

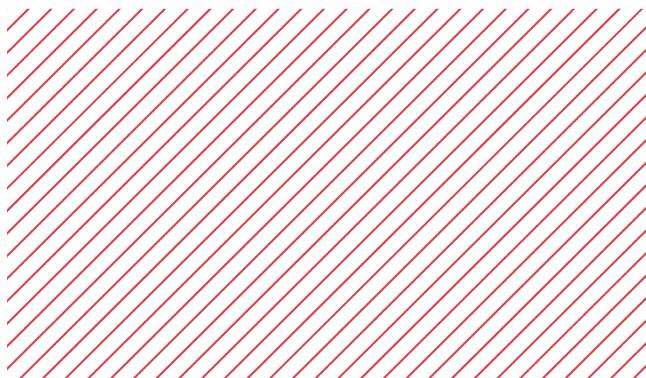
академия
больших
данных



Распознавание текста

Иван Карпухин

Ведущий программист-исследователь в
команде машинного зрения





План лекции

- Пайплайн OCR
- Моделирование последовательностей (RNN, CRNN, Self-Attention)
- Обучение: CTC loss
- Языковое моделирование
- Beam search

Пайплин OCR

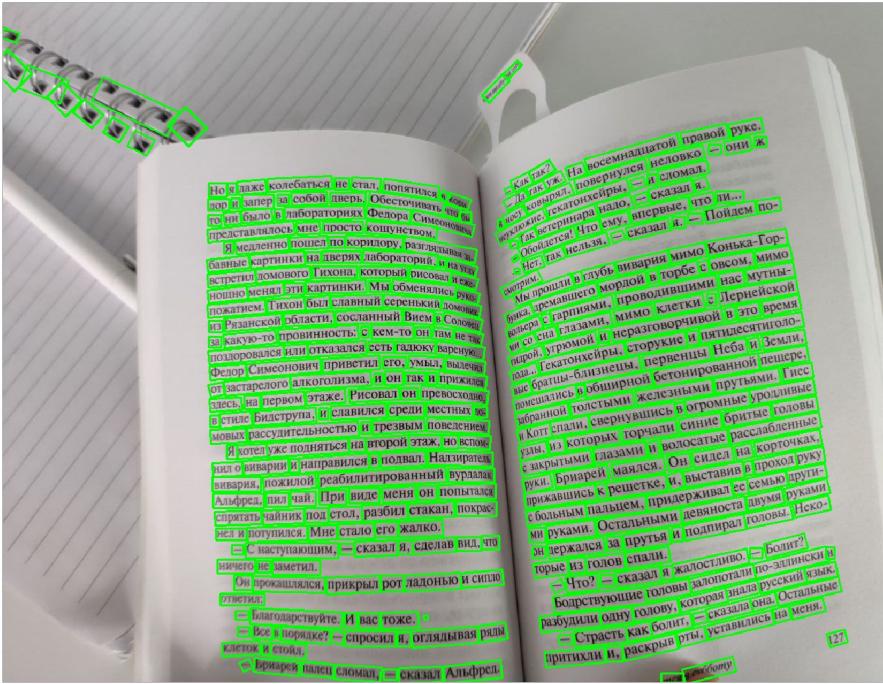
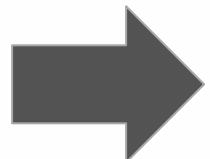


Схема распознавания текста

Считаем, что каждое слово хорошо описывается прямоугольником



Детектирование

Распознавание
и форматирование

Liberals
Please continue on 1-40
until you have left our
GREAT STATE OF
TEXAS
BURKETT

Детектирование текста

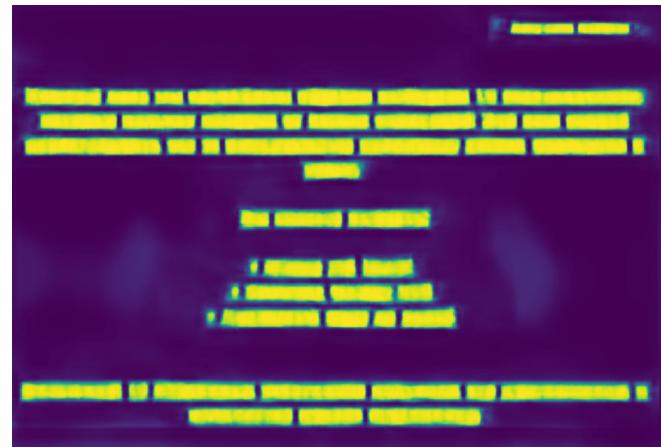
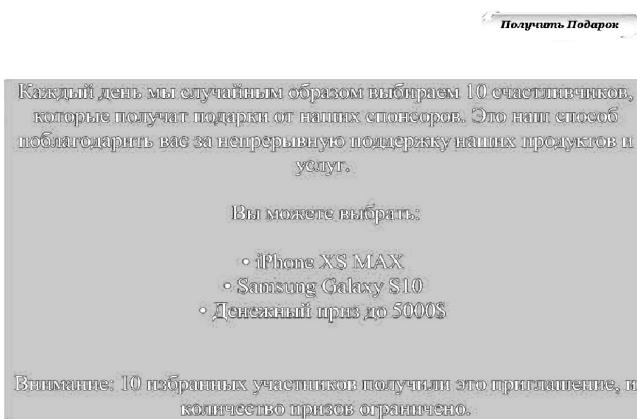
Первый путь: детектировать bounding boxes
(R-CNN, YOLO)

- Слова могут быть вытянуты и повернуты
- Плохое попадание в anchor boxes или receptive field



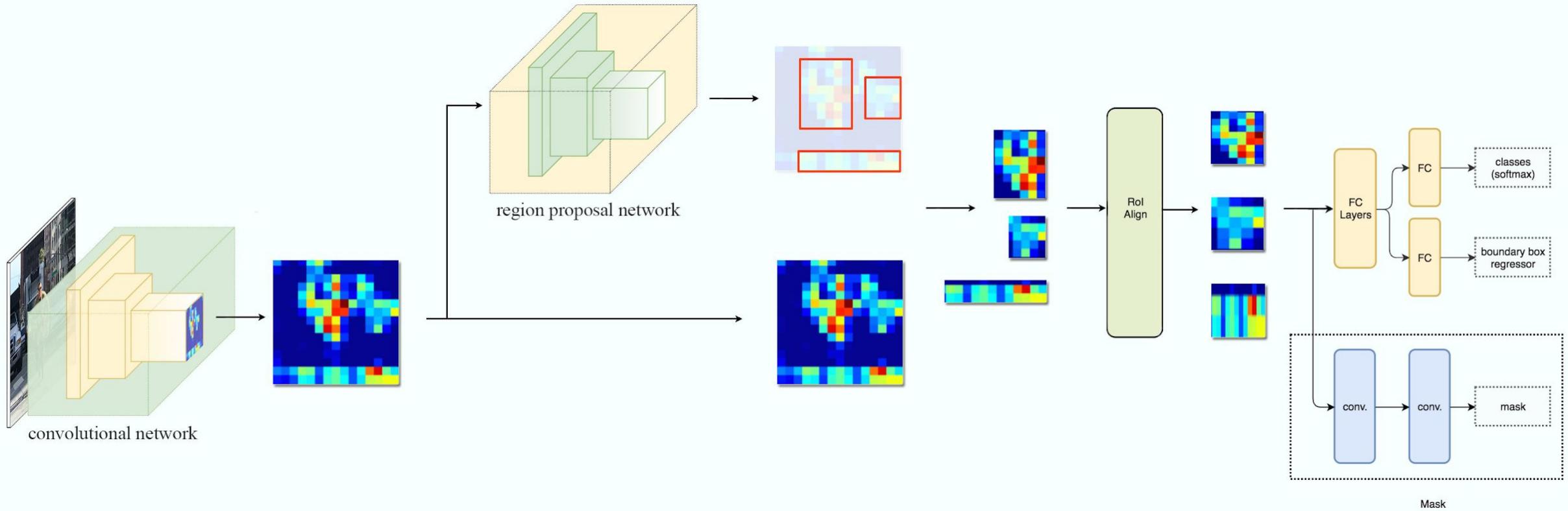
Детектирование текста

Второй путь: сегментировать текст и строить bbox по маске
(Mask R-CNN, FPN, U-net)

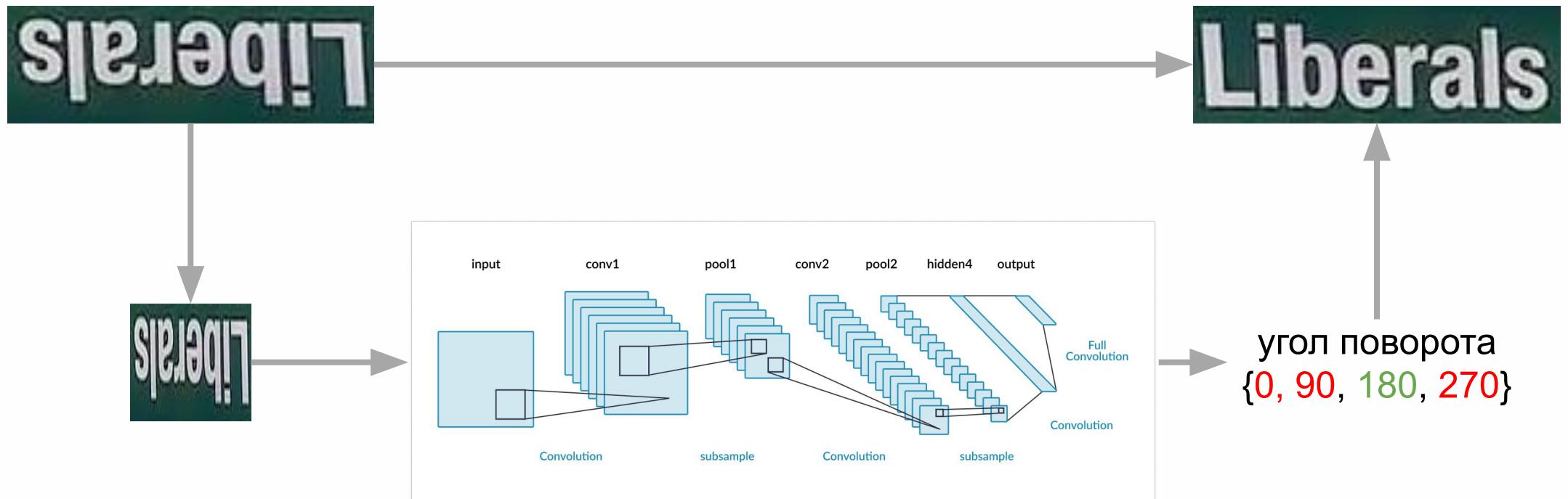


Mask R-CNN

Дополнительная голова для Faster R-CNN



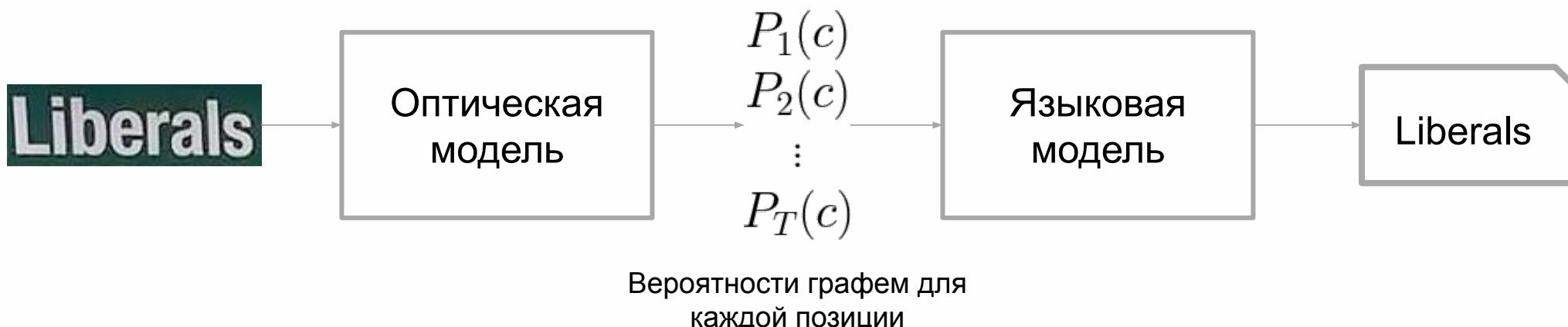
Выравнивание ориентации



Распознавание текста

Декомпозиция оптической и языковой модели

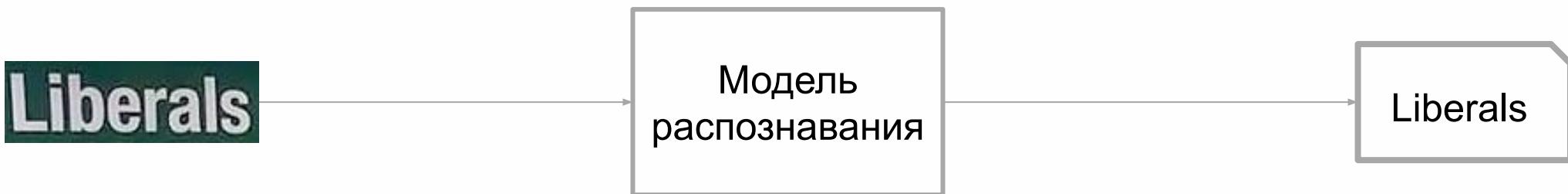
- (+) Языковую модель можно учить на текстах (без картинок)
- (-) Фиксированный список графем - узкое место в передаче информации



