Long Short Term Memory(LSTM)

Dr. Muhammad Safyan

- LSTM

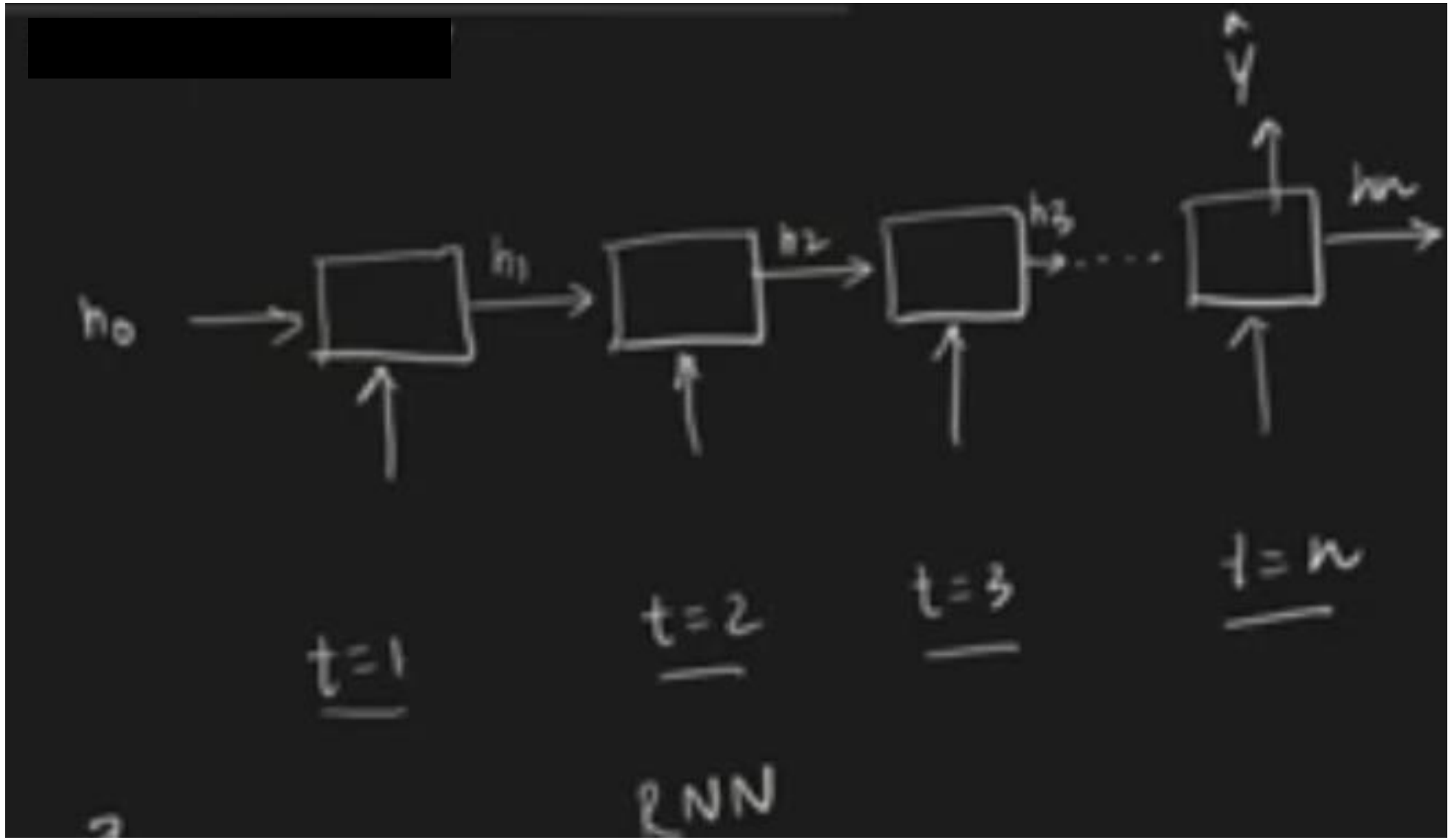
An Example

- Lahore, the cultural heart of Pakistan, is a vibrant city known for its rich history, artistic heritage, and warm hospitality. As the capital of Punjab province, Lahore beautifully blends tradition with modernity — ancient Mughal architecture stands proudly beside bustling markets and contemporary buildings. The city is home to magnificent landmarks like the **Badshahi Mosque**, **Lahore Fort**, and the **Shalimar Gardens**, which reflect its glorious past under Mughal rule. Lahore is also famous for its lively **food streets**, where the aroma of spicy kebabs, biryani, and naan fills the air, making it a paradise for food lovers. The city's educational institutions, such as the **University of the Punjab** and **Lahore University of Management Sciences (LUMS)**, make it a center of learning and innovation. Above all, Lahore's people are known for their friendliness, passion for cricket, and deep love for art, music, and poetry — making it a city that truly embodies the spirit and soul of Pakistan. Language spoken is ----------

Long Term Dependency Problem

- If the chain is long, it forgot the initial context.
- Because of Vanishing/Exploding Gradient Descent.

LSTM Core Idea



A Hypothetical Story

- Is the story was good or bad
- Here is core idea of LSTM

There are two context

  - Long Term Context LTC
  - Short Term Context STC

- You build Long Term context

from short Term context.

