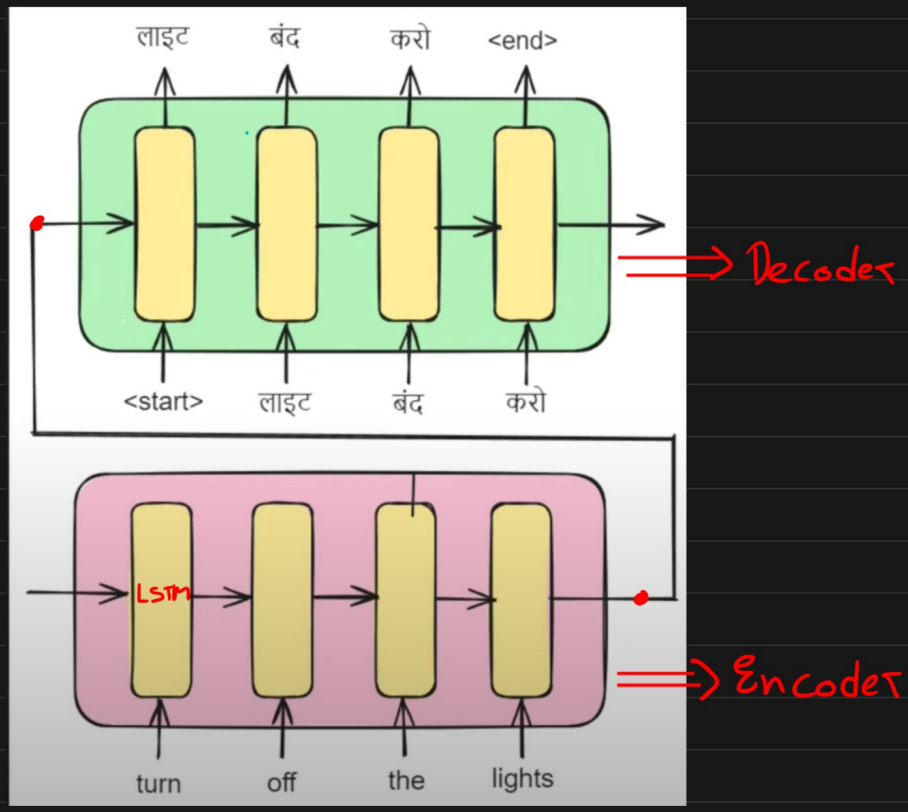
Attention Mechanism

Disadvantage of Encoder - Decoder

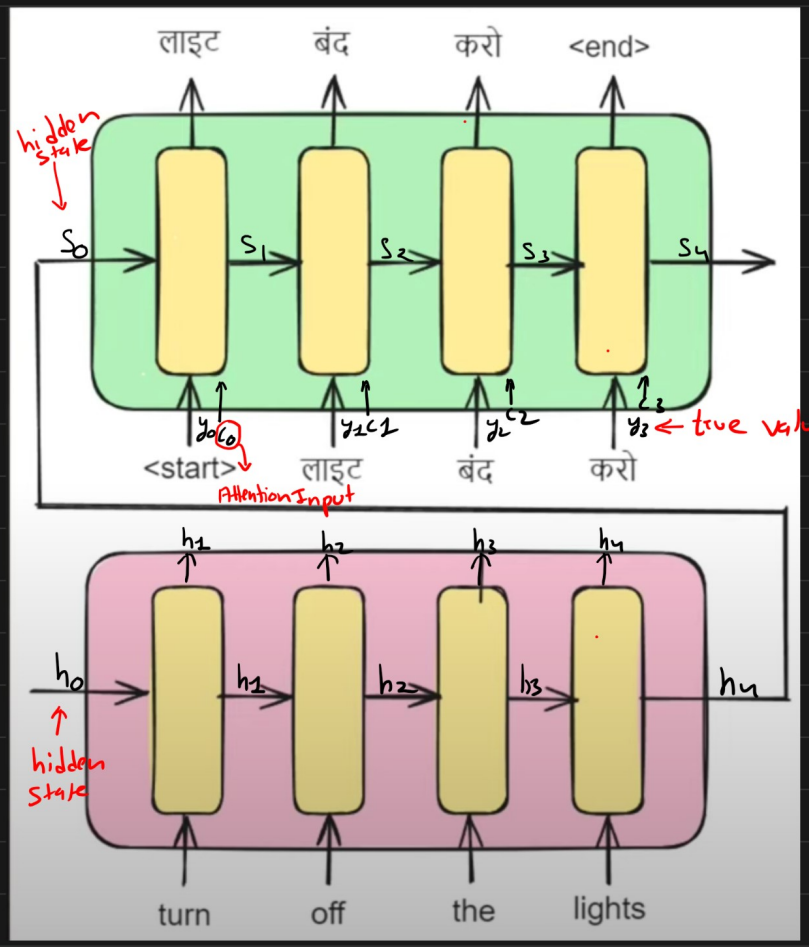


1> In Encoder - Decoder architechture the encoder process the whole sentence and the it passes content vector to the decoder i.e. unnecessary we are giving huge load to encoder the whole sentence that is only represented by context vector.

