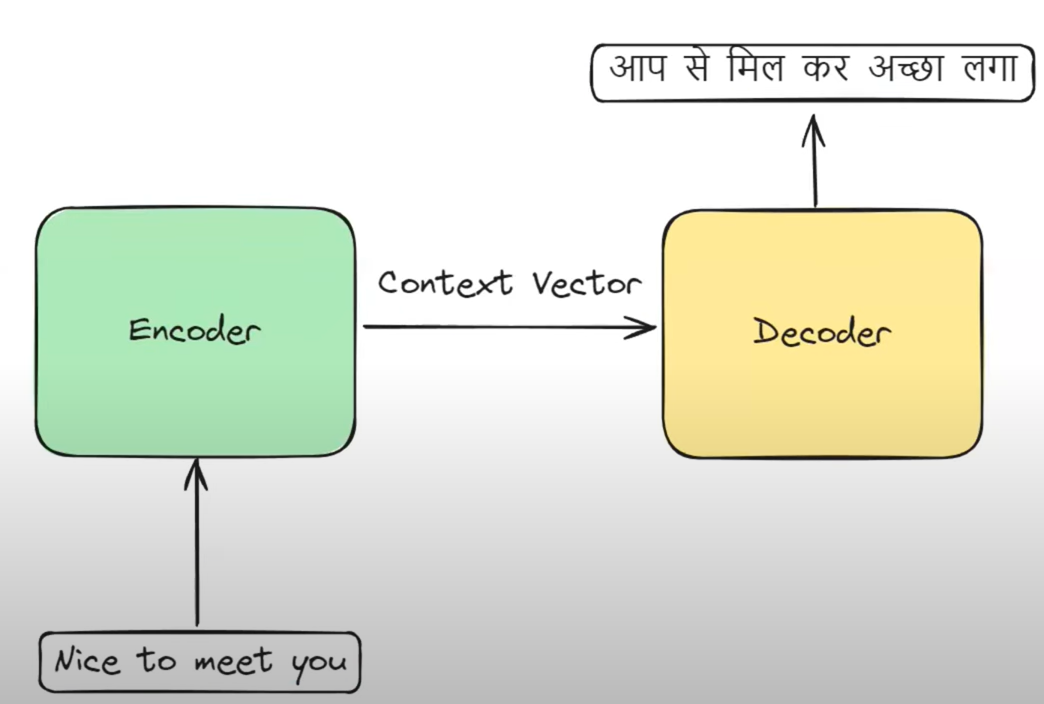
## horizontal line

Encoder Decoder

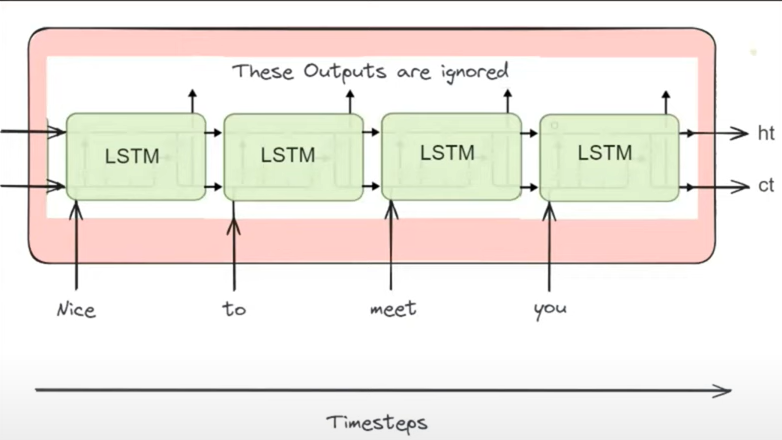
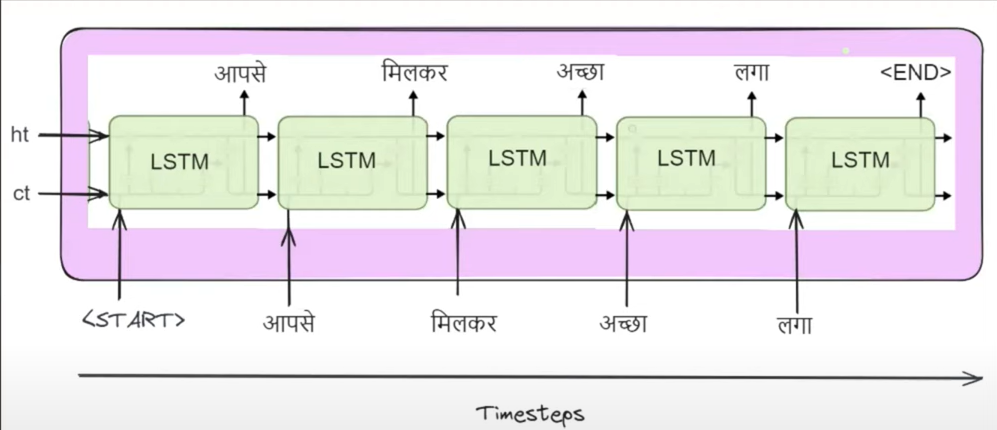
04.06.2025

# Encoder Decoder

Consists of an Encoder block that takes input token (word) by token and updated Ht , Cts after each time stamp.

