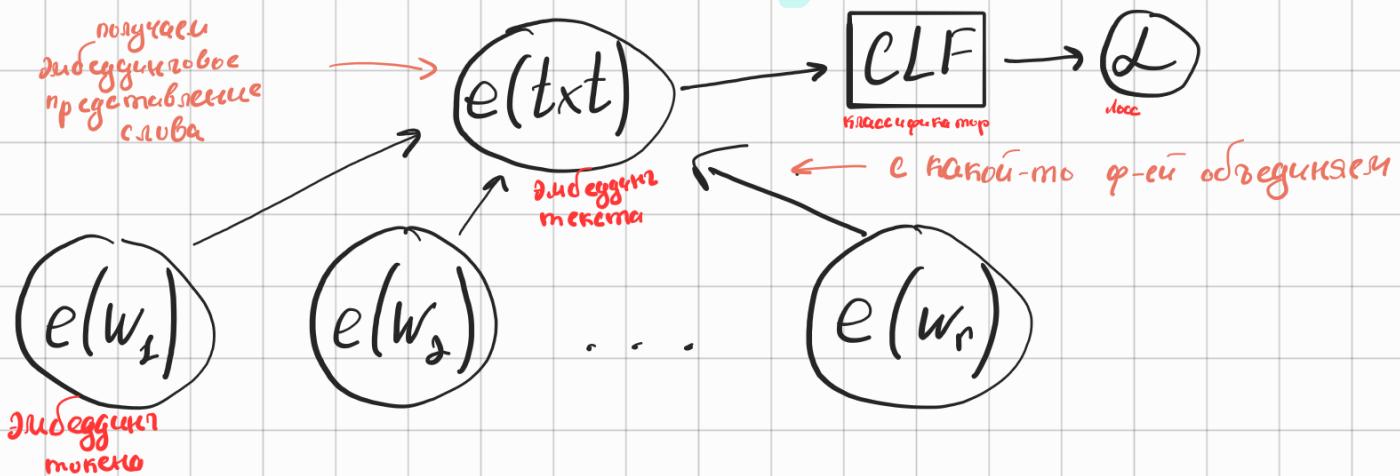


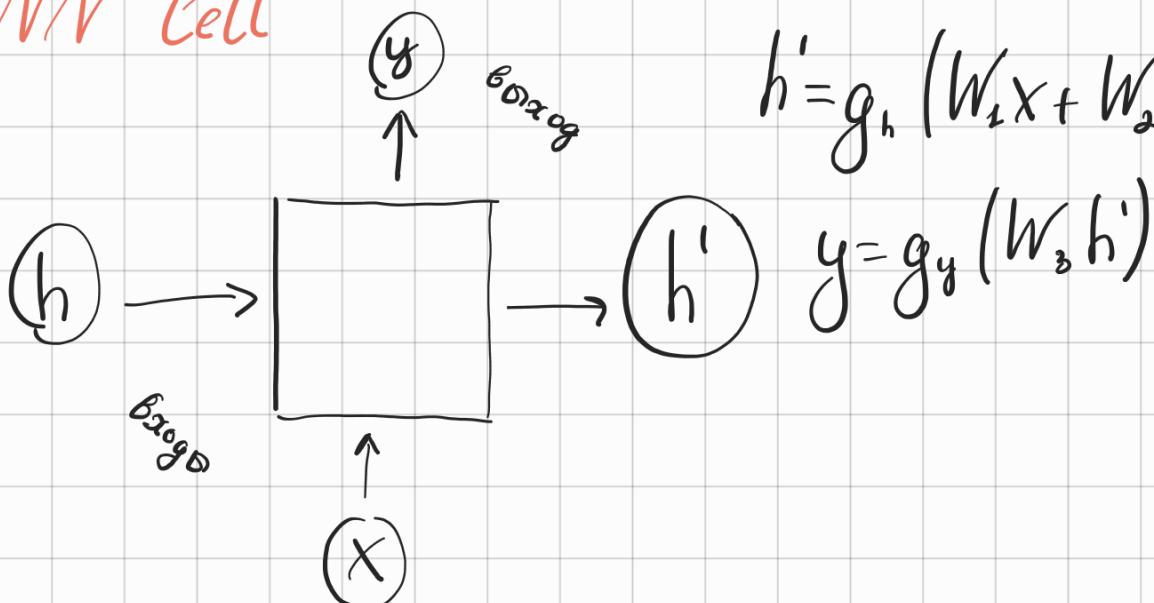
# Лекция 7. Рекуррентные нейронные сети

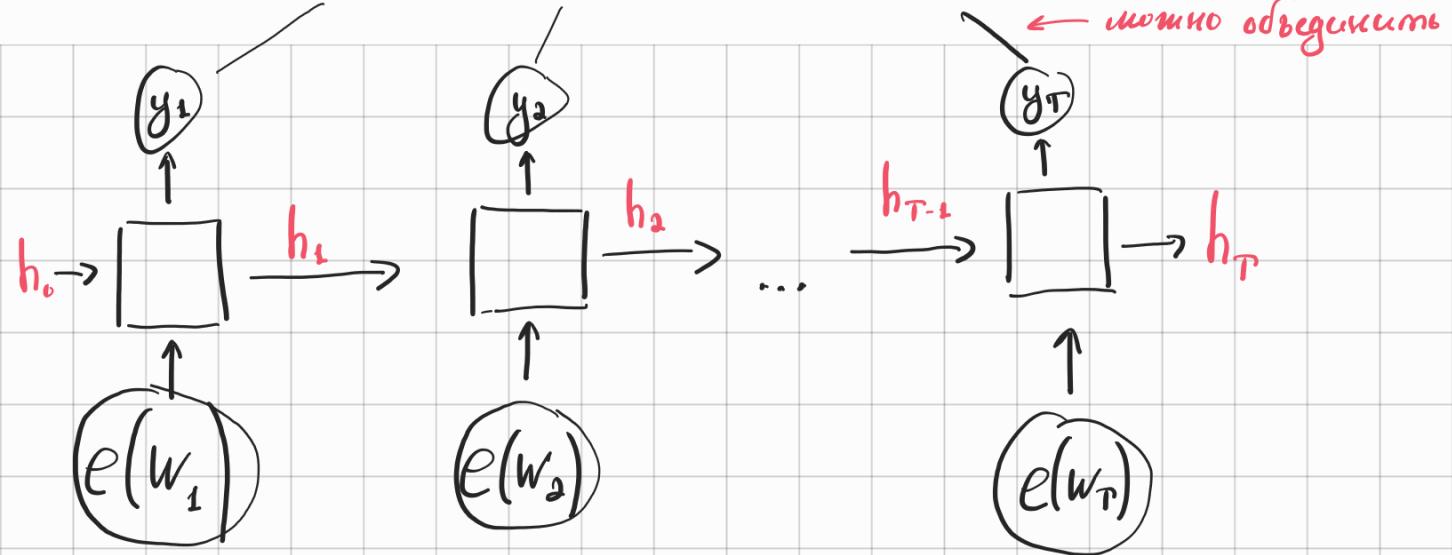


Родник съел чук

Проблемы: текущая архитектура не учитывает контекст, из-за которого может <sup>как ошибки</sup> изменяться смысл слова