2> Unnecessary we are giving all the sentences to decoder at every timestep i.e. at every timestep decoder only required set of words for decoding not the whole sentence.

Attention Mechanism

⇒ Dynamically we need to generate the information that at every timestep of decoder 'which timestep or set of timestep of encoder' are important CURRENTLY for it.



Vanilla Encoder - Decoder	Attention Encoder - Decoder
1) There are only 2 inputs provided to decoder at every timestep.	1) There are 3 inputs provided to decoder at every timestep.

'i' timestep

a_{t_i} $[y_{i-1}, s_{i-1}]$

Previous timestep output — Previous hidden state

'i'th timestep

a_{t_i} $[y_{i-1}, s_{i-1}, c_i]$

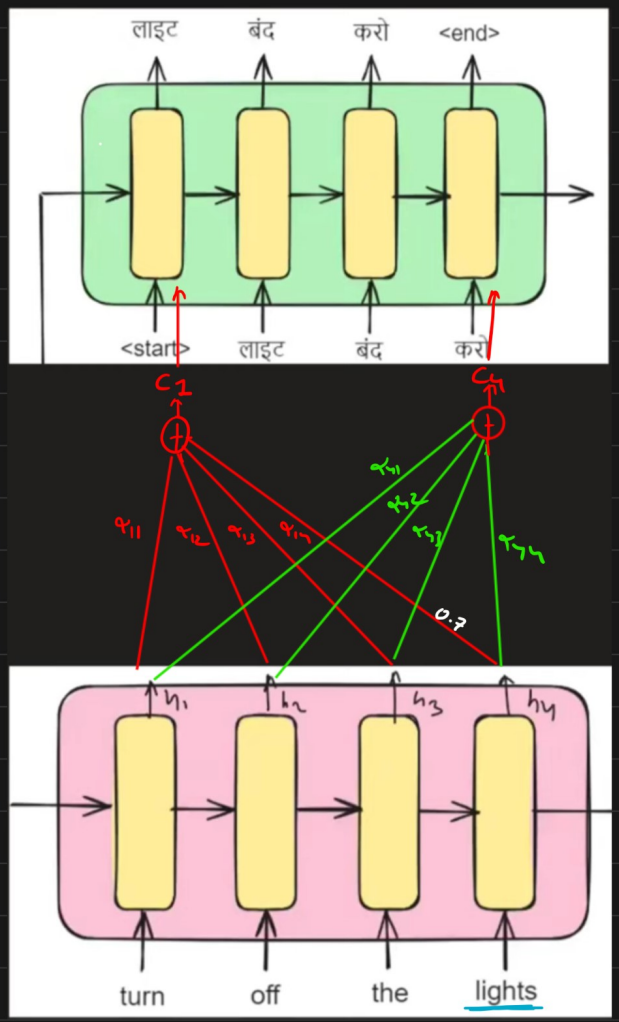
Previous timestep output — Previous hidden state — attention input

Attention Input [C_i]

C_i ⇒ Attention input at i^{th} timestep of decoder. i.e. weighted sum of the hidden states of encoder.

i^{th} timestep attention input <vector> → $C_i = \sum \alpha_{ij} \times h_j$ ← j^{th} timestep hidden input <vector>

α_{ij} ← weight <scalar>



Example

$$C_1 = \alpha_{11} h_1 + \alpha_{12} h_2 + \alpha_{13} h_3 + \alpha_{14} h_4$$

$$C_4 = \alpha_{41} h_1 + \alpha_{42} h_2 + \alpha_{43} h_3 + \alpha_{44} h_4$$

Total weights = $i \times j$ ⇒ $4 \times 4 = 16$ ← No. of weights

Q) What are weights 'α'?

Ans)

α_{ij} ⟶ Similarity score

↓

It tells for decoder i^{th} timestep output how much contribution does encoder j^{th} timestep has.