Context Vector takes a summary of all the inputs and the final Ht , Ct.

Decoder block generates output token by token.

Encoder Decoder

Final Ht , ct from encoder act as input for decoder.

Encoder , Decoder made up of LSTMs (or GRUs).

# 

# Training the Architecture

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

E.g. First we need to convert it into vectors .

| <start> | [1,0,0,0,0,0,0] |
| --- | --- |
| सोच | [0,1,0,0,0,0,0] |
| लो | [0,0,1,0,0,0,0] |
| अंदर | [0,0,0,1,0,0,0] |
| आ | [0,0,0,0,1,0,0] |
| जाओ | [0,0,0,0,0,1,0] |
| <end> | [0,0,0,0,0,0,1] |

| Think | [1 , 0 , 0, 0, 0] |
| --- | --- |
| About | [0, 1 , 0 , 0, 0] |
| It | [0, 0 , 1 , 0, 0] |
| Come | [0, 0 , 0, 1 ,0] |
| In | [0, 0, 0 , 0, 1] |

Now let’s take a row “Think about it” —--> सोच लो 

Now see word by word input sent and summarized .

At decoder the word <start> is sent that initializes the O/p generation.

Based on the final ht , Ct the input sent to the softmax layer that tells probability of each O/p. The word with max probability will be the O/p .

Y\_true = [0,1,0,0,0,0,0]

Y\_pred = [0.1,0.5,0.2,0.3,0.3,0.2,0.1] This may be possible that pred is wrong since we start with random weights. But during training it starts giving true labels . Although the y\_pred is gone as input for next LSTM but for now during training we give true labels to the decoder to predict O/p correctly.

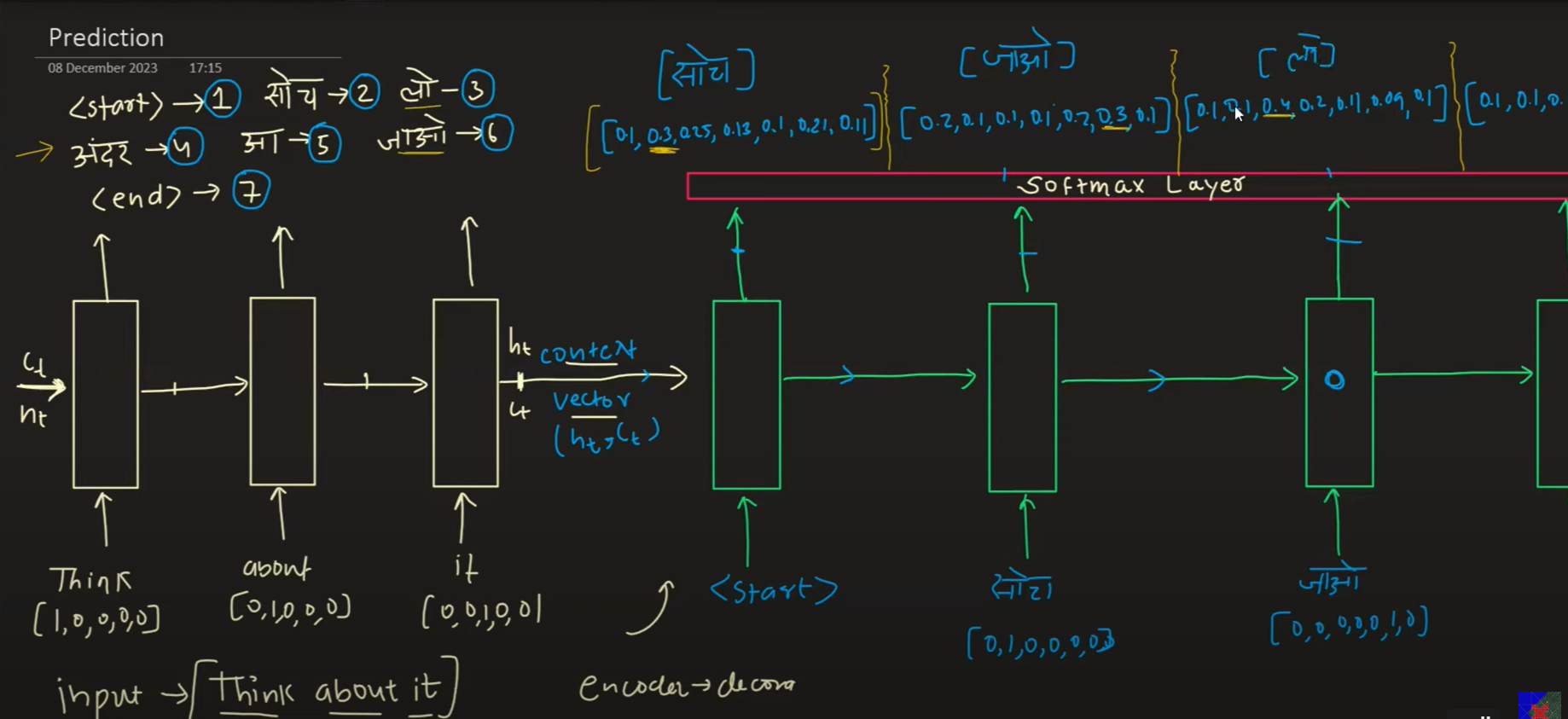
Similarly we get probabilities for all.

