

WB

Stat Quest

①

## Seq2Seq Encoder - Decoder NNS

language - to - language = Seq2Seq

uses encoder - decoder

↳ precursor to transformers

not all input sentence will be the same length

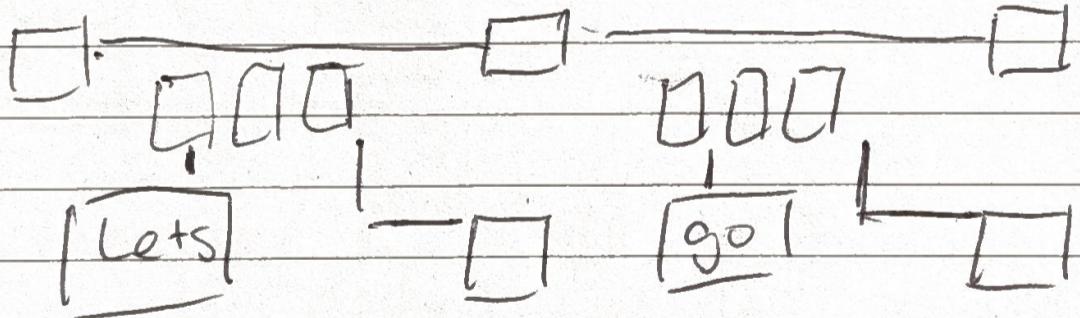
↳ Same for output

↳ Input  $\neq$  output length

need to be able to handle variable lens

long short-term memory arch already tells us how to handle variable lens

by "unrolling" the LSTM



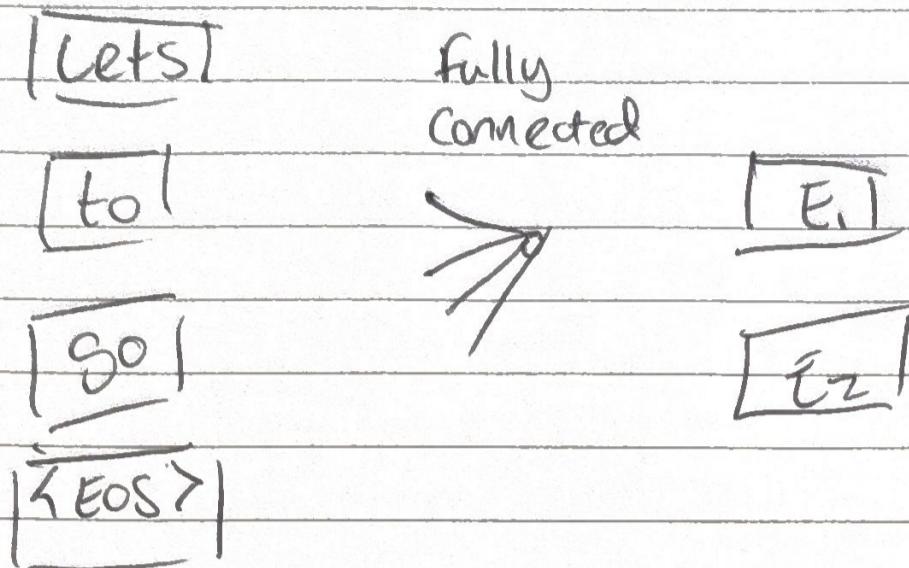
Start w/ sentence but use embed layer to convert words into numbers