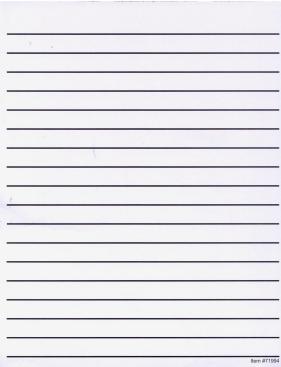
Распознавание текста

End-to-end

- (+) Потенциально лучше качество
- (-) Нужно больше размеченных изображений
- (-) Труднее адаптировать к новым задачам



Подготовка синтетических данных

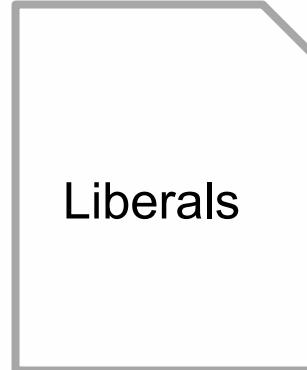


Фон



Times New Roman
Times New Roman Italic
Times New Roman Bold
Times New Roman Bold Italic

Шрифт



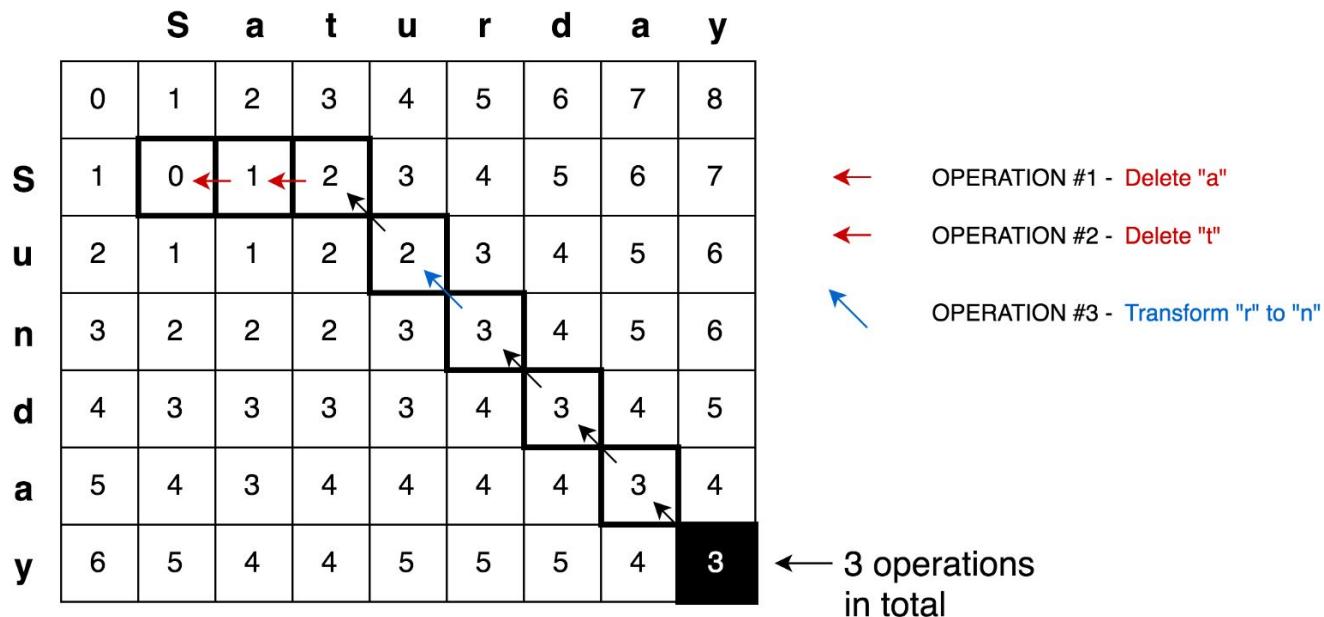
Текст



Аугментации

Оценка OCR

Редакционное расстояние (Левенштейна)

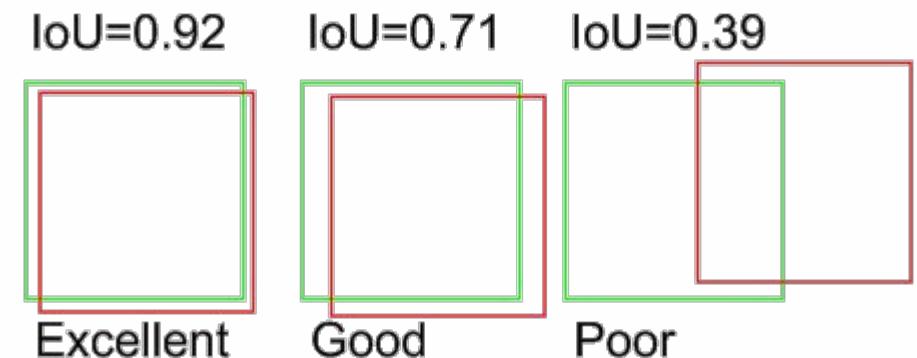
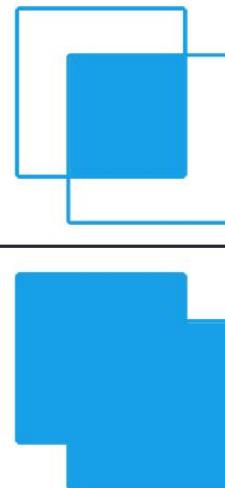


* <https://github.com/trekhleb/javascript-algorithms/tree/master/src/algorithms/string/levenshtein-distance>

Оценка OCR

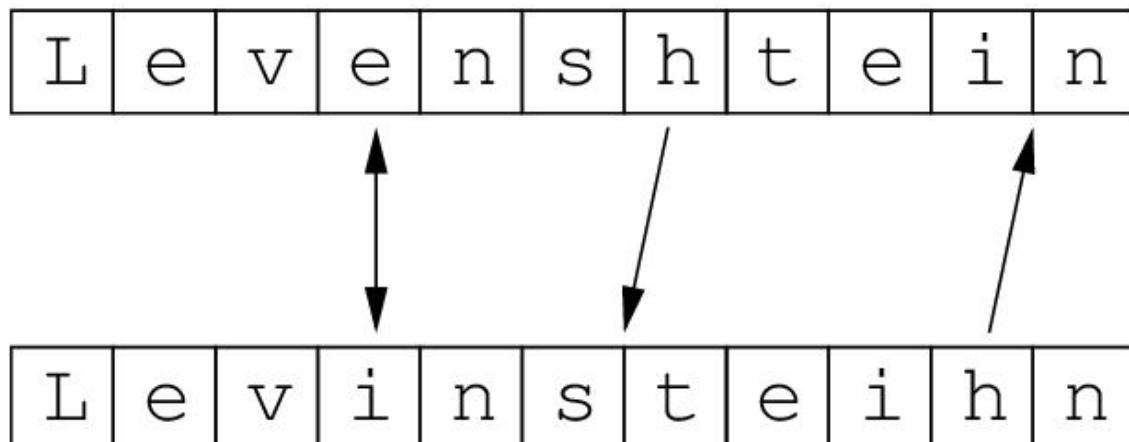
- Детекция
 - mAP
 - Dice

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



Оценка OCR

- Детекция
 - mAP
 - Dice
- Распознавание
 - Доля неверно распознанных символов (CER)
 - Доля неверно распознанных слов (WER)





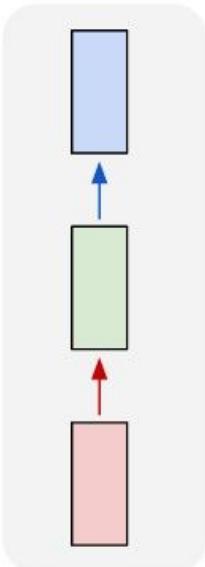
Оценка OCR

- Детекция
 - mAP
 - Dice
- Распознавание
 - Доля неверно распознанных символов (CER)
 - Доля неверно распознанных слов (WER)
- Общие метрики

