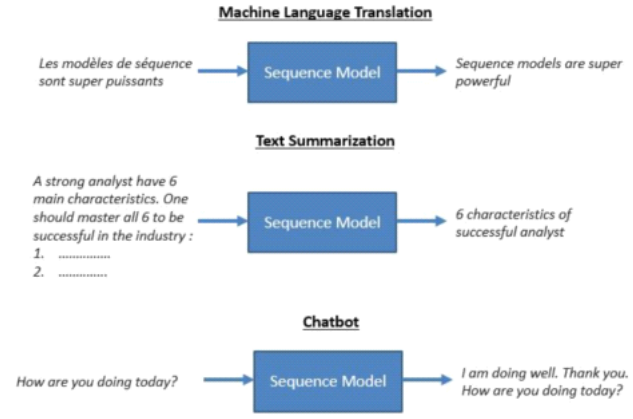
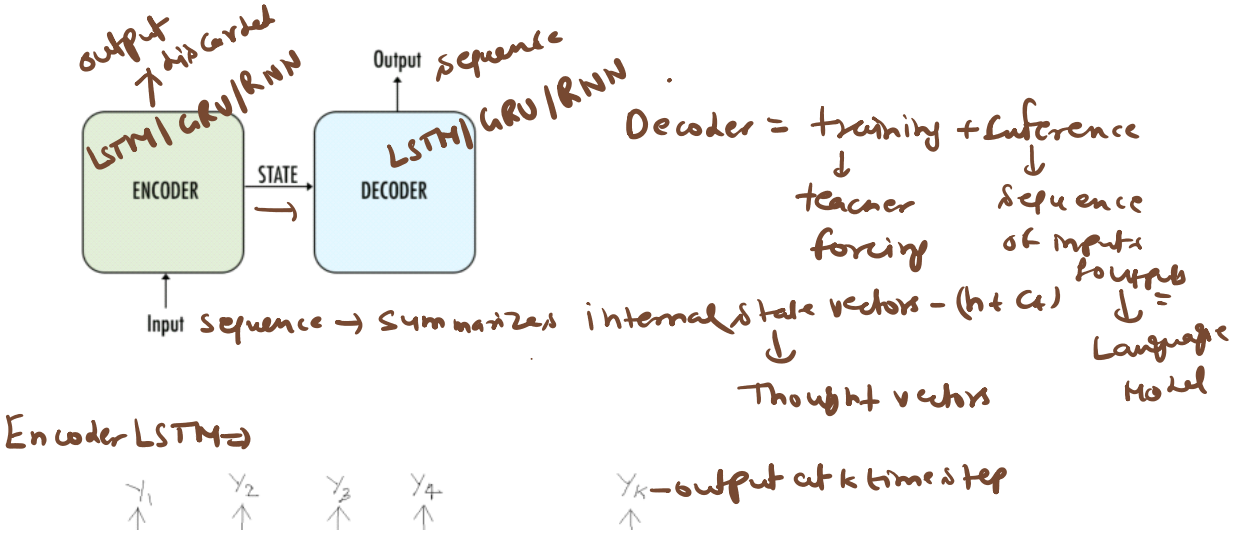
# Encoder-Decoder - Seq2Seq Models

Saturday, October 23, 2021 1:08 PM

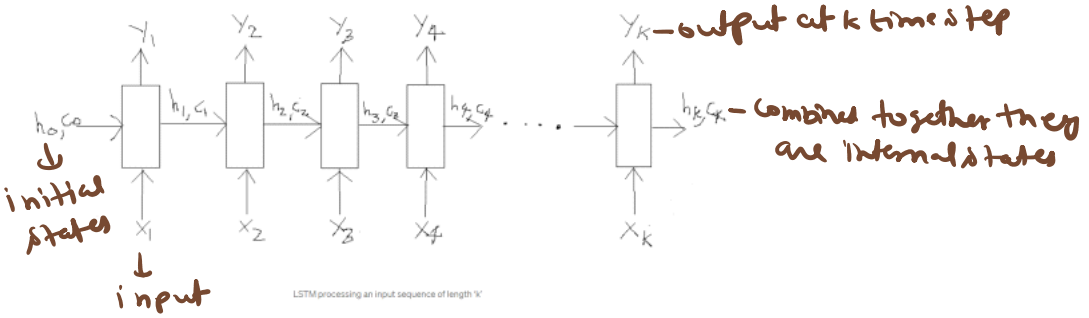
| Input | | Target | | Use Cases |
|---|---|---|---|---|
| Type | Elements | Type | Elements | |
| Scalar | One | Trends | Many | Pattern generation |
| | | Audio | Many | Music Generation |
| | | Text | Many | Text Generation |
| | | Image | Many | Image generation |
| Trends | Many | Scalar | One | Stock Trading decisions |
| | | | | Forecasting KPI for fixed duration |
| | | Trends | Many | DNA Sequence analysis |
| | | | | Time series forecasts |
| Text | Many | Scalar | One | Sentiment Classification |
| | | | | Topic Classification |
| | | | | Answer Selection |
| | | Text | Many | Text Summarization |
| | | | | Machine translation |
| | | | | Chatbots |
| | | | | Name Entity Recognition |
| | | | | Subject Extraction |
| | | | | Part of Speech Tagging |
| | | | | Textual Entailment |
| | | | | Relation Classification |
| | | Trends | Many | Path Query Answering |
| | | Audio | Many | Speech Generation |
| Image | Many | Scalar | One | Facial expression tagging |
| | | | | Entity classification |
| | | Text | Many | Image Captioning |
| | | Image | Many | Image Modification |
| Audio | Many | Scalar | One | Sentiment Classification |
| | | | | Number of speaker tagging |
| | | | | Topic Classification |
| | | Text | Many | Speech Recognition |
| | | | | Conference Summarization |
| | | Audio | Many | Speech Assistant |
| Video | Many | Scalar | One | Activity Recognition |
| | | Text | Many | Subtitle generation |




