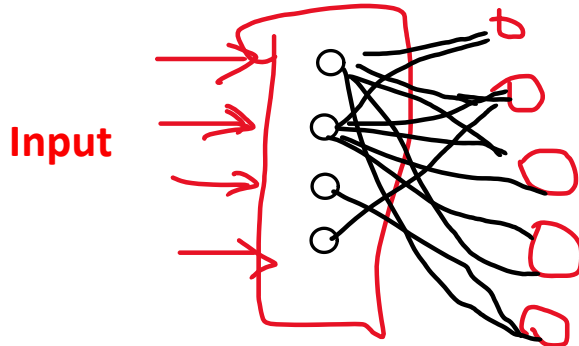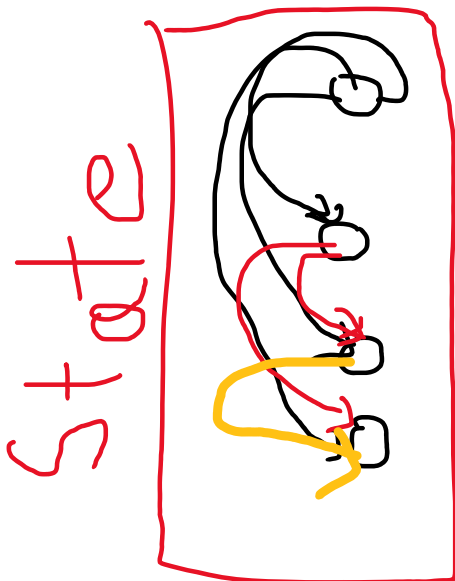# LSTM

Thursday, February 20, 2025   12:52 PM

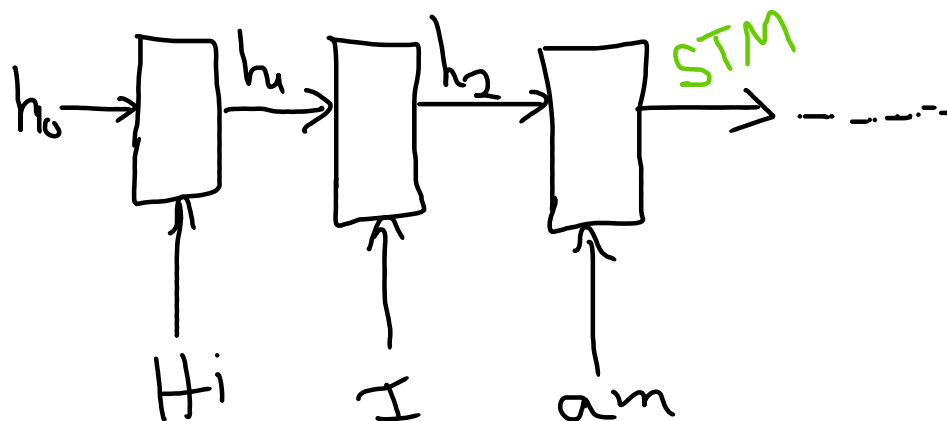## ► How ANN works and What was the problems?

Input

- Inputs are not given in a sequence.
- The meaning of different words is different in different position.
- That's why we need a way solve the problem.

