# LSTM Part 1: The What?

16 October 2024 19:10

## LSTM : The Why?

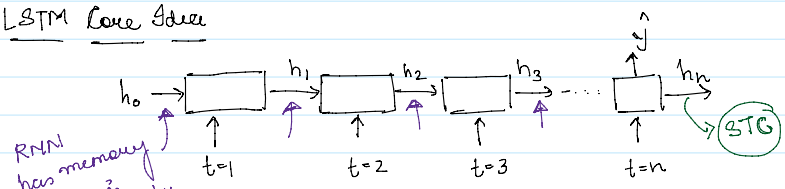
↓

Long Short Term Memory

↓

They were made because RNNs couldn't handle long term dependency

### LSTM Core Idea

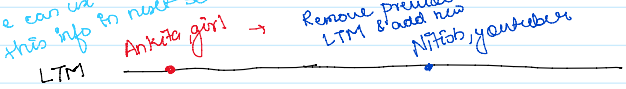


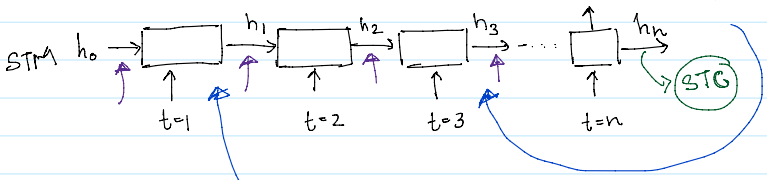

RNN has memory and it is maintained by the recurrent unit → so it is upto the same unit to maintain LTC & STC

↓ mathematically we saw in BPTT, in RNN due to vanishing/exploding gradient only STC dominates

This is when scientists thought that why not maintain a connection to get LTC

We can use this info to reset sentence




— Nitish is an youtuber.
— lives in India

Jab current input ke lie lagta hai ki wo Long term me important hai, hum use long term memory me daal denge

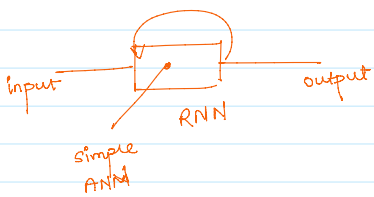
Example

— Ankita is a good girl. — ? — lives in India

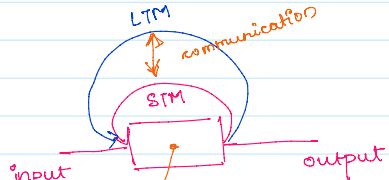
She

predict

LTM remains until it is removed and it is propagated till end




simple RNN




LSTM

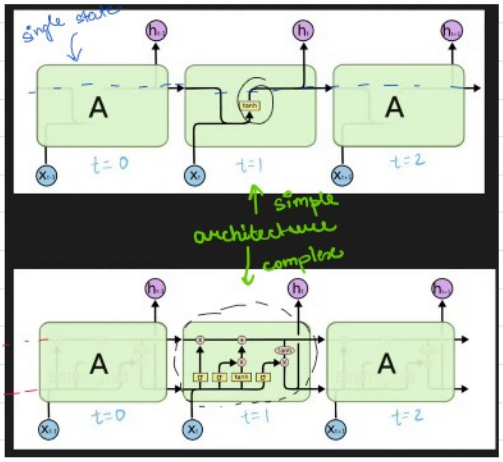
different structure

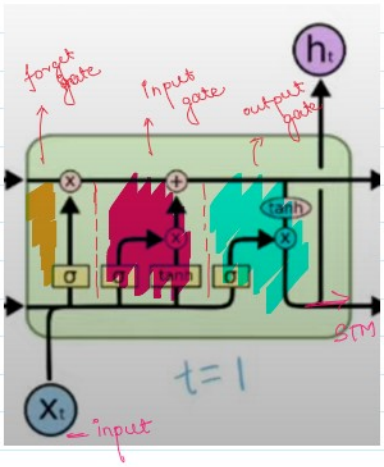
(because STM & LTM communicate to decide whats important)

2 differences

— 2 states (LTM + STM)

