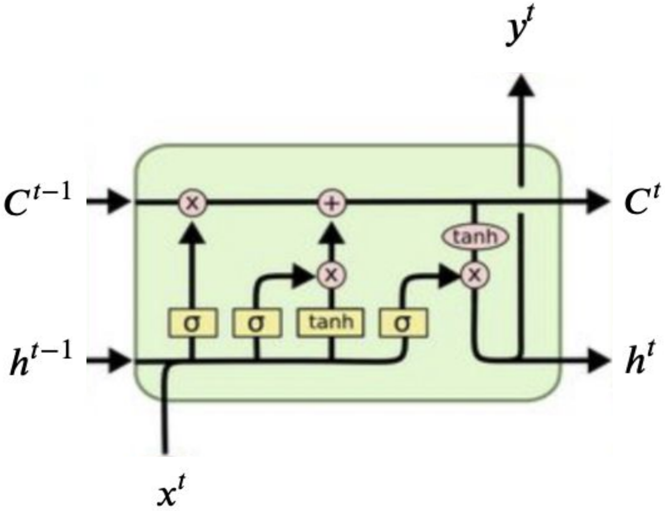


Пример: пусть мы решаем задачу классификации текста по грамматической правильности (правильный/неправильный текст с точки зрения грамматики)

“Человек в шляпе с красивым попугаем на плече, которого вчера видели на пересечении Косого переулка и улицы Роз, зашел в бар”

Когда мы стоим на слове “попугаем”, то при формировании вектора h^t из C^t мы хотим вытащить информацию о форме слова “красивым”, но не хотим добавлять информацию о форме слова “человек”

LSTM

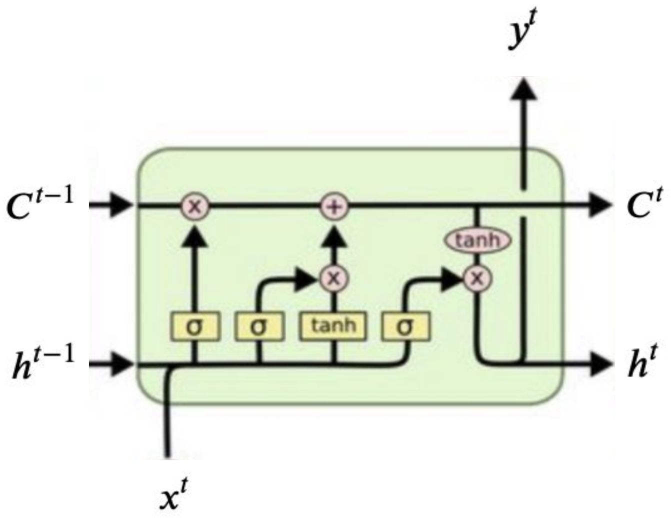


Пример: пусть мы решаем задачу классификации текста по грамматической правильности (правильный/неправильный текст с точки зрения грамматики)

“Человек в шляпе с красивым попугаем на плече, которого вчера видели на пересечении Косого переулка и улицы Роз, зашел в бар”

Когда мы стоим на слове “на”, то мы можем удалить из C^t информацию о словах “красивым попугаем”, и добавить информацию о слове “на”