Similarity Score [α_{ij}]

'j'th timestep encoder hidden state

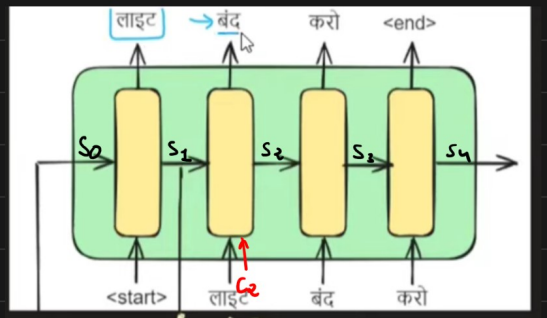
Previous hidden state of decoder

$$\alpha_{ij} = f(h_j, S_{i-1})$$

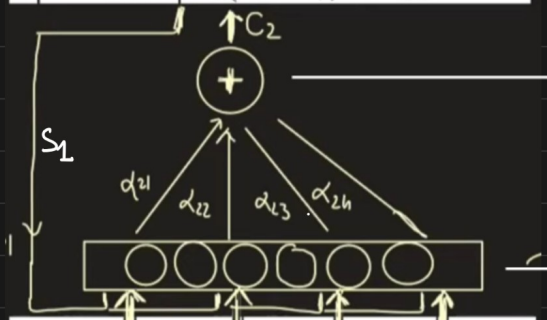
Similarity score

function

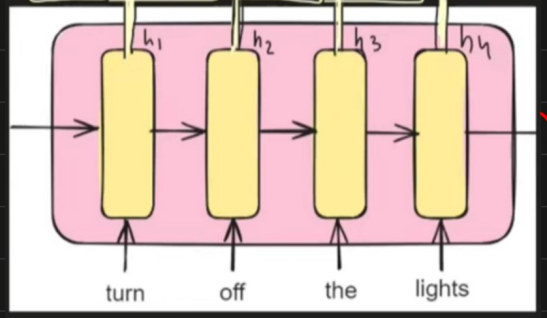
ANN → Universal function Aproximation



$$C_2 = \alpha_{21} h_1 + \alpha_{22} h_2 + \alpha_{23} h_3 + \alpha_{24} h_4$$



Time Distributed Dense Layer

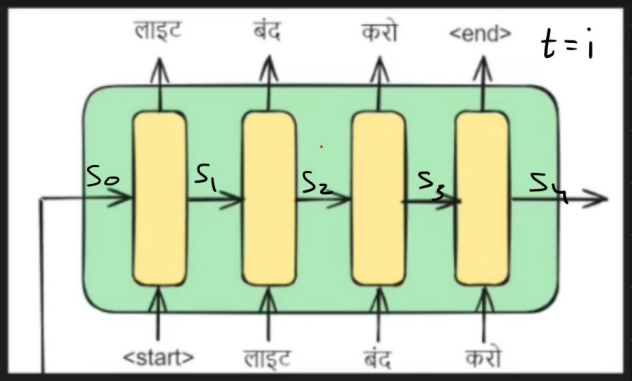


X No context vector

In Attention Mechanism we do not pass Context vector of last time step to decoder

How to find α

⇒ In order to find Similarity score of word to word we can use :



$$\alpha = f(h_j, S_{i-1}) \Rightarrow \text{math function}\left(\text{Encoder hidden state}, \text{Decoder Previous hidden state}\right)$$

