Encoder – Decoder

#1) Seq2Seq Data

Input Sequence ⟶ Output Sequence

Application → Machine Translation

Example

Nice to meet you (Input Sequence) —Machine Translation→ आप से मिल कर अच्छा लगा (output Sequence)

Q. 1) What are the problems faced in Seq2Seq data ?

Ans) 1) Variable Length Input

↳ The input can be of any length & it can vary sample to sample

I/p ⟶ Sentence ⟶ English ⟶ variable Length

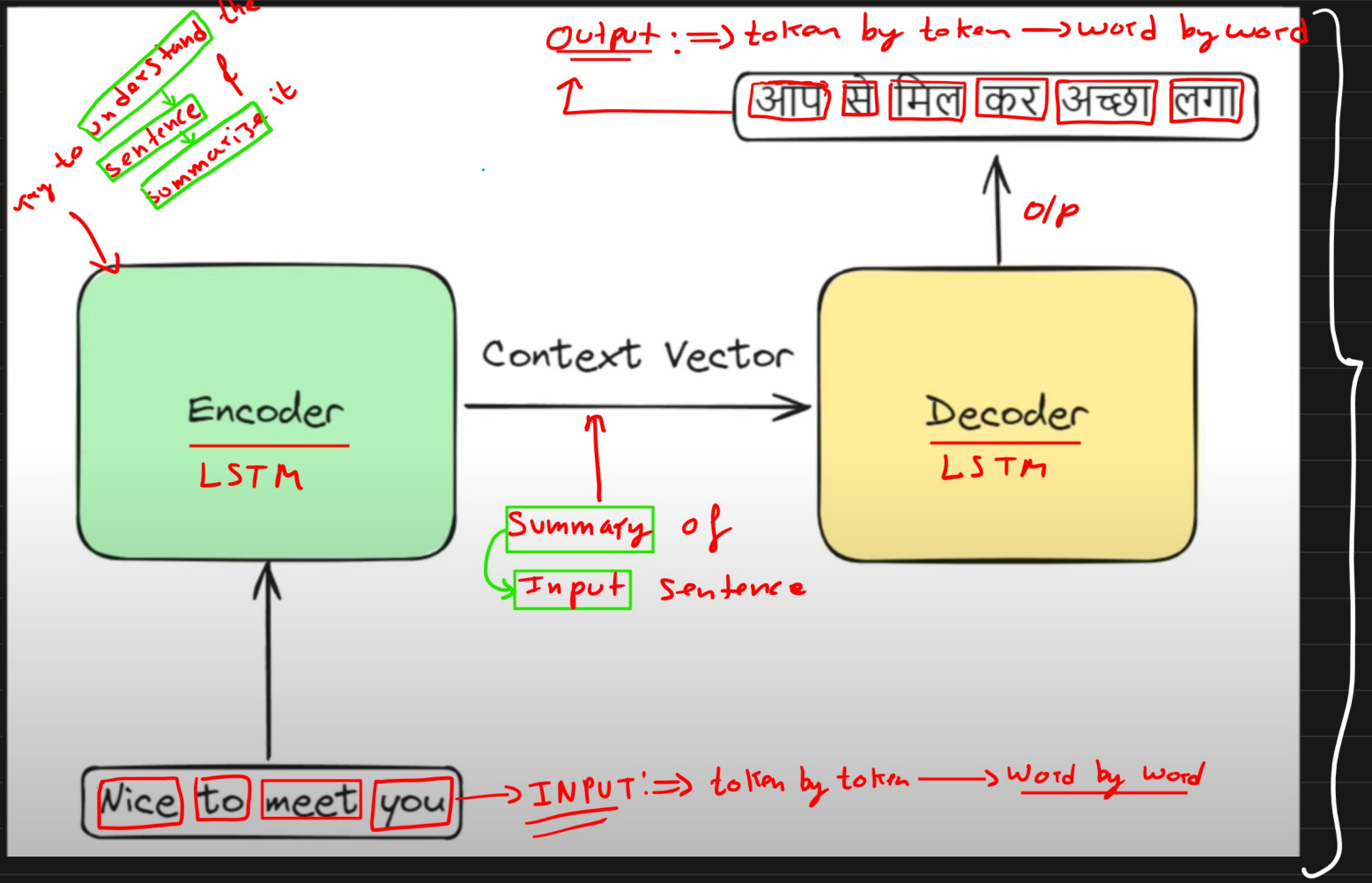
2) Variable Length Output

↳ The output can be of any length & it can vary sample to sample

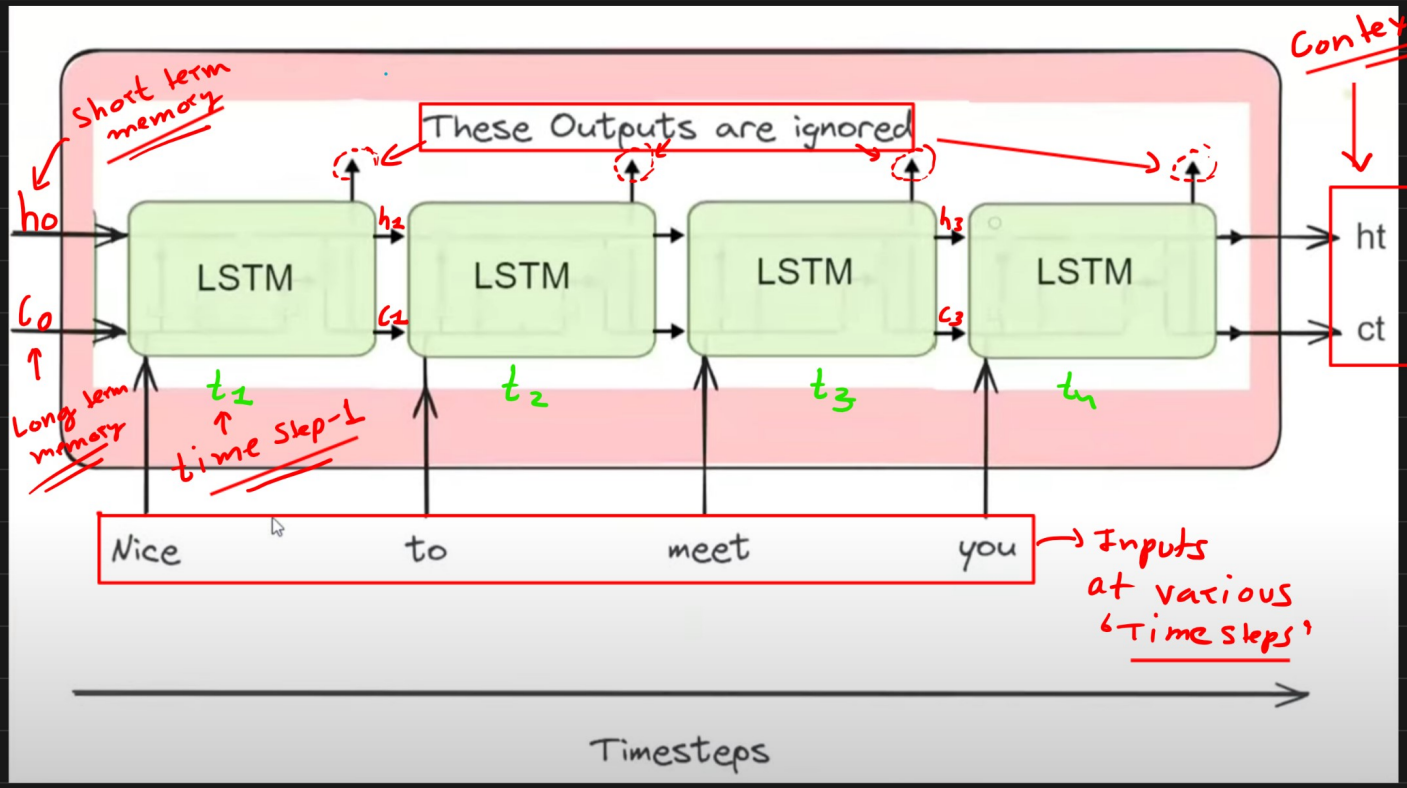
3) No Guarntee that input & output for a sample will be of same length

↳ Nice to meet you (Input Length = 4) —Machine Translation→ आप से मिल कर अच्छा लगा (output Length = 6)

