

Нейросетевые рекомендации

RecSys

lecturer: Mollaev D. E.
Sber AI Lab

План лекции

- Постановка задачи
- Общие архитектуры
 - RNN
 - Transformers
- Адаптация архитектур для рекомендательных систем
 - GRU4REC
 - SASREC
 - BERT4REC
- Другое

План лекции

- Постановка задачи
- Общие архитектуры
 - RNN
 - Transformers
- Адаптация архитектур для рекомендательных систем
 - GRU4REC
 - SASREC
 - BERT4REC
- Другое

Постановка задачи

В предыдущих методах мы напрямую не учитывали последовательную структуру наших рекомендаций:

Представьте, что вы рекомендуете человеку магазины и видите человека с такой упорядоченной историей:

**зоомагазин, супермаркет, метрополитен, зоомагазин, кофейня,
супермаркет, развлекательный сервис**

Порекомендуете ли совершить следующую покупку в зоомагазине?

Постановка задачи

В предыдущих методах мы напрямую не учитывали последовательную структуру наших рекомендаций:

Представьте, что вы рекомендуете человеку магазины и видите человека с такой упорядоченной историей:

**зоомагазин, супермаркет, метрополитен, зоомагазин, кофейня,
супермаркет, развлекательный сервис, зоомагазин**

Порекомендуете ли теперь совершить следующую покупку в зоомагазине?

Постановка задачи

Представим теперь, что у наша последовательность не отсортирована и мы просто знаем, что человек покупал в магазинах

зоомагазин, супермаркет, кофейня, развлекательный сервис

Пора ли ему рекомендовать зоомагазин теперь?

Постановка задачи

User_id	Item_id	Date
5	4	2024-02-24
6	1	2024-02-24
5	3	2024-02-22
1	2	2024-02-13
5	1	2024-02-10
4	2	2024-02-08
2	3	2024-02-08
2	1	2024-02-07
....
1	4	2023-12-31



U/I	item 1	item2	item 3	item 4
User 1	0	1	0	1
User 2	1	0	1	0
User 3	0	1	0	1
User 4	0	1	0	0
User 5	1	0	1	1
User 6	1	1	0	1
User 7	0	1	1	0
User 8	1	0	0	1

User_id	Item_id	Date
5	4	2024-02-24
6	1	2024-02-24
5	3	2024-02-22
1	2	2024-02-13
5	1	2024-02-10
4	2	2024-02-08
2	3	2024-02-08
2	1	2024-02-07
....
1	4	2023-12-31



User1: [4,1]
User2: [2,3]
User3: [2, 4]
User4: [2]
User5: [1,3,4]
User5: [2,3,1]
User7: [2,3]
User8: [1,4]



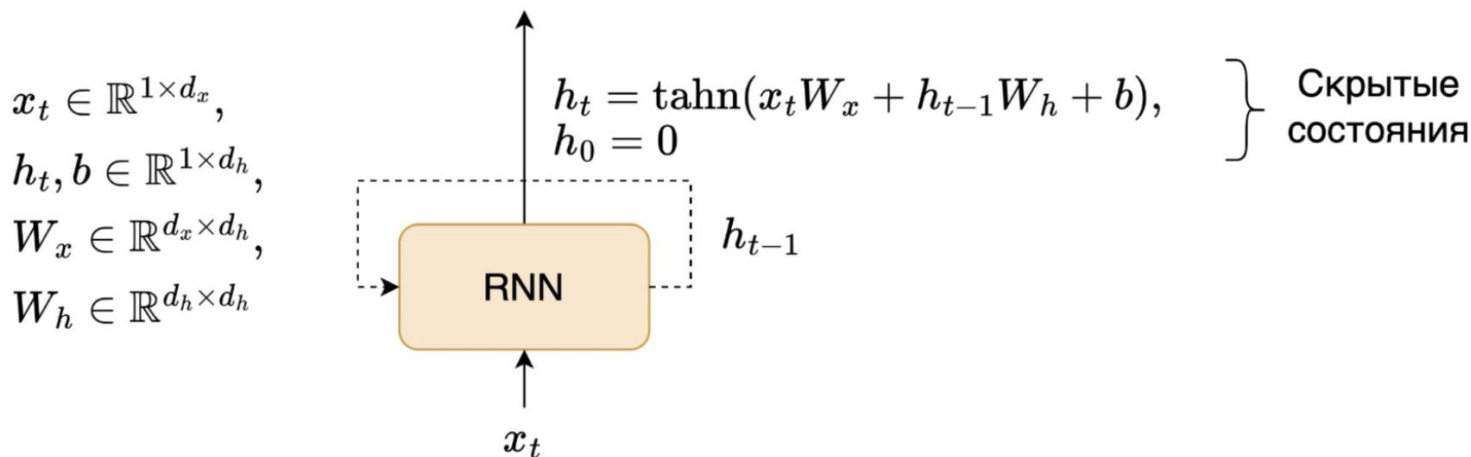
Время

План лекции

- Постановка задачи
- Общие архитектуры
 - RNN
 - Transformers
- Адаптация архитектур для рекомендательных систем
 - GRU4REC
 - SASREC
 - BERT4REC
- Другое

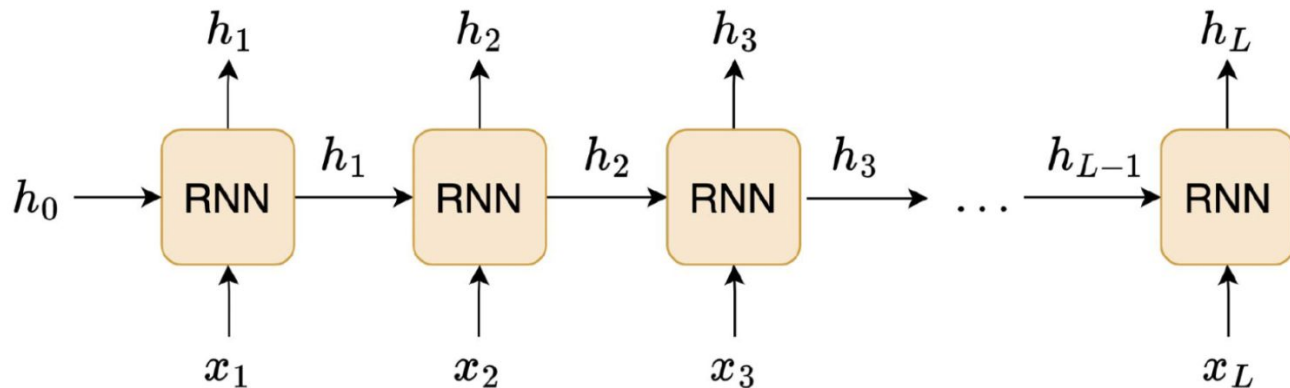
Общие архитектуры: RNN

Цель: обработать последовательность x_1, x_2, \dots, x_L зависимых наблюдений **нефиксированной** длины.



- Состояние h_t можно рассматривать как внутреннюю память модели на шаге t .
- Состояние h_t неявно зависит от всех x_1, \dots, x_t , то есть хранит информацию о входах до шага t включительно.

