

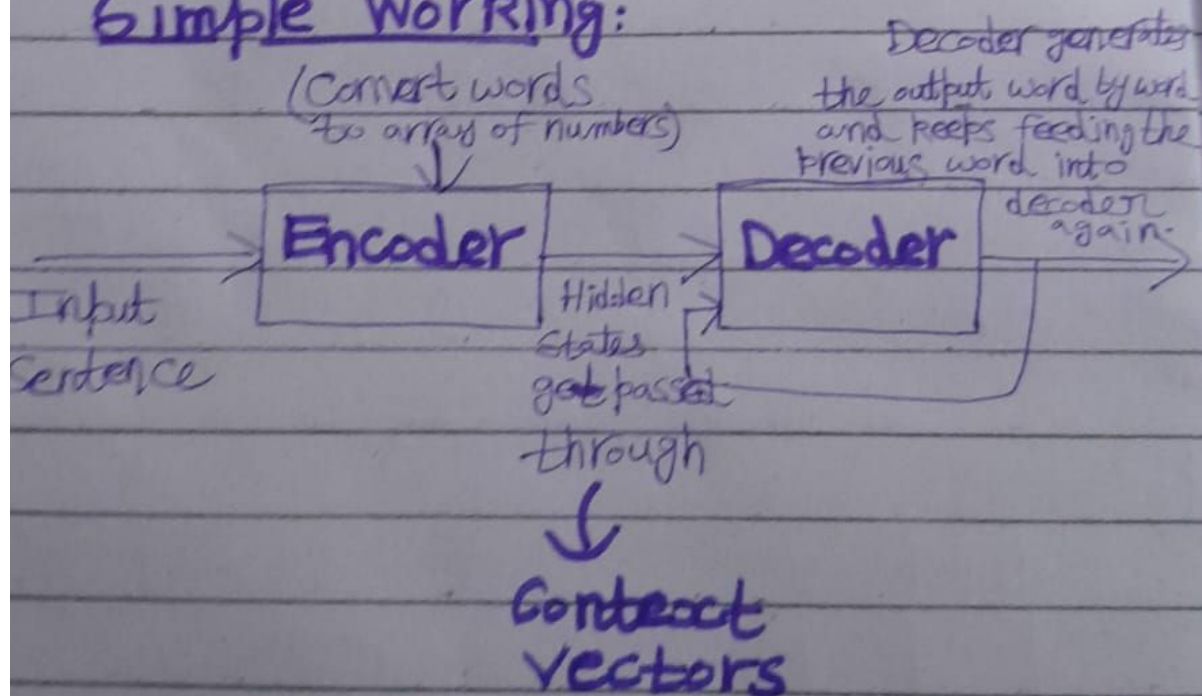
⇒ Encoder and Decoder:

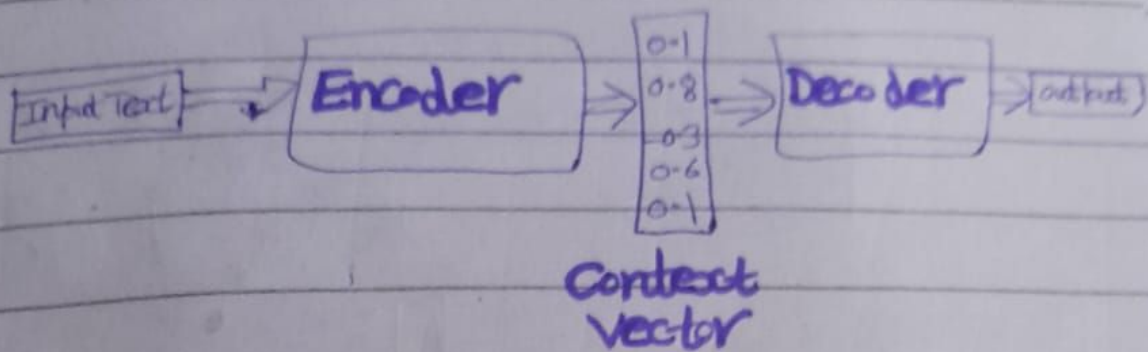
When we have many to many RNN and both input and output is sequential means sequence matters in both

e.g. One Language to other
English → French

In both input and output sequence is important. In such case we use **Encoder and Decoder**

Simple Working:





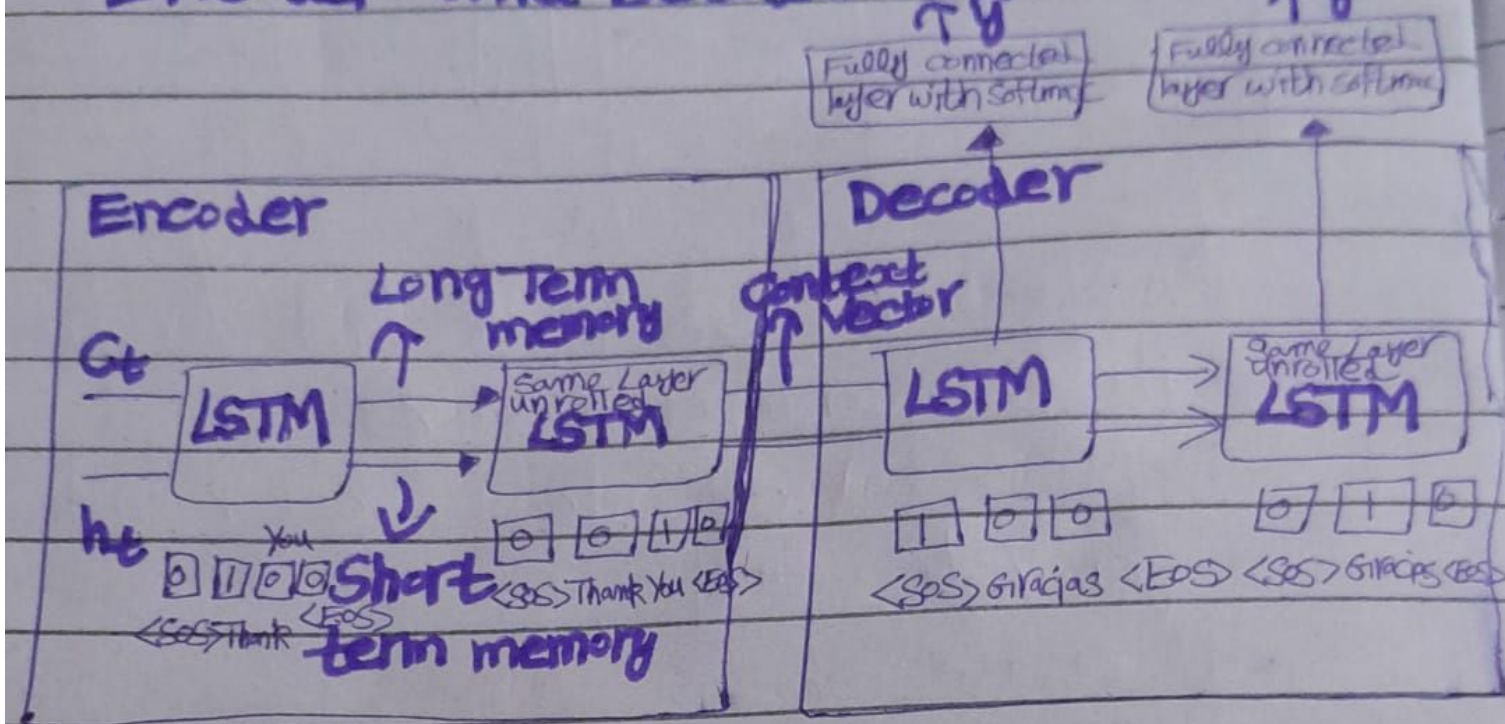
Encoder (Takes input and convert it to context vector)

Decoder (Use context vector and generate output)

• Uses:

- ① Language Translation
- ② Text Generation
- ③ Text Suggestion

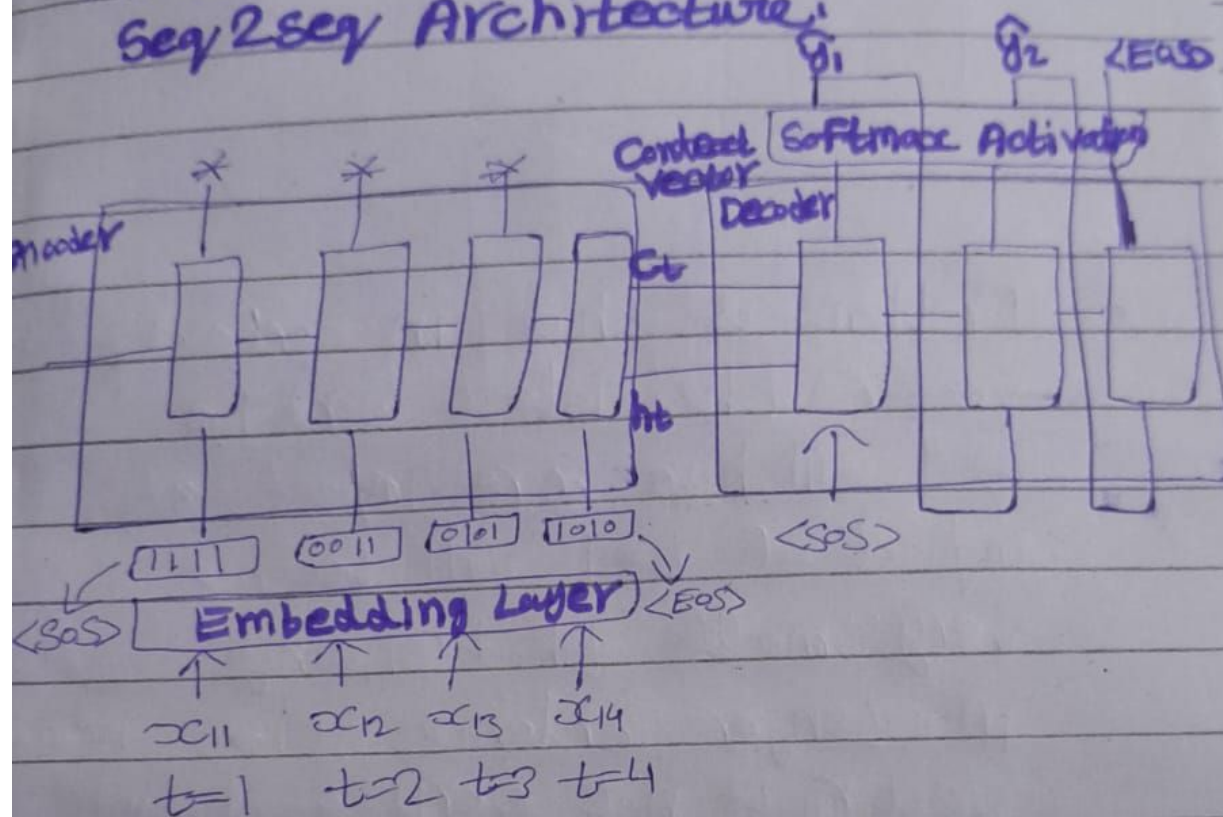
⇒ Sequence to Sequence (Seq2Seq) Encoder and Decoder Neural Network:



$$\text{Loss} = y - \hat{y} \quad (\text{Reduce it using optimizers})$$

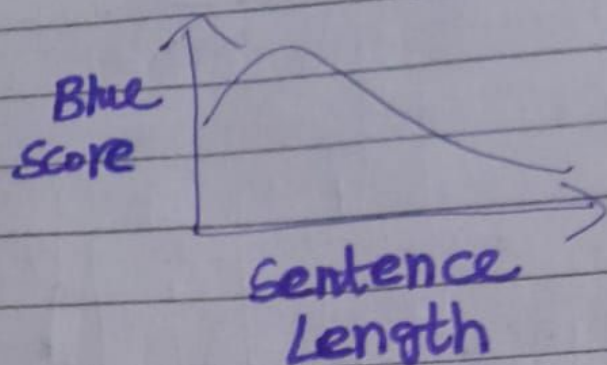
(For more Detailed Diagram See notes in repo)

Problems with Encoder and Decoder Seq2seq Architecture:



- **Context Vector** is combination of C_t (Long Term Memory) and h_t (Short term memory)
- **Context Vector** \Rightarrow Represents the entire sentence

Problem: Researchers try sentences of varying length and observed as the length of sentence become large blue score decreases



- Problem is that in context vector the latest word i.e. $t=4$ will have more impact and impact of $t=1$ will become very small that's why in longer sentence the context vector is not very good and **Blue score decreases.**

⇒ **Attention Mechanism (In Repo)**