1000 year ago, there was once a noble lion king named **Leo** who ruled a peaceful jungle. Leo was brave and strong, yet gentle and kind to all the animals who lived under his protection.

Nearby, in another jungle, ruled **Ragnar**, a fierce and cruel lion king. Ragnar often invaded Leo's land, harming the innocent animals and spreading fear.

One day, a great battle took place between Leo and Ragnar. Sadly, Leo was killed while defending his kingdom. Leo's son, **Arion**, was heartbroken. Determined to avenge his father, Arion prepared himself to face Ragnar. He was just as courageous and kind as his father. But fate was cruel, and in the battle that followed, Arion was also slain.

Years passed, and Leo's youngest grandson, **Zane**, grew up hearing stories of his heroic ancestors. Unlike Leo and Arion, Zane was not very strong or fearless — but he was extremely clever. When the time came, Zane confronted Ragnar. Using his intelligence instead of sheer strength, he outsmarted the wicked lion and finally defeated him, bringing peace back to the jungle.

RNN

- RNN does not has the capability to memorize the things or not capable enough to maintain the state of the things.
- With Normal RNN, short term context dominant on long term context.
- Only recent talk keep in memory.

LSTM Core Idea

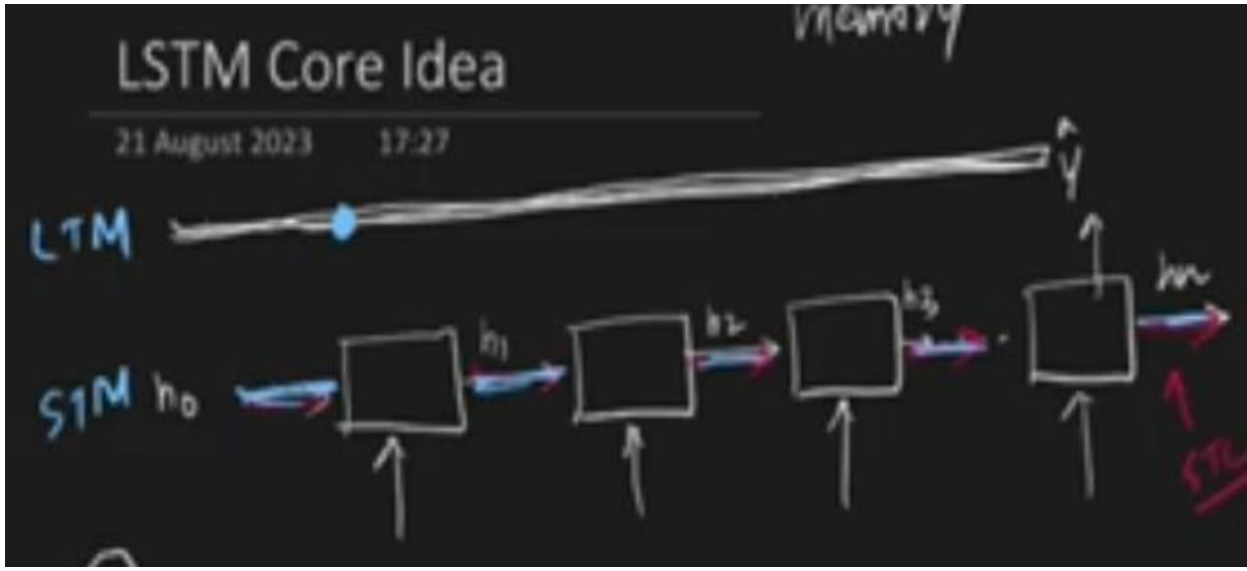
- Maintain long term memory path with Short Term memory Path.
- Amina is a great girl, she/he …….
- Amina & girl are two important things. So add them in LTM.
- Agha ali is youtuber, he makes several video on data science
- When agaha ali came in, model

