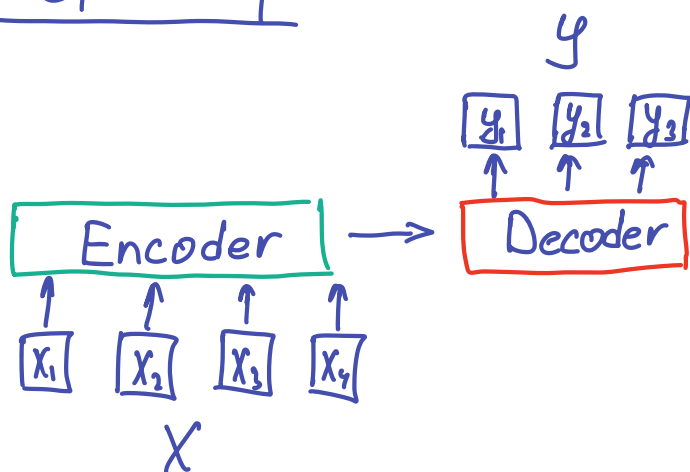


Seq 2 Seq



X - input sequence

$$P_{\theta}(Y|X) = \prod_{i=1}^n P_{\theta}(y_i | X, y_{<i}) \rightarrow \max_{\theta}$$

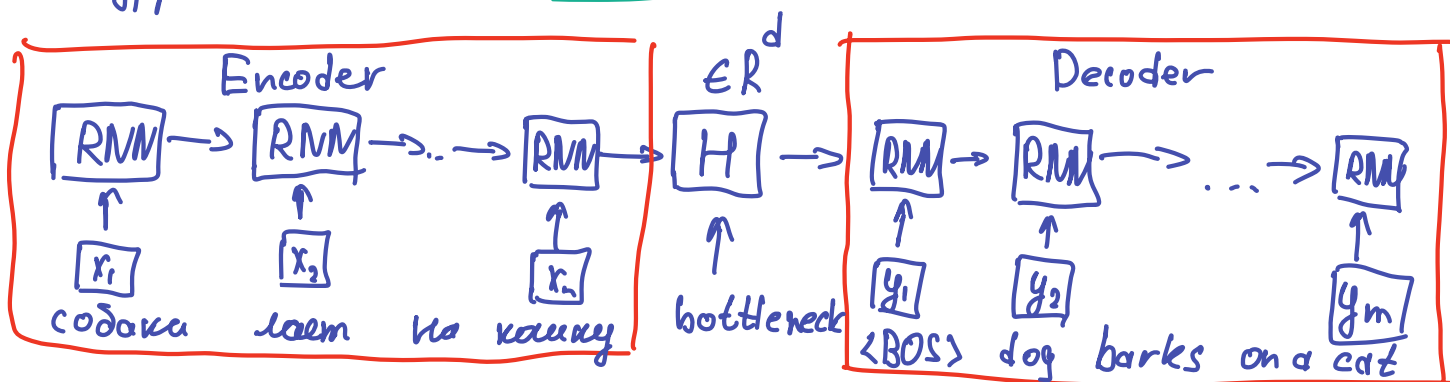
↑
output sequence

$$\hat{y}_i = \operatorname{argmax}_{y_i} P_{\theta}(y_i | X, \hat{y}_{<i}) - \text{Жадная итерация}$$

$$P_{\theta}(\hat{Y}|X) = \prod_{i=1}^n \max_{y_i} P_{\theta}(y_i | X, \hat{y}_{<i}) \neq \max_Y \prod_{i=1}^n P(y_i | X, y_{<i})$$