Function Approximation

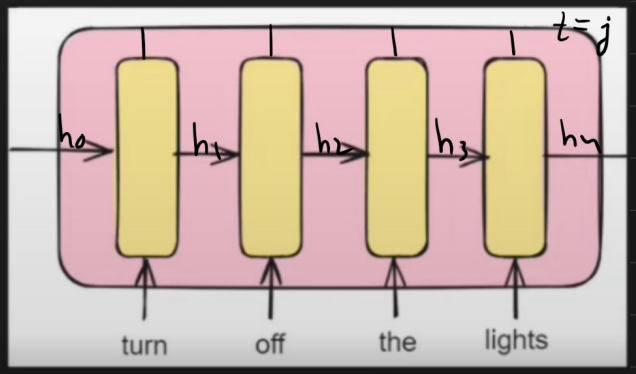
ANN

Bahdanau Attention

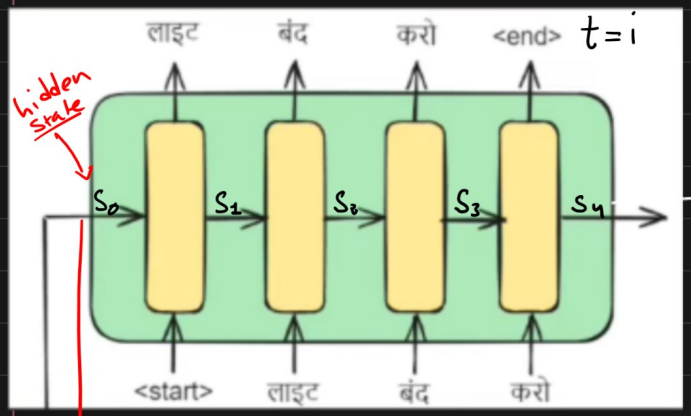
Luong Attention

$$c_{ij} = \sum_i \alpha_{ij} h_j$$

α_{ij} : similarity score



Bahdanau Attention



→ hidden state = $S_i = [e, f, g, h]_{1\times 4}$ (vector)

$$C_i = \sum \alpha_{ij}\, h_j \Rightarrow \alpha_{ij} \Rightarrow f(h_j, S_{i-1})$$

$[\alpha_{11}, \alpha_{12}, \alpha_{13}, \alpha_{14}] \rightarrow C_1$

Softmax

$[e_{11}, e_{12}, e_{13}, e_{14}]$

→ Output layer <1 units> — 4×1

3×1

→ hidden layer <3 units> — 4×3

8×3

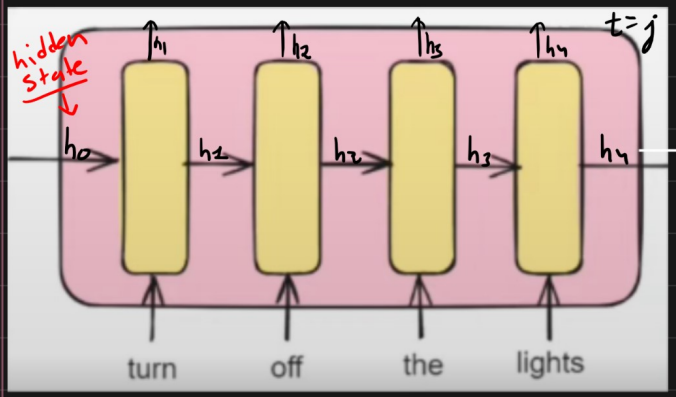
→ Input layer → batch → 4×8

$$\begin{bmatrix} S_0, h_1 \\ S_0, h_2 \\ S_0, h_3 \\ S_0, h_4 \end{bmatrix}$$

Concatenation

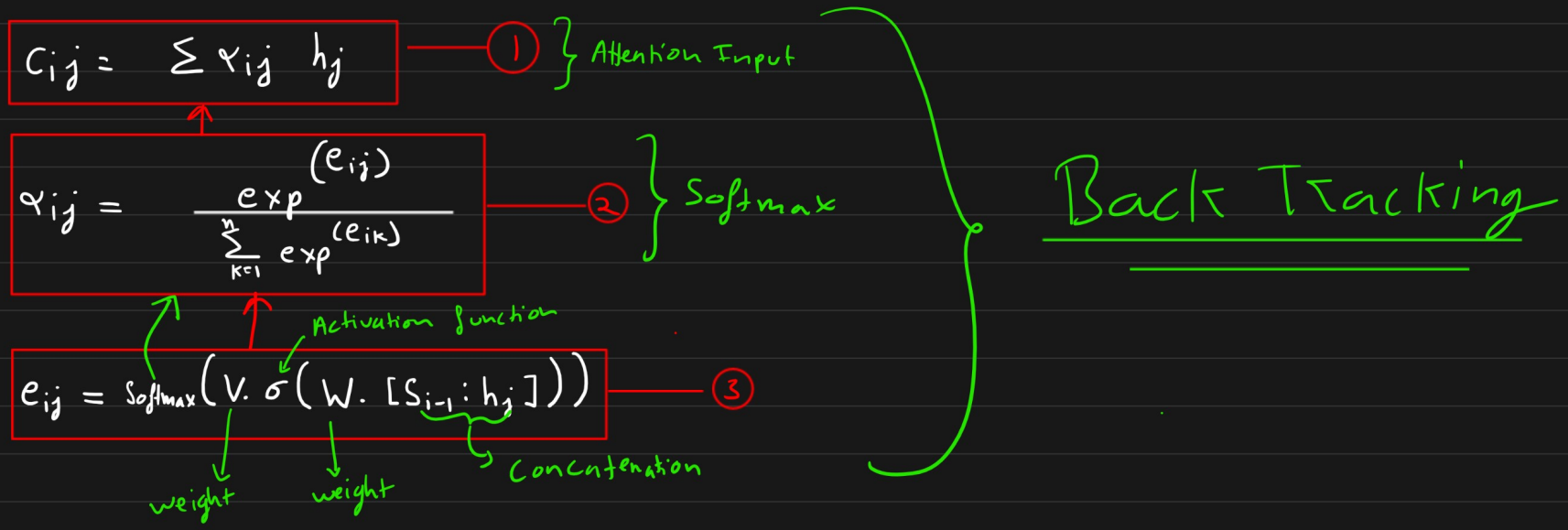
$$\begin{bmatrix} S_{0e}, S_{0f}, S_{0g}, S_{0h}, h_{1a}, h_{1b}, h_{1c}, h_{1d} \\ S_{0e}, S_{0f}, S_{0g}, S_{0h}, h_{2a}, h_{2b}, h_{2c}, h_{2d} \\ S_{0e}, S_{0f}, S_{0g}, S_{0h}, h_{3a}, h_{3b}, h_{3c}, h_{3d} \\ S_{0e}, S_{0f}, S_{0g}, S_{0h}, h_{4a}, h_{4b}, h_{4c}, h_{4d} \end{bmatrix}_{4\times 8}$$