LSTM

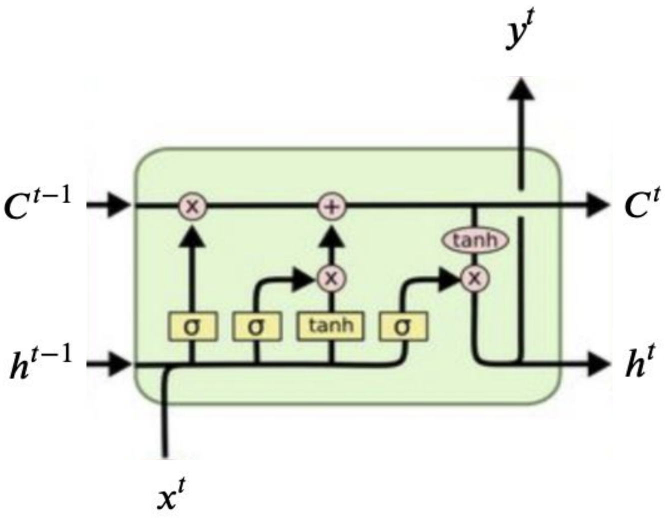


Пример: пусть мы решаем задачу классификации текста по грамматической правильности (правильный/неправильный текст с точки зрения грамматики)

“Человек в шляпе с красивым попугаем на плече, которого вчера видели на пересечении Косого переулка и улицы Роз, зашел в бар”

Когда мы стоим на слове “плече”, то при формировании вектора h^t из C^t мы хотим вытащить информацию о форме слова “на”, но не хотим добавлять информацию о форме слова “человек”

LSTM

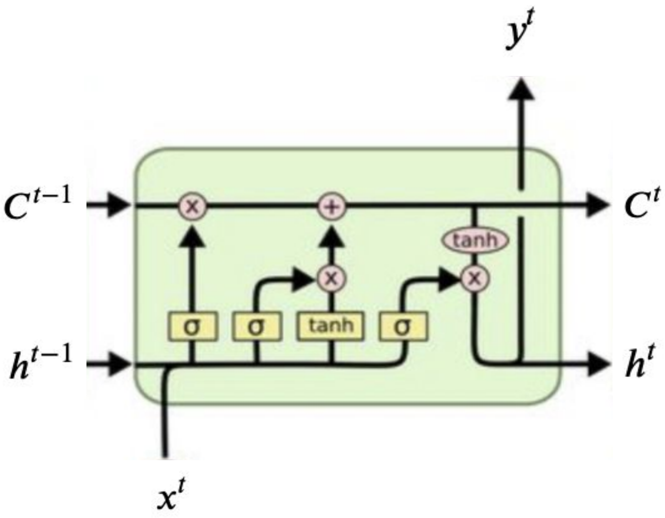


Пример: пусть мы решаем задачу классификации текста по грамматической правильности (правильный/неправильный текст с точки зрения грамматики)

“Человек в шляпе с красивым попугаем на плече, которого вчера видели на пересечении Косого переулка и улицы Роз, зашел в бар”

Когда мы стоим на слове “плече”, то при формировании вектора h^t из C^t мы хотим вытащить информацию о форме слова “на”, но не хотим добавлять информацию о форме слова “человек”

LSTM

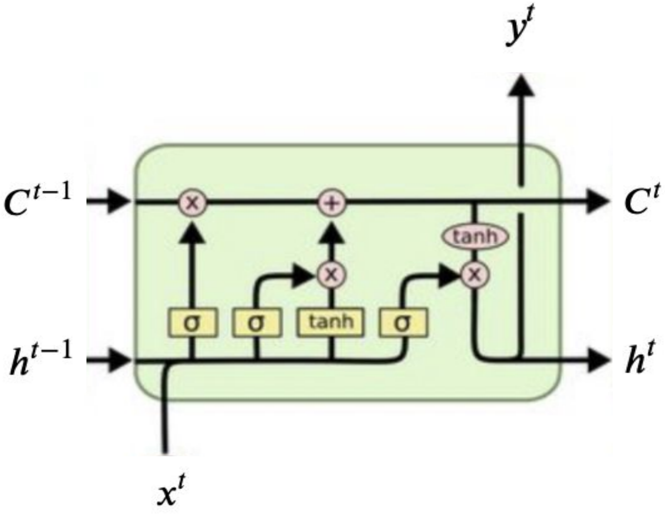


Пример: пусть мы решаем задачу классификации текста по грамматической правильности (правильный/неправильный текст с точки зрения грамматики)

“Человек в шляпе с красивым попугаем на плече, которого вчера видели на пересечении Косого переулка и улицы Роз, зашел в бар”