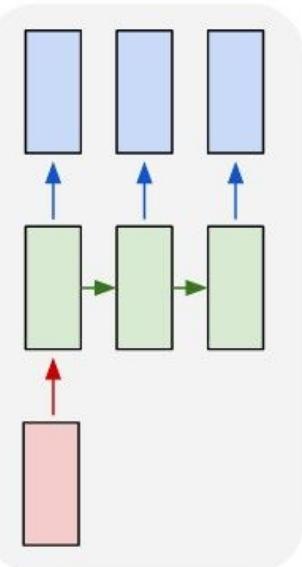
Моделирование последовательностей

Рекуррентные сети

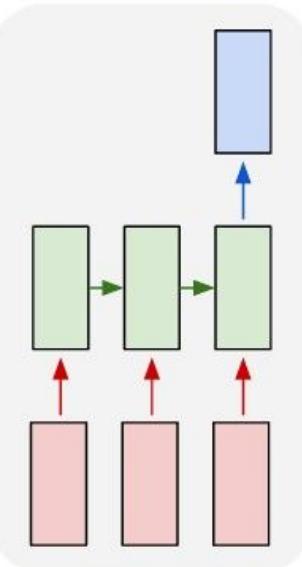
one to one



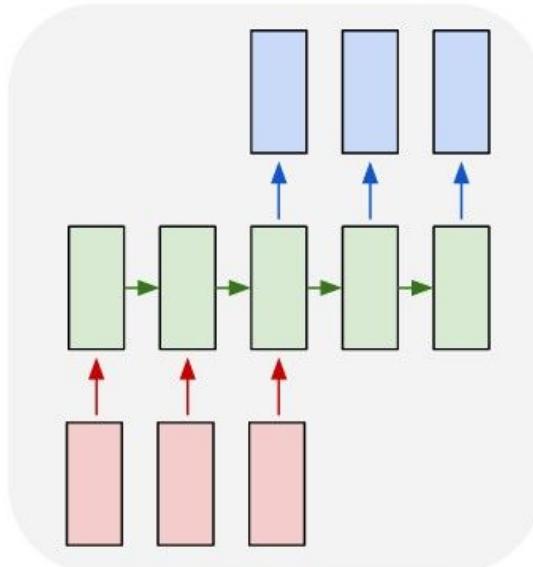
one to many



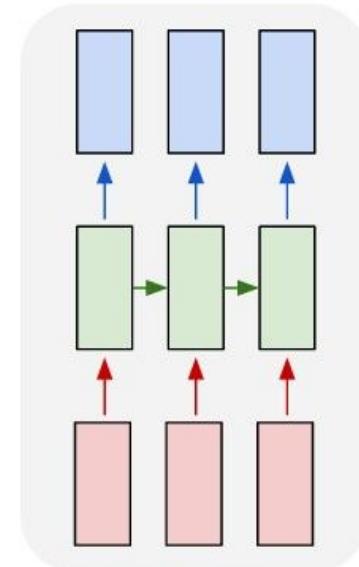
many to one



many to many

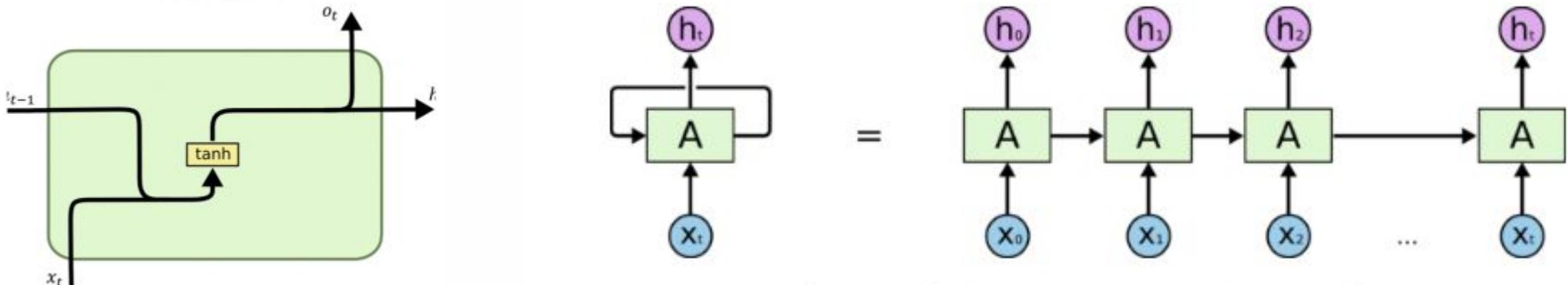


many to many



Рекуррентные сети

RNN: сеть с состоянием

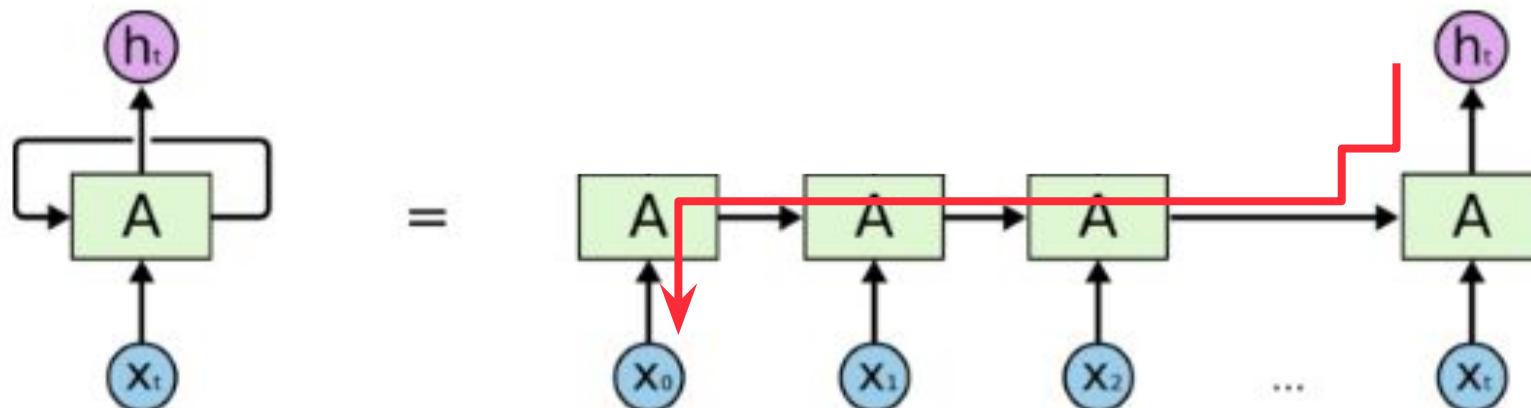


An unrolled recurrent neural network.

$$h_t = g(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$
$$o_t = h_t$$

Рекуррентные сети

RNN: vanishing gradient



An unrolled recurrent neural network.

Рекуррентные сети

LSTM

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

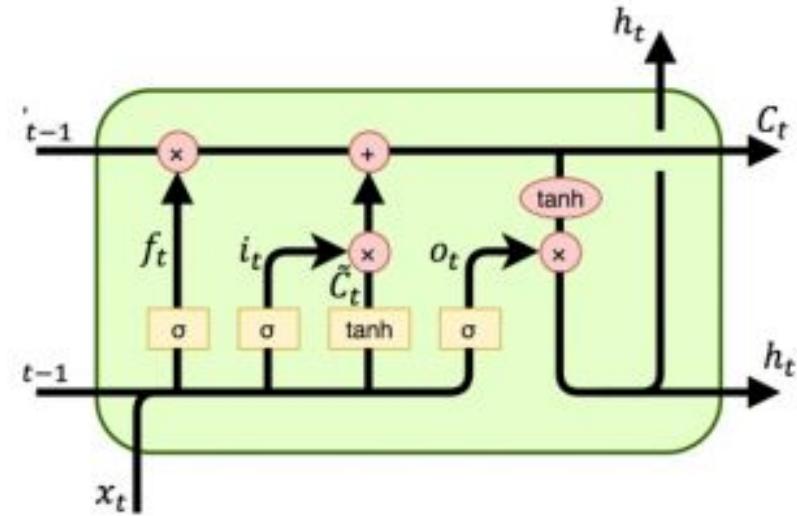
$$o_t = g(W_{ho}h_{t-1} + W_{xo}x_t + b_o)$$

$$f_t = g(W_{hf}h_{t-1} + W_{xf}x_t + b_f)$$

$$\tilde{c}_t = g(W_{hc}h_{t-1} + W_{xc}x_t + b_c)$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

$$h_t = o_t * g(c_t)$$



Рекуррентные сети

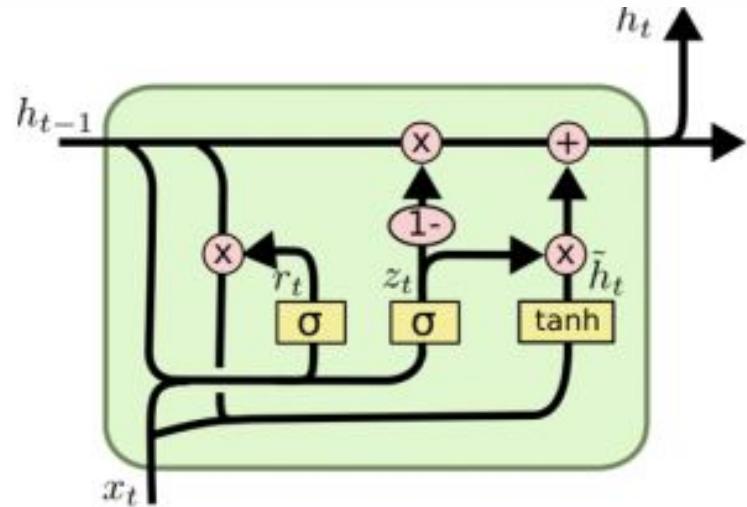
GRU

$$r_t = g(W_{hr}h_{t-1} + W_{xr}x_t + b_r)$$

$$z_t = g(W_{hz}h_{t-1} + W_{xz}x_t + b_z)$$

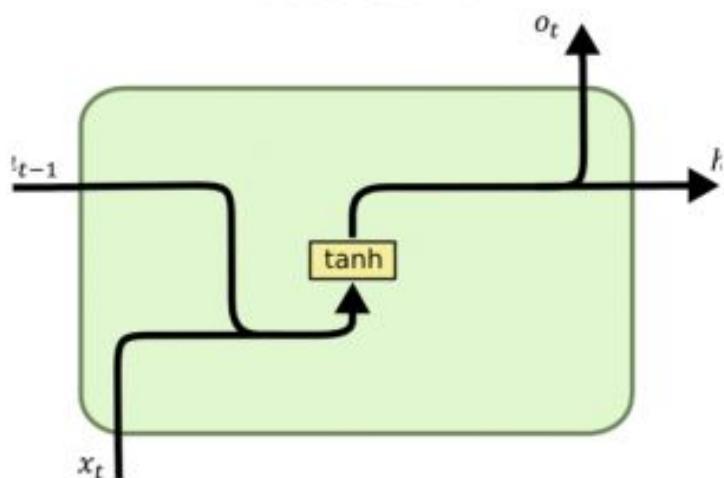
$$\tilde{h}_t = g(W_{hh}(r_t * h_{t-1}) + W_{xh}x_t + b_h)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

