МИСИС
УНИВЕРСИТЕТ

Трансформер. Часть 1

Корнет Мария Евгеньевна
старший преподаватель кафедры
Инженерная кибернетика

План

- Модели внимания RNN, seq2seq
- Self-attention
- Трансформеры

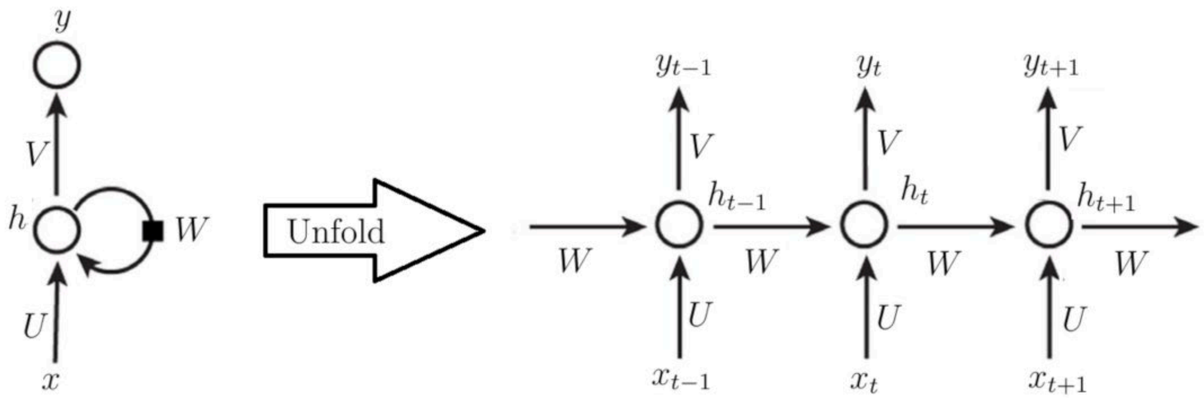
Напоминание: RNN

x_t - входной вектор в момент времени
$t = 1, \dots, T$
h_t - вектор скрытого состояния в момент t
y_t - выходной вектор в момент времени t

$$h_t = \sigma(Ux_t + Wh_{t-1} + b_h)$$
$$y_t = \sigma(Vh_t + b_y)$$

Обучение RNN:

$$\sum_{t=0}^{T} L(U, V, W) \to \min_{U,V,W}$$



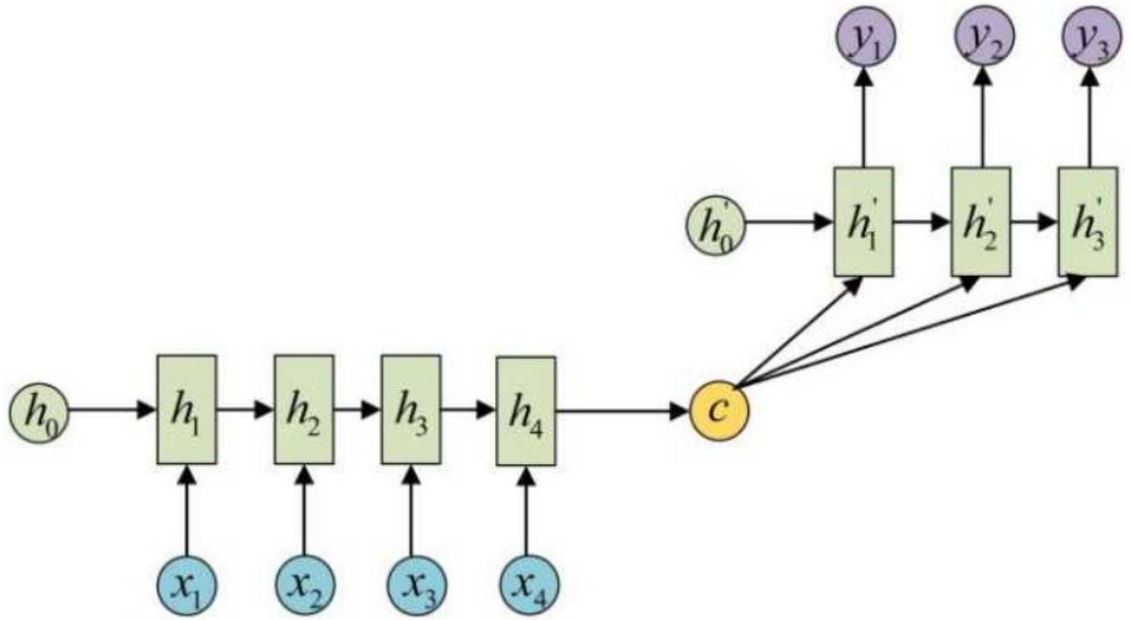
Недостатки:

- Длина входной и выходной последовательностей должна быть одинаковой, что неестественно для перевода.
- Последовательный однонаправленный просмотр.

Напоминание: seq2seq

$X = (x_1, \dots, x_n)$ - входная последовательность
$Y = (y_1, \dots, y_m)$ - выходная последовательност
***c* — контекстный вектор, используемый декодером (context vector)**

$$h_i = f_{in}(x_i, h_{i-1})$$
$$h'_t = f_{out}(h'_{t-1}, y_{t-1}, c)$$
$$y_t = f_y(h'_t, y_{t-1})$$



Недостатки:

- «бутылочное горлышко» вектора ***c***
- затухание, взрывы градиента

http://www.machinelearning.ru/wiki/

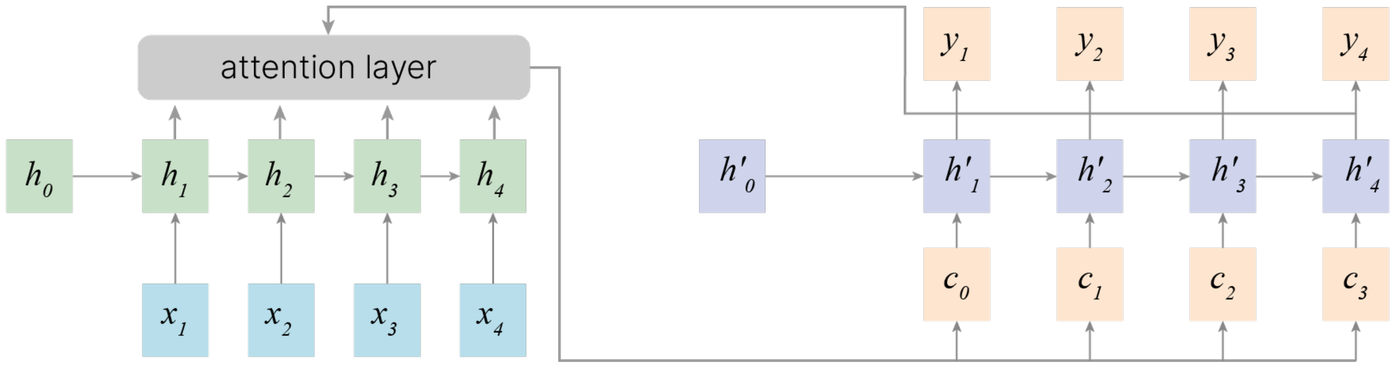
RNN с вниманием

$a(h,\ h')$ - **функция сходства состояний входа h, и выхода h'**

c_t - вектор входного контекста для выхода t

α_{t_i} - **веса внимания (attention score)** важность i – го входа для выхода t

$$h_i = f_{in}(x_i, h_{i-1})$$
$$\boldsymbol{\alpha_{t_i} = norm\, a(h_i, h'_{t-1})}$$
$$\boldsymbol{c_t = \sum_i \alpha_{t_i} h_i}$$
$$h'_t = f_{out}(h'_{t-1}, y_{t-1}, c_t)$$
$$y_t = f_{out}(h'_t, y_{t-1}, c_t)$$