Remove the amina & girl from LTM

Nutshell:

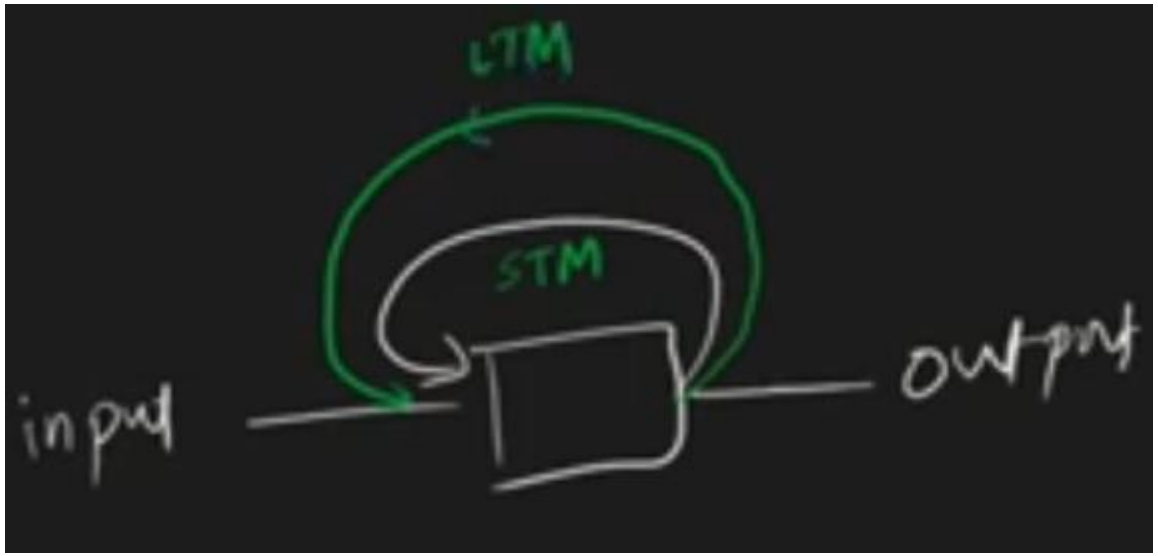
You add a Long Term Memory path

Along with Short Term Memory path



LSTM

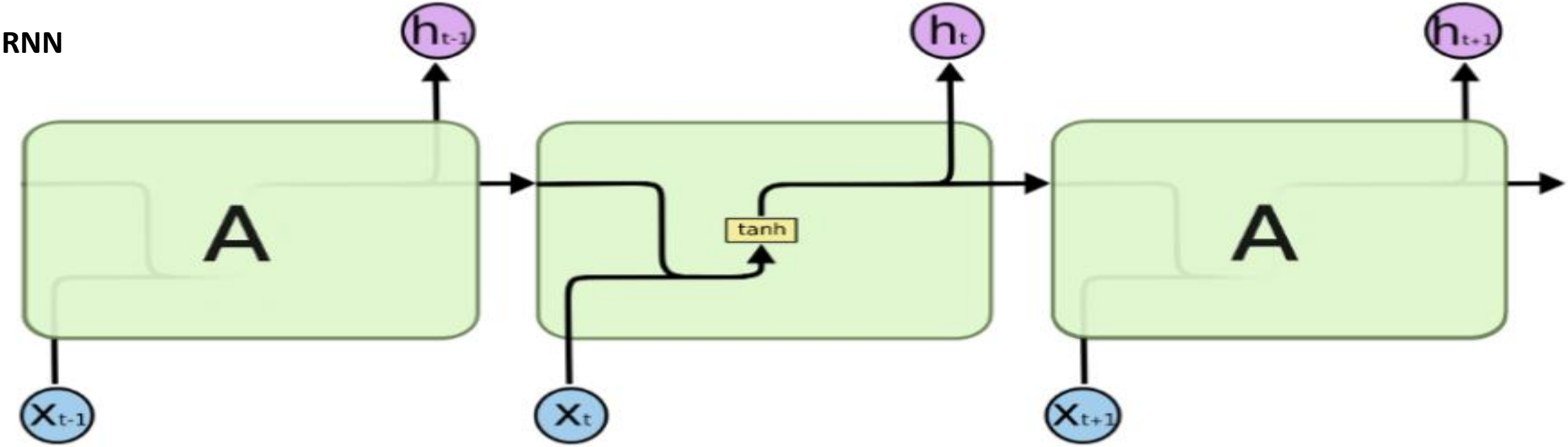
- Things remain in LTM , If they are relevant and even the end of talk.
- **1st difference**