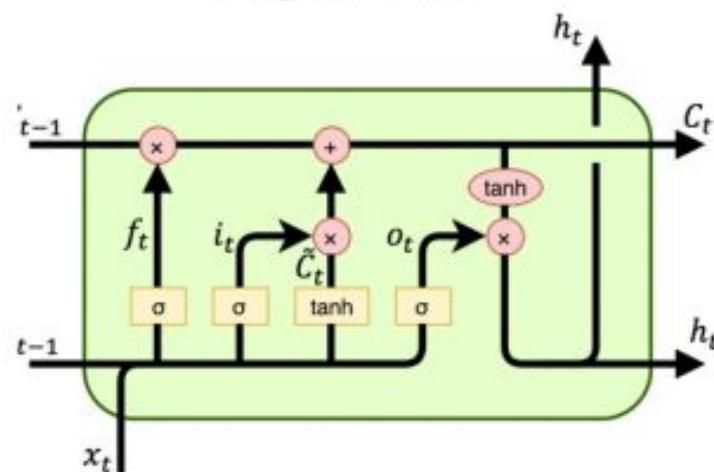


Рекуррентные сети

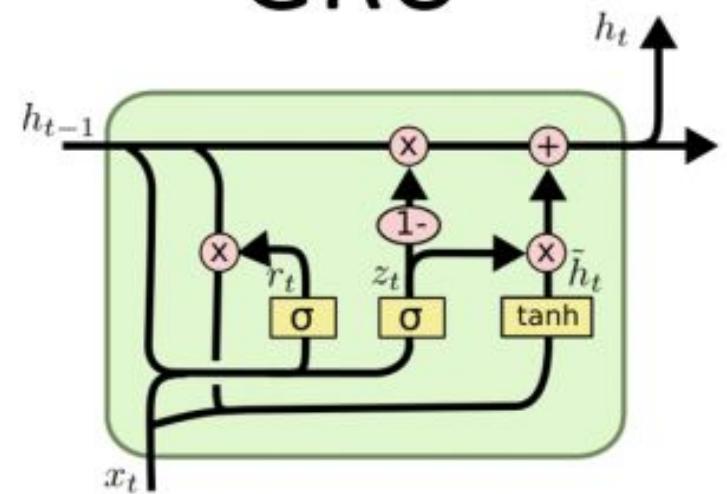
RNN



LSTM

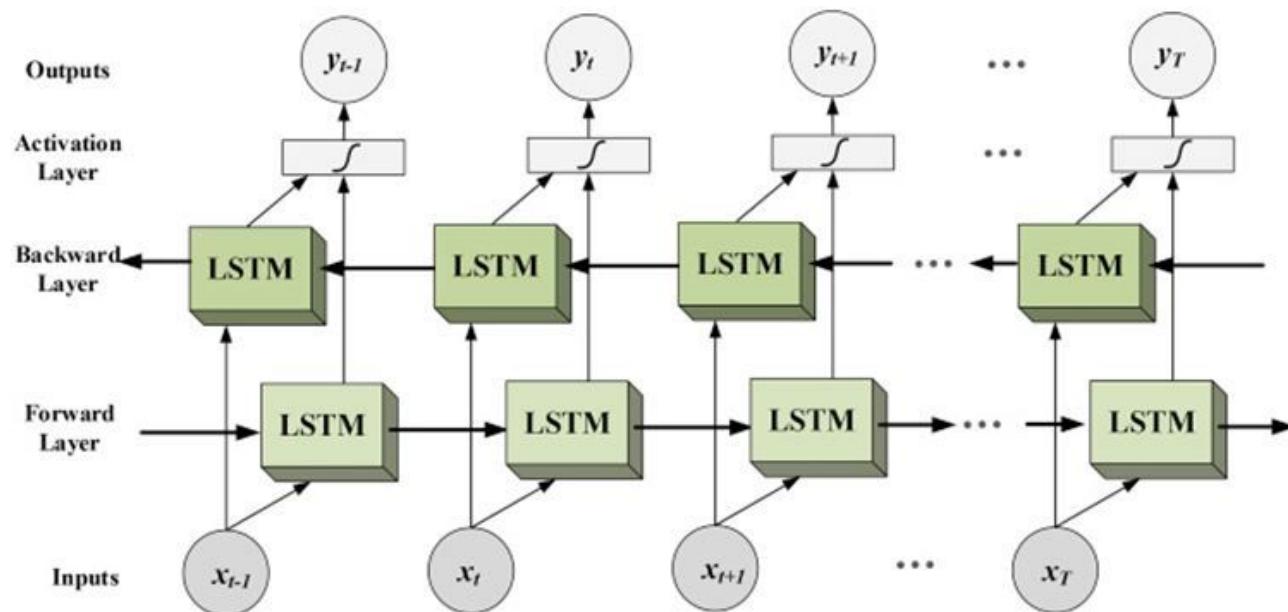


GRU



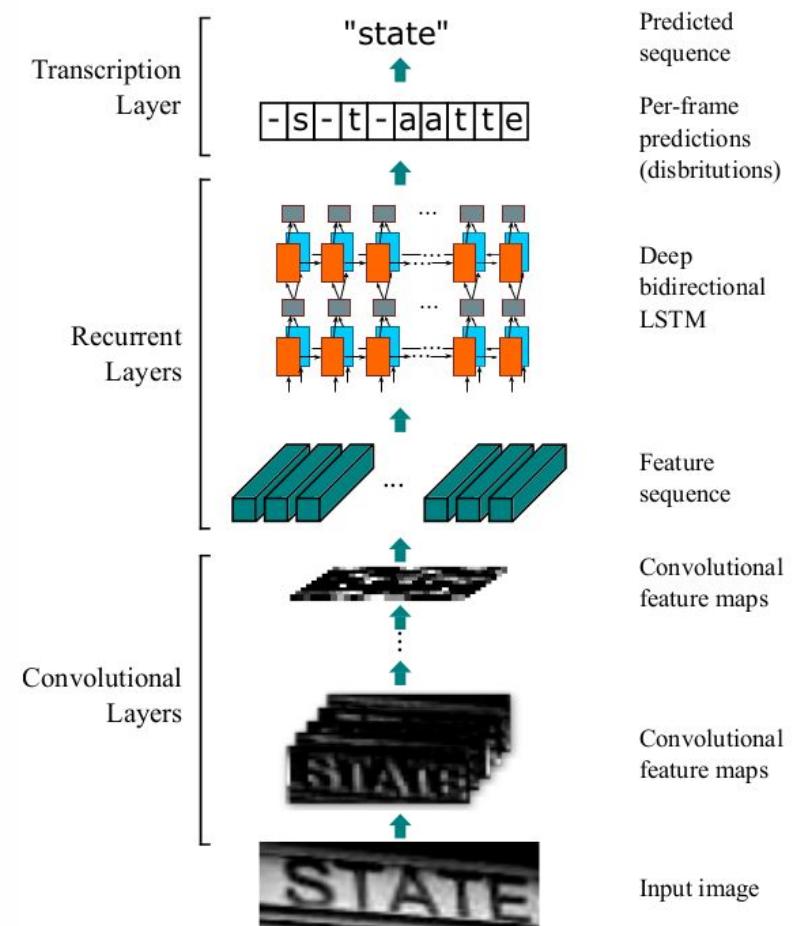
Рекуррентные сети

Двунаправленные сети

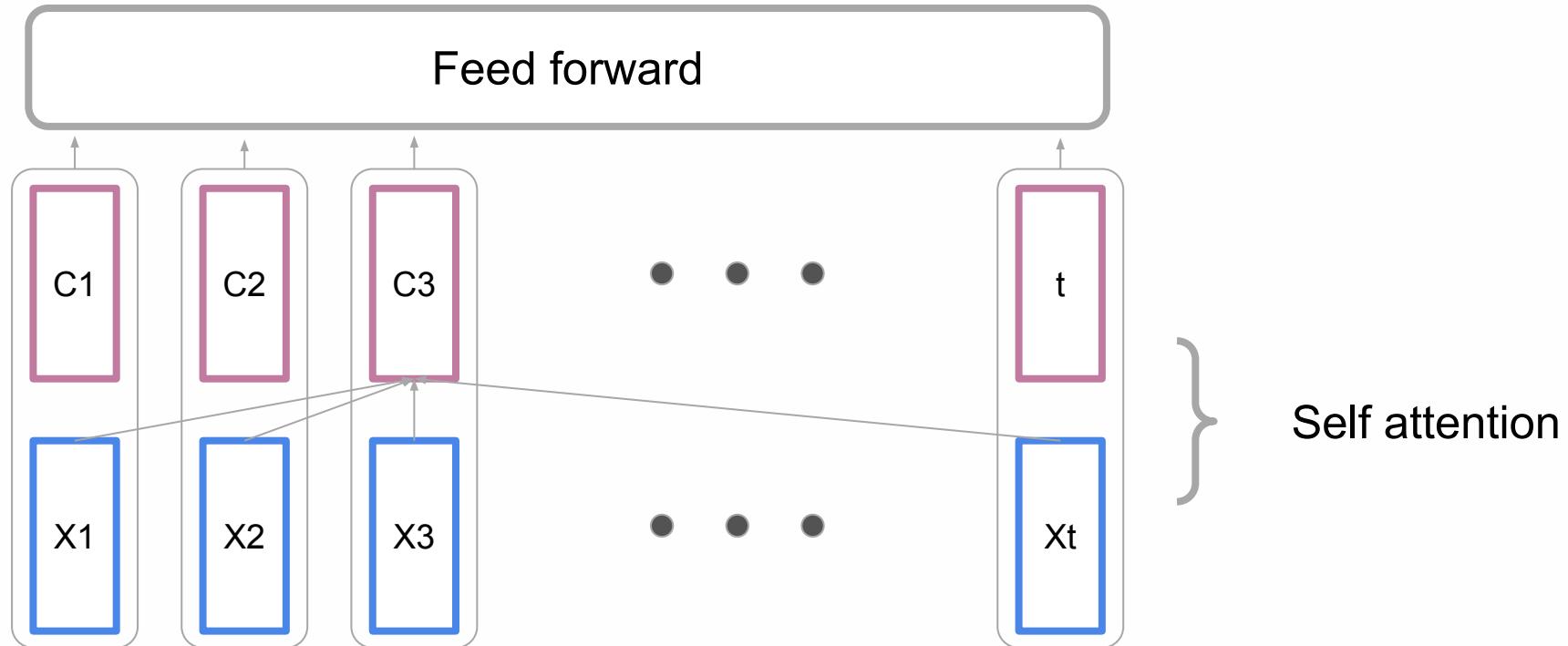


Рекуррентные сети

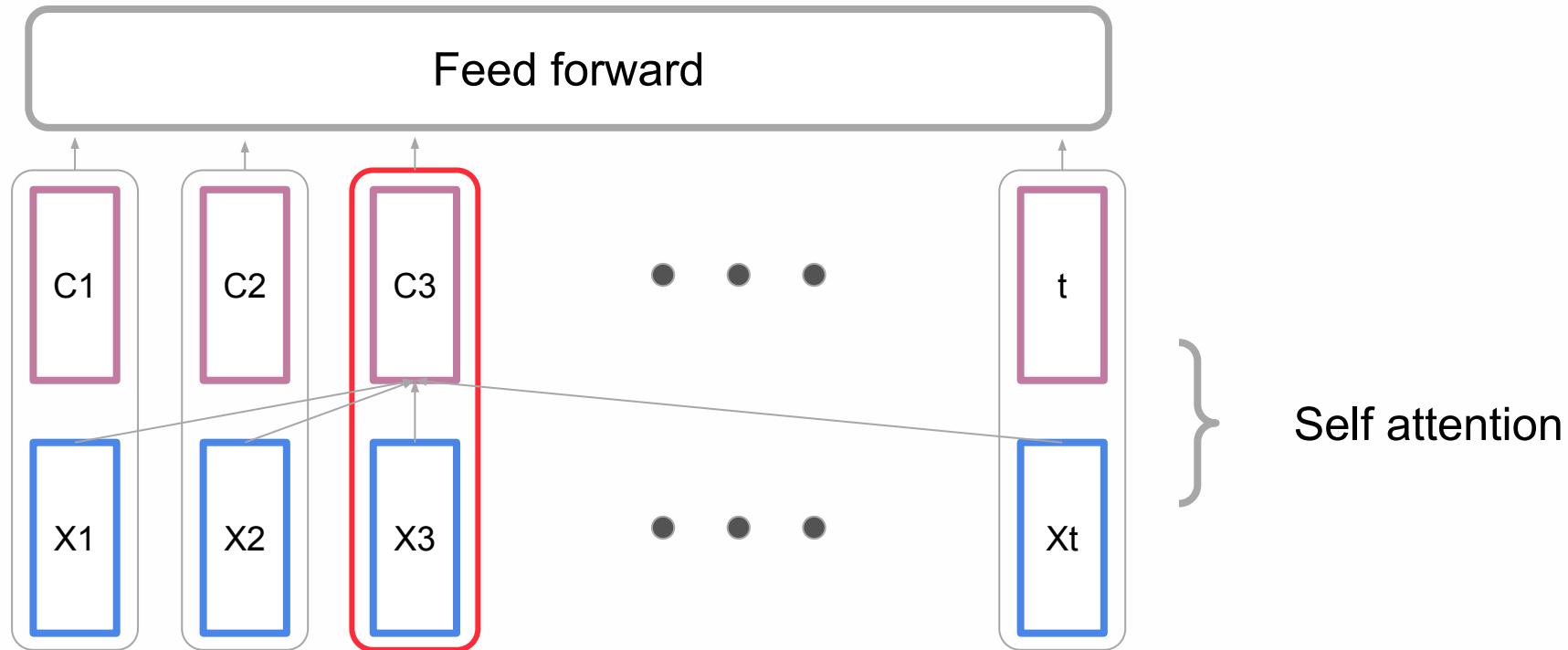
CRNN: комбинация CNN и RNN
(не путать с R-CNN)