#2> High Level Overview : Encoder Decoder



#2.1> Encoder Architecture



Final Representation of Sentence

Context vector

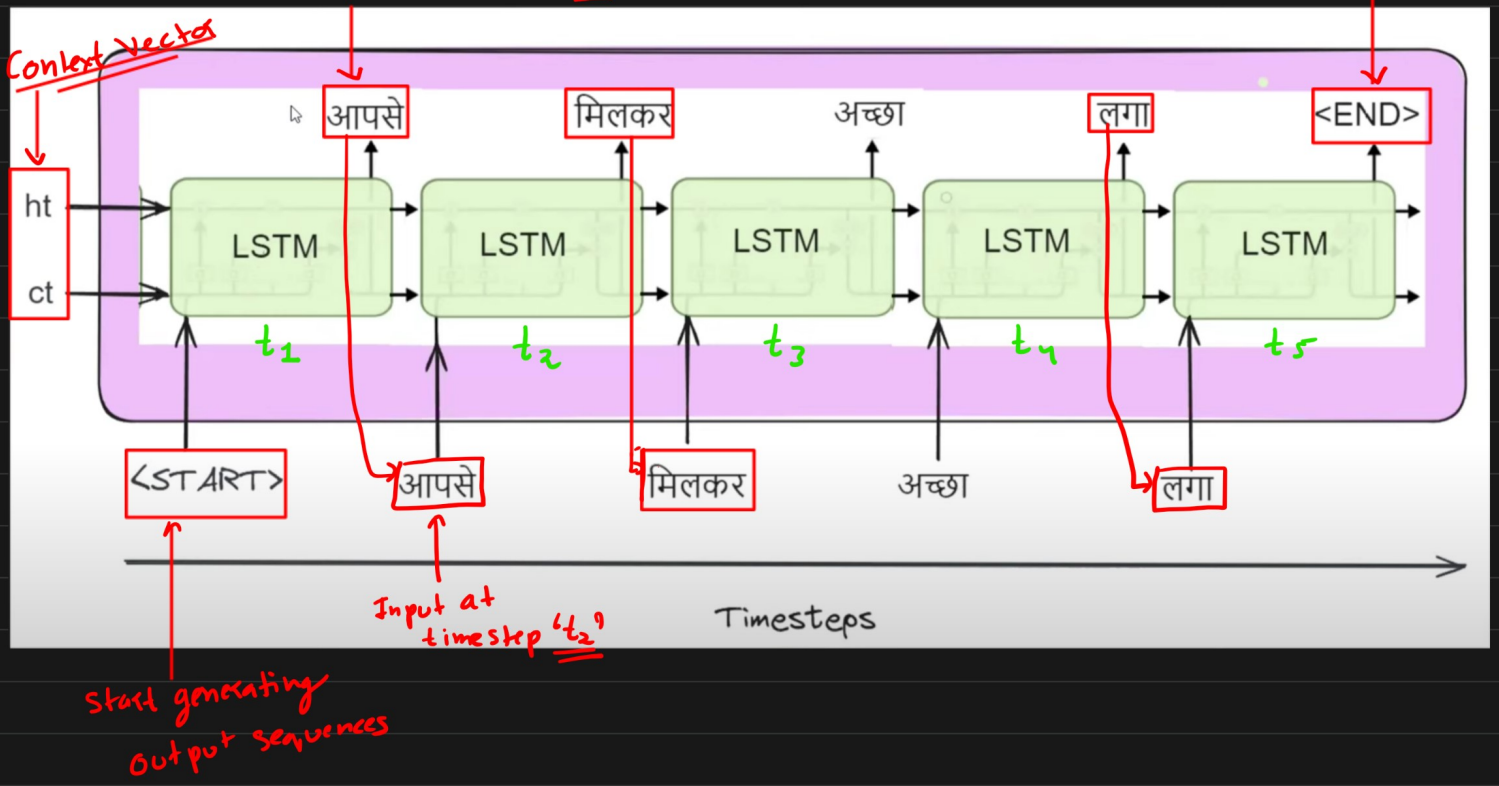
=

Single LSTM

LSTM

Unfold over time

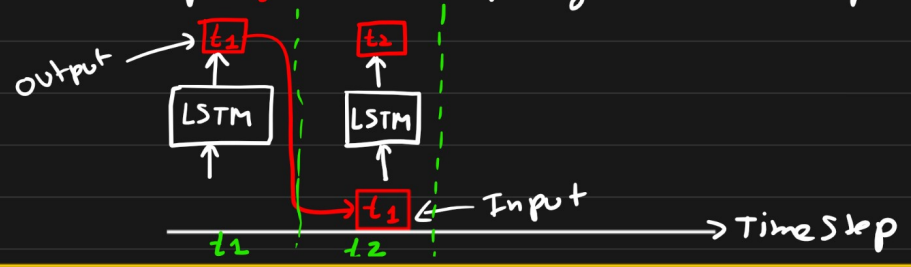
#2.2> Decoder Architecture



NOTES :i> <START> & <END> :=> Represent starting and ending of output sentence.

