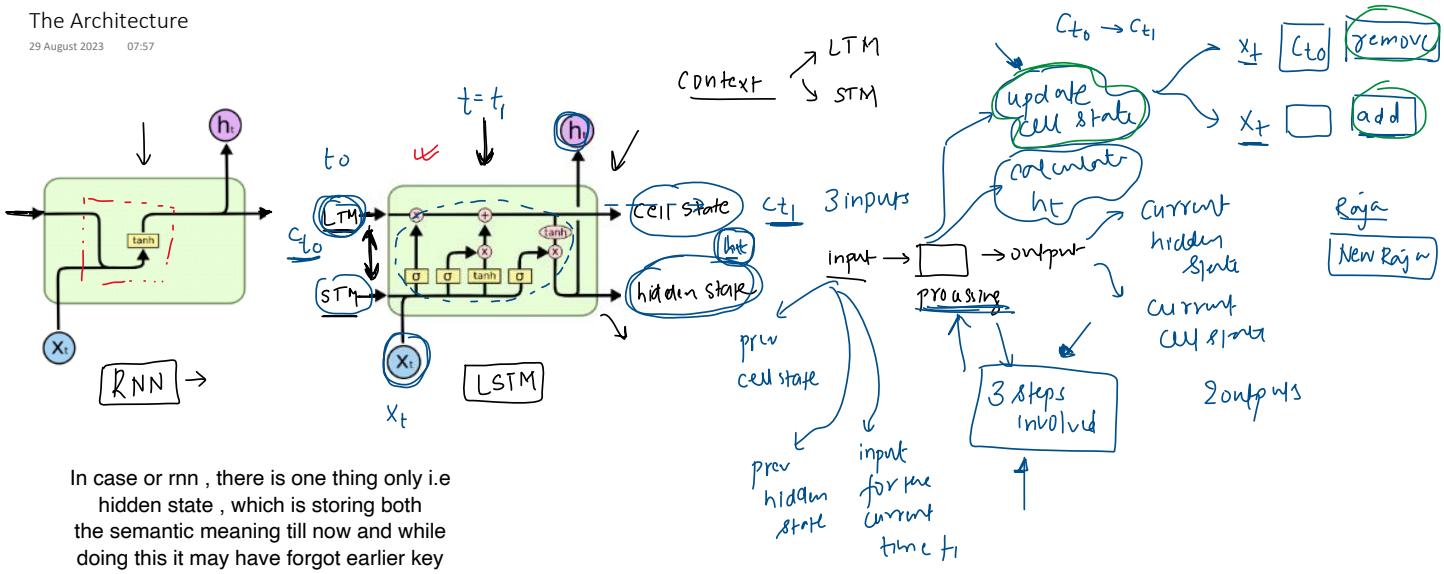
The Architecture

29 August 2023 07:57



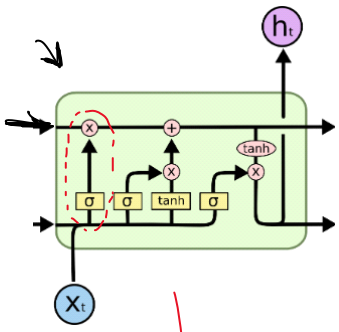
In case or rnn , there is one thing only i.e hidden state , which is storing both the semantic meaning till now and while doing this it may have forgot earlier key things so there is problem of long term dependancy , in some or other way it priotized current or fail to consider the long term .

Also in rnn , we are just performing single operation to get hidden state. (we are not considering ,there things like what to keep from long term or somehting like this)

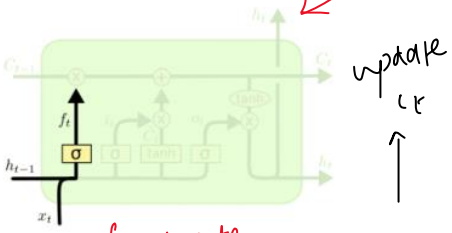
LSTM solves this problem by having two states i.e cell state which act as long term memory and hidden state which act as short term memory .
And we are updating these both state whenever is input added and keep this updated.