Self attention



Self attention



Self attention

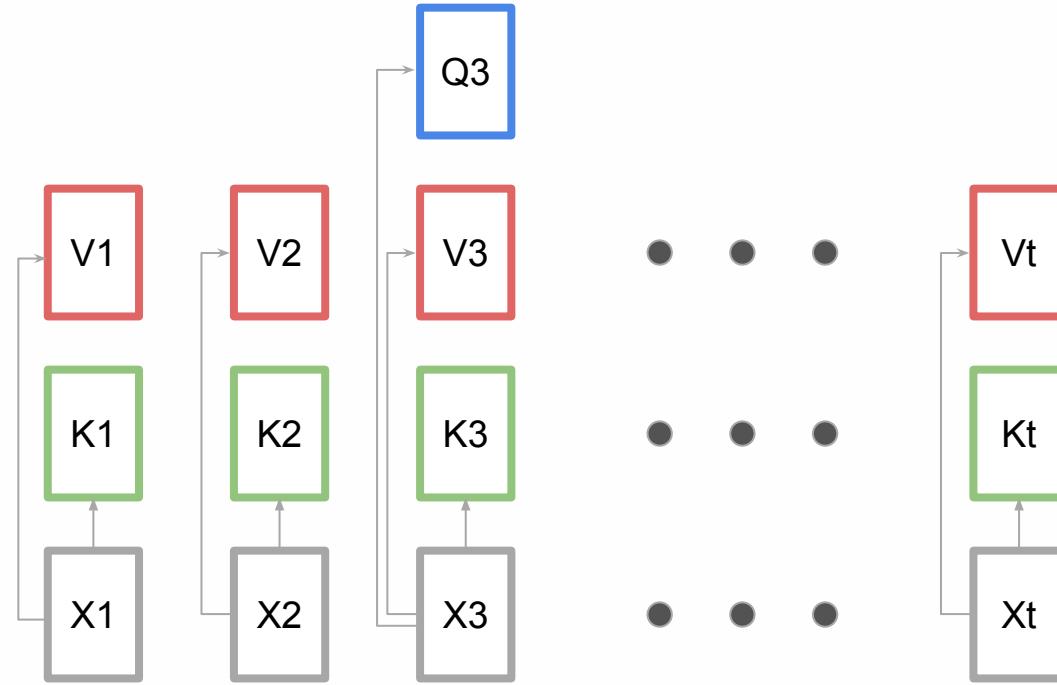
Шаг 1

Для каждой позиции:
ключ, значение, запрос

$$K_t = W_K X_t$$

$$V_t = W_V X_t$$

$$Q_t = W_Q X_t$$

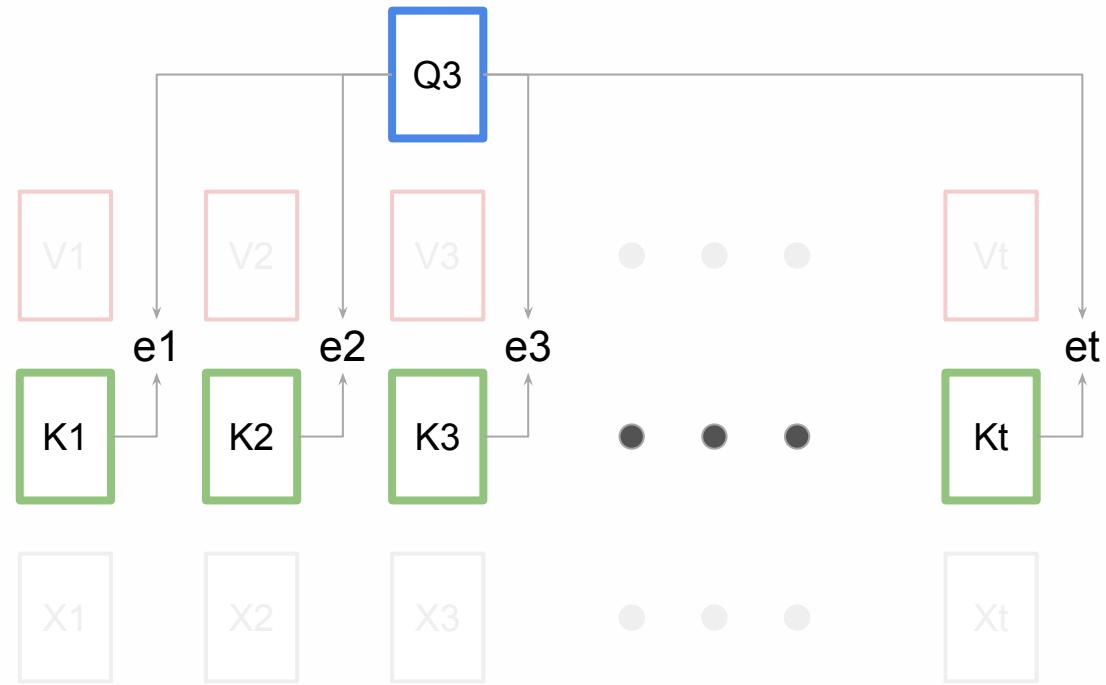


Self attention

Шаг 2

Вычисляем веса
контекстов

$$e_{ij} = \frac{e^{(Q_i, K_j)}}{\sum_j e^{(Q_i, K_j)}}$$

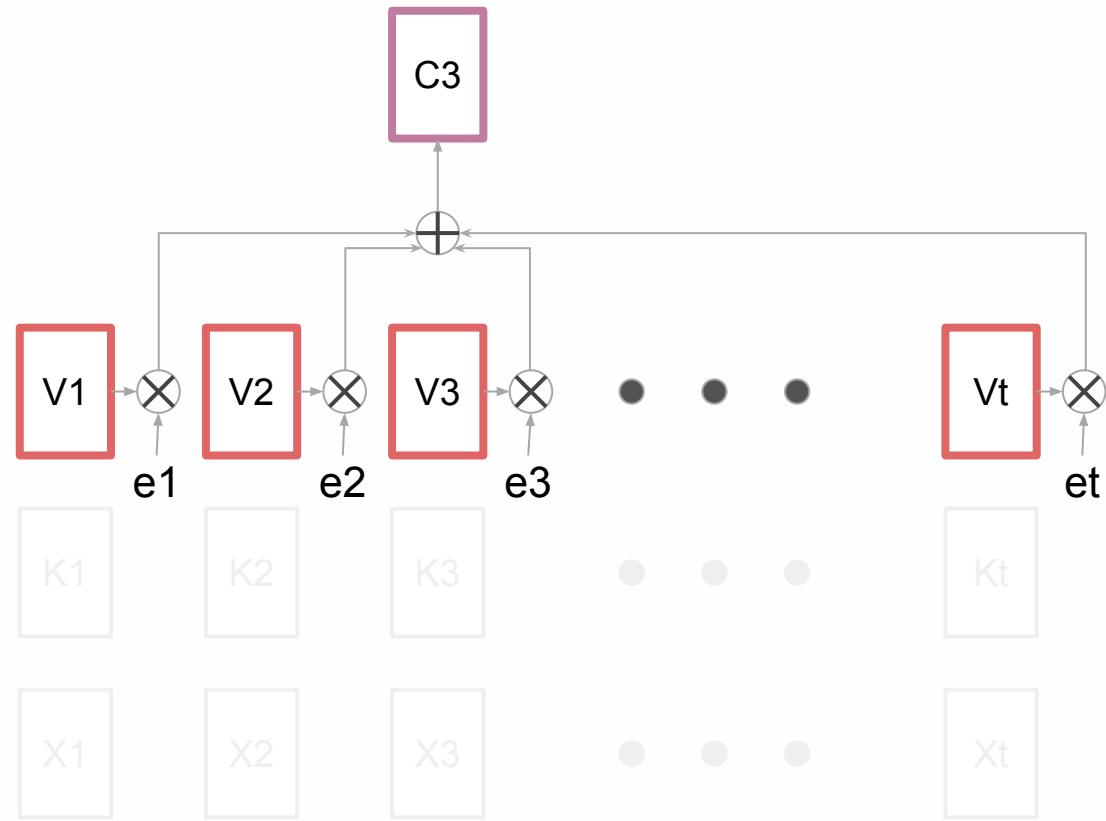


Self attention

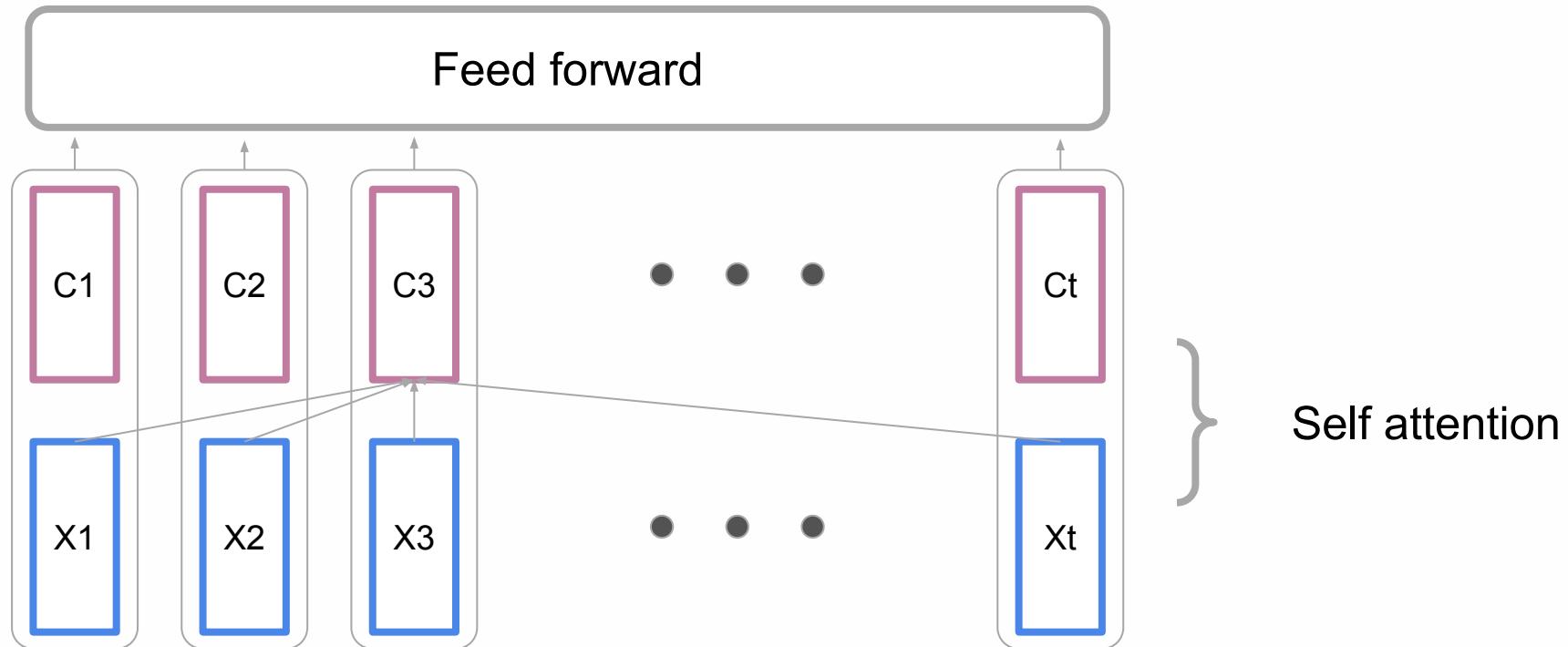
Шаг 3

Контекст - взвешенная
сумма значений

$$c_i = \sum_j e_{ij} V_j$$



Self attention



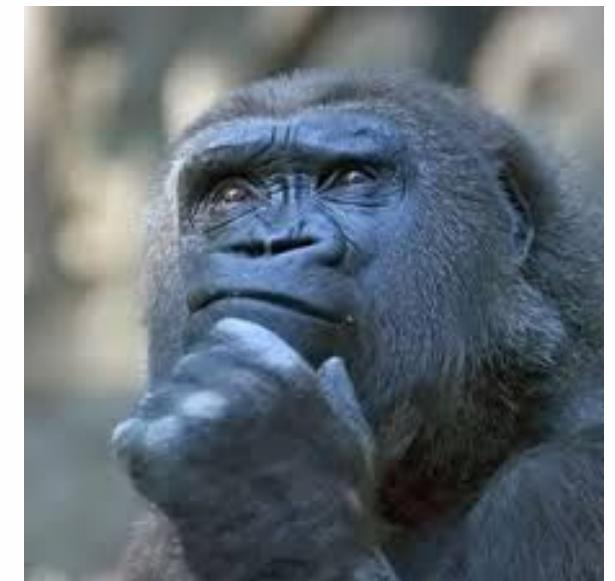
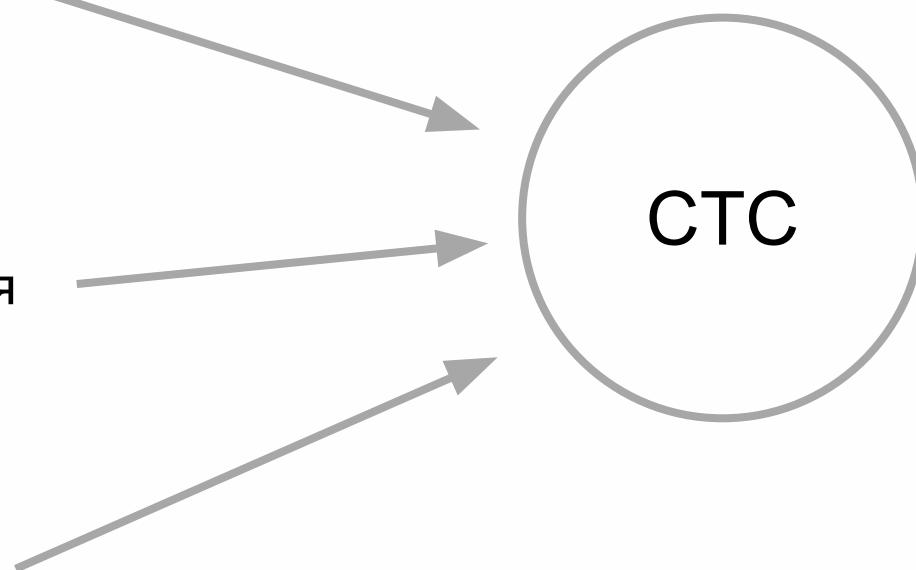
CTC Loss