## ► How RNN Works and what are the problems?

STM- Short term Memory

State

- In theory RNN works well but when the size of the paragraph is long it faces an issue called vanishing gradient .
- For instance when I talk about a city and at last I talked about the city life, this architecture will Remember only the last part and forget the first portion of the paragraph.
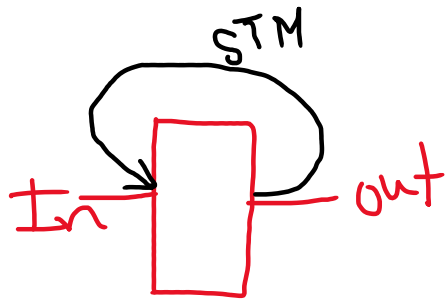
► How LSTM works?
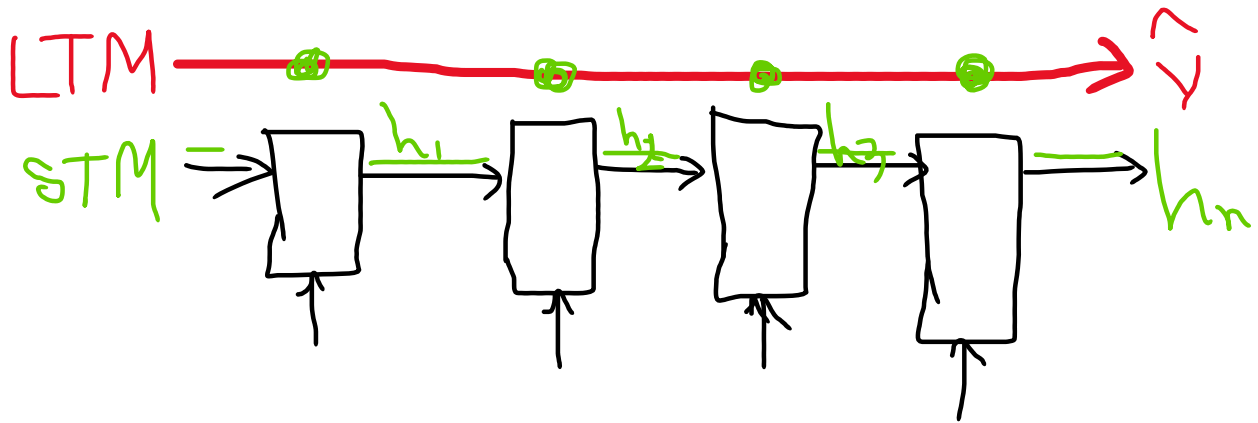