Общие архитектуры: RNN



$$x_t \in \mathbb{R}^{1 \times d_x},$$

$$h_t, b \in \mathbb{R}^{1 \times d_h},$$

$$W_x \in \mathbb{R}^{d_x \times d_h},$$

$$W_h \in \mathbb{R}^{d_h \times d_h}$$

$$h_t = \text{tahn}(x_t W_x + h_{t-1} W_h + b),$$

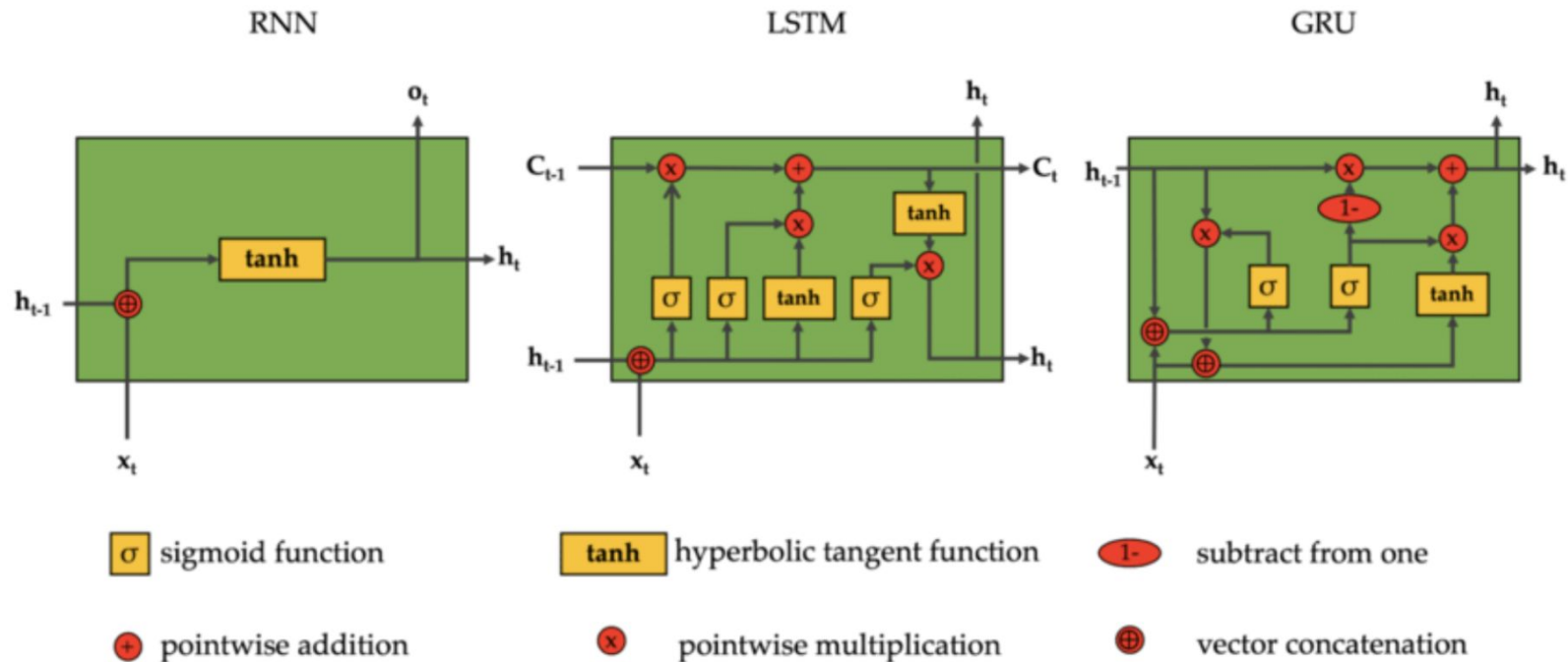
$$h_0 = 0$$

Общие архитектуры: RNN

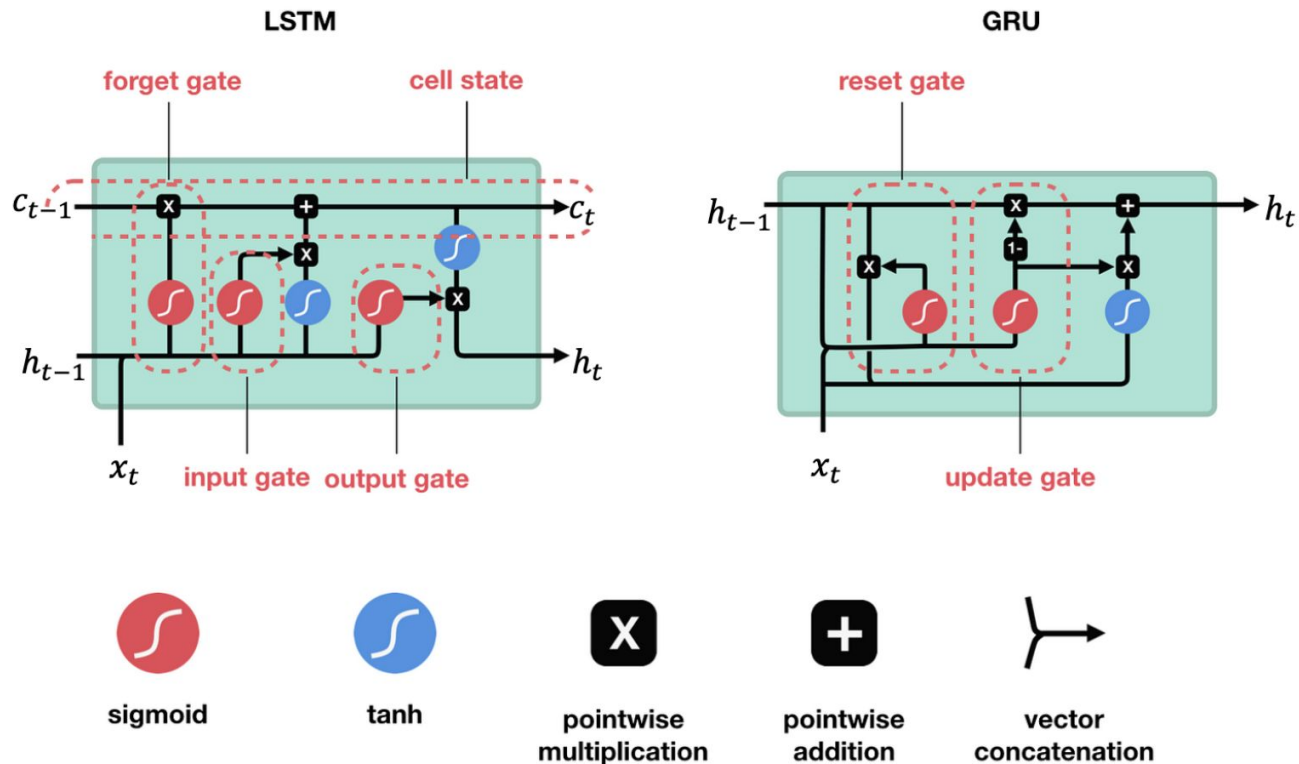
Проблемы:

- Плохо параллелится
- Затухают или взрываются градиенты
- Не может запоминать длинный контекст