ii> The Context vector i.e. [the final states generated by encoder] is pass as the initial states to decoder.

iii> The input at timestep t_2 is the output generated at previous timestep t_1



3> Dataset Preprocessing

Task → Machine Translation

Dataset

English	Hindi
Think about it.	सोच लो
Come in	अंदर आ जाओ

Step-1) Tokenize the sentence → Building block of text

Word

Dataset

English	Hindi
[Think, about, it]	[सोच, लो]
[Come, in]	[अंदर, आ, जाओ]

Word → token

Body of Language

Vocublary

Unique tokens

English	Hindi
Think	सोच
about	लो
it	अंदर
Come	आ
in	जाओ

Step-2> Encoding → Conversion of string data to numbers [machines only understand numbers]

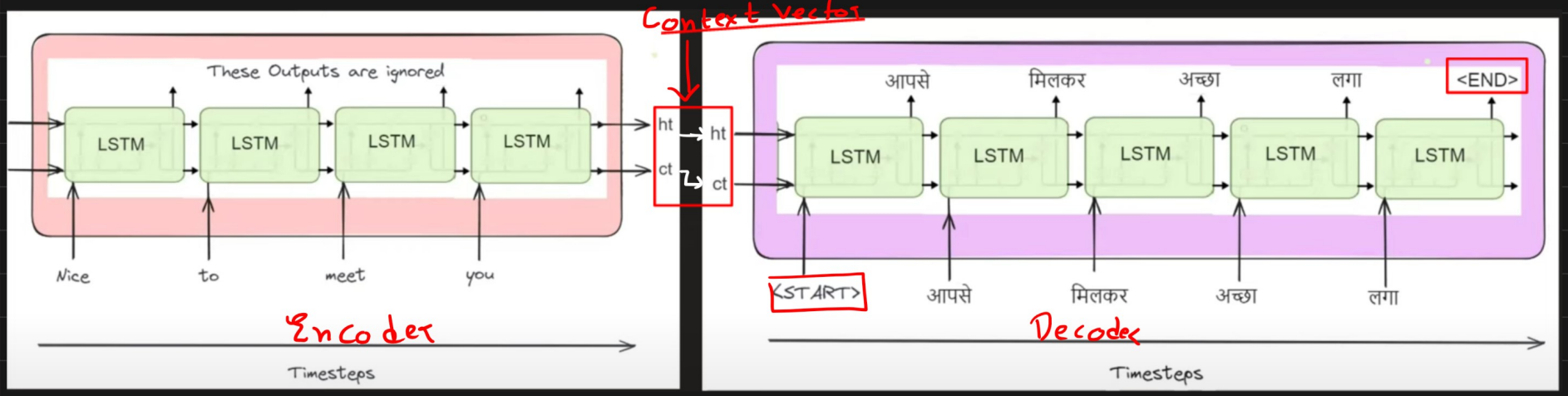
Tokens (word) → Convert → Numbers } Encoding

English	One-hot Encoding
① Think	[1,0,0,0,0]
③ about	[0,1,0,0,0]
② it	[0,0,1,0,0]
④ come	[0,0,0,1,0]
⑤ in	[0,0,0,0,1]
	[①②③④⑤]

Hindi	One-hot Encoding
① <START>	[1,0,0,0,0,0,0]
② सोच	[0,1,0,0,0,0,0]
③ ले	[0,0,1,0,0,0,0]
④ अंदर	[0,0,0,1,0,0,0]
⑤ आ	[0,0,0,0,1,0,0]
⑥ जाओ	[0,0,0,0,0,1,0]
⑦ <END>	[0,0,0,0,0,0,1]
	[①,②,③,④,⑤,⑥,⑦]