RNN Cell



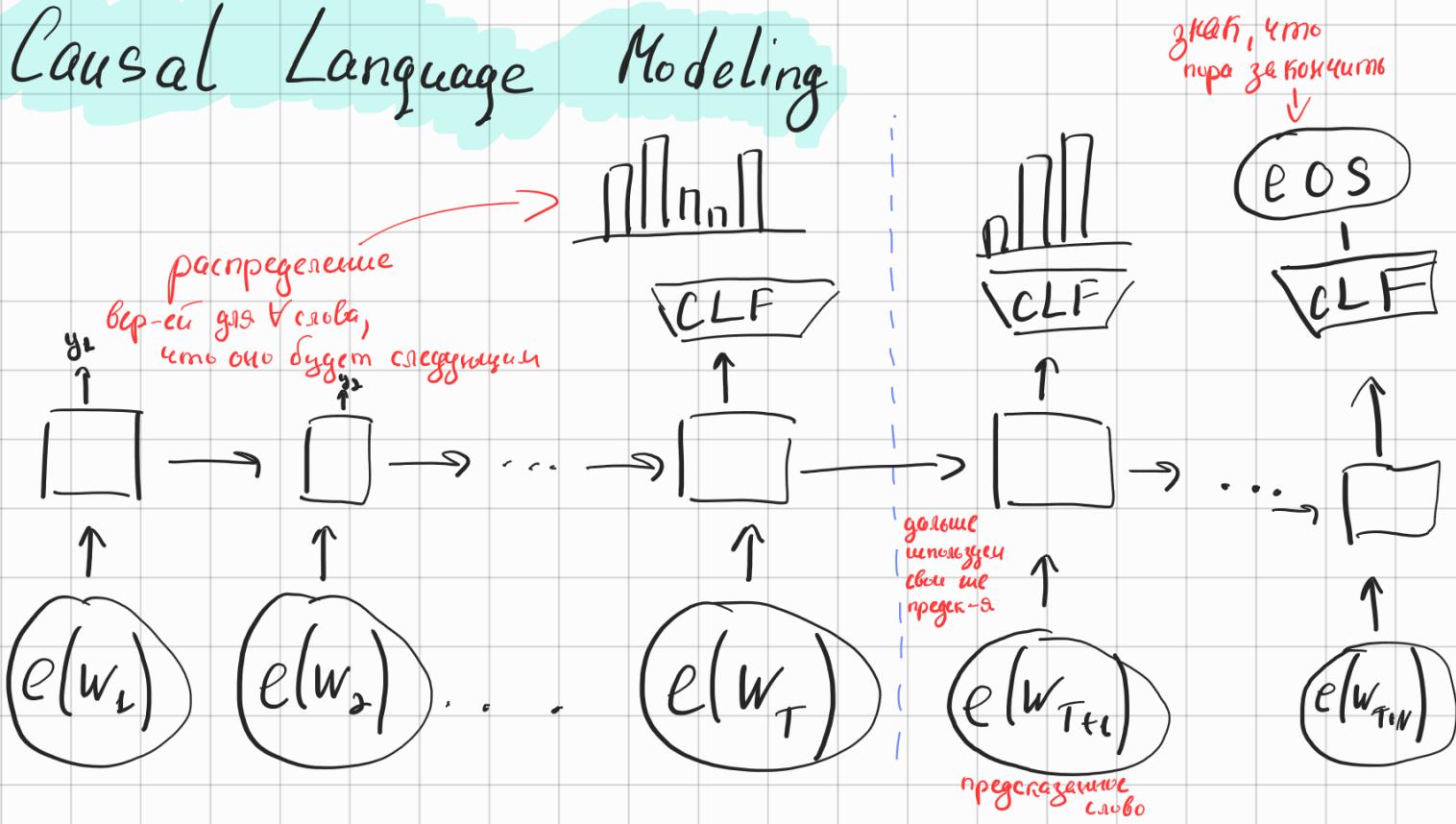


RNN блоки одинаковы, имеют разделяемые параметры  
(в этом плане похожи на CNN)

Удаётся научиться учитывать контекст, благодаря

принципу работы RNN Cell

## Causal Language Modeling



Однако в Питере здесь предсказателем

Авторегрессионный итеративный

Как обучить?

"Однако в Питере ..."

$$\begin{matrix} w_1 \dots w_N \\ \left[ w_1, \dots, w_{N-1} \right] = \left[ y_1, \dots, y_{N-1} \right] \end{matrix}$$

последовательность векторов  
↓  
без заголовка RNNок

Дальше подаем в классификатор. Используем правило:

$\{w_2, \dots, w_N\} \leftarrow$  хотим, чтобы CLF на  $[y_1, \dots, y_{N-1}]$  предсказал их

## Named Entity Recognition

Именованные сущности (named entity) — адреса, чай, бремя и прочее

Хотим извлечь их из текста

Если хотим извлечь члены, можно сделать всё так же, но теперь CLF будет заниматься одинарной классификацией.