Now we calculate loss for all to calculate gradient and update weights and other parameters. Then the updated parameters go for the next sentence.

Since it is a multi class classification problem , we use categorical cross entropy.

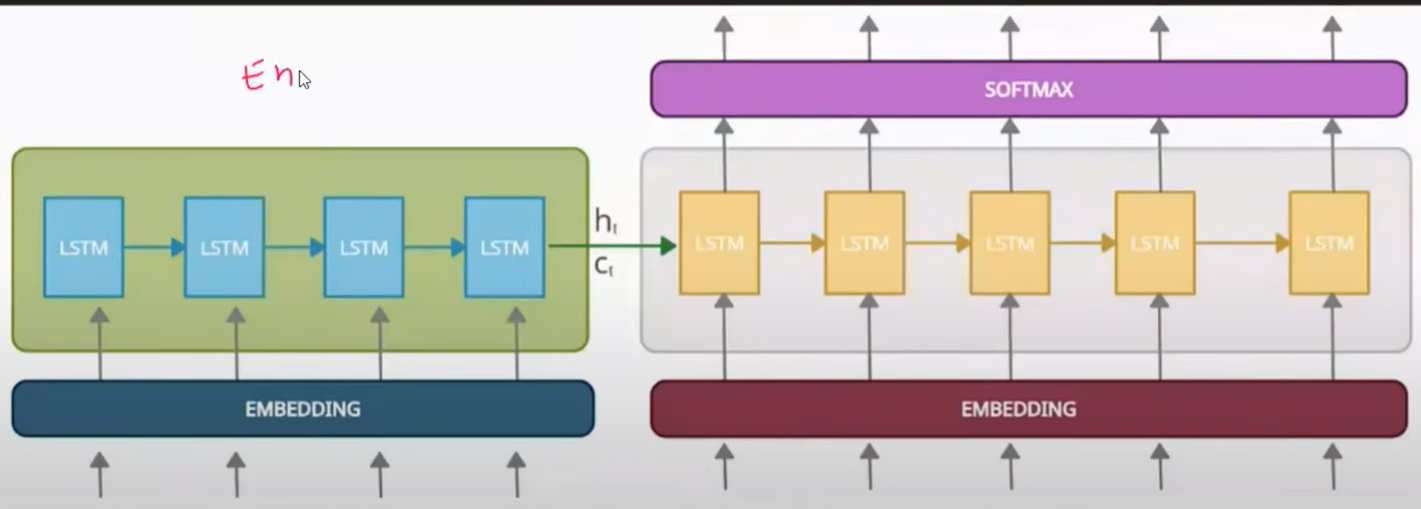
# Prediction

<start> - 1 , सोच - 2, लो - 3 , अंदर - 4 , आ - 5 , जाओ - 6 , <end> - 7. We have trained our data on this dataset. Now we make predictions . let the test data be the same as “Think about it”.