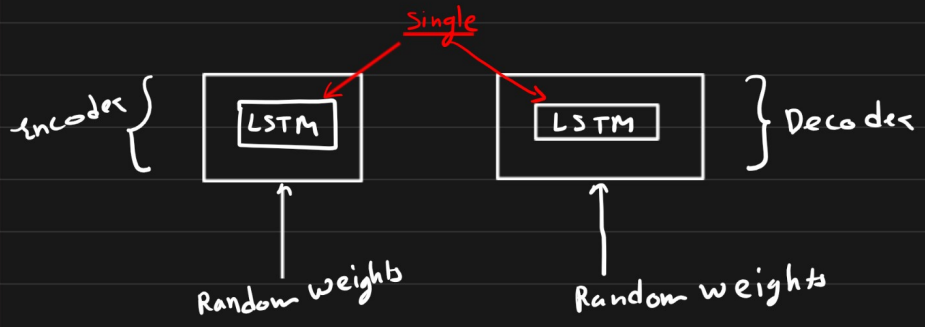
Model Training

→ The training of Encoder & decoder is done Simultaneously.



Training Process

1> At initial stage the learning parameters (weights + Bias) of encoder & decoder are initialized randomly.

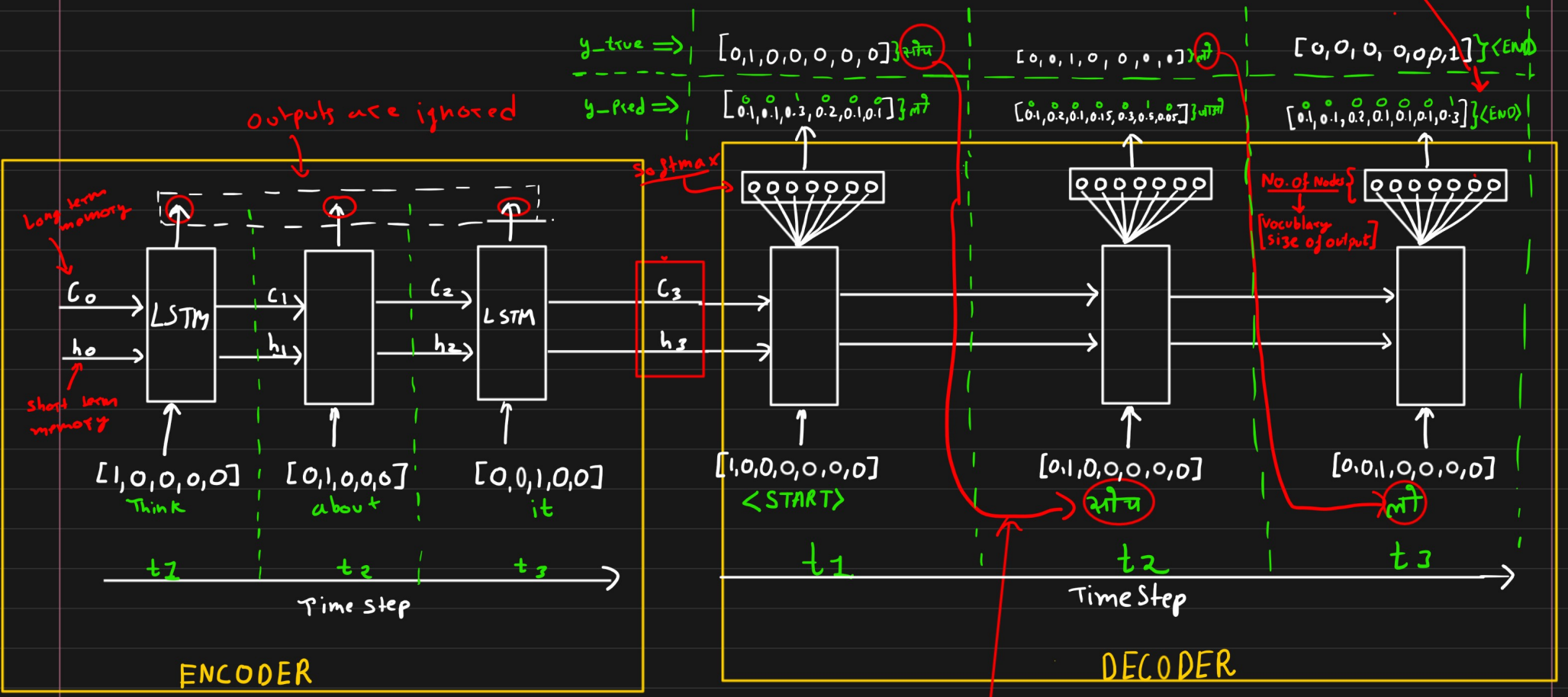


i> Forward Propogation

Sample-1) [Think about it] ⟶ [सोच मी <END>]

features — Target

stop the output prediction if 'y-pred' ⟶ <END>



Teacher Forcing ⇒ { Fast Training (how ⟶ Fast Convergence)