- In LSTM, system store LTM line along with the sort term memory space.
- When system think that some specific words are important for long term then it transfer the words from STM to the LTM.
- If the words are not important then it will be removed eventually from the LTM or it will be stored till the end of the process.
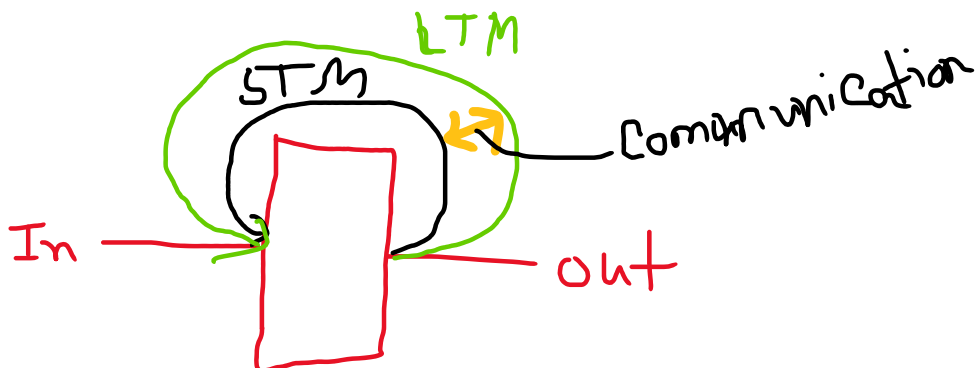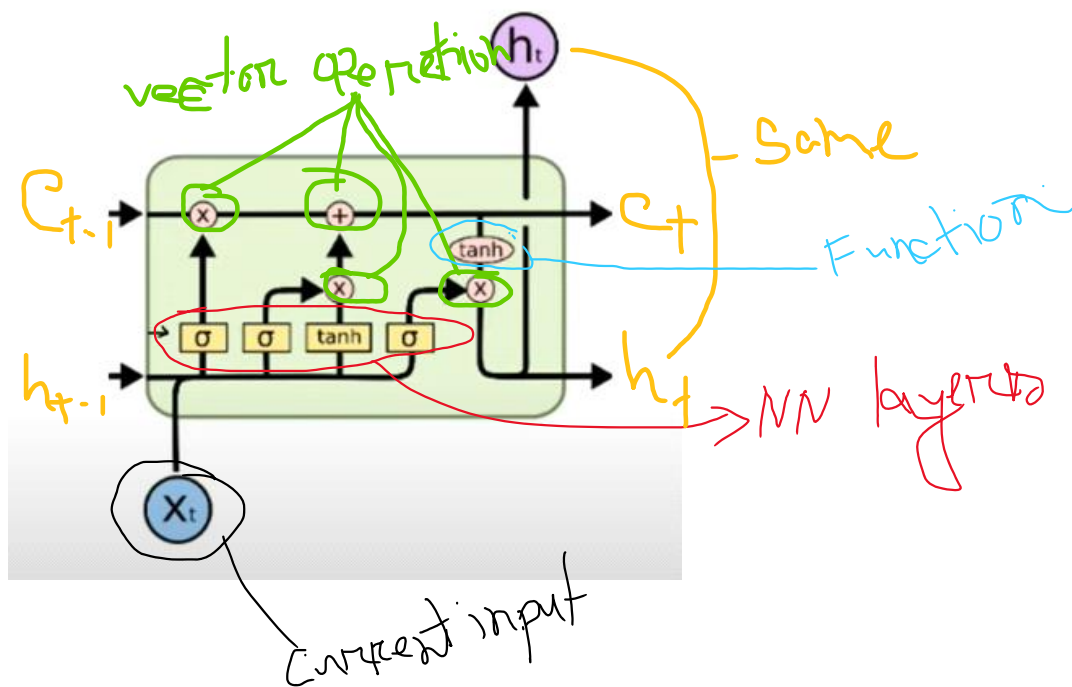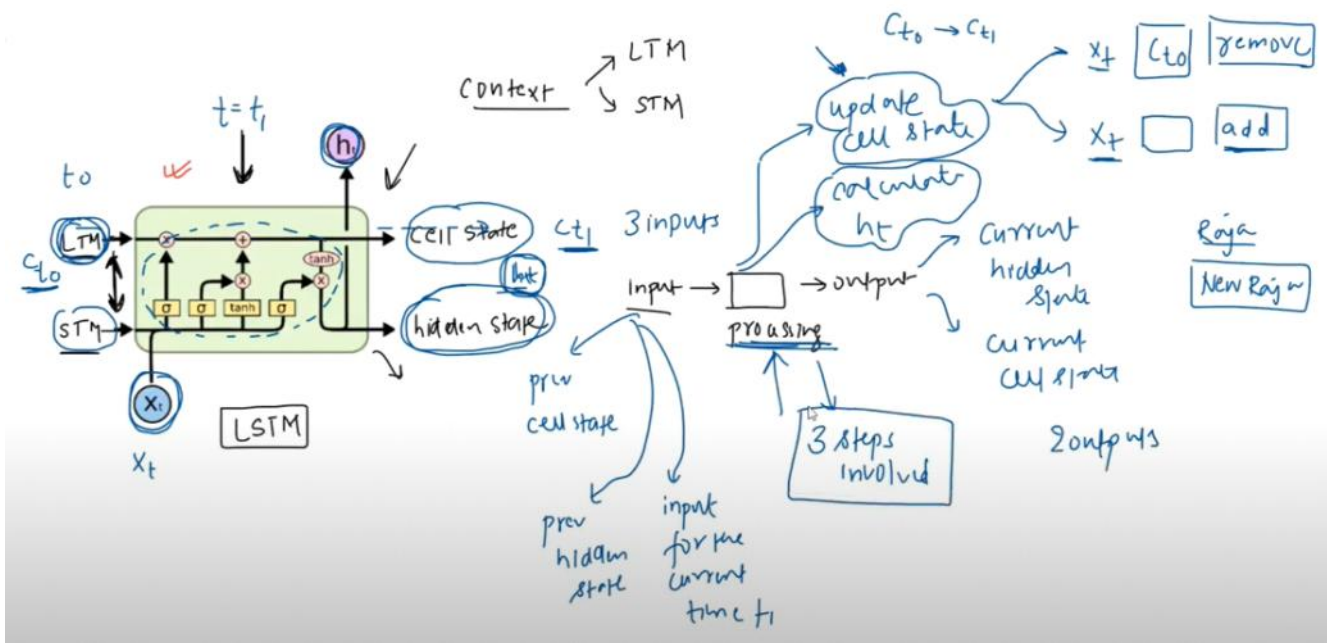

