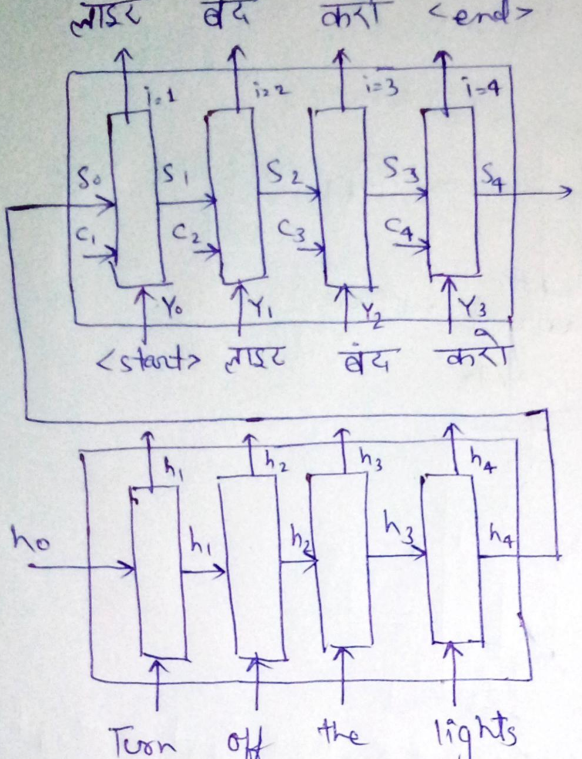
06 March 2025

ATTENTION MECHANISM



i = time step

C_i = attention input

in vanilla encoder decoder

input = $[Y_{i-1}, S_{i-1}]$

in vanilla encoder-decoder with attention mechanism

input = $[Y_{i-1}, S_{i-1}, C_i]$

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

$\alpha_i \rightarrow$ weight (scalar)

$h_j \rightarrow$ encoder's hidden state (vector)

Same for other attention input

(i) Bahdanau Attention

$$C_i = \sum \alpha_{ij} h_j$$

hence

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

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

$$C_3 = \alpha_{31} h_1 + \alpha_{32} h_2 + \alpha_{33} h_3 + \alpha_{34} h_4$$

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

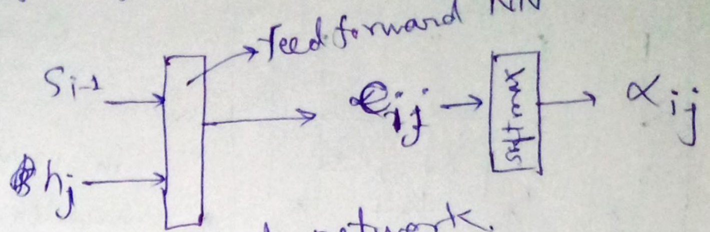
Now how to calculate α?