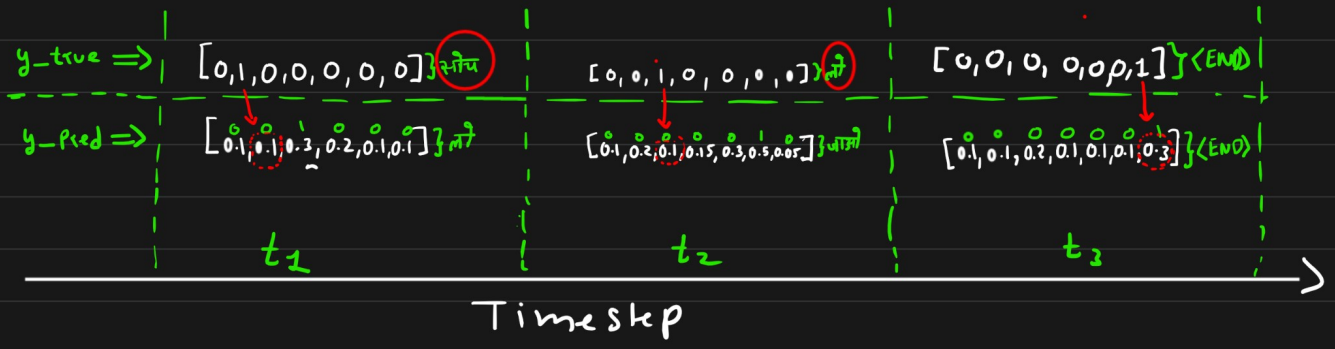
↳ It means we will pass the correct output 'y_true' at timestep t_n as input to next timestep t_{n+1}, irrespective Prediction output 'y_pred' at timestep t_n

ii) Calculate Loss

Loss function :=> Categorical Crossentropy $= -\sum_{i=1}^{c} y_i \log(\hat{y}_i)$

(c: no. of classes; $\hat{y}_i$: Predicted; y_i: true)

'Since at each timestep we need to output one of the category [vocublary of output]



Total Loss → $L = \sum_{i=1} L_{ti}$ ← Loss at each time step

$$L_{t=1} = -1 \times \log(0.1) = -1 \times (-1) = 1$$

$$L_{t=2} = -1 \times \log(0.1) = -1 \times (-1) = 1$$

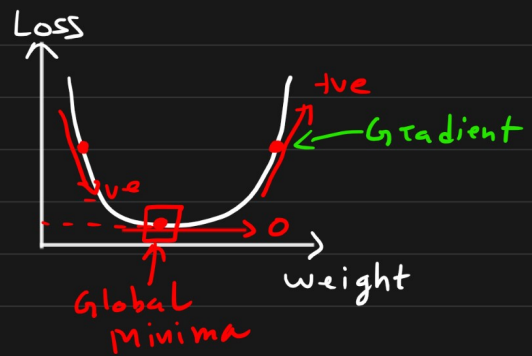
$$L_{t=3} = -1 \times \log(0.3) = -1 \times (-0.52) = 0.52 \leftarrow \text{Minimum Loss}$$

$$L_{total} = L_{t_1} + L_{t_2} + L_{t_3} = 1 + 1 + 0.52 = \boxed{2.52} \leftarrow \text{total Loss}$$

iii) Back Propogation

1) Gradient Calculation :=> { Represent : How much each learning parameter (weight + bias) contribute to loss function }

=> Provide the direction of gradient i.e. [-ve, 0, +ve]