The Gates

29 August 2023 08:09



3 gates

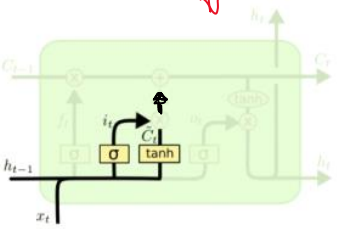


update C_t

forget gate

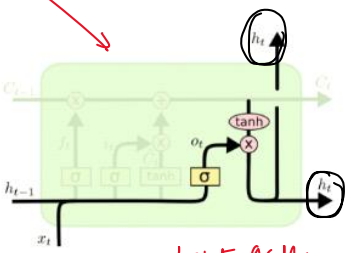
to remove something from C_t

+



input gate

to add something to the C_t



output gate

Calculat hidden state h_t

How this multiplay wala works and like how it is helping there to forget , so like ct : [1,2,3,4] no when we multiply by it [0.6,0.5,0.4,1] , so it is forgetting like 0.4 of infro from 1 and 0 from 4. In this way it works.

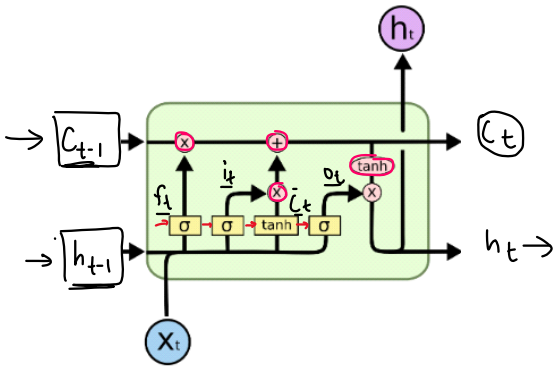
Ct bar : is the potential cell state.

Here we are creating next hidden state by using new cell state , prev hidden state and current input.

Three gate are there in LSTM ;
1. Forget Gate (to remove something from ct)
2. Input Gate (add somthing to ct)
3. Output Gate (gives ht)

What are C_t and h_t

29 August 2023 08:08



h_t C_t → [0.1 0.3 0.9]

3d vector

vectors

rule

same

h_t C_t dim equal

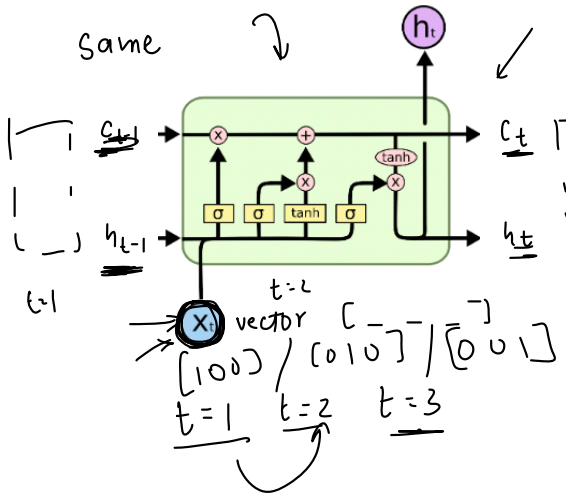
h_t [0.1 0.45 0.6]

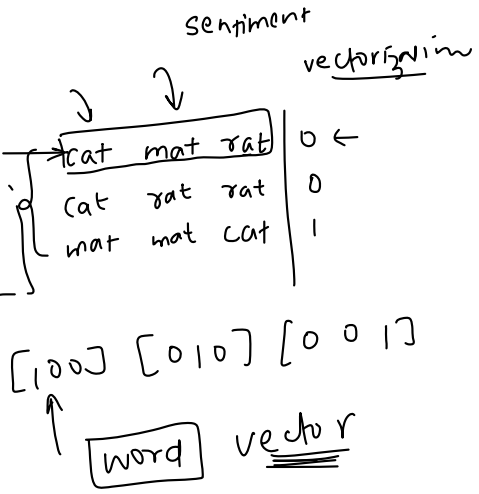
C_t [0.55 0.6 0.0]