lets take an example α_{21}

$\alpha_{21} \rightarrow$ alignment/similarity score

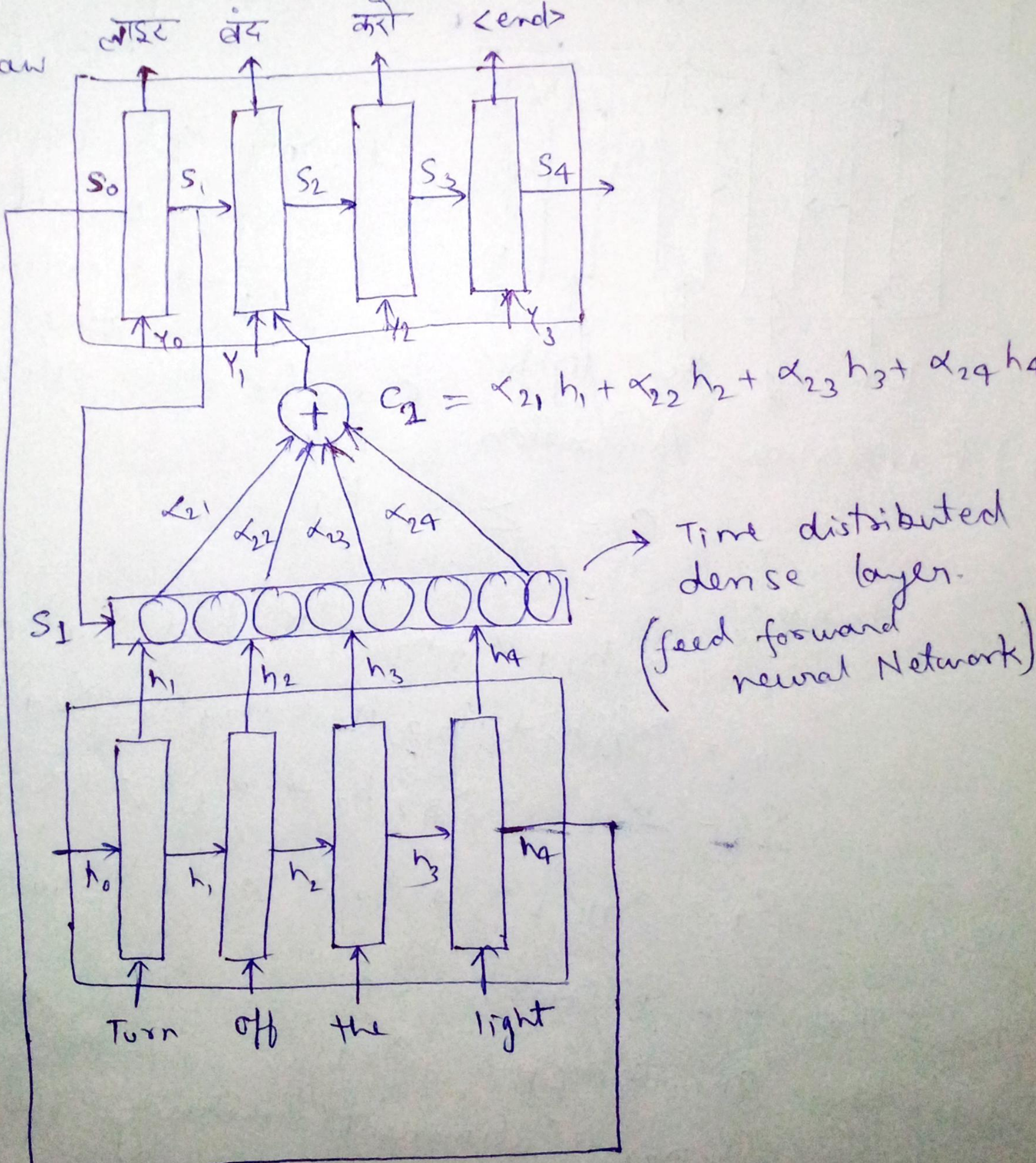
- α_{21} depends on h_1 and S_1 (previous hidden state of decoder)

- $\alpha_{21} \rightarrow f(h_1, S_1)$
  or $\alpha_{ij} = f(h_j, S_{i-1})$ in general



artificial neural network.

assume we are now at i = 2



- in original paper researcher use the bidirectional LSTM

07 Mar 2025

BAHDANAU ATTENTION Vs LUONG ATTENTION

Two attention mechanism
- Bahdanau Attention
- Luong Attention.

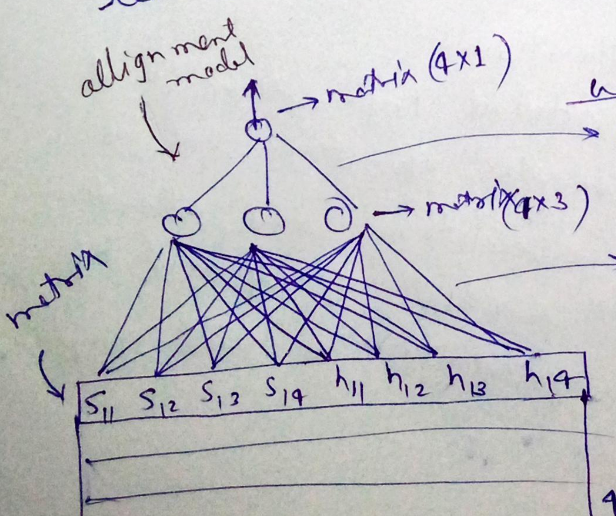
$s_1 = [e\ f\ g\ h] =$ four dimension vector

now concatenat s_1 with h_1, h_2, h_3 and h_4
means make a matrix (4 rows / 8 columns)

$$[s_{i-1}, h_j] = \begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} & h_{11} & h_{12} & h_{13} & h_{14} \\ s_{11} & s_{12} & s_{13} & s_{14} & h_{21} & h_{22} & h_{23} & h_{24} \\ s_{11} & s_{12} & s_{13} & s_{14} & h_{31} & h_{32} & h_{33} & h_{34} \\ s_{11} & s_{12} & s_{13} & s_{14} & h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix}$$

now put this matrix in feed forward neural network using a batch operation

let us assume our feed forward NN architecture:



$e_{ij} = [e_{21}\ e_{22}\ e_{23}\ e_{24}]$

$\alpha = \text{softmax}(e)$

$\alpha_{ij} = [\alpha_{21}\ \alpha_{22}\ \alpha_{23}\ \alpha_{24}]$

Now

$[s_1, y_1, c_2^i] \rightarrow \boxed{\text{LSTM}} \rightarrow$ बंद, s_2

at time step 2 $(i=2)$

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

- Now at ~~three~~ first iteration all the weights value are same they are update in next iteration.
- weights value are update with the help of backpropogation. (further iteration)
- weights are update till convergence to minimize the error (prediction of word)
- the Bahdanau attention is also called the additive attention.