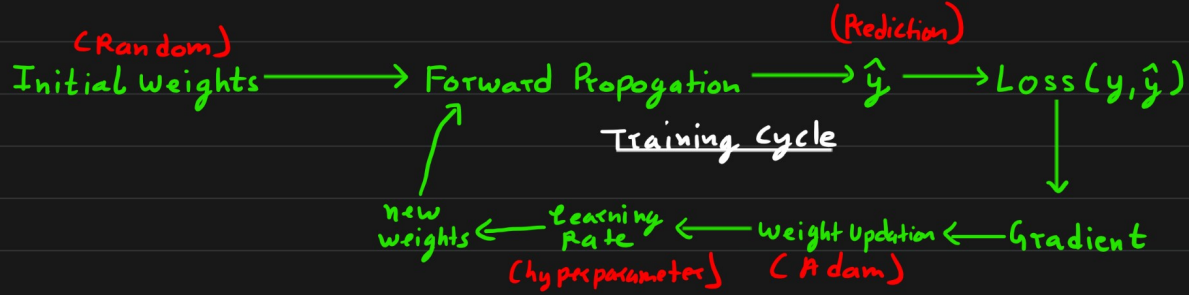
2) Weight Updation

for k iteration :

$$W_{1new} = W_{1old} - \eta \frac{\delta L}{\delta W_{1old}}$$

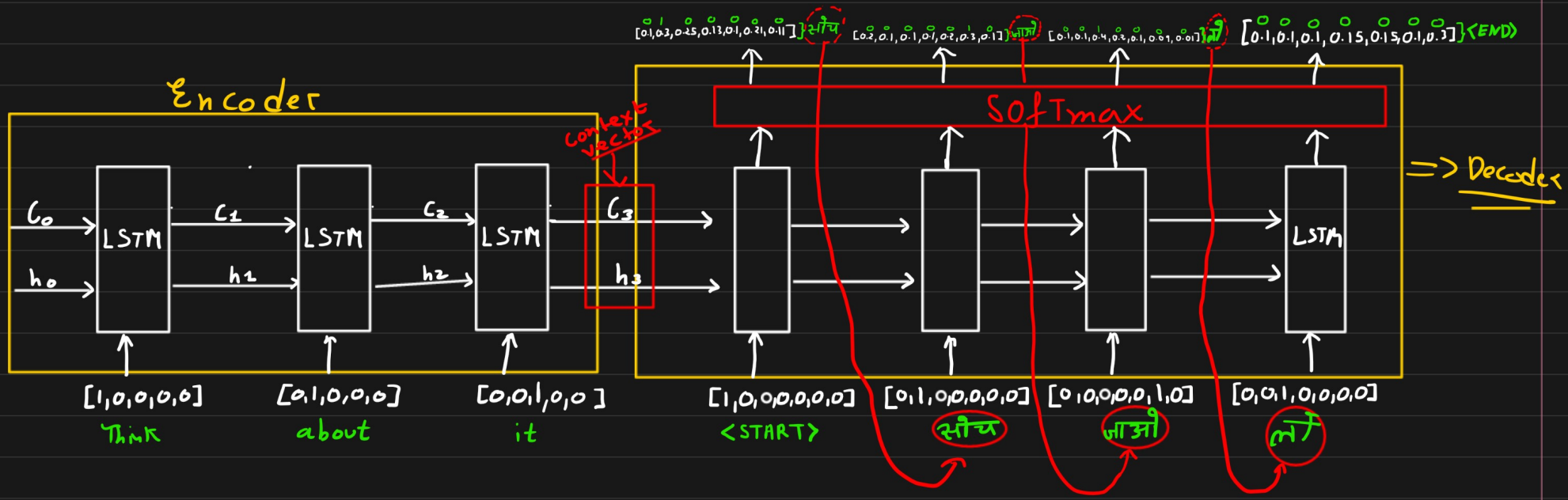
Partial derivative of 'Loss' w.r.t W_1

η : Learning Rate



iv) Prediction

I/p: Think about it



Input	Machine Translation	Prediction
Think about it		सोच जाओ तो

y-Pred

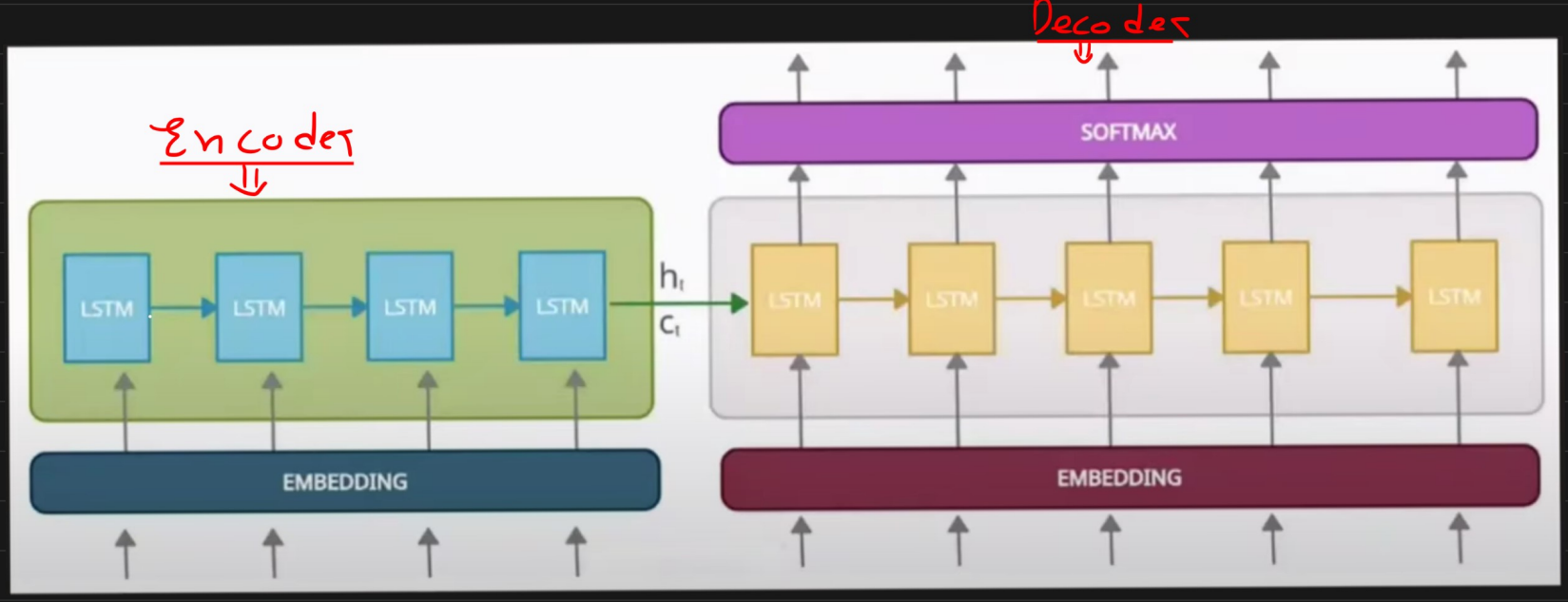
Improvement → Embedding

Embedding
- Low Dimensional dense Representation.