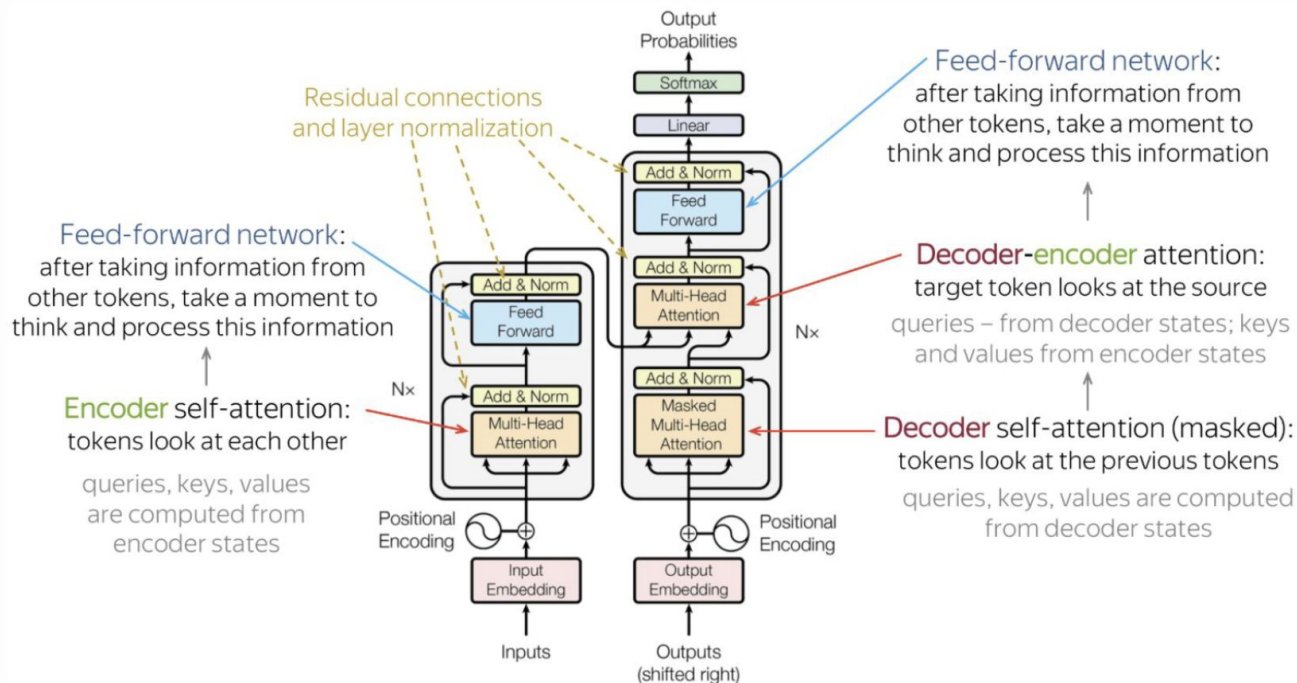
Общие архитектуры: RNN



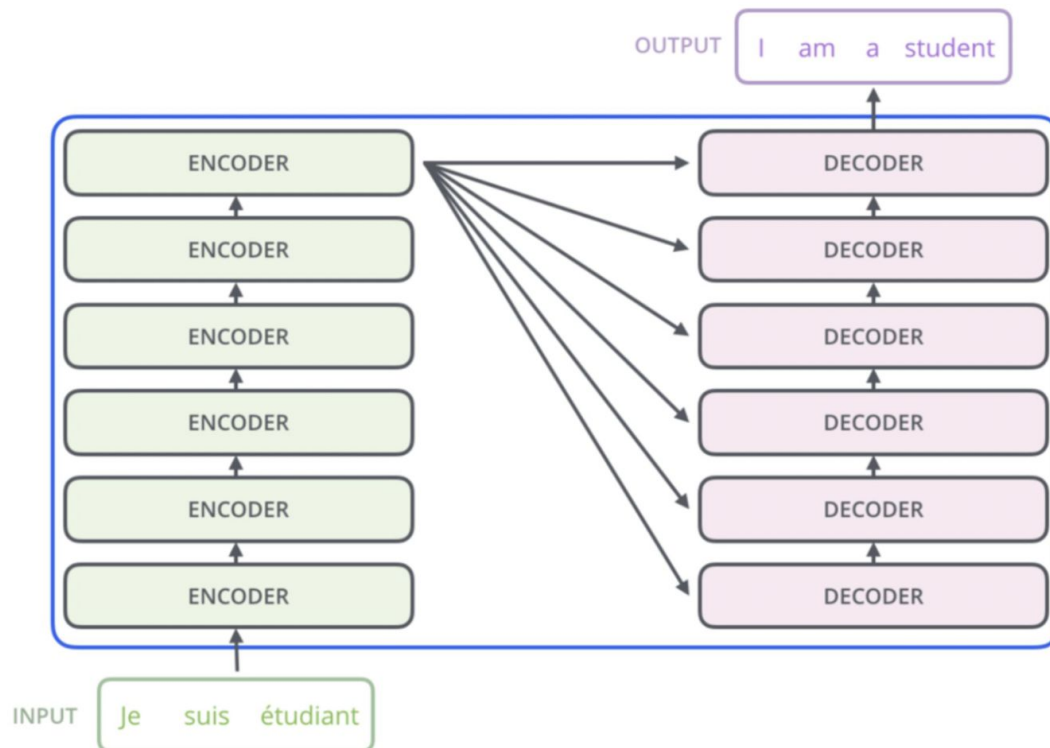
Общие архитектуры: RNN



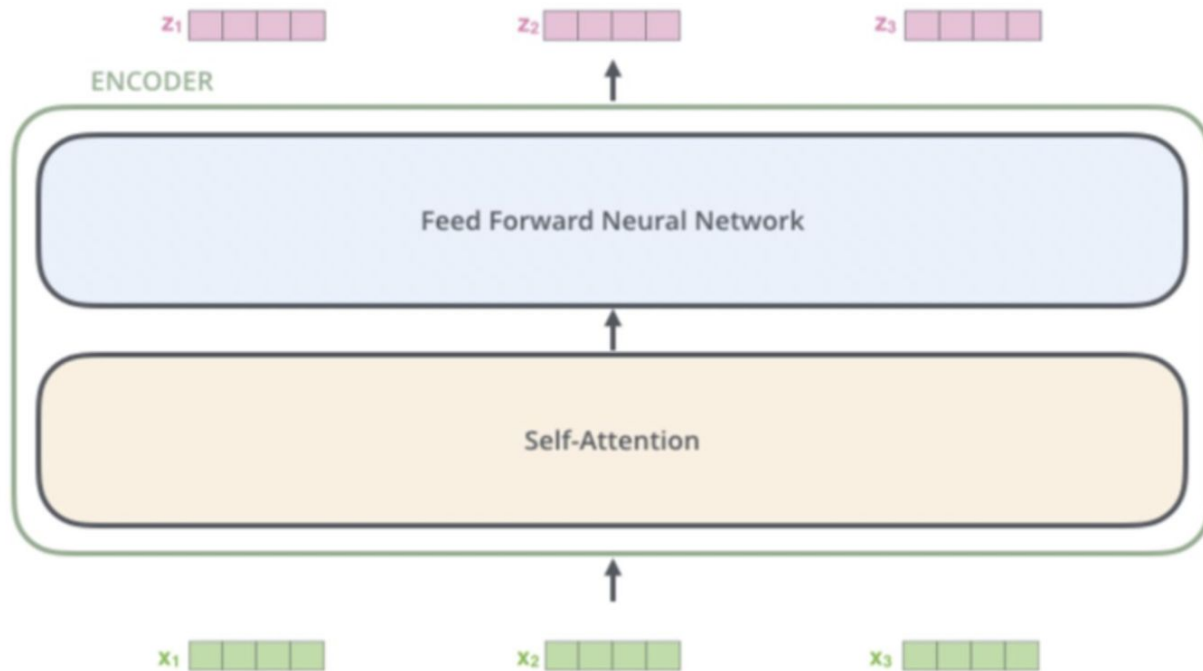
Общие архитектуры: Transformer



Общие архитектуры: Transformer



Общие архитектуры: Transformer



Transformer: self-attention

Each vector receives three representations ("roles")

$$[W_Q] \times \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \end{bmatrix} = \begin{bmatrix} \text{blue} \\ \text{blue} \\ \text{blue} \end{bmatrix}$$

Query: vector **from** which the attention is looking

"Hey there, do you have this information?"

$$[W_K] \times \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \end{bmatrix} = \begin{bmatrix} \text{yellow} \\ \text{yellow} \\ \text{yellow} \end{bmatrix}$$

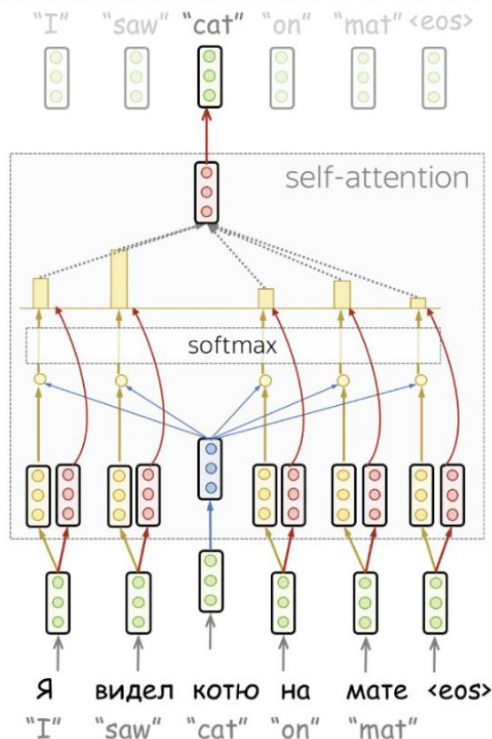
Key: vector **at** which the query looks to compute weights

"Hi, I have this information – give me a large weight!"

$$[W_V] \times \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \end{bmatrix} = \begin{bmatrix} \text{red} \\ \text{red} \\ \text{red} \end{bmatrix}$$

Value: their weighted sum is attention output

"Here's the information I have!"