Если хотим адреса (ул. Строителей 9), то можно сделать

Уме классы - то же з класса, BIO разметка - Beginning, Inside, Outside (B I Стройтесь (g O ...))

Если несколько сущностей, то используем многоклассовую

BIO-разметку, то есть теперь будем несколько разных типов B u I

## Sequence-to-Sequence Modeling

много объектов  
один объект

CLF - Many-to-one

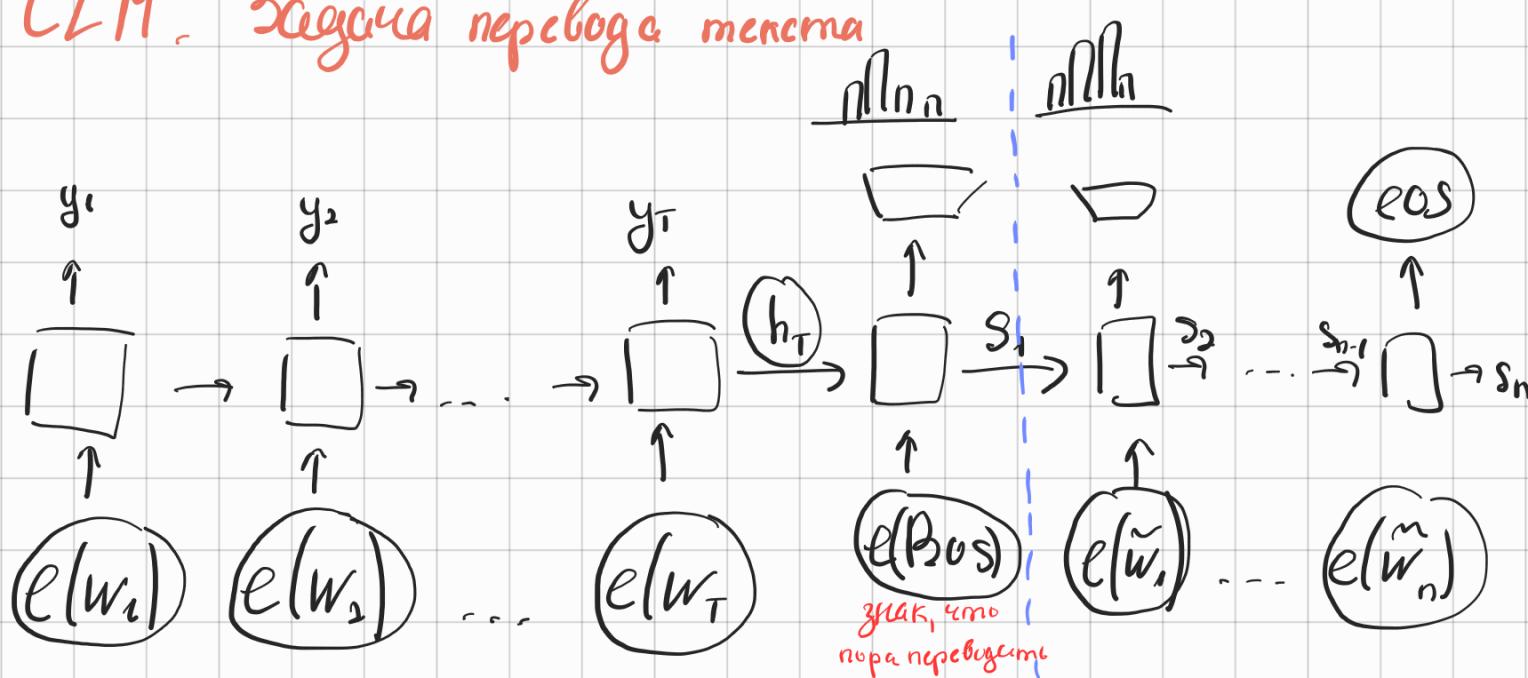
на конкретный  
объект

NER - Many-to-many

CLM - Many-to-many

S2S - Many-to-many

CLM. Задача перевода текста



English

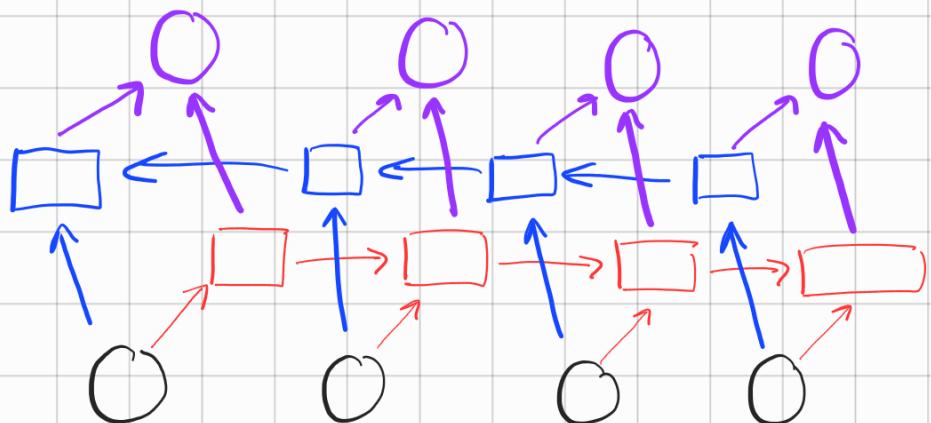