- Similar words have similar vectors

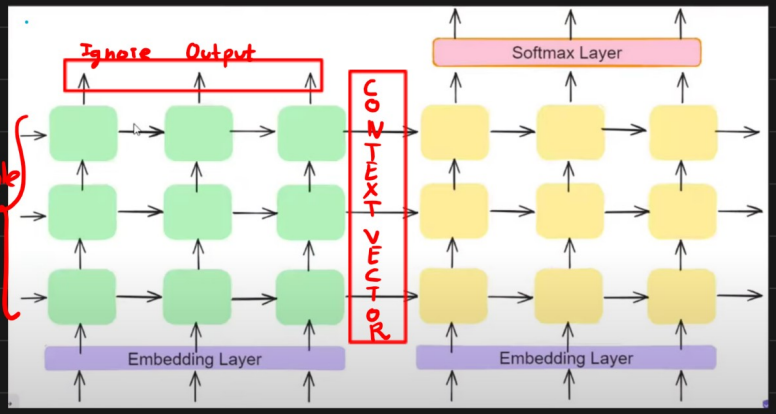
#Example :

Consider you have English vocublary size is 13K.

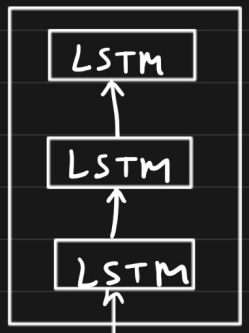
	One-hot Encoding	3-dimensional Embedding
think	[1,0,0,....,0]	[2.3, 0.4, 1.8]
about	[0,1,0,....,0]	[2.1, 3.6, 1.2]
it	[0,0,1,....,0]	[0.3, 0.9, 1.1]
	1x13K (Sparse vector, shape)	1x3 (Dense vector, shape)



2) Improvement → Deep LSTM



Captures Long term dependency



Encoder

Unfold over time