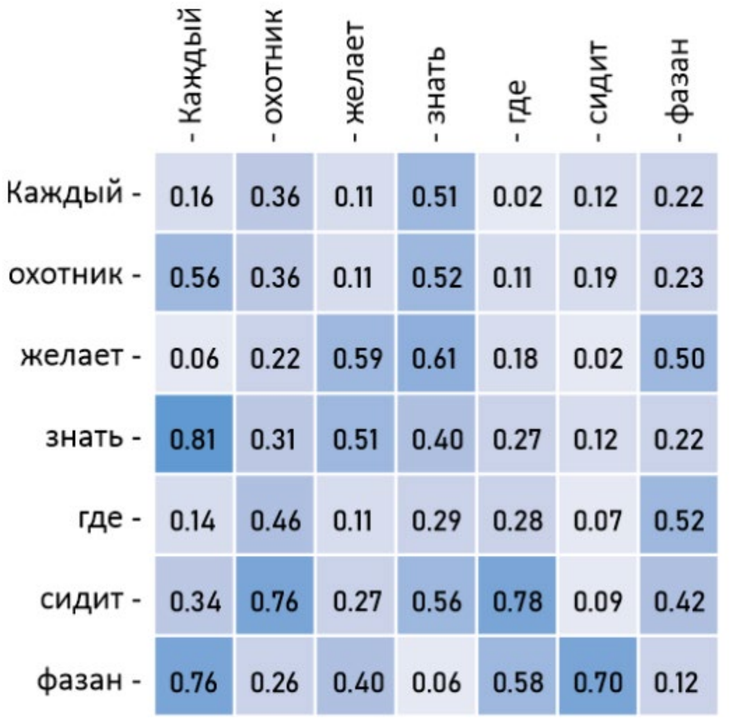


http://www.machinelearning.ru/wiki/images/1/19/Voron-ML-Attention-slides.pdf

Матрица внимания

Матрица внимания — двумерная таблица, в которой каждая ячейка соответствует значению вектора attention score и обновляется во время обучения.

Матрица показывает на какие слова входной последовательности x_i модель обращает внимание, генерируя слова выходной последовательности y_i



Функции сходства

1. Скалярное произведение:

$$a(h,\ h') = h^T h'$$

2. Экспоненциальное скалярное произведение:

$$a(h, h') = exp(h^T h')$$

3. Билинейное внимание:

$$a(h, h') = h^T W h'$$

W — обучаемая матрица, которая позволяет моделировать сложные взаимодействия

4. Аддитивное внимание:

$$a(h, h') = w^T tanh(Uh + Vh')$$

$w\ \ , U, V$ – обучаемые параметры

Query-Key-Value attention

Функция внимания вычисляет взвешенную сумму значений, при этом веса определяются мерой соответствия между запросом и ключами:

$$a(h_i, h'_{t-1}) = \frac{(\boldsymbol{W_k}\, h_i\,)^T (\boldsymbol{W_q} h'_{t-1})}{\sqrt{d}},$$

d — размерность векторов (для стабилизации градиентов)

$$\alpha_{t_i} = softmax\; a(h_i, h'_{t-1})$$

$$c_t = \sum_i \alpha_{t_i} \boldsymbol{W_v} h_i$$

$\boldsymbol{W_k}, \boldsymbol{W_q}, \boldsymbol{W_v}$ - матрицы **key, query, value**

Три вектора Query, Key, Value:

Query (Запрос) – текущее состояние декодера. Можно представить как вопрос, который токен задаёт сети: «Что мне нужно знать, чтобы понять контекст?».

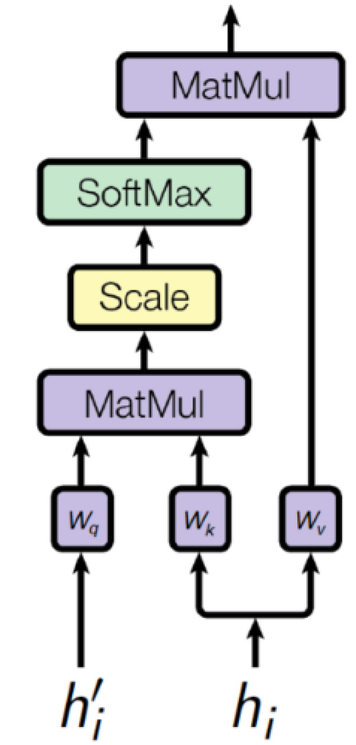
Key (Ключ) - это то, что содержит данный токен, его содержимое.

Value (Значение) - хранит информацию, которая может быть передана другим токенам.