Похожие на UNet

need language

Интересно, что  $e(w_i)$ -могут быть как какие-то заранее обученные, полученные энSEMBЛИ, так и случайные векторы, т.к. модель сама обучит энSEMBЛИ

## Модификации RNN

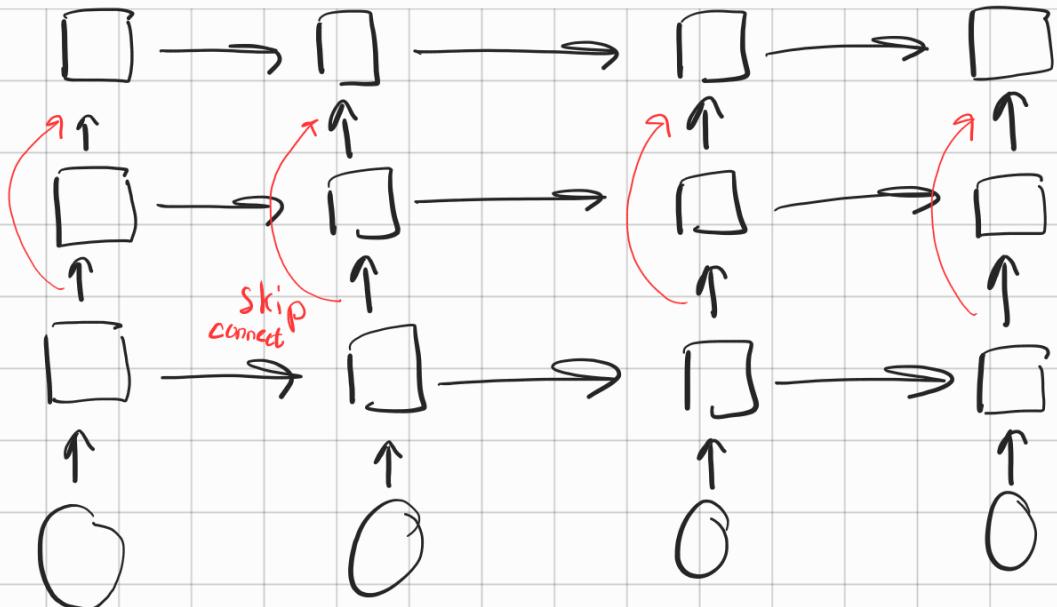
Токены знают только о своих соседях слева. Но слова определяются и правыми контекстами, что надо учитывать



Bidirectional RNN. Не подходит для задачу предсказания след. токена

Многослойное RNN + Skip Connection

Инициализация как в MLP



## Регуляризация RNN

### Напоминание Batch Norm.

Зачем:

- + кебольшой шум даёт статистики по мини-батчам
- исправляет разные масштабы признаков
- не даёт градиентов взрываться/затухать (учитывает этот факт)

задачи

активации лучше работают с такими нормир. данными, которые  
„скользят“

$(B, H)$

$$y_{ih} = \gamma_h \frac{x_{ih} - \mu_h}{\sqrt{\sigma_h^2 + \epsilon}} + \delta_h$$

$$\mu_h = \frac{1}{B} \sum_{i=1}^B x_{ih}$$

$$\sigma_h^2 = \frac{1}{B} \sum_{i=1}^B (x_{ih} - \mu_h)^2$$

задачи с картами

$(B, C, H, W)$

мерким

$(B, C, T)$

↑  
група  
слова

↑  
група  
момента

$$\mu_h = \frac{1}{BT} \sum_{i=1}^B \sum_{t=1}^T x_{i,t,h}$$

↓  
 сумма  
+  
норма

$$\sigma_h^2 = \frac{1}{BT} \sum_{i=1}^B \sum_{t=1}^T (x_{i,t,h} - \mu_h)^2$$

Почему же незадогум Batch Norm' гэс мөрсөв?

•  $\mu, \sigma^2$ -сүлжесийн нэгэн тохиолдлыг одошиг,  $T$ -мөрсөн

Дэл мөрсөнүүд нэг незадогум Batch Norm

## Layer Norm

$$\mu_{it} = \frac{1}{C} \sum_{k=1}^C x_{itk}$$

↓  
 признаковы  
размерности

$$\sigma_{it}^2 = \frac{1}{C} \sum_{k=1}^C (x_{itk} - \mu_{it})^2$$

То сэто счимаси  
теперь чисто бодь признаковы  
размерности все стандартны

Договоримся с взвешами/затуханием градиентов

## Drop Out

(6 гэ)

## Tokenization

Токен = символ / слово

- длинный вход
- + короткий вход
- + маленький слово
- большой слово

Byte-pair encoding — идея с прошлой (6) лекции

В RNN есть проблема затухающих градиентов, потому что обычно много слоёв, поэтому вычисления получаются большими в глубину

LSTM (long short term memory)

символа, таким образом все числа от 0 до 1

Forget gate:  $f = \sigma(W^f h_{t-1} + U^f x_t + b^f)$

input gate:  $i = \sigma(W^i h_{t-1} + U^i x_t + b^i)$

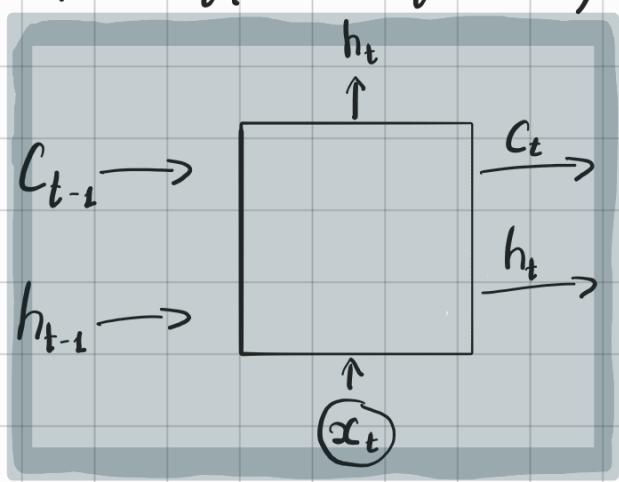
output gate:  $o = \sigma(W^o h_{t-1} + U^o x_t + b^o)$