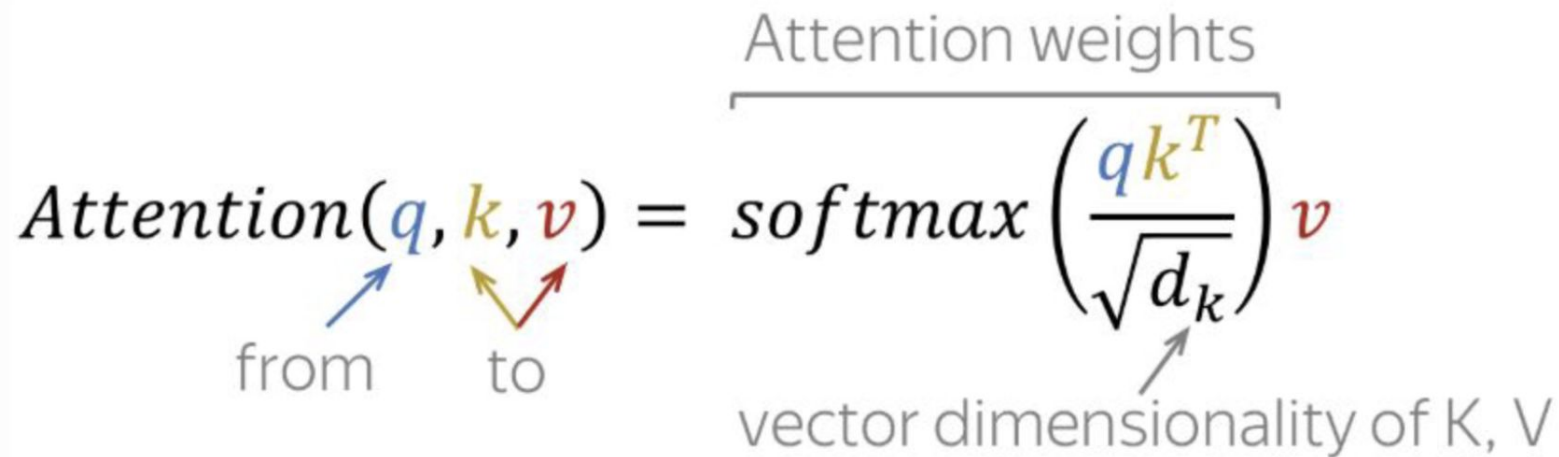
- **query** - asking for information
- **key** - saying that it has some information
- **value** - giving the information

Transformer: self-attention

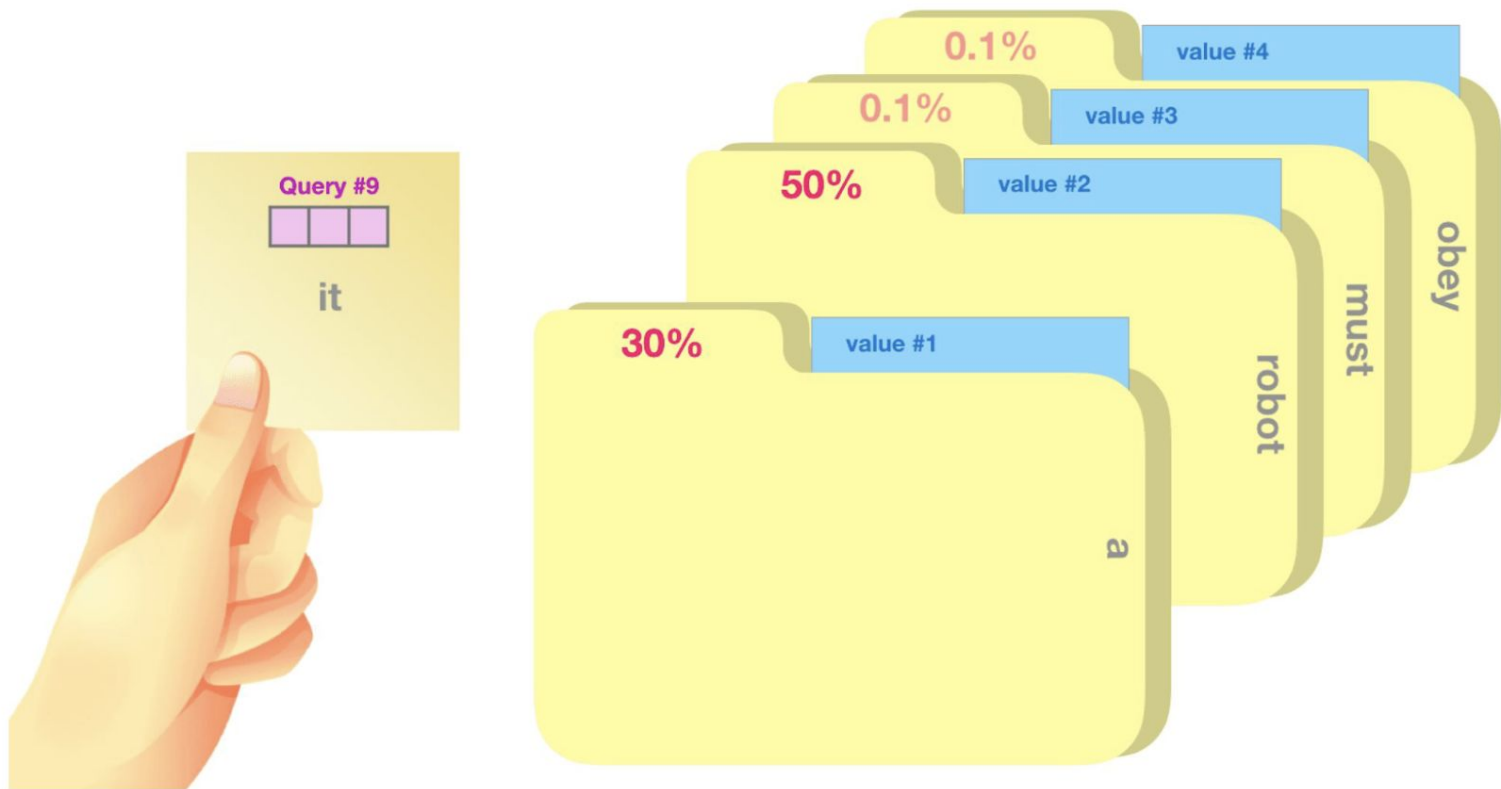
$$\textit{Attention}(q, k, v) = \overbrace{\textit{softmax}\left(\frac{qk^T}{\sqrt{d_k}}\right)}^{\text{Attention weights}} v$$

from to

vector dimensionality of K, V

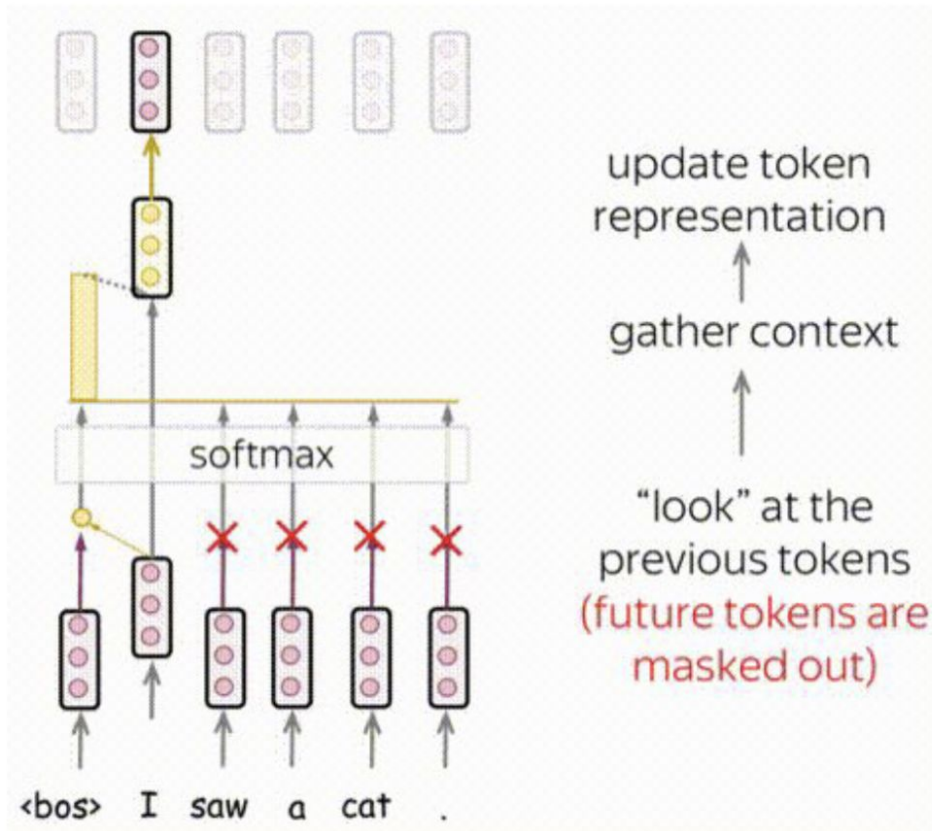
The diagram shows the self-attention formula with several annotations. A blue arrow points from the word 'from' to the query vector 'q'. A yellow arrow points from the word 'to' to the key vector 'k'. A red arrow points from the word 'to' to the value vector 'v'. A horizontal line above the softmax function is labeled 'Attention weights'. A grey arrow points from the text 'vector dimensionality of K, V' to the denominator $\sqrt{d_k}$ in the softmax argument.