— architectural differences due to communication




forget gate

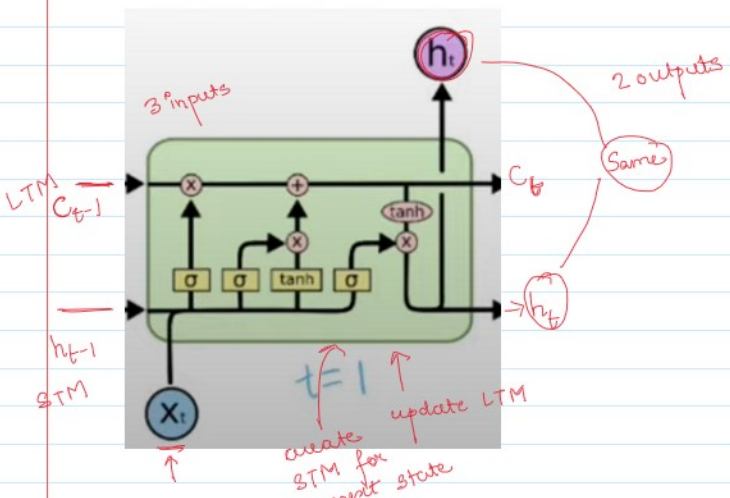
based on current input + STC ↓ decide what to remove from LTM

input gate

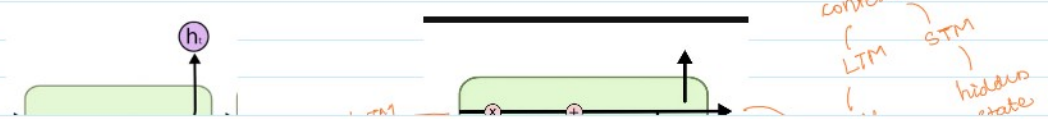
based on current input decide what to add in LTM

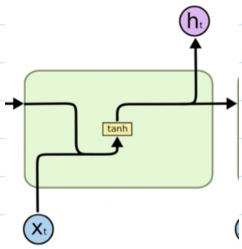
output gate

based on current input and LTM gives output → in intermediate stages it gives STM

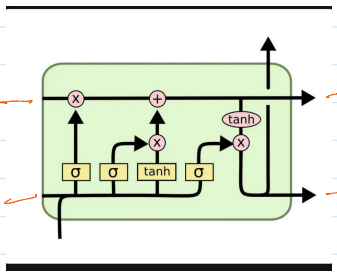



## Architecture of LSTMs





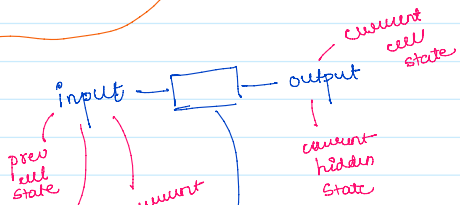

RNN




LSTM

LTM → cell state

STM → hidden state




$C_{t_0} \rightarrow C_{t_1}$

2 parts

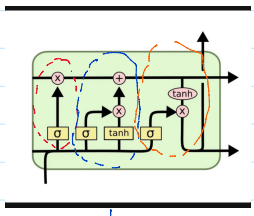
based on $X_t$: decide what to remove from LTM

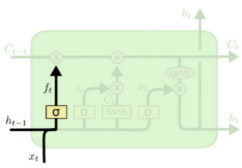
based on $X_t$: decide what to add to LTM

- update cell state $C_{t_0} \rightarrow C_{t_1}$
- calculate $h_t$ ($h_t$)

## The Gates

3 gates



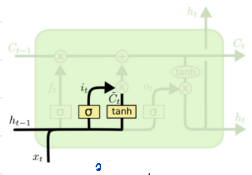




forget gate

remove something from cell state

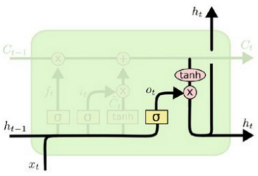
$C_{t_0} \rightarrow C_{t_1}$ update

+




input gate

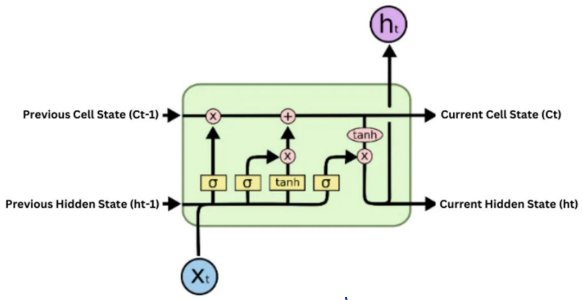
add something to cell state




output gate

calculate $h_t$

## What are $h_t$ and $C_t$?




[1 0 0] one word into vector

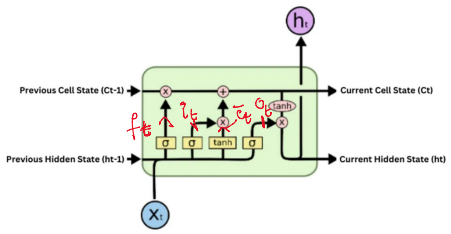
t=1 t=2 t=3

- They are vectors
- They have same dimension

| | |
|---|---|
| cat mat rat | 0 |
| cat rat rad | 0 |
| mat mat cat | 1 |

↓ vectorize OHE

| | |
|---|---|
| [1 0 0] [0 1 0] [0 0 1] | 0 |
| [1 0 0] [0 0 1] [0 0 1] | 0 |
| [0 1 0] [0 1 0] [1 0 0] | 1 |




What are $f_t, i_t, o_t, \tilde{c}_t$?