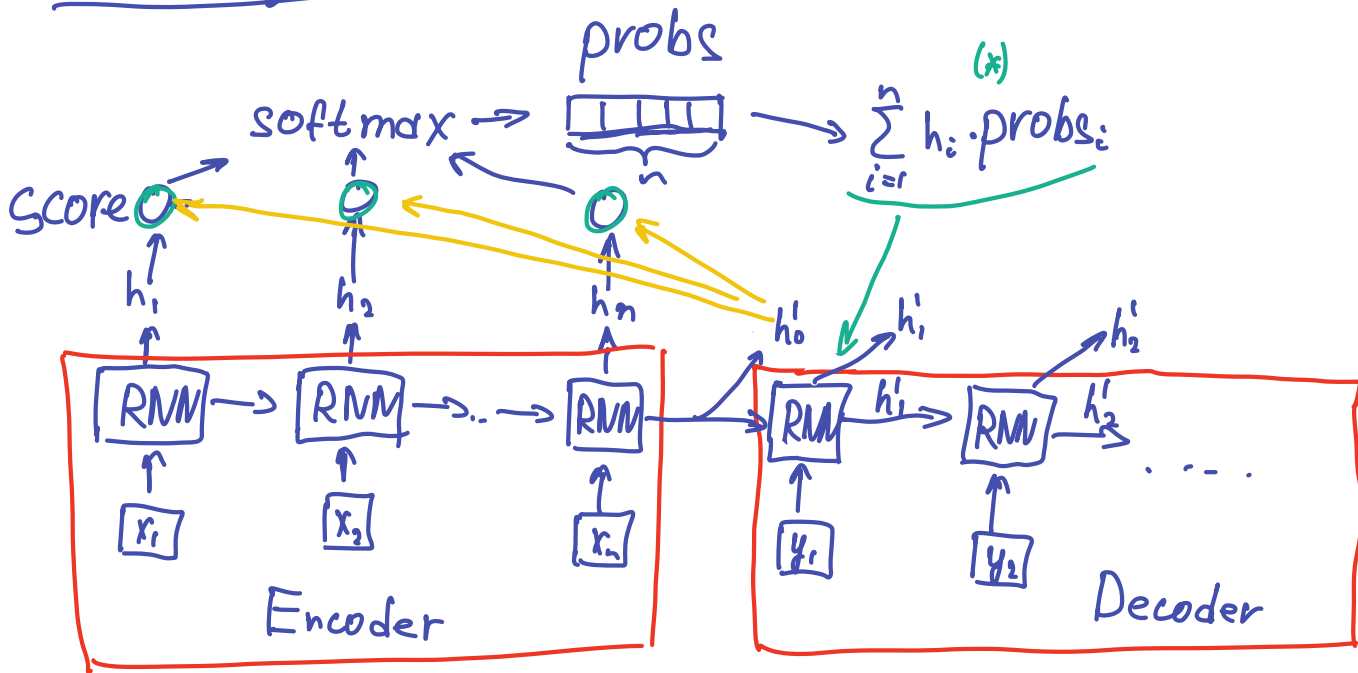
Для итерации лучше Beam Search

Рекуррентные сети не позволяют извлечь всю информацию



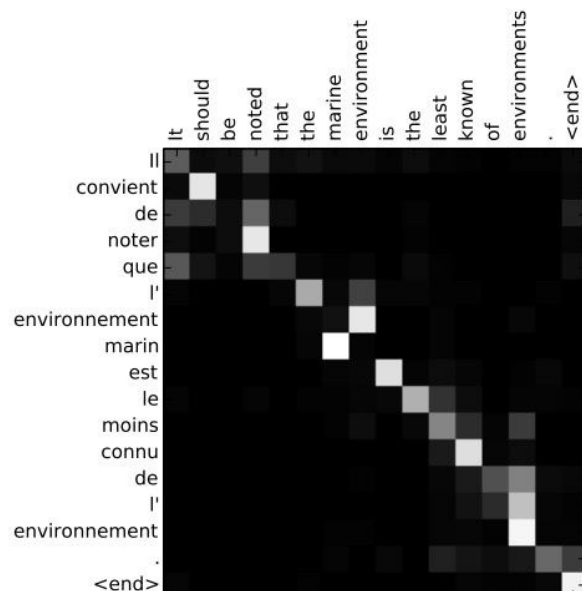
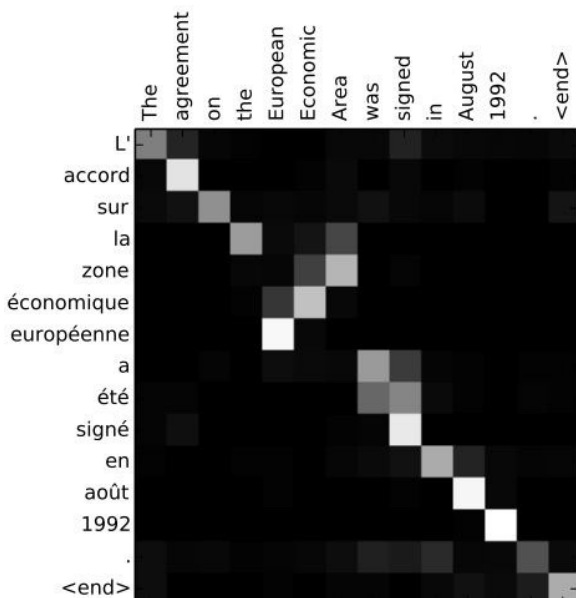
Обучение: Cross-Entropy