CTC Loss

Представление
выравниваний

Способ обучения

Механизм
декодирования

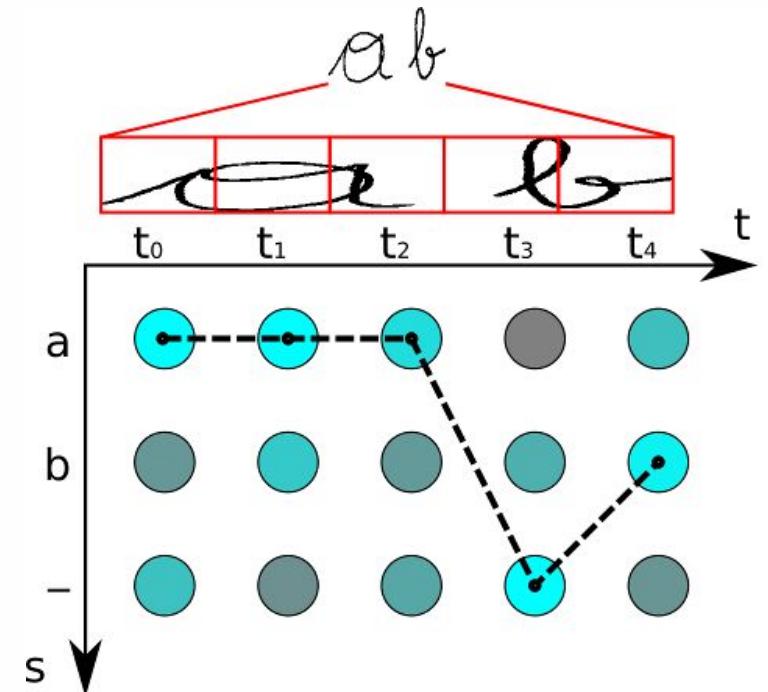


Выравнивания

- Сеть выдает вероятности букв для каждого кадра:
 $C_{t,k} = P(c_t = k | image)$
- Метка содержит буквы слова L_i

Для слова “ab” и семи кадров есть три выравнивания:

	$P(\pi C)$
a bbb	0.1
aa bb	0.2
aaa b	0.4



“|” - разделитель, “π” - выравнивание

Вероятность слова

По формуле полной вероятности:

$$P(w|C) = \sum_{\pi} P(w|\pi)P(\pi|C) = \sum_{\pi} P(\pi|C)$$

P(π|C)

T - число кадров

| a | bbb | 0.1

L - число букв

| aa | bb | 0.2

N - число выравниваний

| aaa | b | 0.4

$$N = C_{T-L}^{L-1}$$

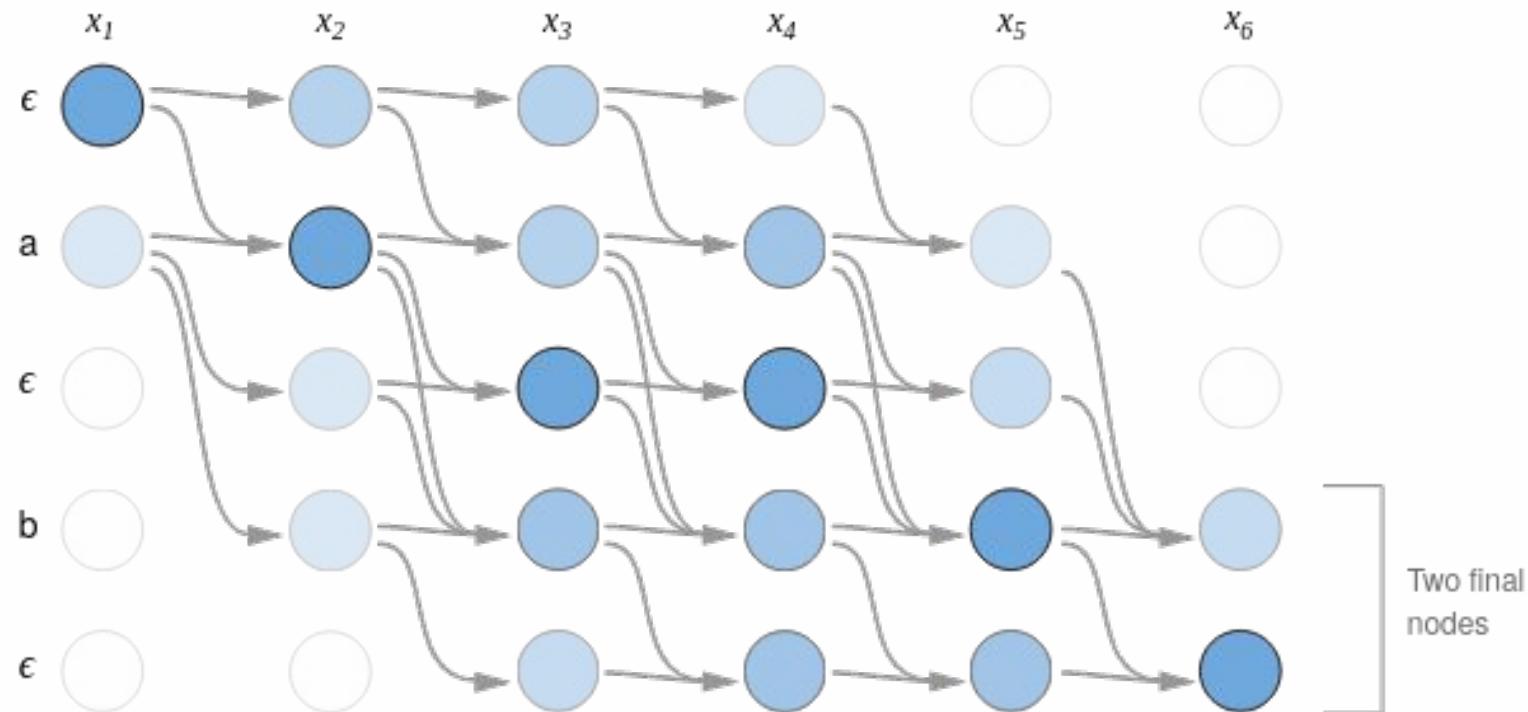
Комбинаторный взрыв

P("ab") 0.7

другие слова 0.3

CTC Loss

Динамическое программирование



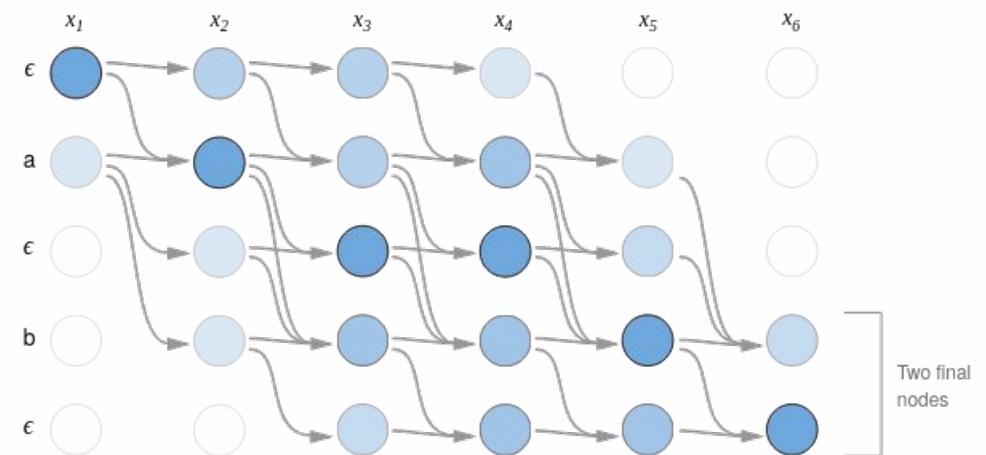
CTC Loss

$$\text{MLE: } \arg \max_{\theta} \prod_{w,C} P(w|C, \theta)$$

Почему не пробрасывать градиенты из final?

- Тяжело вычислять из-за max
- Vanishing gradients
- HMM EM legacy?

Идея: получить вероятность слова при условии прохождения пути через каждое состояние на каждом кадре



CTC Loss

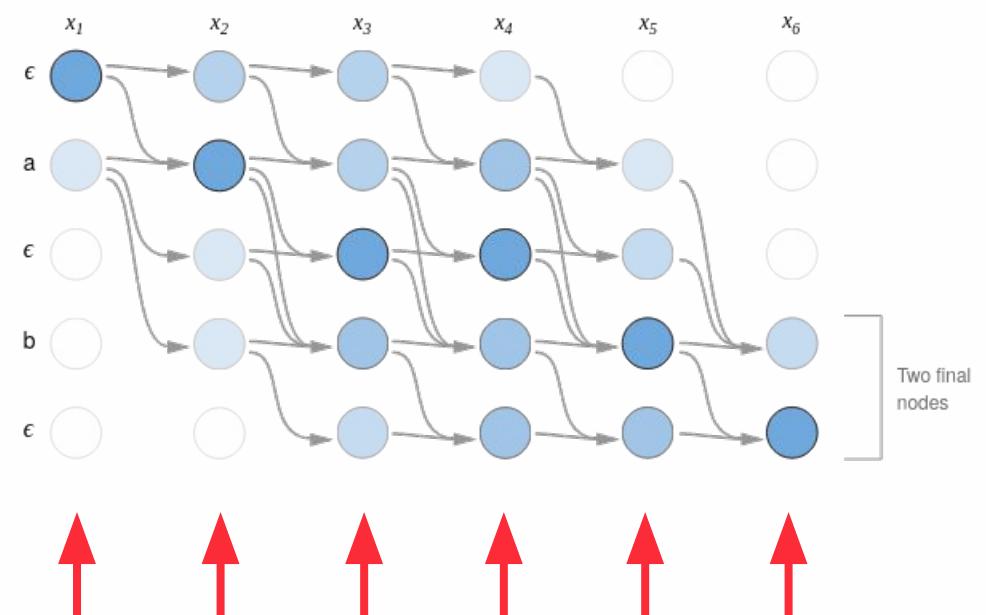
Forward-backward (аналогия с НММ)

$$\alpha_t(s) = \sum_{\pi_{1:t} \sim L_{1:s}} P(\pi_{1:t} | C_{1:t})$$

$$\beta_t(s) = \sum_{\pi_{t:T} \sim L_{s:|L|}} P(\pi_{t:T} | C_{t:T})$$

$$\frac{\alpha_t(s)\beta_t(s)}{C_{t,L_s}} = \sum_{\pi: \pi_t = L_s} P(\pi | C), \forall t$$

$$\arg \max_{\theta} P(L | C, \theta) = \arg \max_{\theta} \sum_s \frac{\alpha_t(s)\beta_t(s)}{C_{t,L_s}}, \forall t$$