Key и **Value** — это выходы декодера.

Scale dot product attention

На вход подаются **запросы** и **ключи** (размерностью d_k) и **значения** (размерностью d_v) .
Далее считается *скалярное произведение (Matmul)* запросов со всеми ключами.
Scale: каждый результат делится на $\sqrt{d}$, потом применяется *Softmax* чтобы получить веса значений.



Модель внимания

2017 Google «Attantion is all you need»

Искомый вектор контекста релевантного запросу:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$$c = Attention(q, k, v) = \sum_i v_i Softmax_i a(k_i, q)$$

Q, K, V – вектора запроса (query), ключа (key), значения (value).

$a(k_i, q)$ – функция сходства, оценка релевантности ключа k_i запросу q

self-attention – частный случай, когда входная и выходная последовательности совпадают.

https://arxiv.org/abs/1706.03762

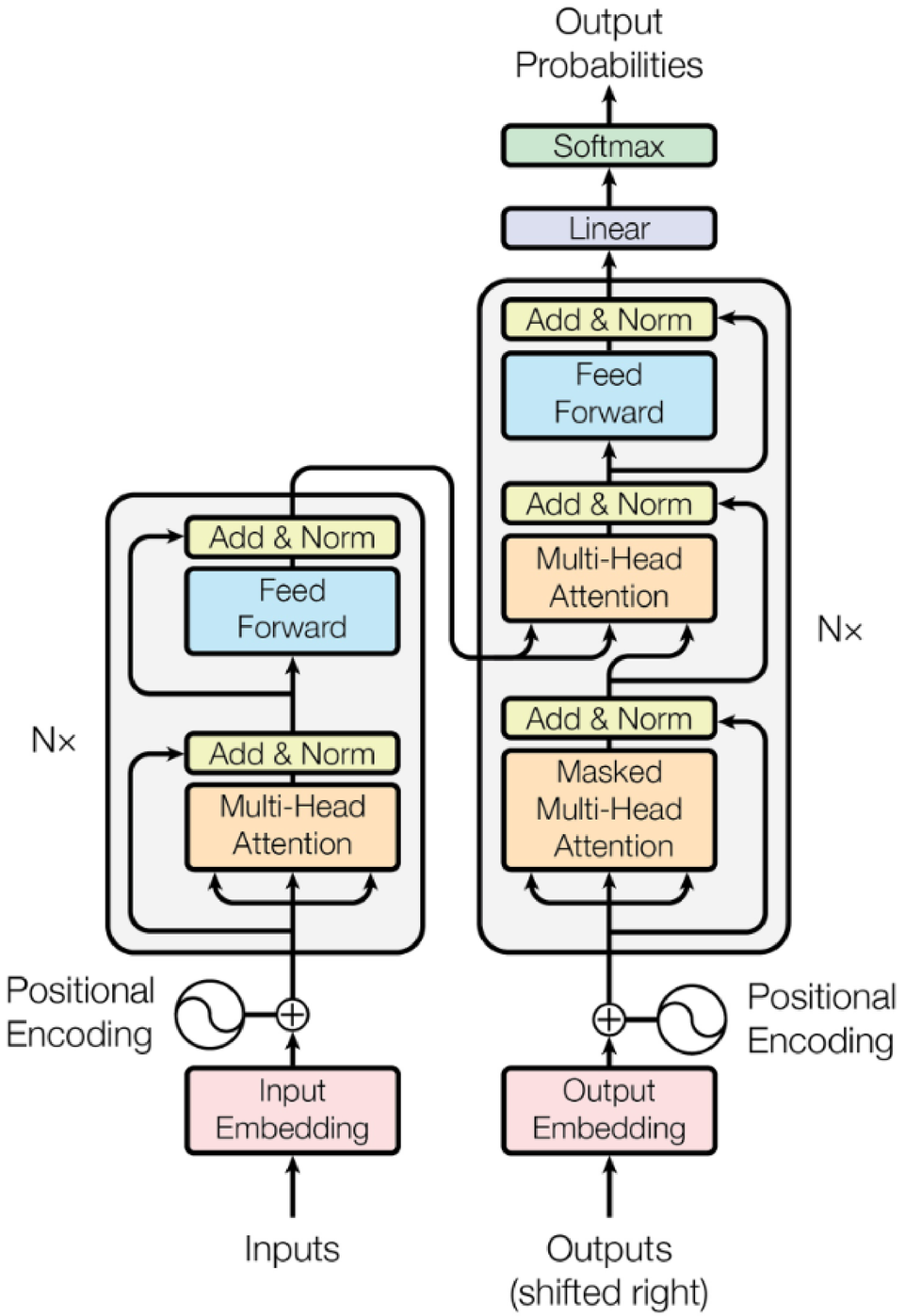
Архитектура Transformer

Энкодер:

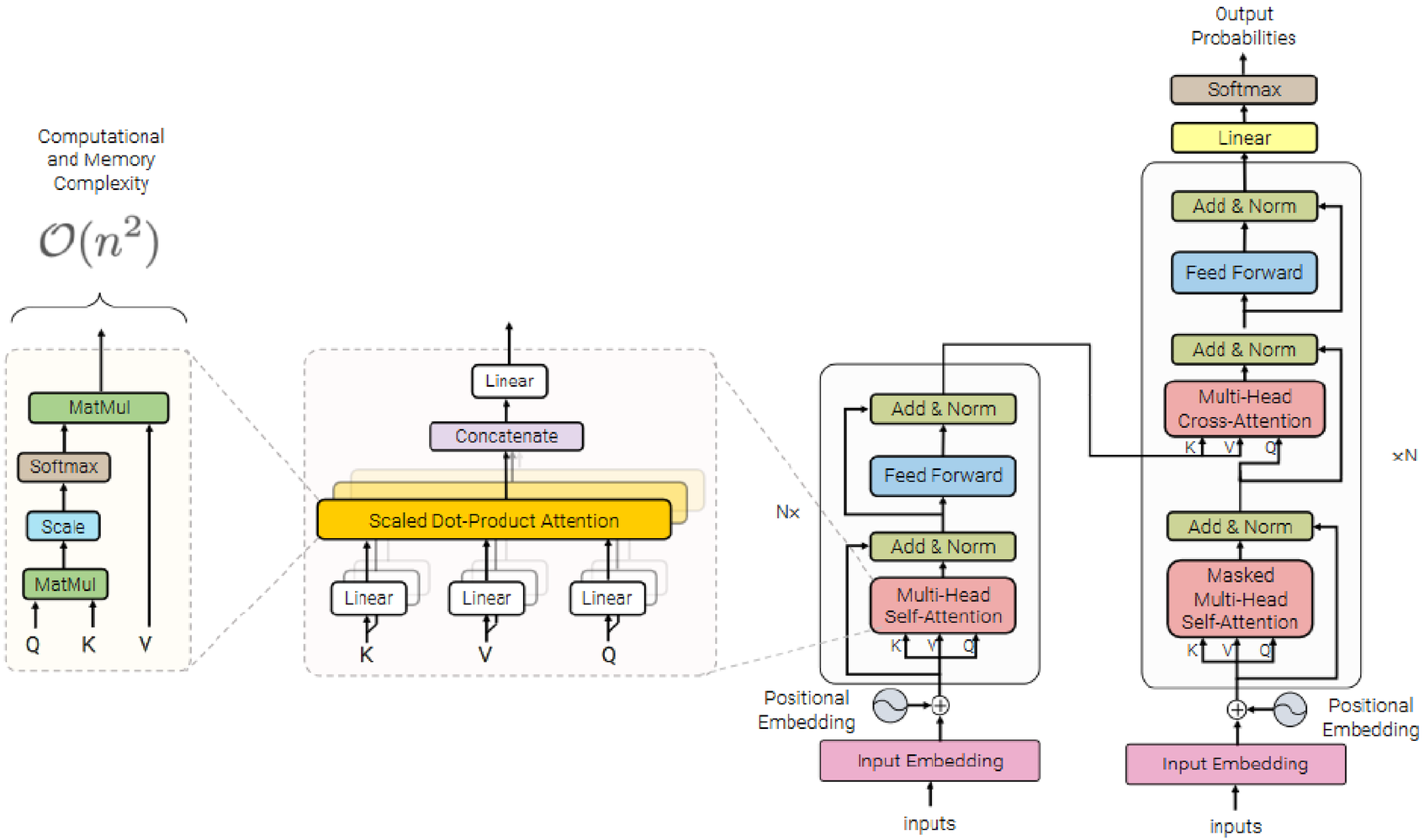
1. Векторное представление входной последовательности
2. Positional encoding
3. Multi-head self-attention слой:
- Полносвязный слой (feed-forward network);
- Нормализацию и остаточные связи (LayerNorm + residual connections).