Механизм внимания. (2014)



Score functions:

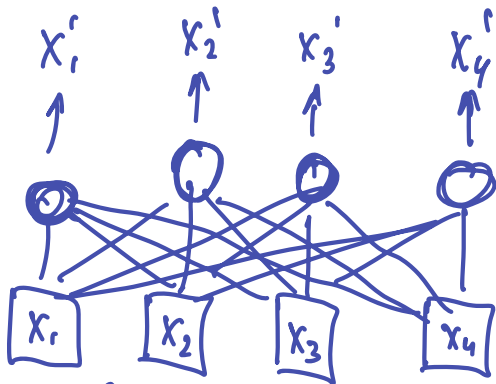
- 1) Dot-product: $s(h, h') = h^T h'$
- 2) Bilinear: $s(h, h') = \underbrace{h^T}_{R^d} \underbrace{W_s}_{R^{d \times d}} \underbrace{h'}_{R^d}$
- 3) MLP: $s(h, h') = \vec{w}_s^T \tanh \left(W_s \begin{bmatrix} h^T \\ h' \end{bmatrix} \right)$



Learned Attention Matrices

Self-attention

Encoder



на позиции значения весит значения
 pos embed. $\begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$

$$\begin{cases} q_i = W_q X_i \\ k_i = W_k X_i \\ v_i = W_v X_i \end{cases}$$

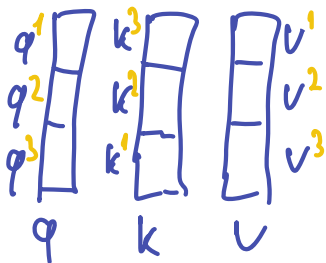
$$\text{score}_{ij} = \frac{q_i^T k_j}{\sqrt{d}}$$

$$\text{attention}_i = \text{softmax}(\text{score}_i)$$

$$X'_i = \sum_{j=1}^n \underset{\substack{\uparrow \\ \text{scalar}}}{\text{attention}_{ij}} v_j$$

Multi-head Attention

Идея: разобьем q, k, v на n_{heads} векторов. Каждая голова будет извлекать свою информацию. Потом склеим все выходы вместе.



$$\text{attention}_i^S = \text{softmax} \left(\frac{q_i^S k_j^S}{\sqrt{d}} \right)$$

$$X_i^S = \sum_{j=1}^n \text{attention}_i^S v_j^S$$

$$\begin{bmatrix} x_i^1 \\ x_i^2 \\ x_i^3 \end{bmatrix} = x_i'$$

Masked Self-Attention (Decoder)

$$\text{attention}_{ij} = \text{softmax}\left(\frac{q_i^T k_j}{\sqrt{d}} + \text{mask}_{ij}\right)$$

$$\text{mask}_{ij} = \begin{cases} 0, & i \leq j \\ -\infty, & i > j \end{cases}$$

$$\Rightarrow \forall i > j: \text{attention}_{ij} = 0$$

Загружаем внимание так,
чтобы декодер при обучении
не мог смотреть вперед