$f_t$ — forget gate

$i_t$ $\tilde{c}_t$ — input gate; $\tilde{c}_t$ — candidate cell state

$o_t$ — output gate

They have same dimension as $c_t, h_t$

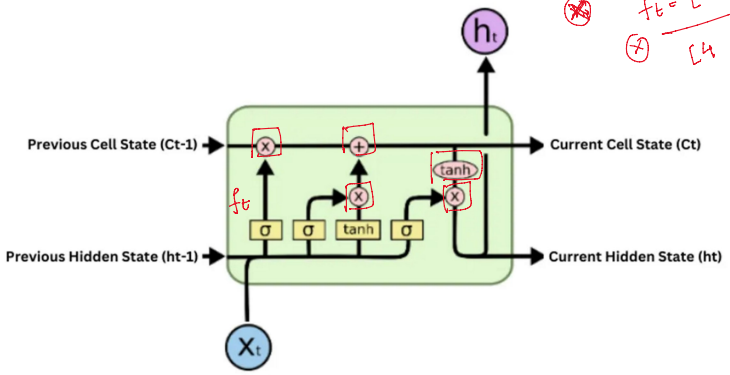
## Pointwise Operation

$C_{t-1} = [4\ 5\ 6]$

$f_t = [1\ 2\ 3]$

$\otimes$ (pointwise multiplication)

$\oplus$ $[4\ 10\ 18]$




$\oplus$ $C_t = [4\ 5\ 6]$

$f_t = [1\ 2\ 3]$

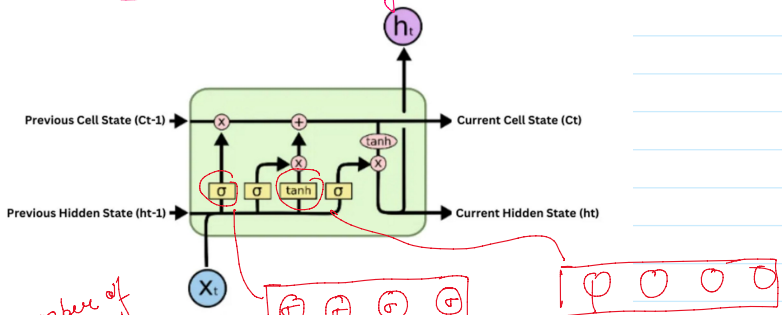
$[5\ 7\ 9]$

tanh $C_t = [4\ 5\ 6]$

$\tanh(4)$ $\tanh(5)$ $\tanh(6)$

$[\ -\ \ -\ \ -\ ]$

## Neural Network Layers



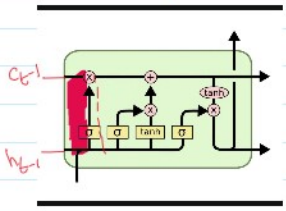

- The number of nodes in the NN layer are hyperparams
- The nodes will be same for all the 4 NNs

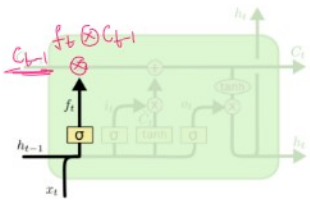
## The Forget Gate


input

output

## The Forget Gate




**input**
$h_{t-1}, C_{t-1}$
$x_t$

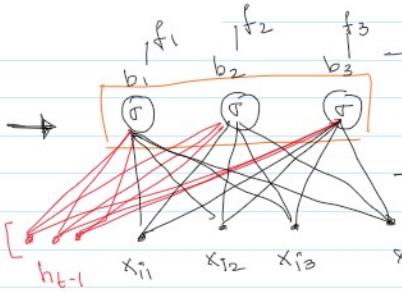
**output**
remove from cell state

1) Calculate $f_t$
2) $C_{t-1} \otimes f_t$

mem units → (3)

$X_t \rightarrow$ 4 dim vector
$[x_{i1}\ x_{i2}\ x_{i3}\ x_{i4}]$

The dimension of $C_{t-1}, h_{t-1}$ will be same as number of nodes in NN layer




$\rightarrow [f_1\ f_2\ f_3] = f_t$ (1×3)

$\rightarrow W_{3\times7} = W_f$

21 weights + 3 biases

concat $h_{t-1}$ & $x_t$

$(3\times7)(7\times1) = (3\times1) + (3\times1)$
$= (3\times1)^T = f_t$

$$f_t = \sigma\left(W_f [h_{t-1}, x_t] + b_f\right)$$

$3\times7 \otimes 7\times1 \rightarrow 3\times1$
$3\times1 \oplus 3\times1 \rightarrow (3\times1)\ f_t$