Декодер:

+ masked self-attention
+ attention к выходу декодера
+ линейный слой и softmax

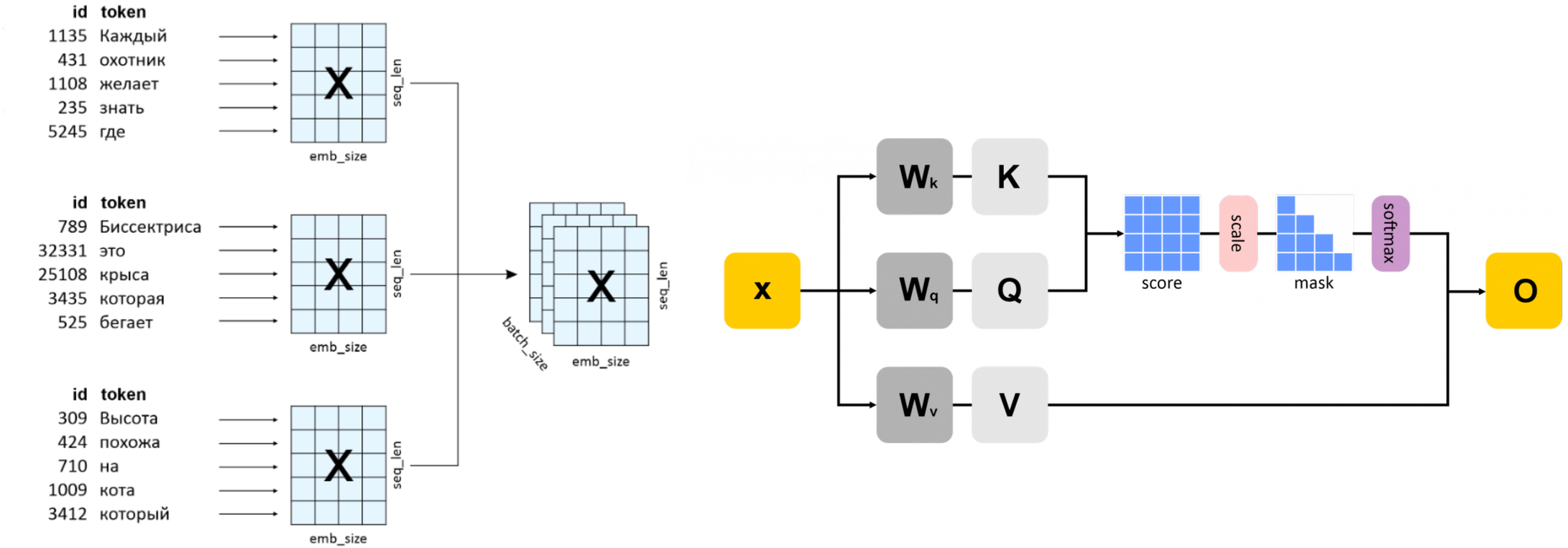


Архитектура Transformer

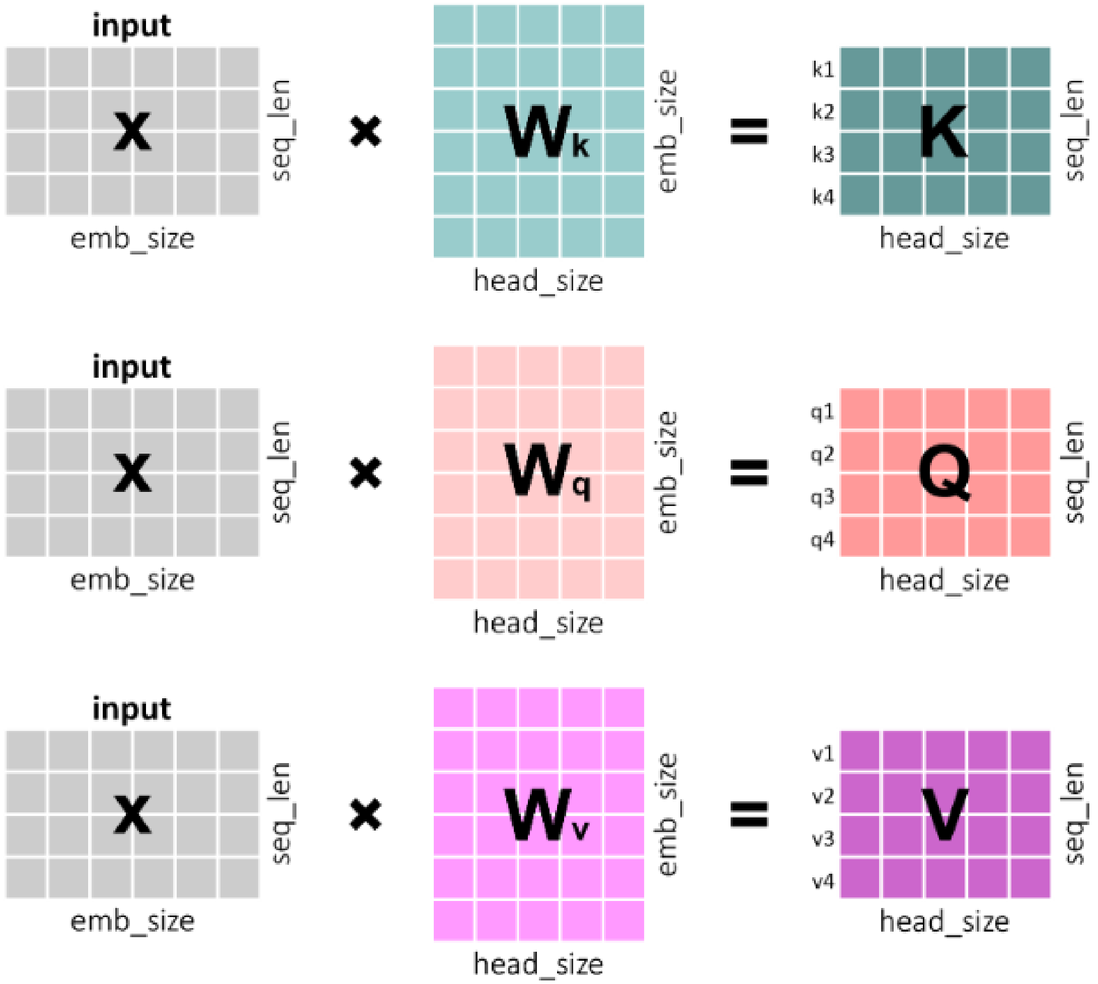