*получим конечное пересечение векторов*

$$g_t = \tanh(W^g h_{t-1} + U^g x_t + b^g)$$

$C_t = f \circ C_{t-1} + i \circ g_t$

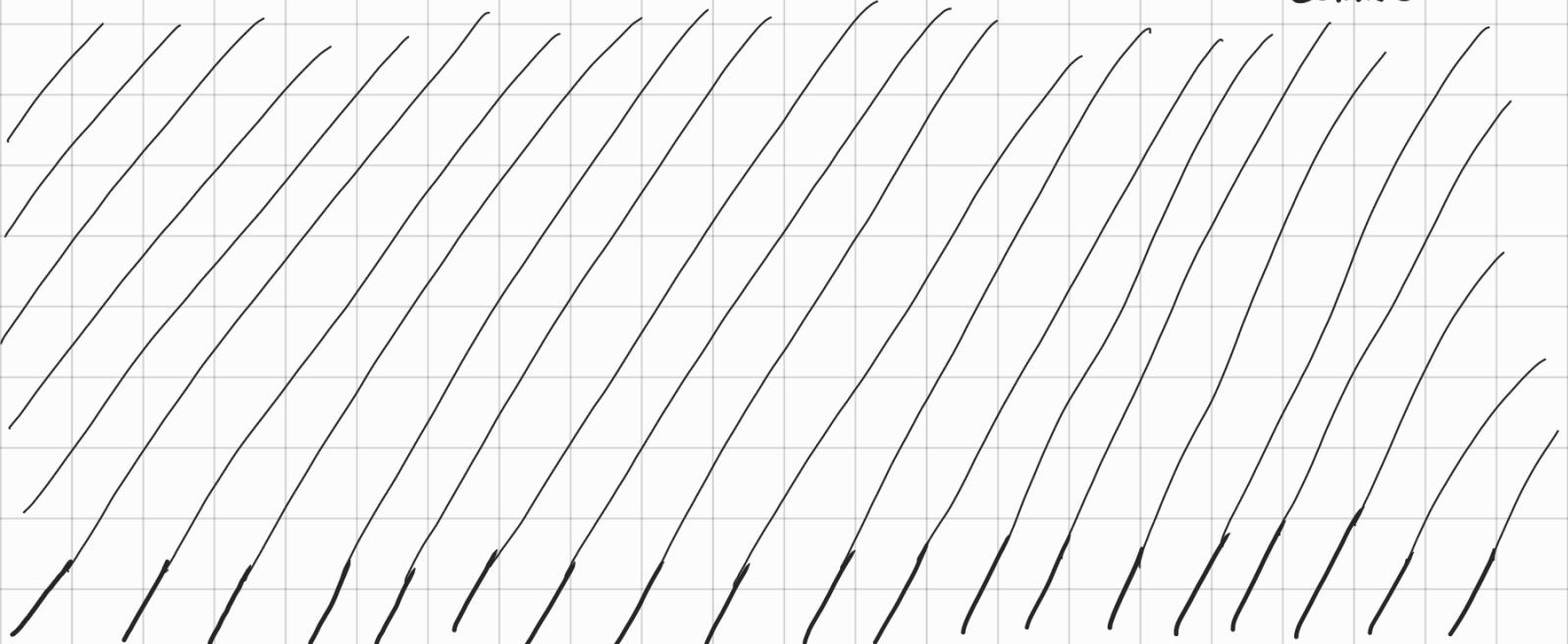
$h_t = o \circ \tanh(C_t)$



$C_t$ :