decoder previous hidden state | encoder hidden state



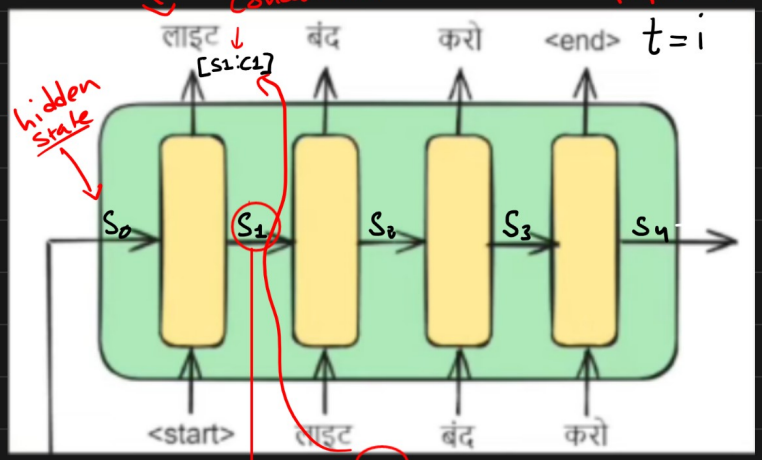
→ hidden state → $h_j = [a, b, c, d]_{1\times 4}$ (vector)

Bahdanau Attention Mathematics



Loung Attention

② difference



Mathematical function

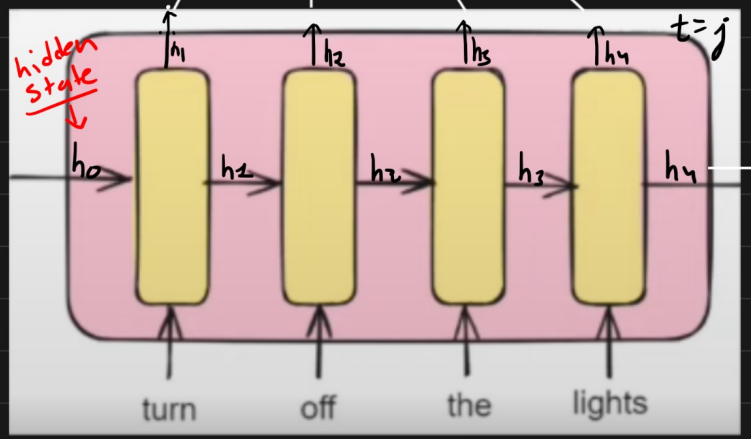
$$C_{ij} = \Sigma \alpha_{ij} h_i$$

$$\alpha_{ij} = f(S_{i-1}, h_j)$$

- S_{i-1} → Latest hidden state
- ① difference
- dot product (·) ⇒ $S_i^T \cdot h_j$

dot product $(S_i^T, h_j) \Rightarrow$ $S_i \Rightarrow [e,f,g,h]$, $h_j = [a,b,c,d]$

$$S_i^T \cdot h_j = \begin{bmatrix} e \\ f \\ g \\ h \end{bmatrix}_{1\times 4} \cdot [a,b,c,d]_{4\times 1} = \alpha_{ij}$$ → scalar



→ h_i

Advantage

i) It is fast as we use (·) → dot product for mathematical function.

ii) We use latest information (context vector) i.e. S_i instead of previous hidden state h_{i-1}.