What is X_t

29 August 2023 17:40

RNN



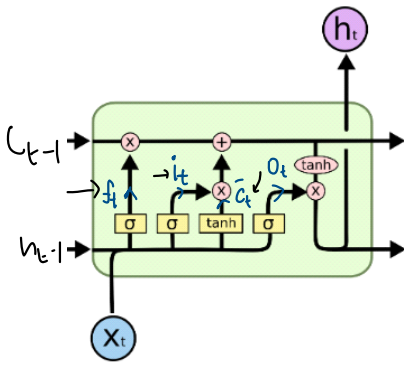


OHE

Cat	mat	rat
1	0	0
0	1	0
0	0	1

What are f_t, i_t, o_t and $\bar{C}t$

29 August 2023 08:09



f_t i_t $\bar{C}_t$ → Candidate cell state O_t

f_t: forget gate

i_t $\bar{C}_t$: input gate

O_t: output gate

vectors → C_t h_t

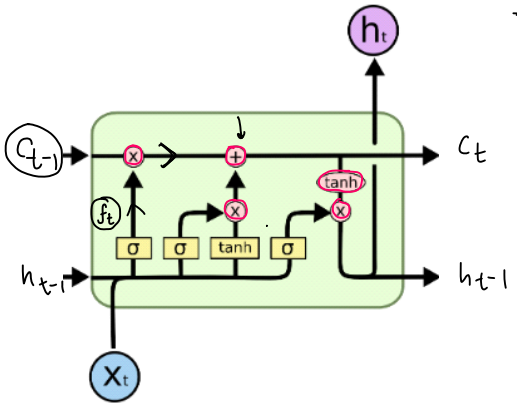
f_t i_t $\bar{C}_t$ O_t

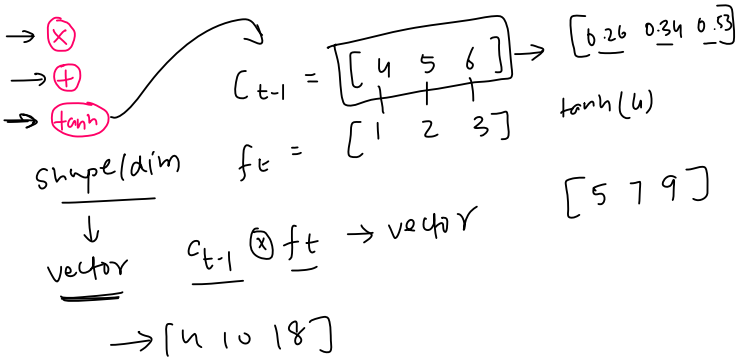
[x y z]

[ ]

Pointwise Operations

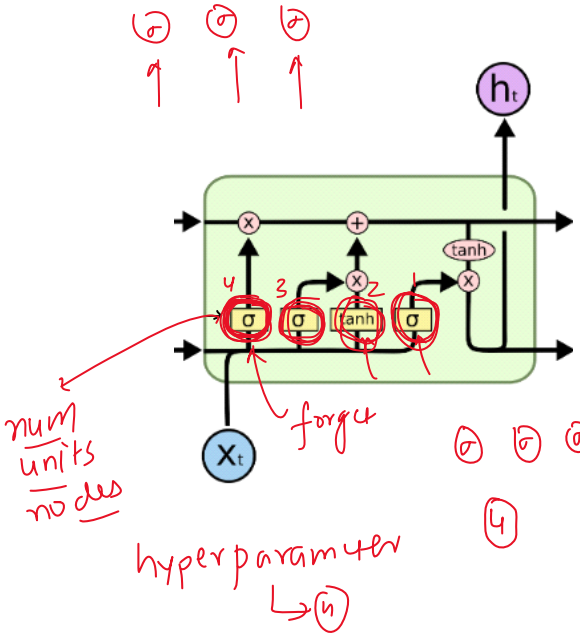
29 August 2023 18:26

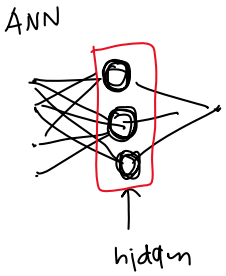




→Neural Network Layers

29 August 2023 18:34

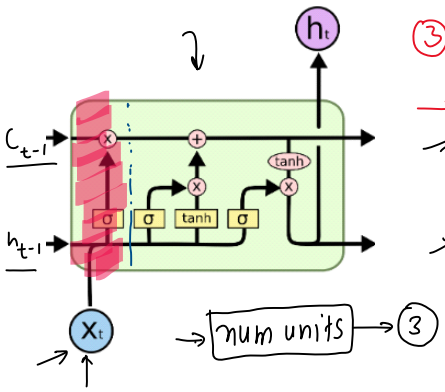


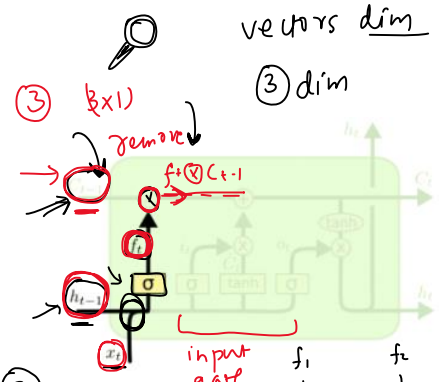


hyperparameter

The Forget Gate

29 August 2023 19:58





input
output → cell state → remove

(3x1) ⊗ (3x1)