(ii) Luong Attention

Here $\alpha_{ij} = f(S_i \cdot h_j)$

lets $S_i = [a\ b\ c\ d]$

$h_j = [e\ f\ g\ h]$

dot product of S_i and h_j

$e_{ij} = ae + bf + cg + dh$ = attention value

$\alpha_{ij} =$ softmax (e_{ij})

ans why Luong attention use the current state for calculating the attention

- because we got a updated information and we use the less complex function use in luong attention to calculation the attention value that is dot product of S_i and h_j
- and Luong attention architecture is less complex.

$$e = [s_1 h_1 \quad s_1 h_2 \quad s_1 h_3 \quad s_1 h_4]$$

$$e = [e_{11} \quad e_{12} \quad e_{13} \quad e_{14}]$$

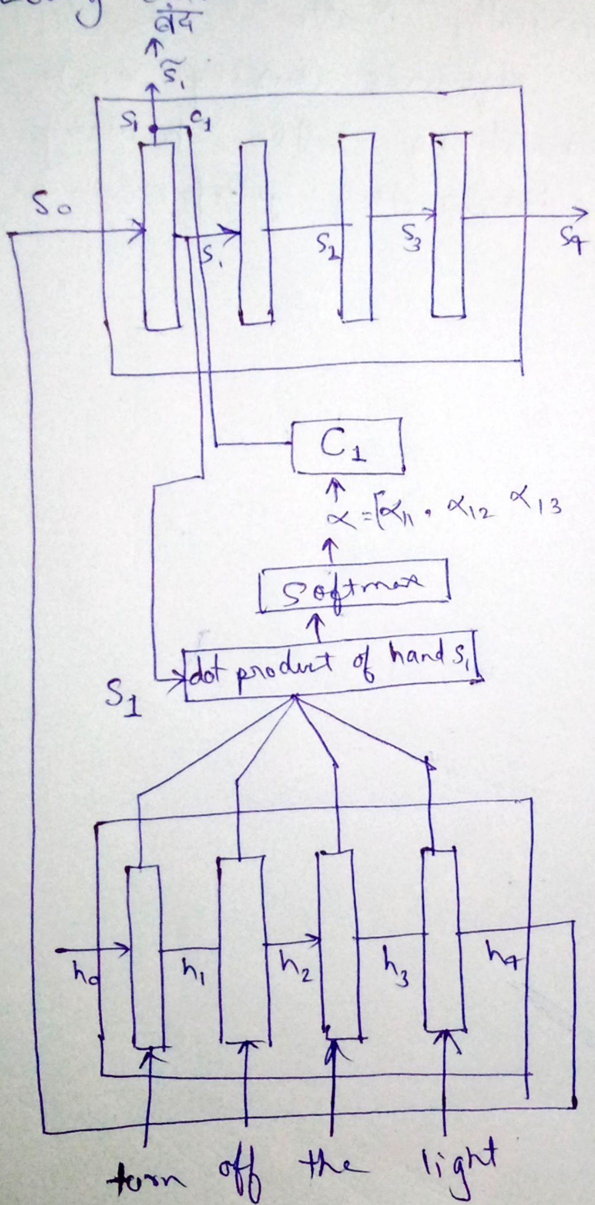
$$\alpha_{ij} = \text{softmax}(e)$$

$$\alpha_{ij} = [\alpha_{11} \quad \alpha_{12} \quad \alpha_{13} \quad \alpha_{14}]$$

Luong attention architecture:

here we assume that we are at time step 1

$i = 1$



- this attention is also called the multiplicative attention why we required the Bahdanau and Luong attention?

- Seq2Seq model with an ecoder - decoder architecture traditionally suffer from the bottle-neck of compressing all input information into a single fixed-length context vector.
- Attention mechanism mitigate this by letting the decoder dynamically "attend" to different part of the input sequence during each decoding step, enabling better handing of long sequences and improving performance.