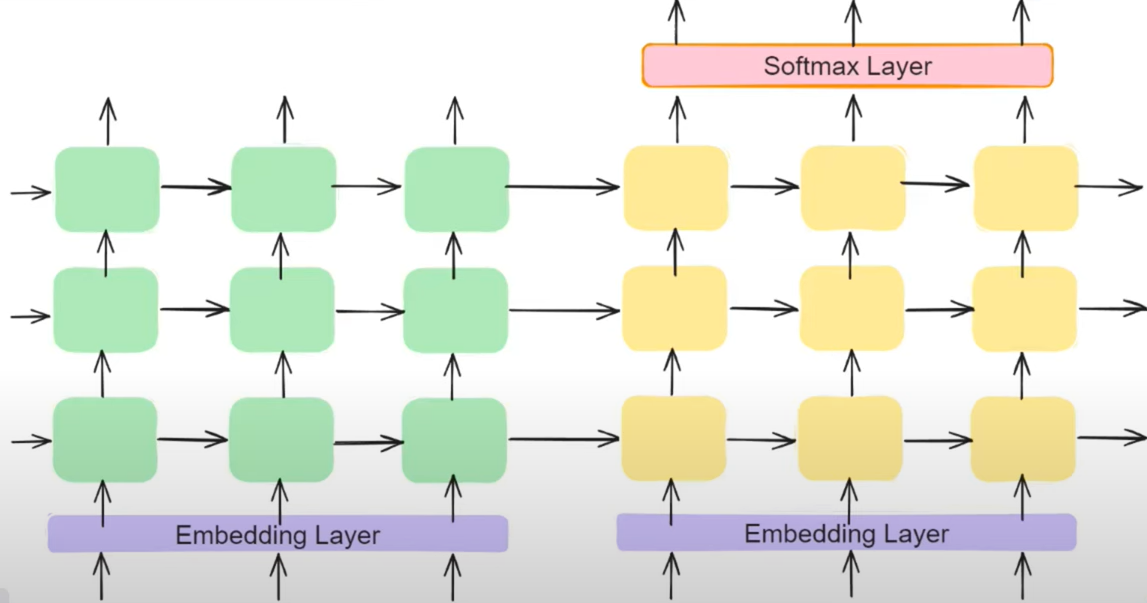


# Improvements Suggested

1. Embeddings

We use “Think” - [1,0,0,0,0] , but we have thousands of sentences in our data. Then we have a large vocab and the OHE of each word may have a very long dimension containing a lot of zeroes. 

* Introducing an embedding layer gives low dimensionality , dense vectors. Easier to process.
* Either we can use pre trained or trained embeddings.

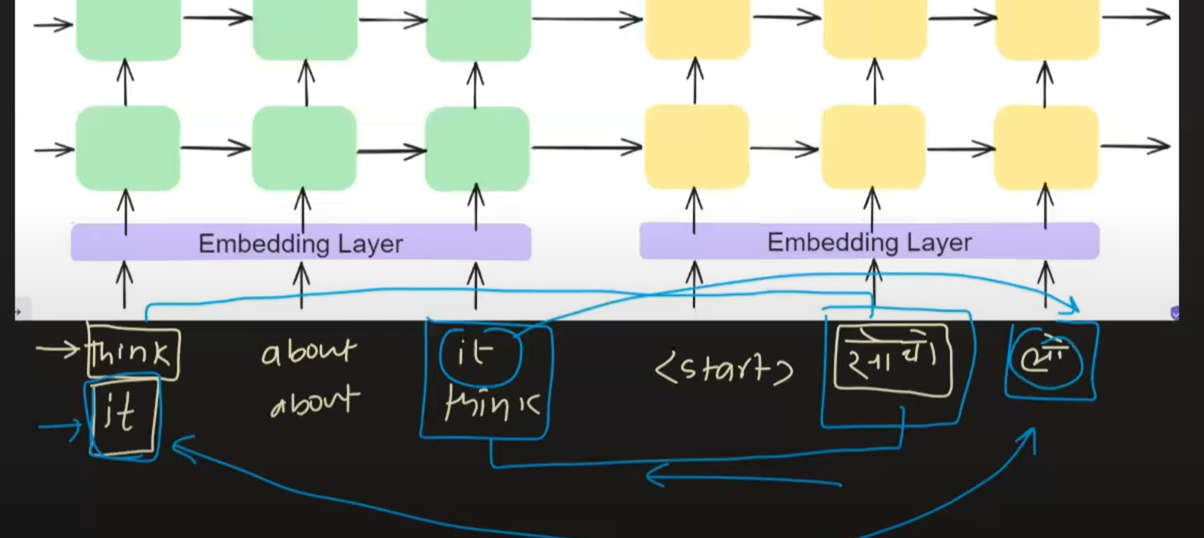
1. Deep RNNs

It provides 3 types of benefits.

Long term dependency

Learn hierarchical representation of data

More parameters , more learning.

1. Reversed inputs 

Used when initial words contain more context.