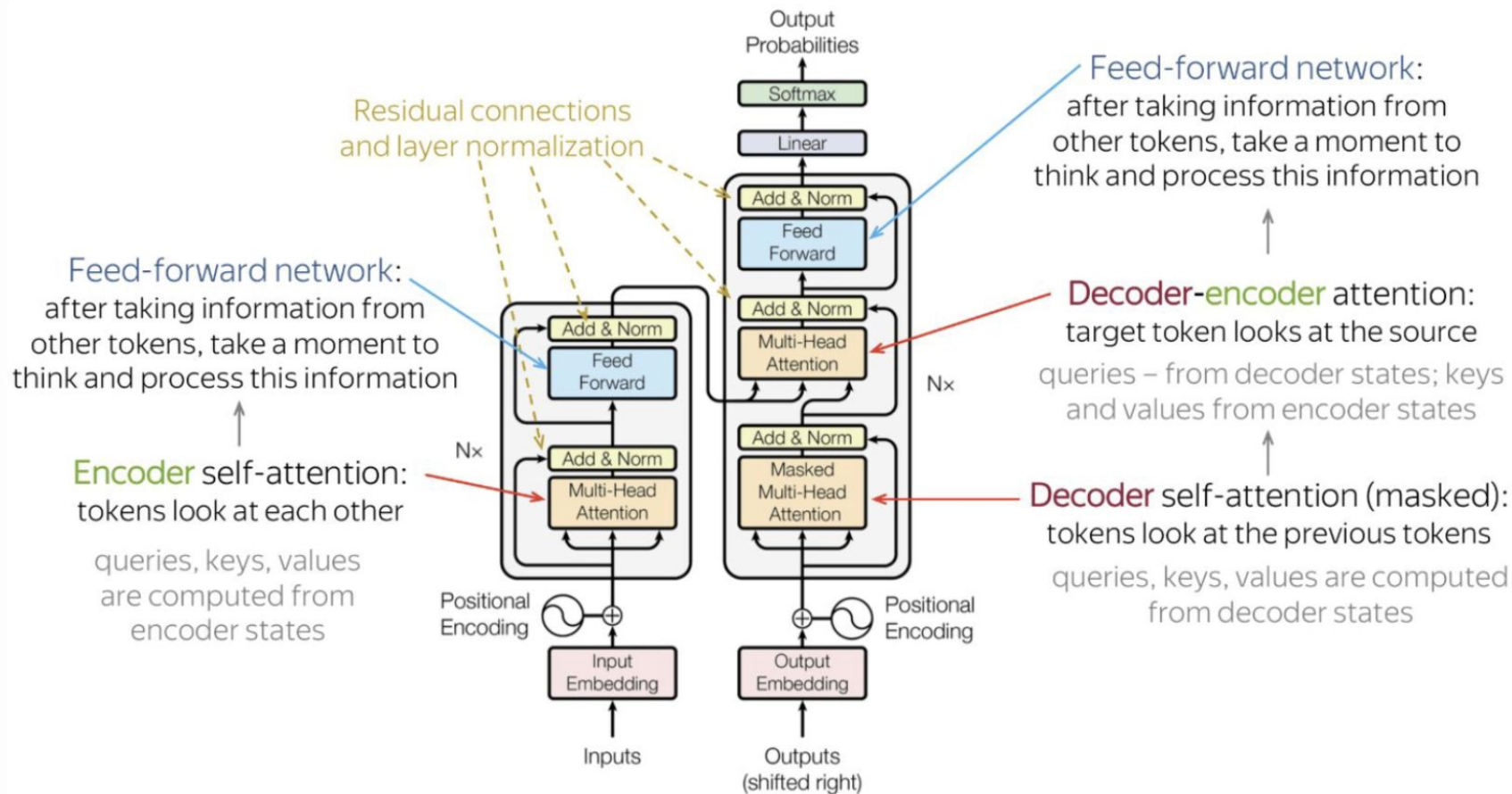
Transformer: self-attention



Transformer: masked self-attention

To forbid the decoder to look ahead, the model uses masked self-attention: future tokens are masked out.

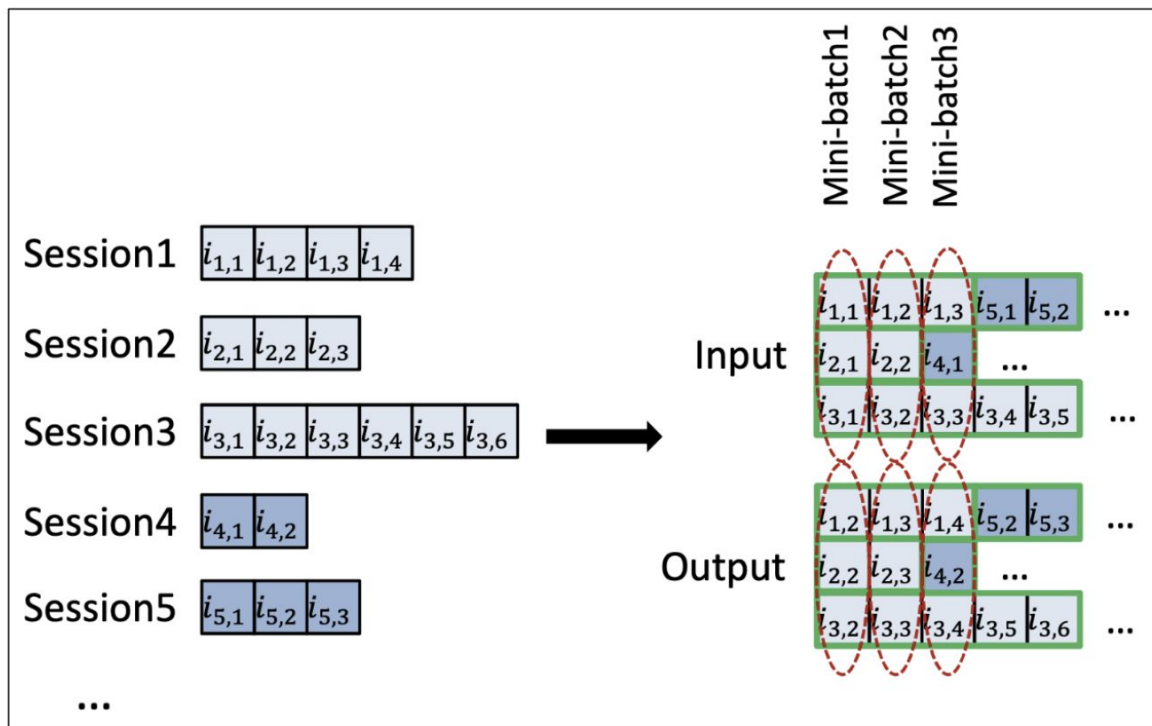




План лекции

- Постановка задачи
- Общие архитектуры
 - RNN
 - Transformers
- Адаптация архитектур для рекомендательных систем
 - GRU4REC
 - SAS4REC
 - BERT4REC
- Другое