$C_{t-1}(3\times1) \otimes f_t(3\times1)$

$\rightarrow (3\times1)$

$C_{t-1}$ [4 5 6] $\cdots \otimes \rightarrow$ [2 2.5 3]

$f_t$ [½ ½ ½]

basically this means we are halving the values in LTM and forgetting half LTM

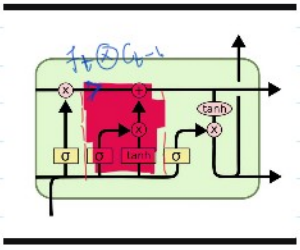
If $f_t = [0\ 0\ 0] \rightarrow$ forget all LTM

$f_t = [1\ 1\ 1] \rightarrow$ remember all forget nothing

$\therefore f_t$ decides how much information to let flow

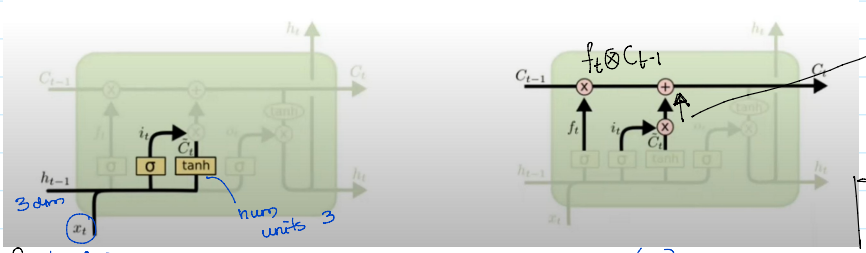
## Input Gate

Add some new important information to $C_t$




Stages

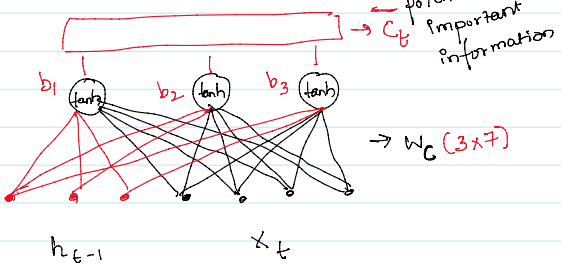
1) $\bar{C}_t$ candidate cell state
2) $i_t$ decides what values from $\bar{C}_t$ to add in $C_t$
3) $C_t$ final cell state



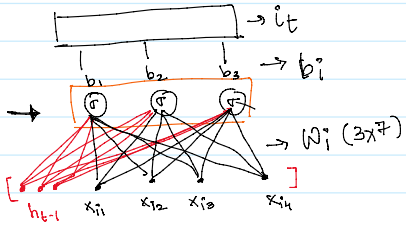

$i_t \otimes \bar{C}_t \rightarrow \bar{C}'_t$ (filtered candidate cell state)
(3×1) (3×1) (3×1)

$C_t = f_t \otimes C_{t-1} \oplus i_t \otimes \bar{C}_t$

$\bar{C}_t$ calculation




$i_t$ calculation




$$\bar{C}_t = \tanh\left(W_C[h_{t-1}, x_t] + b_C\right)$$

(3×7) (7×1) (3×1) (3×1)

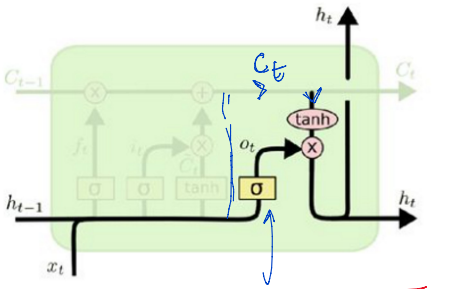
$\tanh(3\times1) = (3\times1)\ \bar{C}_t$

$$i_t = \sigma\left(W_i[h_{t-1}, x_t] + b_i\right)$$

(3×7) (7×1) (3×1)

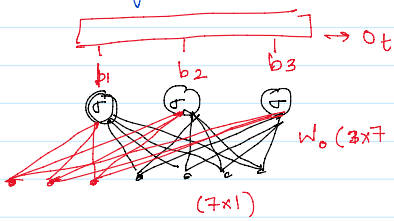
$\sigma(3\times1)$ (3×1)$^T$ $i_t$ – filter for candidate cell state

## Output gate




1) apply tanh on $C_t$
2) filteration of above using $O_t$
3) $O_t$ calculation

$h_t = [O_t \otimes \tanh(C_t)]$




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

(3×7) (7×1) + (3×1)

(3×1) $O_t$

$$h_t = O_t \otimes \tanh(C_t)$$

(3×1) (3×1)

(3×1)