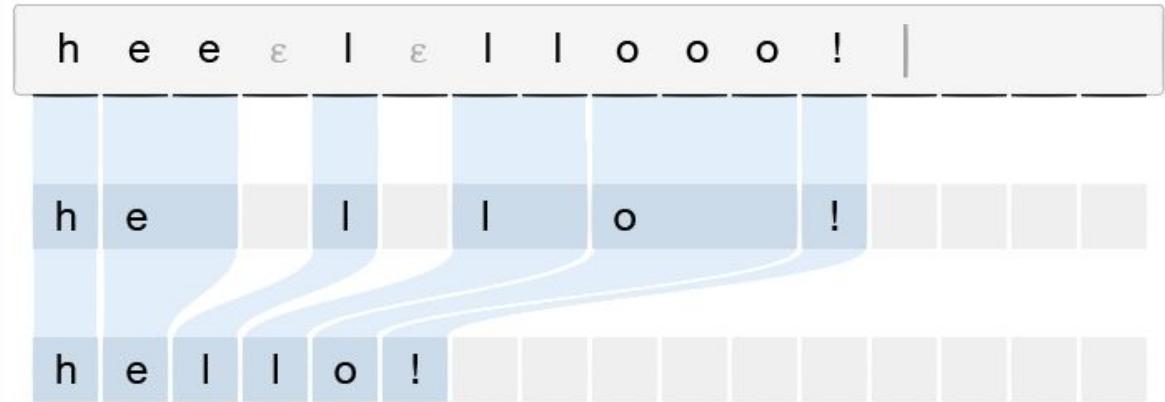


Декодирование

Простой способ:

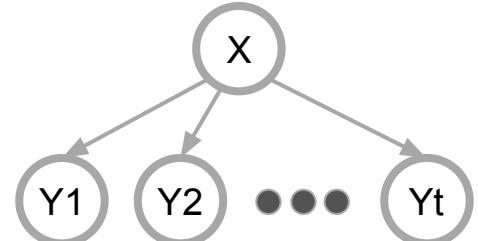
$$c_t = \arg \max_i C_{t,i}$$

- Убираем повторы
- Убираем разделитель



Свойства CTC Loss

- Независимость результатов для разных кадров во время распознавания
(модель способна выдавать несуществующие слова)

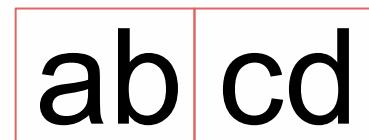


Свойства CTC Loss

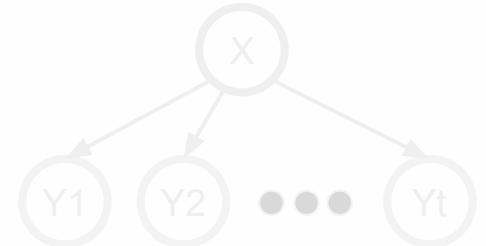
- Независимость результатов для разных кадров во время распознавания
(модель способна выдавать несуществующие слова)
- Каждой букве соответствует один или больше кадров



OK



Not OK



Языковое моделирование в OCR



N-gram

$$P(w_1, w_2, w_3, \dots, w_n) \\ = P(w_1)P(w_2|w_1)P(w_3|w_2, w_1)\dots P(w_n|w_{n-1}, \dots, w_1)$$

1 - Gram:

$$P(w_1)P(w_2)P(w_3)\dots P(w_n)$$

k - Gram:

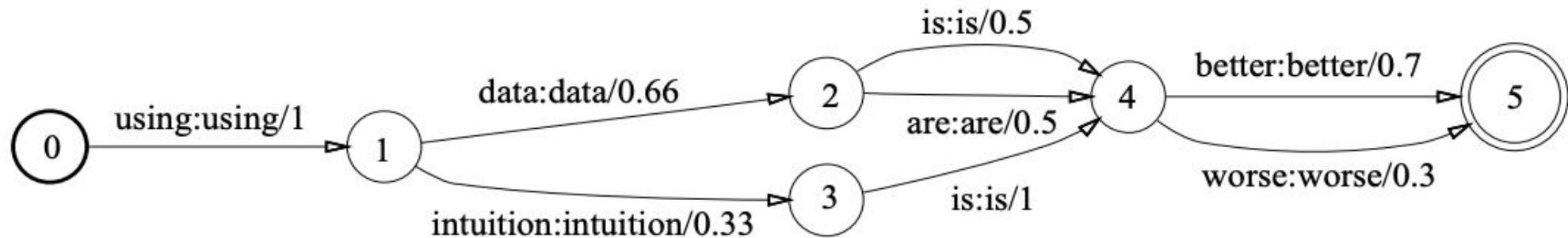
$$P(w_1)P(w_2|w_1)P(w_3|w_2, w_1)\dots P(w_n|w_{n-1}, \dots, w_{n-k+1})$$

АВТОМАТЫ

Weighted finite-state transducers (WFST)

- Грамматики
- N-Gram
- Исправление ошибок

(+) Эффективный beam-search

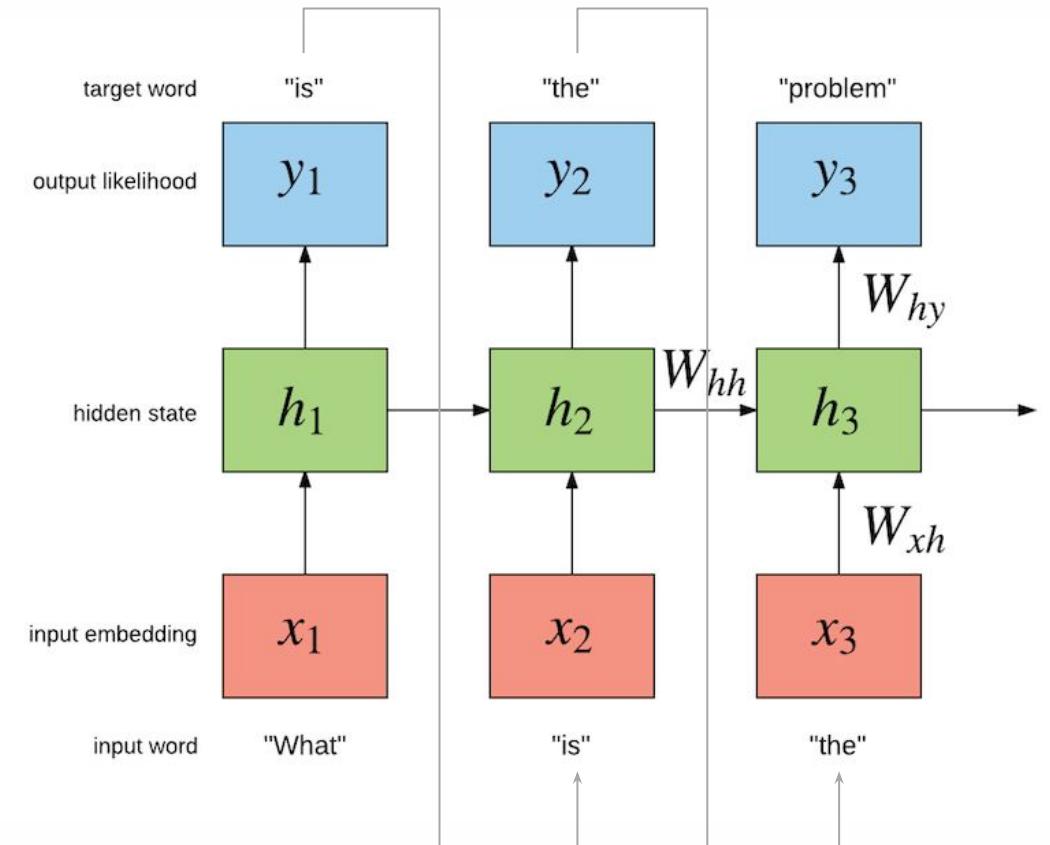


RNN

(+) RNN способна учитывать все предыдущие слова

$$P(w_n | w_{n-1}, \dots, w_1)$$

(-) Сложнее организовать beam search



Декодирование



Beam search

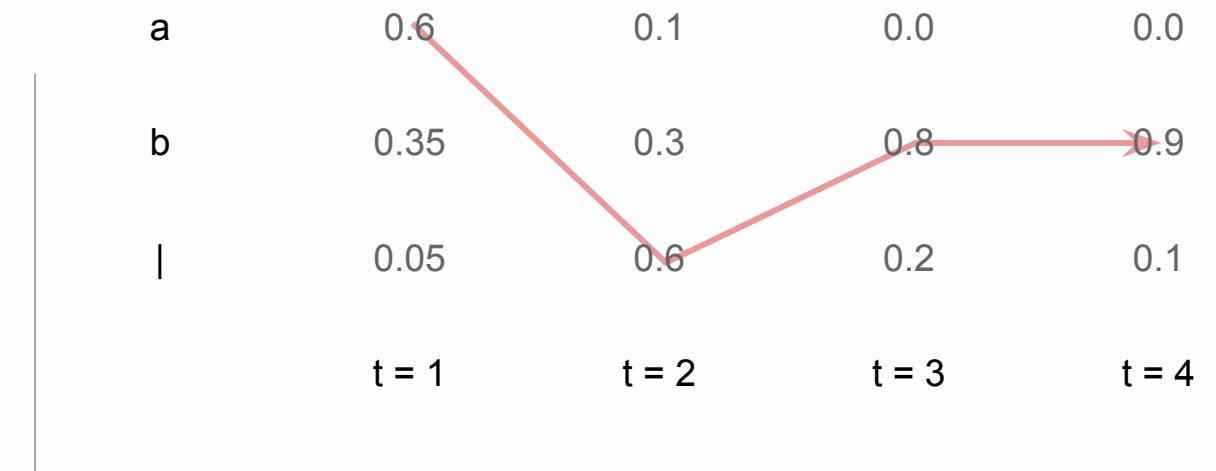
Декодирование с использованием языковой модели

a	0.6	0.1	0.0	0.0
b	0.35	0.3	0.8	0.9
	0.05	0.6	0.2	0.1
	t = 1	t = 2	t = 3	t = 4

Beam search

Декодирование с использованием языковой модели

		$P(w)$	$P(\pi C)$
a b b	“ab”	0.2	0.26



Beam search

Декодирование с использованием языковой модели

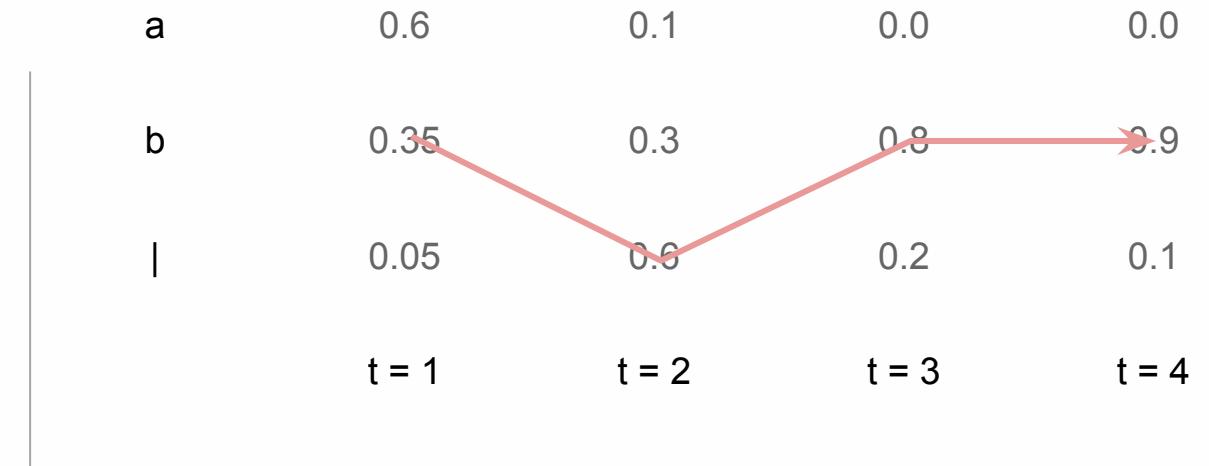
		$P(w)$	$P(\pi C)$
a b b	“ab”	0.2	0.26
a a b	“ab”	0.2	0.01