- Need to handle an architecture for the communication between the STM and LTM.
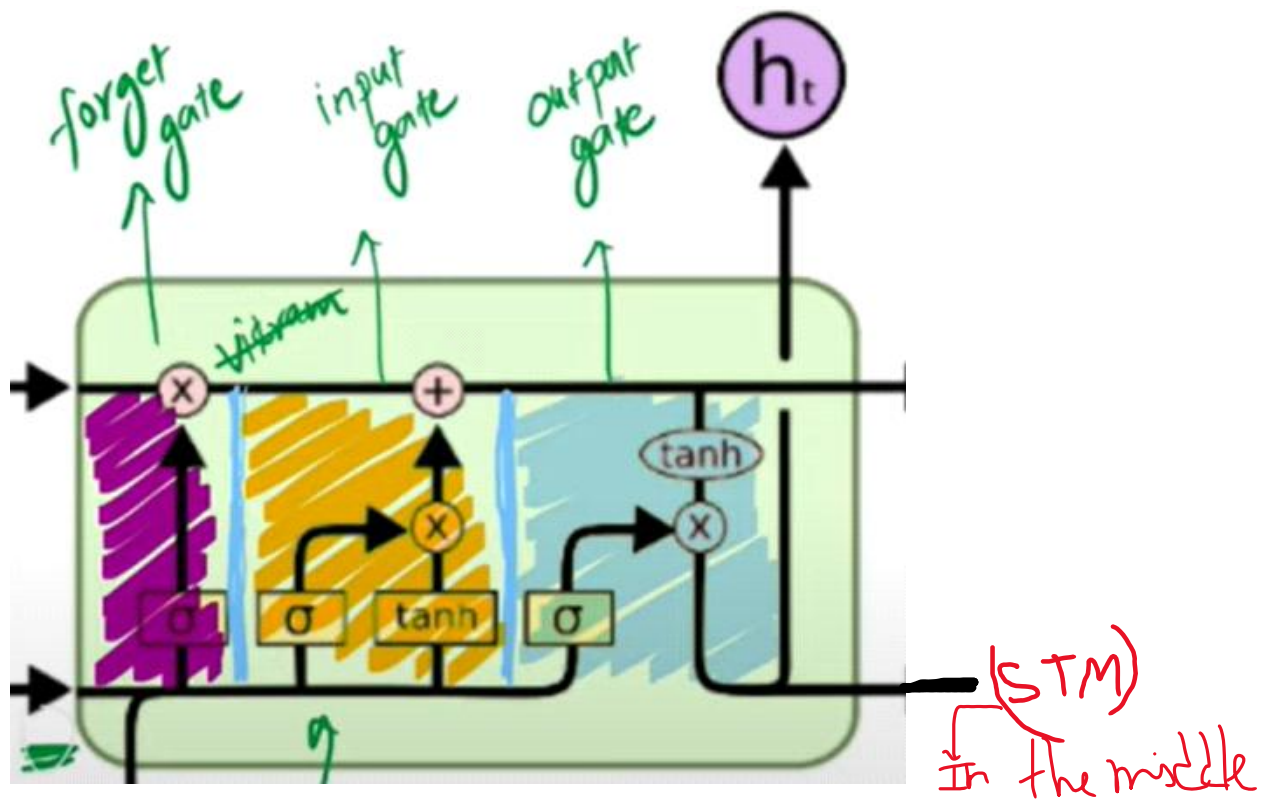
- Forget gate remove the words that are not important for future, based on current input.
- Input gate add the new words based on current input.
- Output gate provide the output at the last based on the current input and the LTM data.