Когда мы стоим на запятой, мы должны добавить в память информацию о том, что начался причастный оборот

LSTM

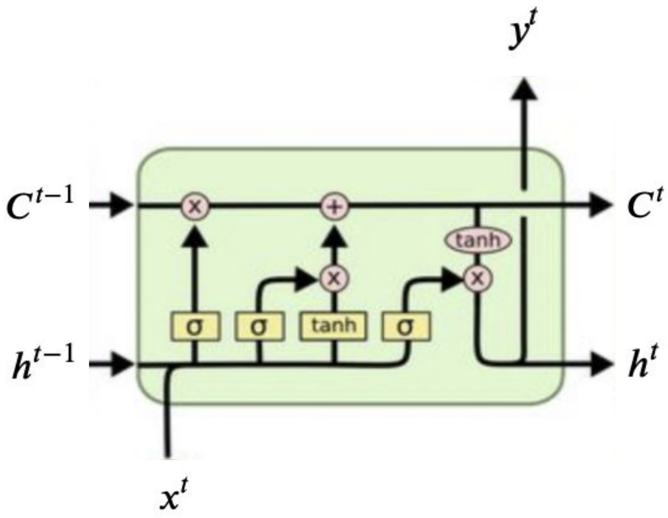


Пример: пусть мы решаем задачу классификации текста по грамматической правильности (правильный/неправильный текст с точки зрения грамматики)

“Человек в шляпе с красивым попугаем на плече, которого вчера видели на пересечении Косого переулка и улицы Роз, зашел в бар”

Когда мы стоим на слове “которого”, то при формировании вектора h^t из C^t мы хотим вытащить информацию о форме слове “человек” и о том, что сейчас идет причастный оборот

LSTM



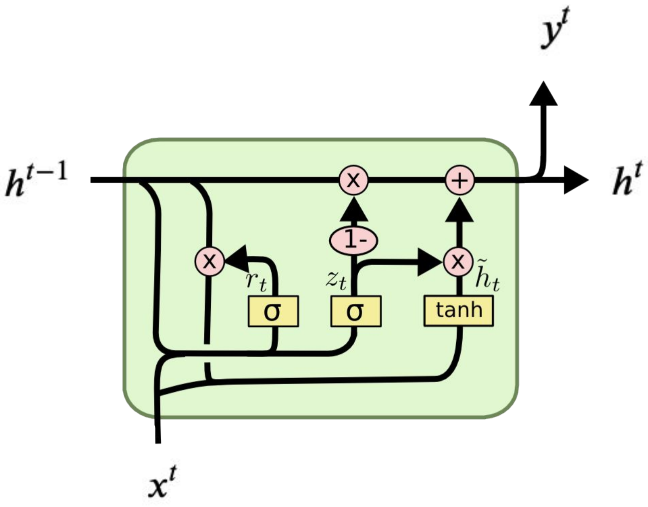
Пример: пусть мы решаем задачу классификации текста по грамматической правильности (правильный/неправильный текст с точки зрения грамматики)

“Человек в шляпе с красивым попугаем на плече, которого вчера видели на пересечении Косого переулка и улицы Роз, зашел в бар. После ...”

Когда мы стоим на слове “после”, то мы можем удалить из C^t информацию о предыдущем предложении, и добавить информацию о слове “после”

GRU

“Облегченный вариант” LSTM



$$z^t = \sigma(W_z \cdot [h^{t-1}, x^t] + b_z)$$

$$r^t = \sigma(W_r \cdot [h^{t-1}, x^t] + b_r)$$

$$h^t_{add} = tanh(W \cdot [r^t * h^{t-1}, x^t])$$

$$h^t = (1 - z^t) * h^{t-1} + z^t * h^t_{add}$$

Итоги видео

В этом видео мы познакомились с идеей устройства новых рекуррентных слоев: LSTM и GRU.