individual words in vocab/sentence = tokens

you create embeds from tokens

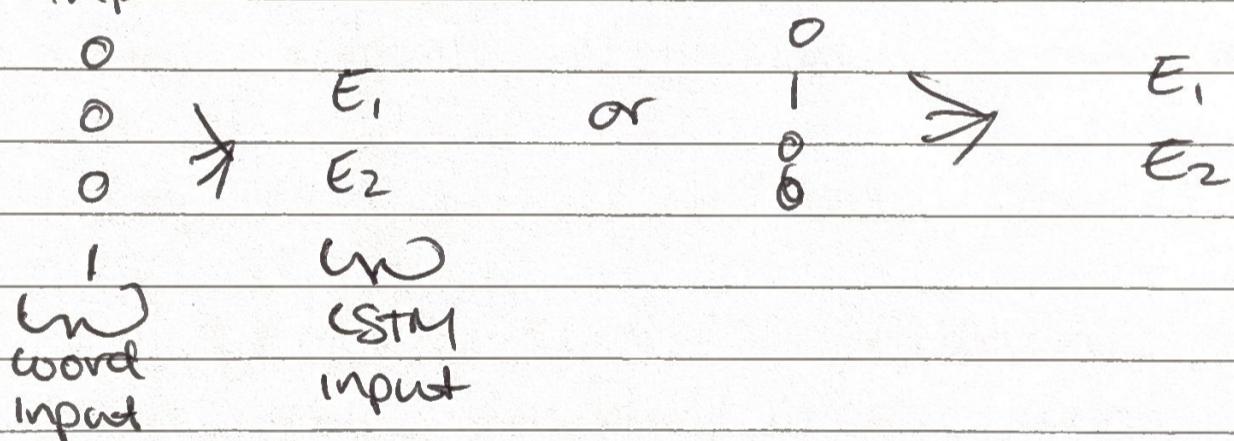
(2)

e.g 4 tokens into 2 embeds



embeds can be feed into LSTM

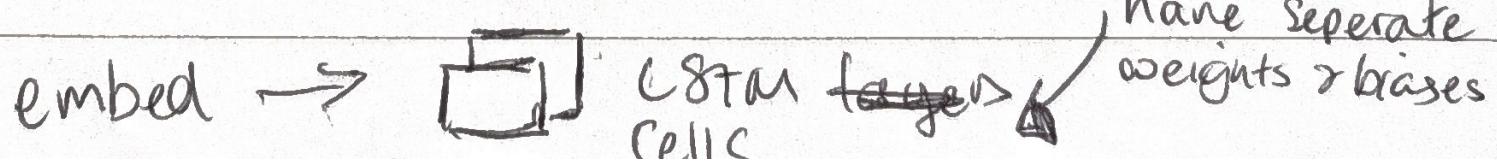
for each word there is a 2-dg embed input



the same weights & biases are used  
in the embed & LSTM regardless of  
the word inputted

► this whole part of the process is  
encoding the input sentence

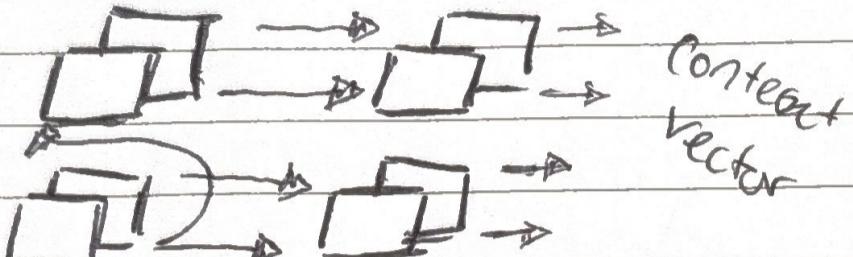
to add complexity more LSTM layers  
can be added to model



① can have additional LSTM cells



② also can have additional LSTM layers



encoder part of  
encode-decode model

the Encoder encodes the input sentence  
into long & short term memories

Aka cell & hidden states

the final cell & hidden state  
from the encoder/LSTM are called  
the content vector

Encoder  $\rightarrow$  Content Vector

now we need to Decode the content vec

Decoder has same LSTM structure to  
accept the content vector as input

but has its own weights & bias

Content vector init the cell & hidden state  
of the decoders

(4)