Source https://www.analyticsvidhya.com/blog/2018/04/sequence-modelling-an-introduction-with-practical-use-cases/

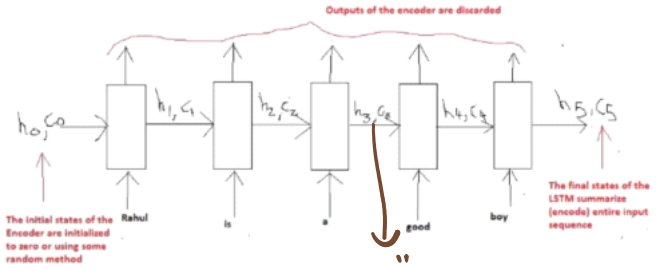


Encoder LSTM




Input sentence (English) => "Rahul is a good boy" — Q – How to represent each word as a vector??

Output sentence (Marathi) => "राहुल चांगला मुलगा आहे"




Encoder LSTM

$h_i, c_i$ =??

—

Rahul is a" till now – summary to 3 previous steps

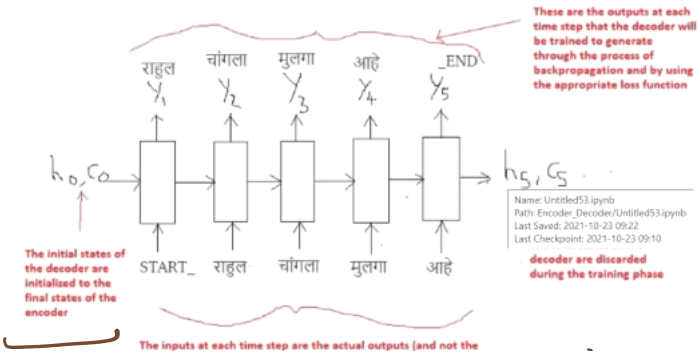
EncoderLSTM – Reads the input sequence word by word and preserve the internal states

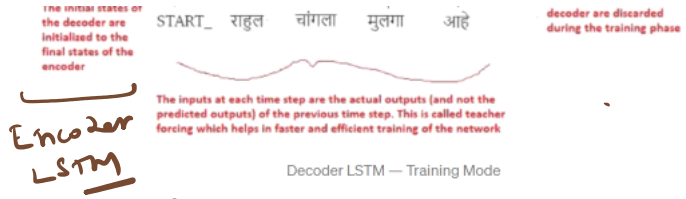
Post $h_k, c_k$ – encoding the input sequence

Decoder LSTM ⇒ ① Training ② Inference phase

↳ output sequence word by word

Start_ < > _END – like language Models

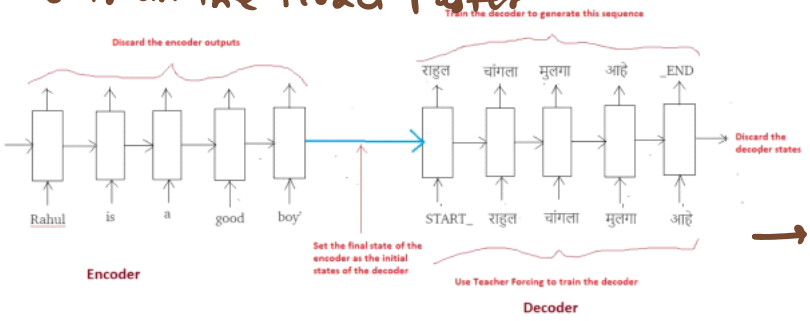





Decoder LSTM — Training Mode

Decoder depends upon the information received by encoder Thought vectors

Teacher Forcing ⇒ Input from previous time step is not the Prediction its actual to train the Model faster