Beam search

Декодирование с использованием языковой модели

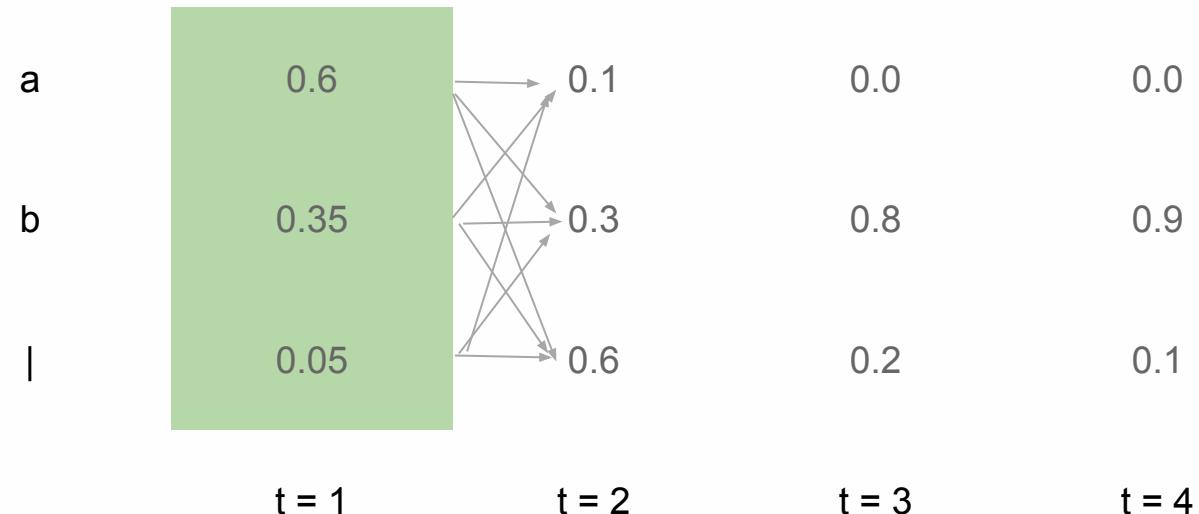
		$P(w)$	$P(\pi C)$
a b b	“ab”	0.2	0.26
a a b	“ab”	0.2	0.01
b b b	“bb”	0.8	0.15



Beam search

Декодирование с использованием языковой модели
на примере 2-Gram

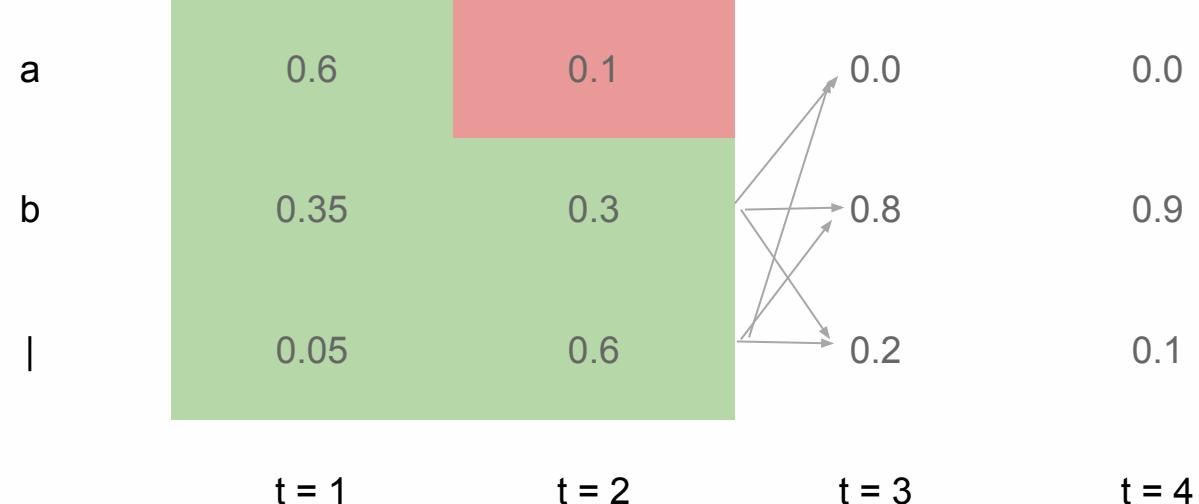
Похоже на forward pass, но
добавляем вероятности
переходов из языковой модели



Beam search

Декодирование с использованием языковой модели
на примере 2-Gram

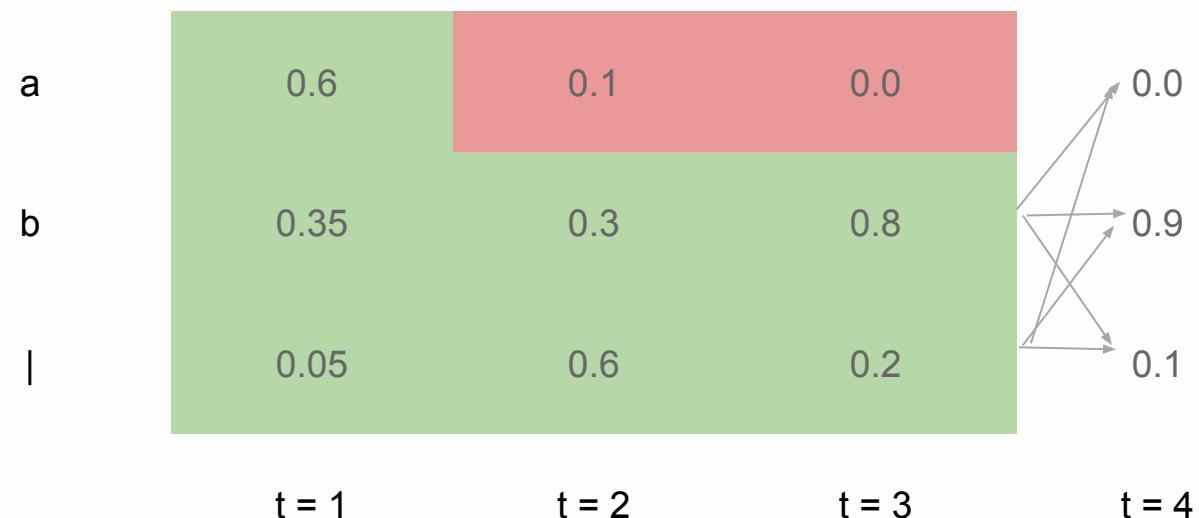
Отсеиваем токены с низкой
вероятностью



Beam search

Декодирование с использованием языковой модели
на примере 2-Gram

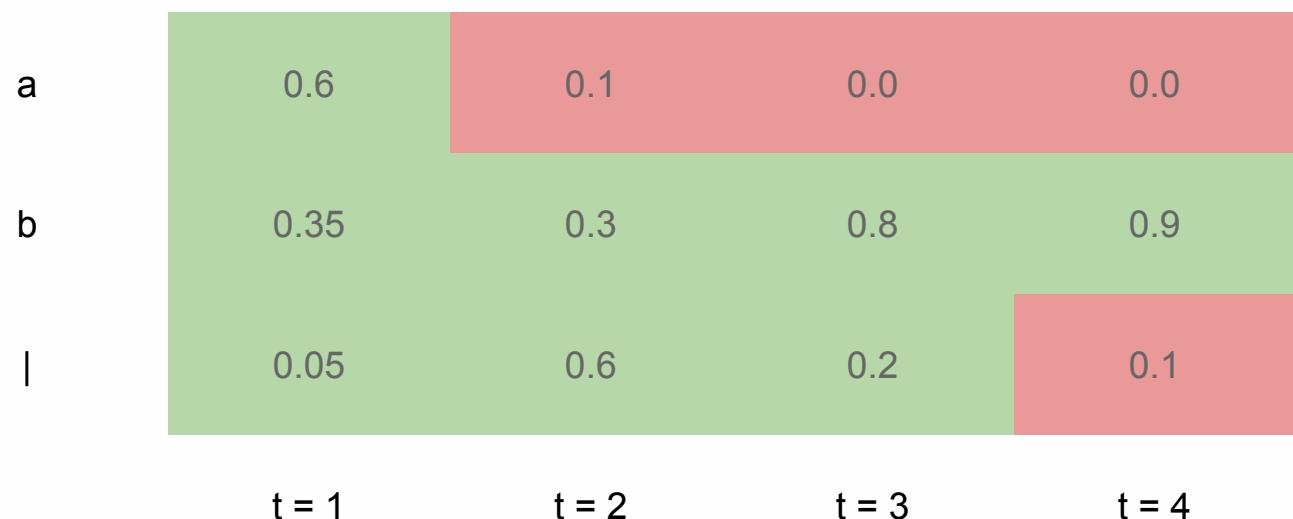
Отсеиваем токены с низкой
вероятностью



Beam search

Декодирование с использованием языковой модели
на примере 2-Gram

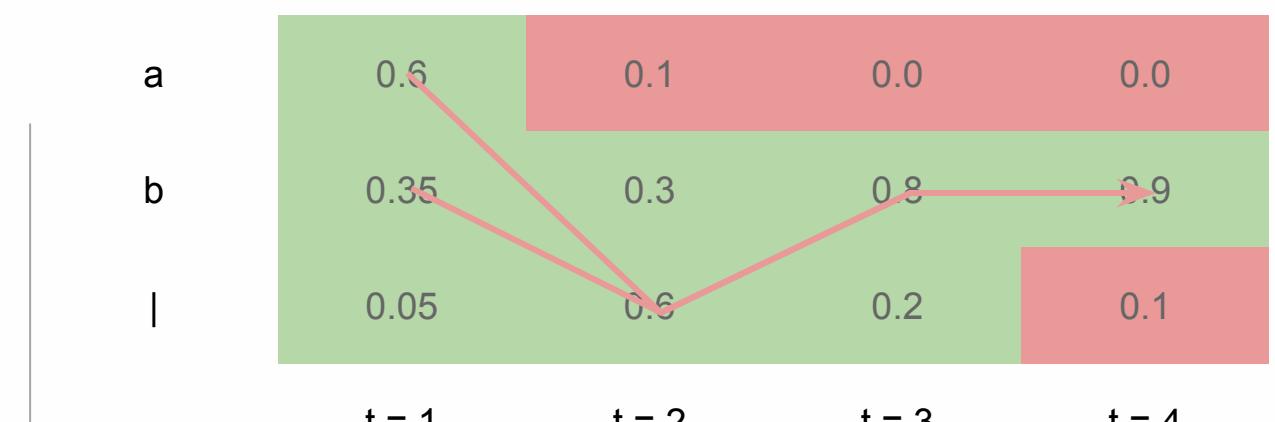
Отсеиваем токены с низкой
вероятностью



Beam search

Декодирование с использованием языковой модели

		$P(w)$	$P(\pi C)$
a b b	“ab”	0.2	0.26
a a b	“ab”	0.2	0.01
b b b	“bb”	0.8	0.15



Заключение



Резюме

- Разобрали устройство OCR: детекция и распознавание
- Детекция эффективно реализуется при помощи сегментации
- Для распознавания данных переменной длины можно использовать RNN или self-attention
- CTC loss позволяет обучаться на данных без выравнивания



Материалы

- OCR

<https://towardsdatascience.com/a-gentle-introduction-to-ocr-ee1469a201aa>

2016 Shi B. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition

- CTC

<https://distill.pub/2017/ctc/>

2006 Graves A. et al. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks

- RNN / Attention

<https://www.coursera.org/learn/nlp-sequence-models>

<https://towardsdatascience.com/attention-and-its-different-forms-7fc3674d14dc>

- Segmentation

https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272