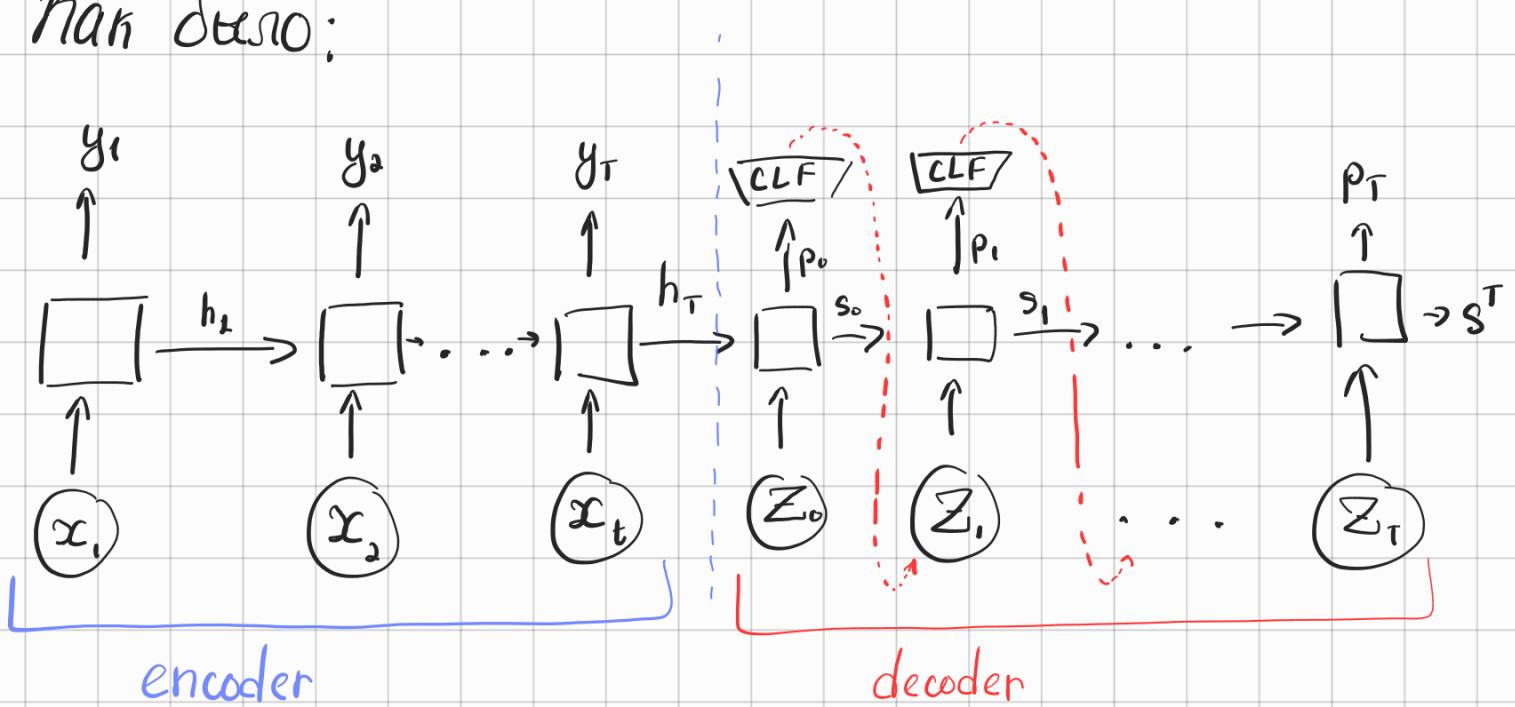
- Вектор контекста, служит роль долгой памяти
- $g$ - "главная" активация
- $C_0 = 0$  - соответствует пустой памяти
- $i$  - интерпретируется как индикатор, указывает важность информации. Если близок к 1, то считают, что информацию надо запоминать. И наоборот
- $f$  - какимто путем забывать старую информацию
- $C_t$  сам по себе - обновленная репрезентация  $X_t$   
 $h_t$  - видение, полученное инфр-я для след. ачейки

При добавлении  $f \circ C_{t-1}$  неявно реализует идентичное skip connection



# Механизм внимания

Как это:



$$s_t = g(W_1 z_t + W_2 s_{t-1})$$

$$p_t = g_y(W_3 s_t)$$

ногу касн

$\downarrow$

$z_{t+1}$

$$z_t := h_t$$

Проблема: модель не способна выдать слова из исходн. посл-ти, которые обращаются только к  $s$ , чего недостаточно

$$S_t = g_s(W_1 z_t + W_2 s_{t-1} + W_u d_t)$$

$$d_t = \sum_{i=1}^T y_i \cdot d_{ti}$$

то есть, на каждом шаге модель определяет, с каким весом слова из исходной посл-ти ей полезно

$$d_{ti} = \frac{\exp(\text{sim}(s_{t-1}, y_i))}{\sum_{k=1}^T \exp(\text{sim}(s_{t-1}, y_k))}$$

характеризует текущий контекст  
 ↓

нейронная сеть однотипна для каждого слова

$$\text{sim}(x, y) = \langle x, y \rangle$$

большая sim  
 W можно добавить

то это слово сейчас полезно, его надо постараться.