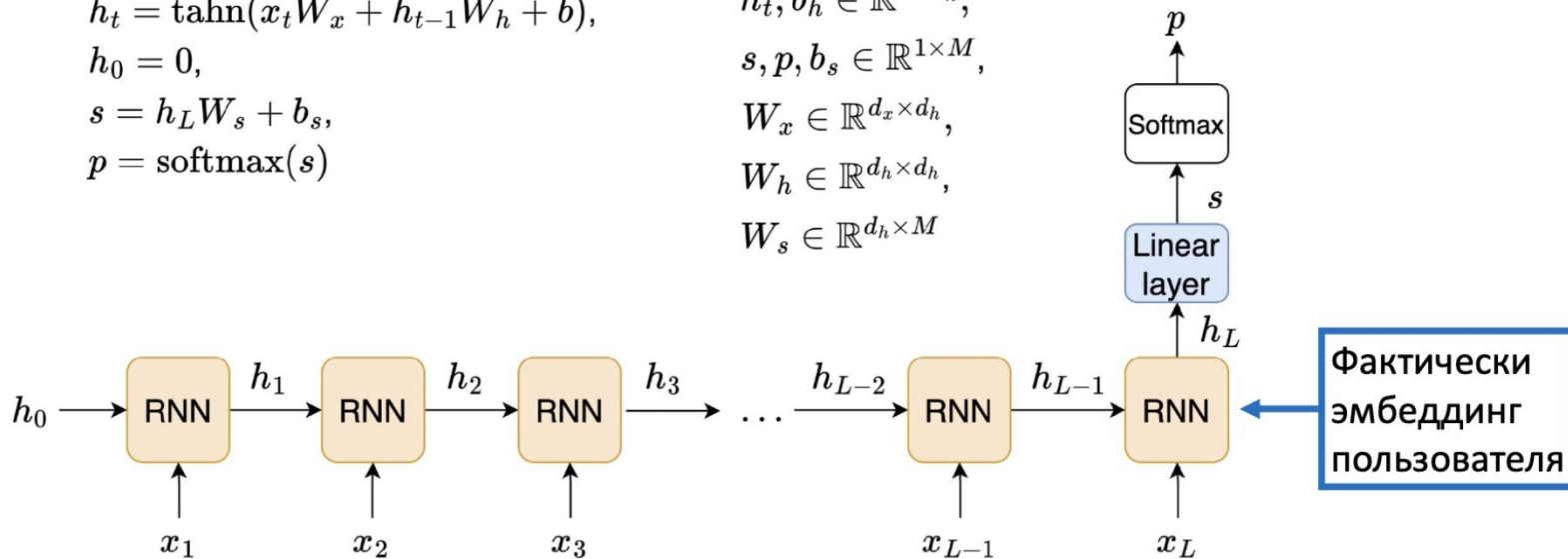
GRU4REC



GRU4REC

$$\begin{aligned}h_t &= \text{tanh}(x_t W_x + h_{t-1} W_h + b), \\h_0 &= 0, \\s &= h_L W_s + b_s, \\p &= \text{softmax}(s)\end{aligned}$$

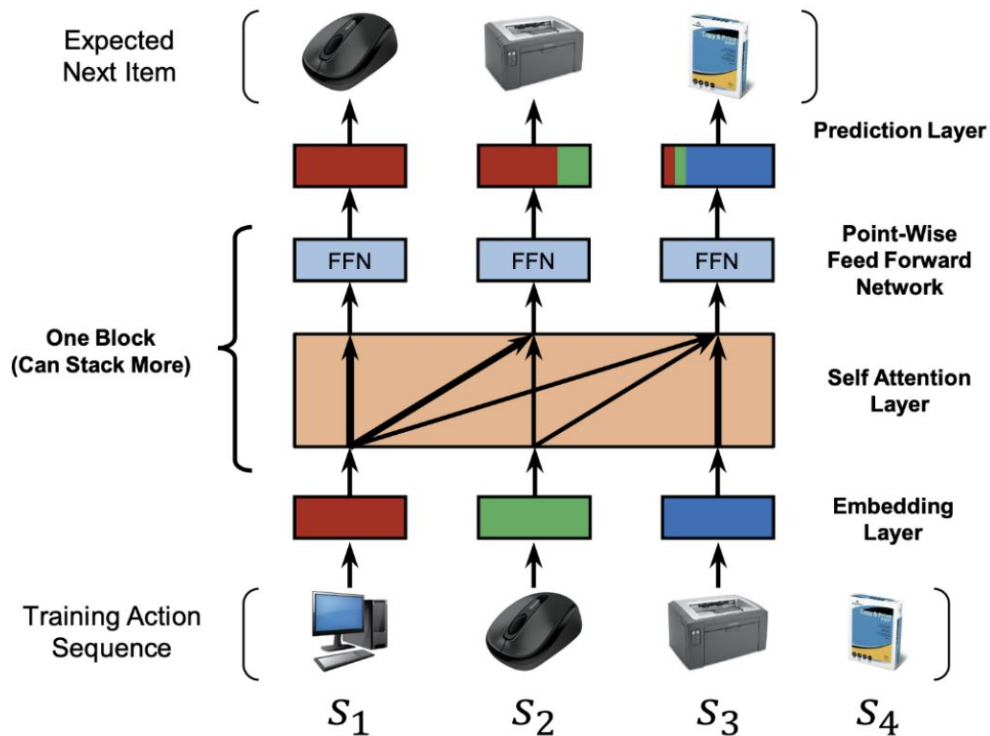
$$\begin{aligned}x_t &\in \mathbb{R}^{1 \times d_x}, \\h_t, b_h &\in \mathbb{R}^{1 \times d_h}, \\s, p, b_s &\in \mathbb{R}^{1 \times M}, \\W_x &\in \mathbb{R}^{d_x \times d_h}, \\W_h &\in \mathbb{R}^{d_h \times d_h}, \\W_s &\in \mathbb{R}^{d_h \times M}\end{aligned}$$



s - логиты,

p - предсказанное распределение (M классов)

SASREC (Self-Attentive Sequential Recommendation)



SASREC

- Shared эмбеддинги айтемов на **входе** и на **выходе**
- BCELoss
- Обучаемые positional embeddings
- Маскируем аттеншн: не можем смотреть в будущее

$$L_{BCE} = - \sum_{\mathcal{S}^u \in \mathcal{S}} \sum_{t \in [1, 2, \dots, n]} \left[\log(\sigma(r_{o_t, t})) + \sum_{j \notin \mathcal{S}^u} \log(1 - \sigma(r_{j, t})) \right].$$

Note that we ignore the terms where $o_t = \text{<pad>}$.

SASREC

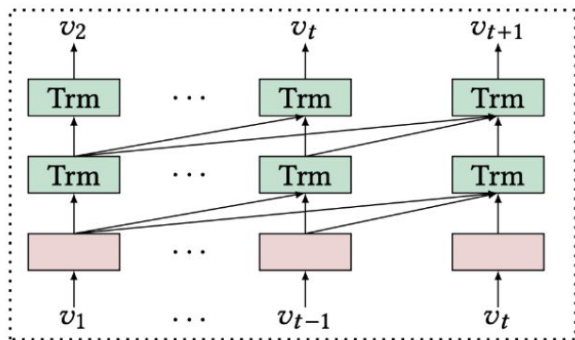
Как посчитать релевантность i -го айтема:

- Прогоняем всю последовательность через модель
- Берем последний hidden layer
- Умножаем на $embedding_i$