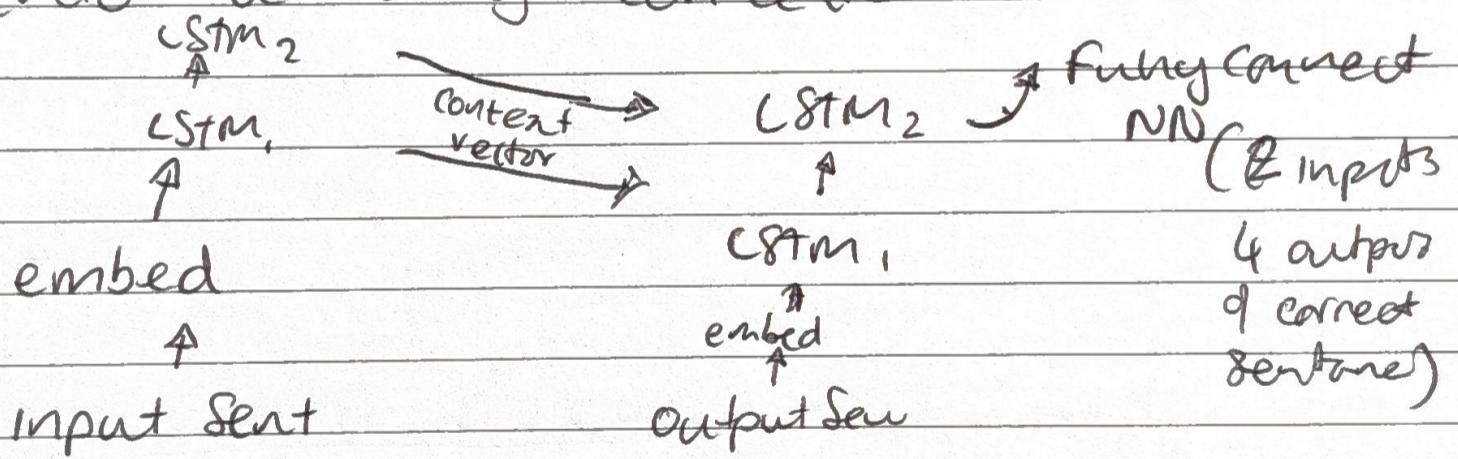
goal of Decoder is to convert Context vector into output sentence

the input to the Decoder LSTM as comes from an embed layer

↳ this time ~~is~~ from the desired output sentence

Input & output embeds have big weights & biases

At the end, the cell & state from the top right LSTM are put into a fully connected NN



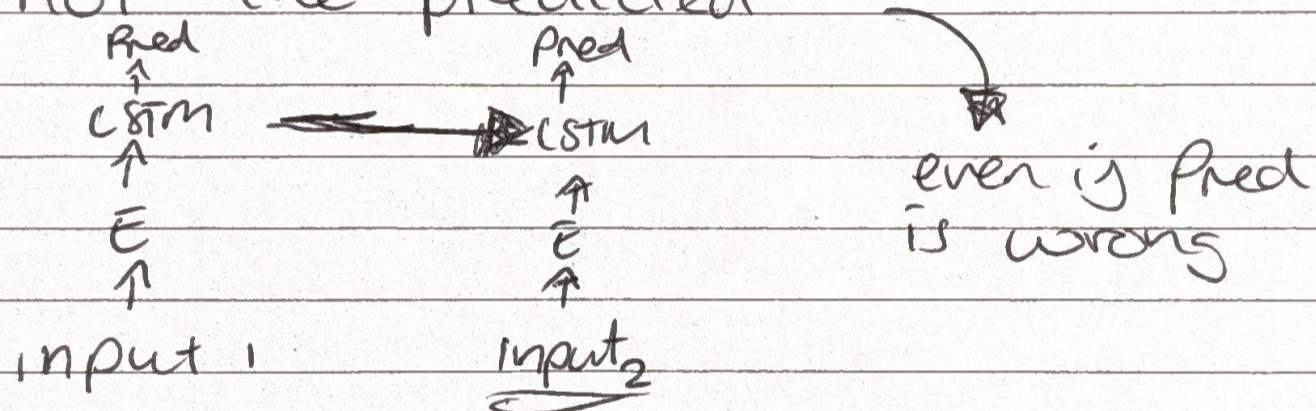
fully connected uses soft max for correct word  
↳ iterate until <eos>

by decoupling the encoder - Decoder, the input & output may be different lengths

All weights & biases are trained using Backprop

important:

- (1) In the decoder training, when unrolling the LSTM, the input token is the correct token, not the predicted



- (2) Decoder stops when true sentence is EOS, even if pred  $\neq$  EOS

### Orig Seq2Seq Paper

- Uses input vocab of 160k tokens & 80k output
- 1000 embeds per token
- 4 LSTM layers
- 1000 cells
- 1000 fully connect inputs
- 80k Fully connect outputs
- 384 million weights & bias