Реализация self-attention

Вход: тензор **x** размерностью batch_size × seq_len × emb_size

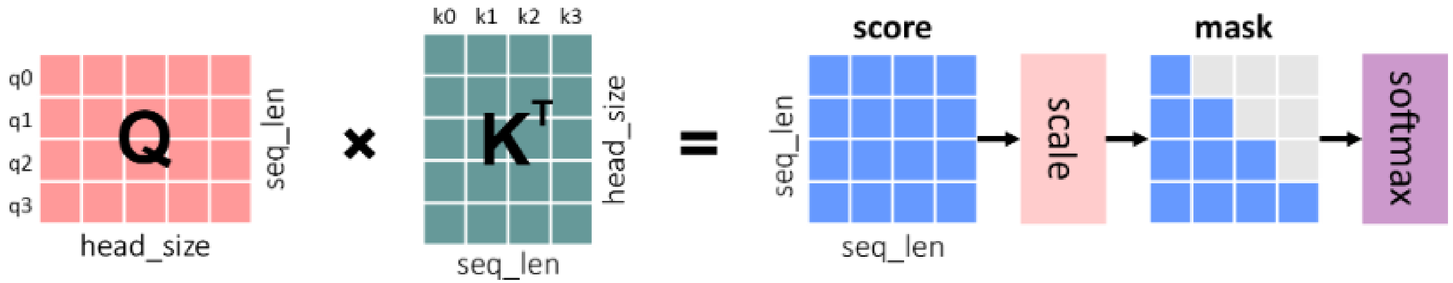


Реализация self-attention

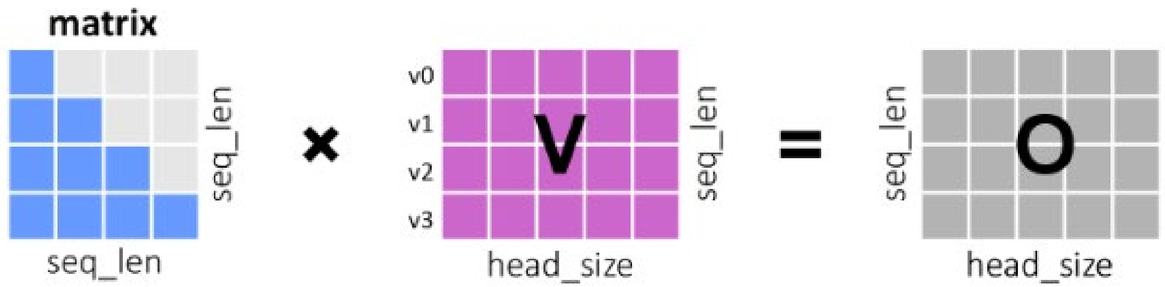