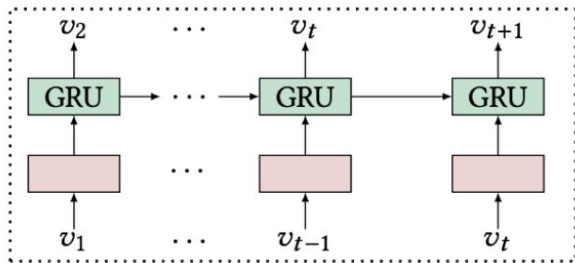
SASREC: качество

Dataset	Metric	(a) PopRec	(b) BPR	(c) FMC	(d) FPMC	(e) TransRec	(f) GRU4Rec	(g) GRU4Rec ⁺	(h) Caser	(i) SASRec	Improvement vs. (a)-(e) (f)-(h)	
<i>Beauty</i>	Hit@10	0.4003	0.3775	0.3771	0.4310	<u>0.4607</u>	0.2125	0.3949	0.4264	0.4854	5.4%	13.8%
	NDCG@10	0.2277	0.2183	0.2477	0.2891	<u>0.3020</u>	0.1203	0.2556	0.2547	0.3219	6.6%	25.9%
<i>Games</i>	Hit@10	0.4724	0.4853	0.6358	0.6802	<u>0.6838</u>	0.2938	0.6599	0.5282	0.7410	8.5%	12.3%
	NDCG@10	0.2779	0.2875	0.4456	0.4680	0.4557	0.1837	<u>0.4759</u>	0.3214	0.5360	14.5%	12.6%
<i>Steam</i>	Hit@10	0.7172	0.7061	0.7731	0.7710	0.7624	0.4190	<u>0.8018</u>	0.7874	0.8729	13.2%	8.9%
	NDCG@10	0.4535	0.4436	0.5193	0.5011	0.4852	0.2691	<u>0.5595</u>	0.5381	0.6306	21.4%	12.7%
<i>ML-1M</i>	Hit@10	0.4329	0.5781	0.6986	0.7599	0.6413	0.5581	0.7501	<u>0.7886</u>	0.8245	8.5%	4.6%
	NDCG@10	0.2377	0.3287	0.4676	0.5176	0.3969	0.3381	0.5513	<u>0.5538</u>	0.5905	14.1%	6.6%

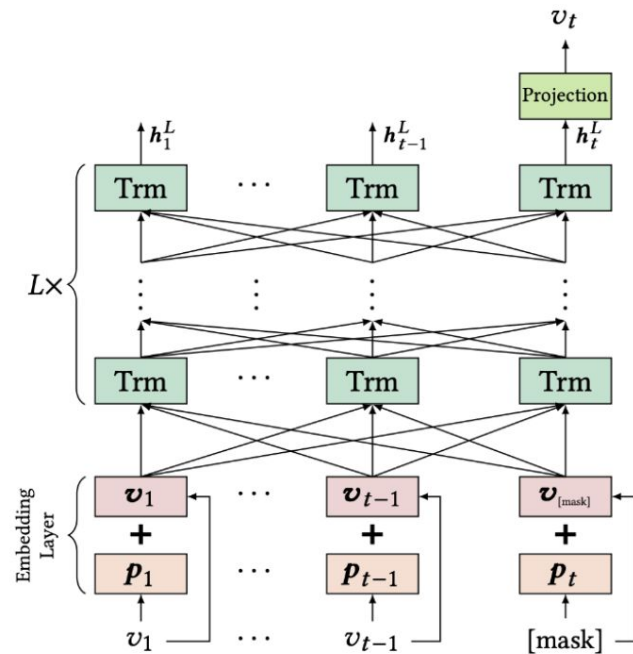
BERT4REC



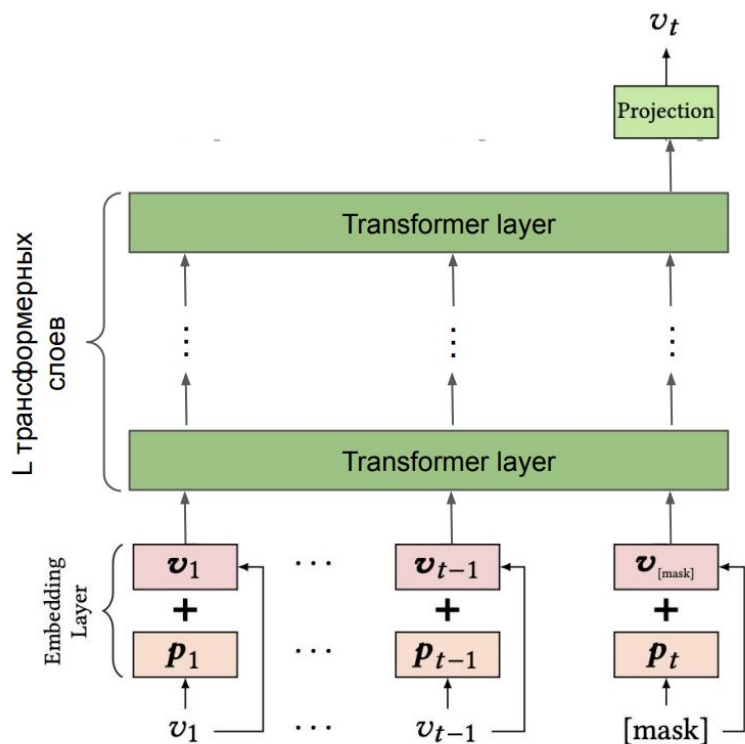
(c) SASRec model architecture.



(d) RNN based sequential recommendation methods.



Архитектура Bert4Rec

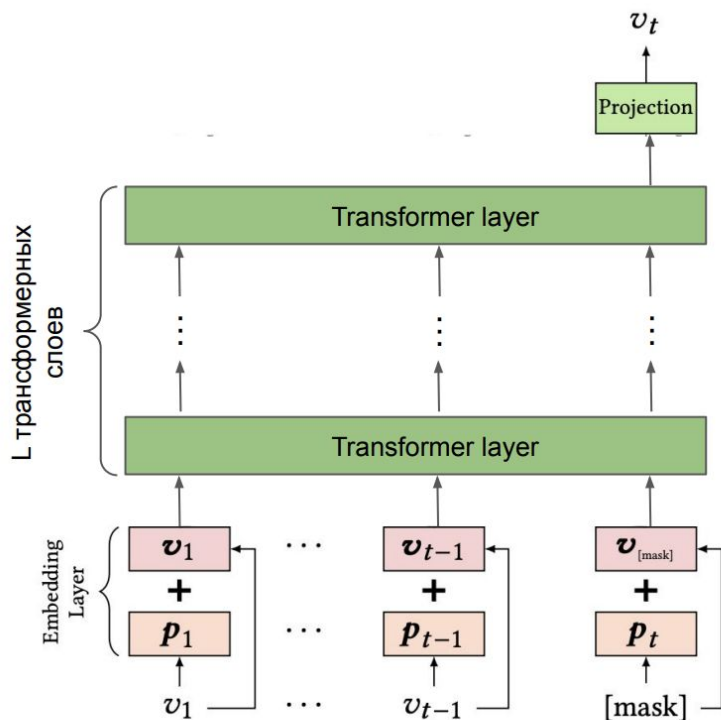


(b) BERT4Rec model architecture.

Модель состоит из

- Слоя эмбедингов айтеров
- L трансформерных слоев
- Проецирующей головы, выполняющей предсказания

Архитектура Bert4Rec



(b) BERT4Rec model architecture.

Модель состоит из

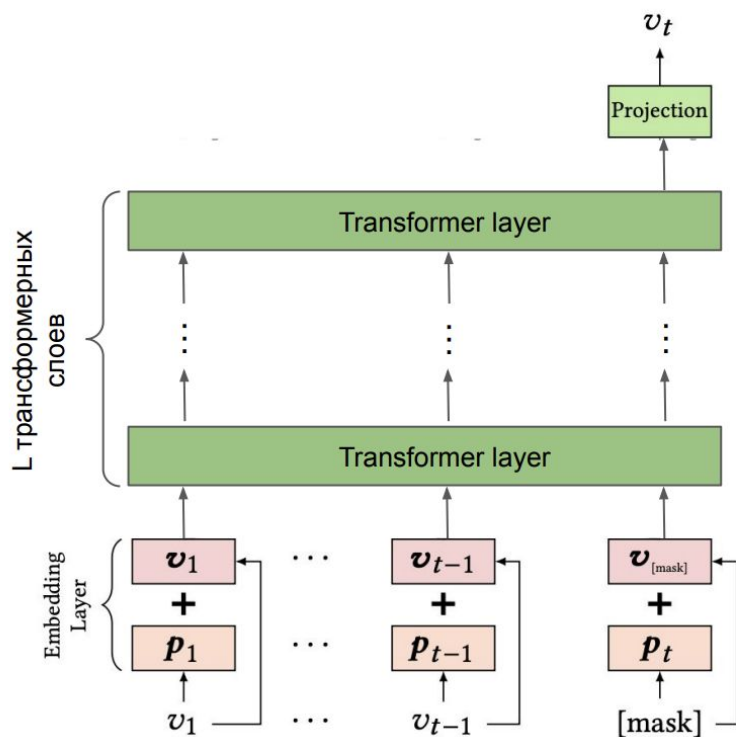
- Слоя эмбеддингов айтемов
- L трансформерных слоев
- Проецирующей головы, выполняющей предсказания

На вход подаются айтемы позитивных взаимодействий пользователя в порядке по времени

В конец истории добавляется специальный айтем [mask]

К выходному эмбеддингу, соответствующему этому специальному айтему, применяем проекционную голову, предсказывающую релевантный следующий айтем

Архитектура Bert4Rec



(b) BERT4Rec model architecture.