→ 1) Calculate f_t

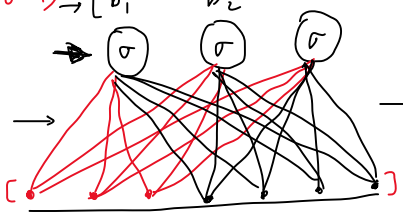
2) $C_{t-1} \otimes f_t$

σ σ σ

f_1 f_2 f_3 → $[f_1\ f_2\ f_3]$ → $f_t(3)$

b_1 b_2 b_3 (3x1) (3x7) (7x1)

→ (3x1) + (3x1) = (3x1)



→ W_{3x7} (21 weights) 21 wts

biases 4 biases

h_{t-1} (3) x_t (4) → 7x1

forget gate ← 0%
50%
100%

remove

$[h_{t-1}, x_t]$

long term

f_t

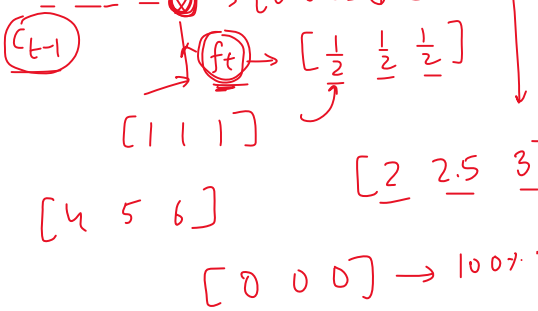
$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$

(3x1) (3x7) (7x1) (3x1) + (3x1) → (3x1) → 3x1

removal

$f_t \otimes C_{t-1}$ → (3x1)

(C_{t-1}) → remove



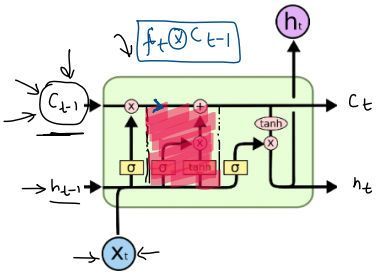
[4 5 6]

[0 0 0] → 100% remove

The Input Gate

30 August 2023 04:38

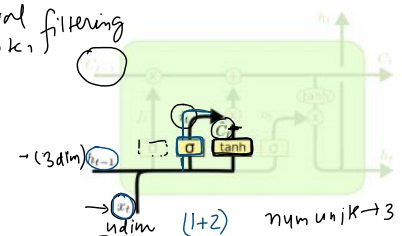
add some new imp info to ml C_t



Stages

1) $\tilde{C}_t$ candidate cell state
2) i_t
3) C_t → cell state current

potential filtering important info



(1+2) number of units → 3

$\tilde{C}_t = (3 \times 1)$

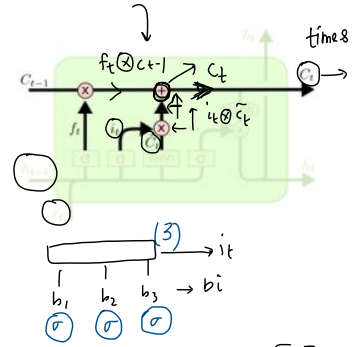
b_1 b_2 b_3

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

W_C (3 x 7)

(3x7) (7x1) (3x1) (3x1) (7x1) → (3x1)

h_{t-1} x_t



timestep

pointwise $i_t \otimes \tilde{C}_t \rightarrow \tilde{C}_t$ (filtered candidate cell state)

(3x1) (3x1)

[0.5 0.5 0.5] [4 5 6]

1 1 1 ↓ 50% 0% 100%

[2, 2.5 3]

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

(3) → i_t

b_1 b_2 b_3 → b_i

σ σ σ

→ [ ] W_i

h_{t-1} x_t

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

(3x7) (7x1) (3x1) (3x1) σ(3x1) → (3x1)

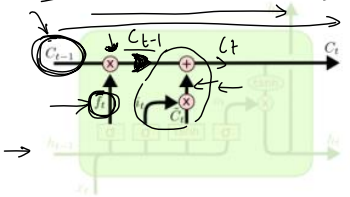
i_t - filter

$\tilde{C}_t \rightarrow C_t$

by

[4 5 6] → [4 5 6]

t = t-1 t = t



$f_t = [1\ 1\ 1]$

$i_t \otimes \tilde{C}_t = [0\ 0\ 0]$

[4 5 6] [0 0 0]

$C_t = [4\ 5\ 6]$

long term context

C_t C_t

The Output Gate

30 August 2023 05:27