Реализация self-attention

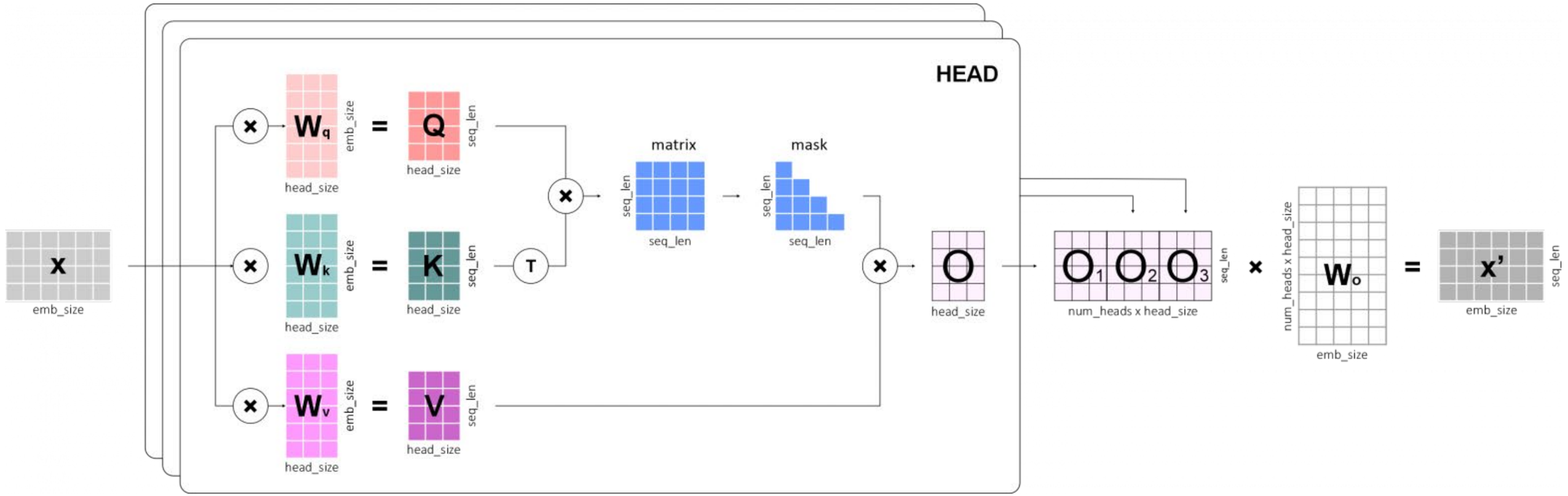
Матрица внимания:



Выходной тензор:



Multi-head attention



МИСИС
УНИВЕРСИТЕТ

Спасибо
за внимание!