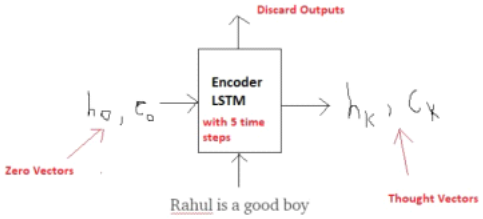

→ Training Mode

Summary of the training process

Decoder LSTM - Inference Mode ⇒ No prediction is happening in Encoder

But decoder has to predict the output

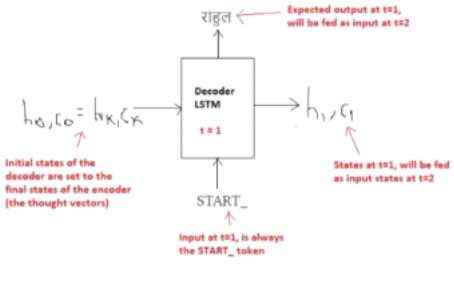
**Step 1**: Encode the input sequence into the Thought Vectors:




Encode the input sequence into thought vectors

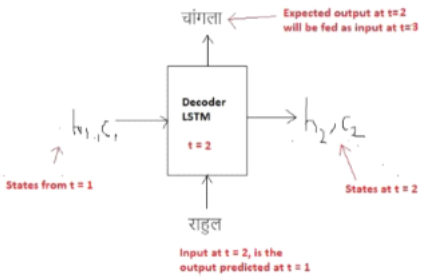
**Step 2**: Start generating the output sequence in a loop, word by word:

At t = 1



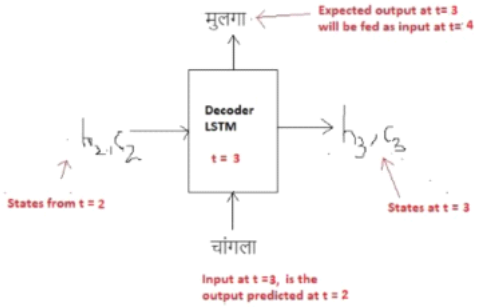

Decoder at t=1

At t = 2



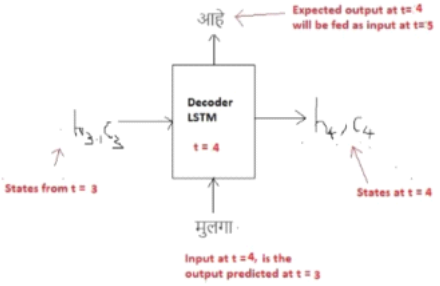

Decoder at t = 2

At t = 3



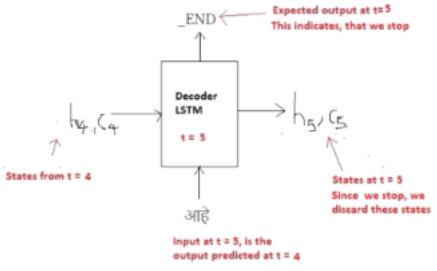

Decoder at t = 3

At t = 4



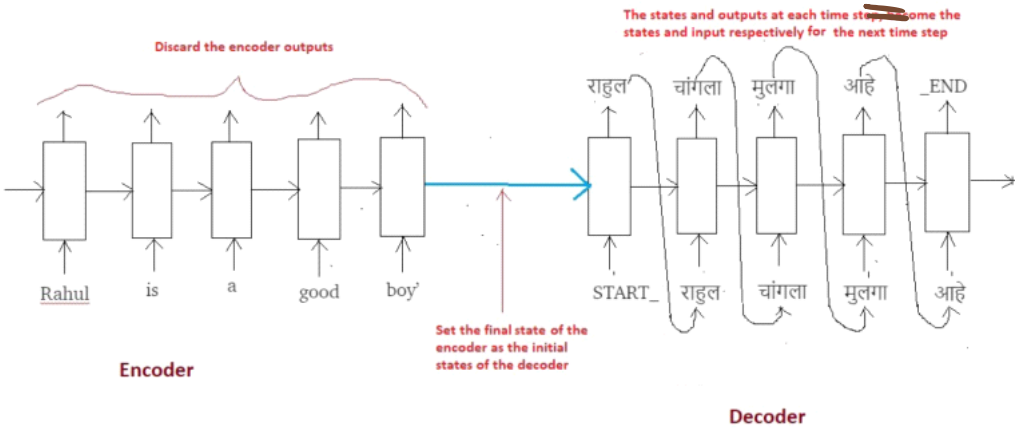

Decoder at t = 4

At t = 5




Decoder at t = 5

(i) One word at a time prediction via a loop — each time step
(ii) at each time step, the predicted output is fed as input in the next time step
(iii) Break the loop when we predict END word token




**Code - Move to Notebook**
**Dataset link -** http://www.manythings.org/anki/