Модель состоит из

- Слоя эмбеддингов айтемов
- L трансформерных слоев
- Проецирующей головы, выполняющей предсказания

На вход подаются айтемы позитивных взаимодействий пользователя в порядке по времени

В конец истории добавляется специальный айтем [mask]

К выходному эмбеддингу, соответствующему этому специальному айтему, применяем проекционную голову, предсказывающую релевантный следующий айтем

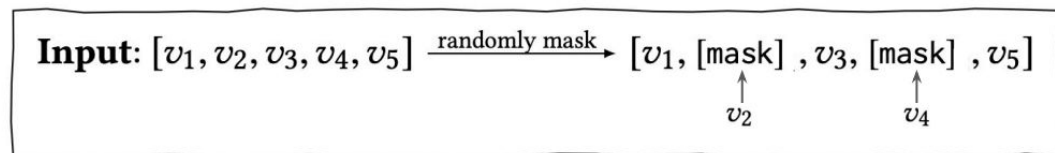
Проекционный слой устроен как

$$P(v) = \text{softmax} \left(GELU(h_i^L W^P + b^P) E^T + b^O \right),$$

где $E \in \mathbb{R}^{|V| \times d}$ – обучаемая матрица айтемов.

Обучение Bert4Rec

- Случайным образом замаскируем долю айтемов ρ из истории пользователя, то есть заменим на специальный айтем [mask]



- Пусть S_u^m – множество замаскированных позиций пользователя u ,
 S'_u – последовательность айтемов с замененными на [mask] айтемами,
 v_m^* – предсказания айтемов на замаскированных позициях

Тогда оптимизируем правдоподобие, то есть:

$$\mathcal{L} = -\frac{1}{|S_u^m|} \sum_{m \in S_u^m} \log P(v_m^* = v_m | S'_u)$$

- Другими словами, обучение похоже на Masked Language Model обучение в текстовых задачах

BERT4REC: качество

Datasets	Metric	POP	BPR-MF	NCF	FPMC	GRU4Rec	GRU4Rec ⁺	Caser	SASRec	BERT4Rec	Improv.
Beauty	HR@1	0.0077	0.0415	0.0407	0.0435	0.0402	0.0551	0.0475	<u>0.0906</u>	0.0953	5.19%
	HR@5	0.0392	0.1209	0.1305	0.1387	0.1315	0.1781	0.1625	<u>0.1934</u>	0.2207	14.12%
	HR@10	0.0762	0.1992	0.2142	0.2401	0.2343	0.2654	0.2590	<u>0.2653</u>	0.3025	14.02%
	NDCG@5	0.0230	0.0814	0.0855	0.0902	0.0812	0.1172	0.1050	<u>0.1436</u>	0.1599	11.35%
	NDCG@10	0.0349	0.1064	0.1124	0.1211	0.1074	0.1453	0.1360	<u>0.1633</u>	0.1862	14.02%
	MRR	0.0437	0.1006	0.1043	0.1056	0.1023	0.1299	0.1205	<u>0.1536</u>	0.1701	10.74%
Steam	HR@1	0.0159	0.0314	0.0246	0.0358	0.0574	0.0812	0.0495	<u>0.0885</u>	0.0957	8.14%
	HR@5	0.0805	0.1177	0.1203	0.1517	0.2171	0.2391	0.1766	<u>0.2559</u>	0.2710	5.90%
	HR@10	0.1389	0.1993	0.2169	0.2551	0.3313	0.3594	0.2870	<u>0.3783</u>	0.4013	6.08%
	NDCG@5	0.0477	0.0744	0.0717	0.0945	0.1370	0.1613	0.1131	<u>0.1727</u>	0.1842	6.66%
	NDCG@10	0.0665	0.1005	0.1026	0.1283	0.1802	0.2053	0.1484	<u>0.2147</u>	0.2261	5.31%
	MRR	0.0669	0.0942	0.0932	0.1139	0.1420	0.1757	0.1305	<u>0.1874</u>	0.1949	4.00%
ML-1m	HR@1	0.0141	0.0914	0.0397	0.1386	0.1583	0.2092	0.2194	<u>0.2351</u>	0.2863	21.78%
	HR@5	0.0715	0.2866	0.1932	0.4297	0.4673	0.5103	0.5353	<u>0.5434</u>	0.5876	8.13%
	HR@10	0.1358	0.4301	0.3477	0.5946	0.6207	0.6351	<u>0.6692</u>	0.6629	0.6970	4.15%
	NDCG@5	0.0416	0.1903	0.1146	0.2885	0.3196	0.3705	0.3832	<u>0.3980</u>	0.4454	11.91%
	NDCG@10	0.0621	0.2365	0.1640	0.3439	0.3627	0.4064	0.4268	<u>0.4368</u>	0.4818	10.32%
	MRR	0.0627	0.2009	0.1358	0.2891	0.3041	0.3462	0.3648	<u>0.3790</u>	0.4254	12.24%
ML-20m	HR@1	0.0221	0.0553	0.0231	0.1079	0.1459	0.2021	0.1232	<u>0.2544</u>	0.3440	35.22%
	HR@5	0.0805	0.2128	0.1358	0.3601	0.4657	0.5118	0.3804	<u>0.5727</u>	0.6323	10.41%
	HR@10	0.1378	0.3538	0.2922	0.5201	0.5844	0.6524	0.5427	<u>0.7136</u>	0.7473	4.72%
	NDCG@5	0.0511	0.1332	0.0771	0.2239	0.3090	0.3630	0.2538	<u>0.4208</u>	0.4967	18.04%
	NDCG@10	0.0695	0.1786	0.1271	0.2895	0.3637	0.4087	0.3062	<u>0.4665</u>	0.5340	14.47%
	MRR	0.0709	0.1503	0.1072	0.2273	0.2967	0.3476	0.2529	<u>0.4026</u>	0.4785	18.85%

План лекции

- Постановка задачи
- Общие архитектуры
 - RNN
 - Transformers
- Адаптация архитектур для рекомендательных систем
 - GRU4REC
 - SASREC
 - BERT4REC
- Другое

Sequential modelling: Loss functions

Original SASRec loss: binary cross entropy with one negative sample for each positive

$$\mathcal{L}_{BCE} = - \sum_{u \in U} \sum_{t=1}^{n_u} \log(\sigma(r_{t,i_t}^{(u)})) + \log(1 - \sigma(r_{t,-}^{(u)})),$$

BERT4Rec loss: full cross entropy

$$\mathcal{L}_{CE} = - \sum_{u \in U} \sum_{t \in T_u} \log \frac{\exp(r_{t,i_t}^{(u)})}{\sum_{i \in I} \exp(r_{t,i}^{(u)})}$$

Sampled cross-entropy from “Turning Dross Into Gold Loss: is BERT4Rec really better than SASRec?”

$$\mathcal{L}_{CE-sampled_N} = - \sum_{u \in U} \sum_{t=1}^{n_u} \log \frac{\exp(r_{t,i_t}^{(u)})}{\exp(r_{t,i_t}^{(u)}) + \sum_{i \in I_N^-(u)} \exp(r_{t,i}^{(u)})},$$

Does It Look Sequential? An Analysis of Datasets for Evaluation of Sequential Recommendations

В данной работе оценивают насколько сильную последовательную структуру имеют некоторые открытые датасеты