► **Knowledge about LSTM Architecture**

Context → LTM / STM

$t = t_1$

t_0

h_1

LTM — c_{t_0}

STM

Cell state — c_{t_1} 3 inputs

link

hidden state

LSTM

X_t

prev cell state

prev hidden state

input for the current time t_1

$c_{t_0} \rightarrow c_{t_1}$

update cell state

calculate h_t

Input → [ ] → output

proussing

3 steps involved

x_t | c_{t_0} | remove

x_t | [ ] | add

current hidden state

current cell state

2 outputs

Raja

New Raj~

---

vector operation h_t

C_{t-1} — C_t — Same

tanh   — Function

h_{t-1} h_t → NN layers

X_t → Current input

Forget Gate: After getting the output of previous state, h(t-1), Forget gate helps us to take decisions about what must be removed from h(t-1) state and thus keeping only relevant stuff. It is surrounded by a sigmoid function which helps to crush the input between [0,1].
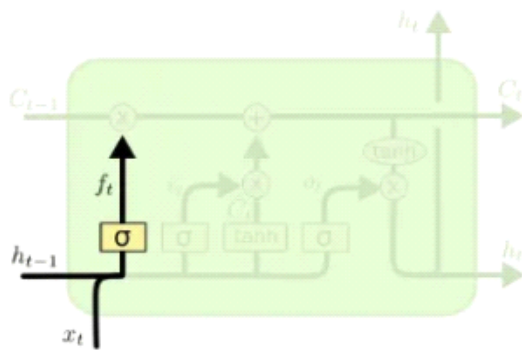
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

Forget Gate, src: Google

Input Gate: In the input gate, we decide to add new stuff from the present input to our present cell state scaled by how much we wish to add them In the above photo, sigmoid layer decides which values to be updated and tanh layer creates a vector for new candidates to added to present cell state.
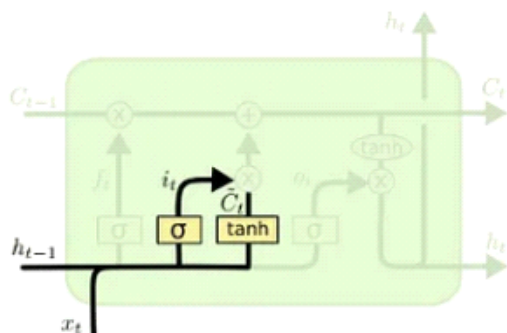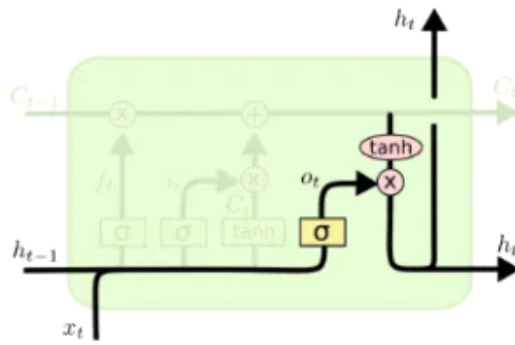

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Input Gate+Gate_gate,photo credits: Christopher Olah

Output Gate: Finally we'll decide what to output from our cell state which will be done by our sigmoid function. We multiply the input with tanh to crush the values between (-1,1) and then multiply it with the output of sigmoid function so that we only output what we want to.

output Gate, source:Google

$$o_t = \sigma \left(W_o \left[h_{t-1}, x_t \right] + b_o \right)$$

$$h_t = o_t * \tanh \left(C_t \right)$$