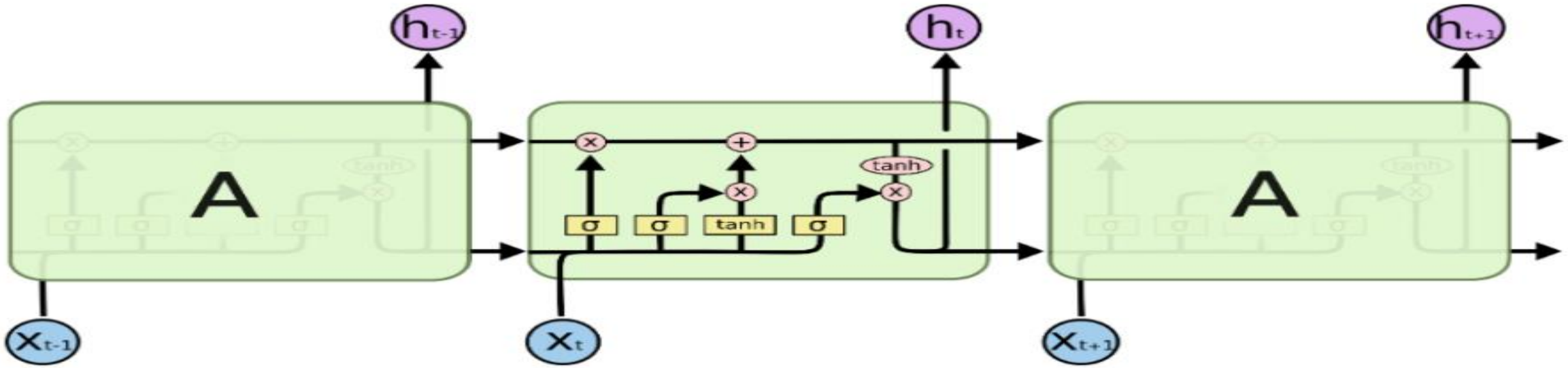
LSTM

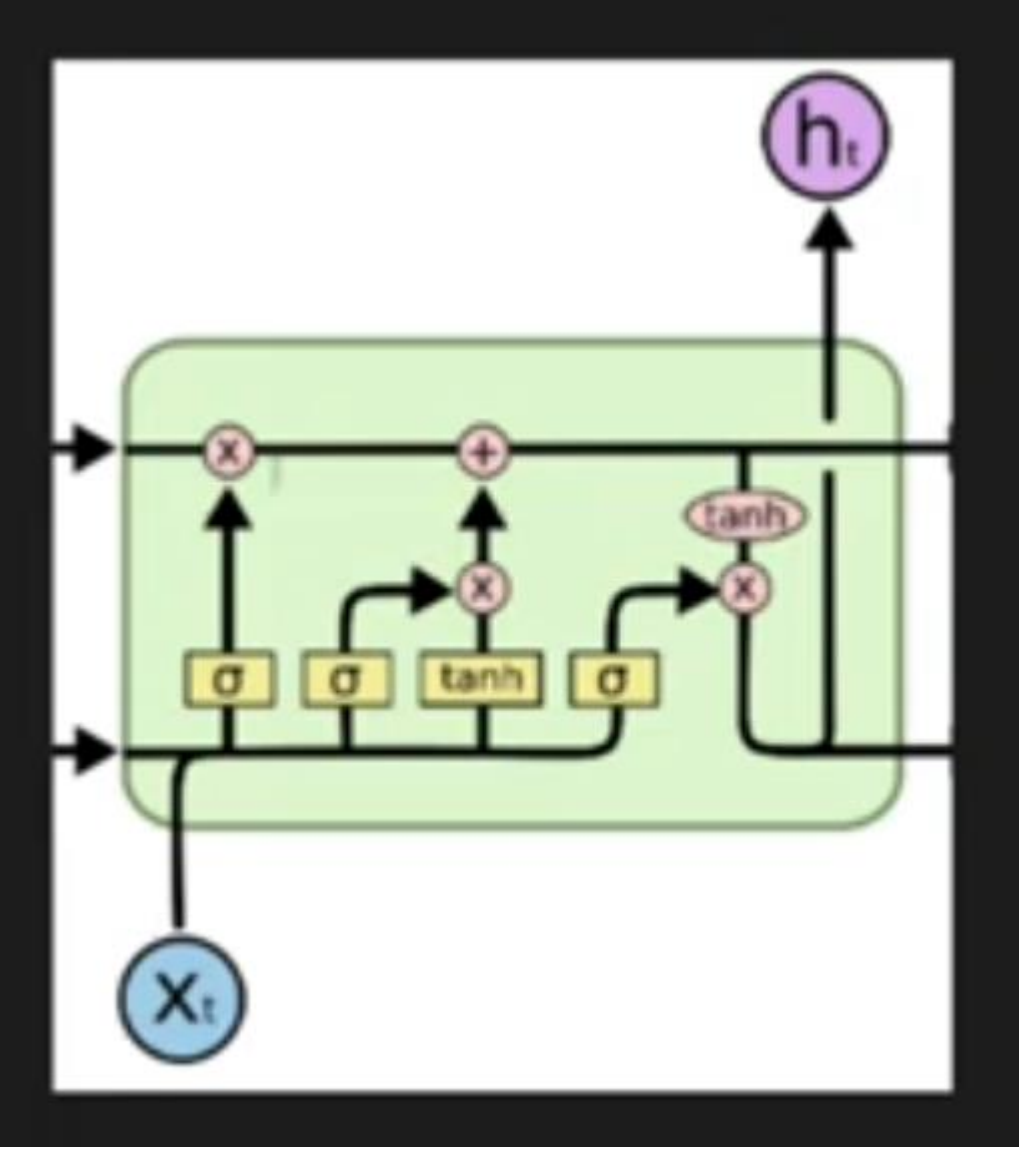
- **2nd Difference**
- RNN is simple but LSTM is difficult one. You need to maintain a link/relation/chat between the two states.
  - For this , you need a architectural change.

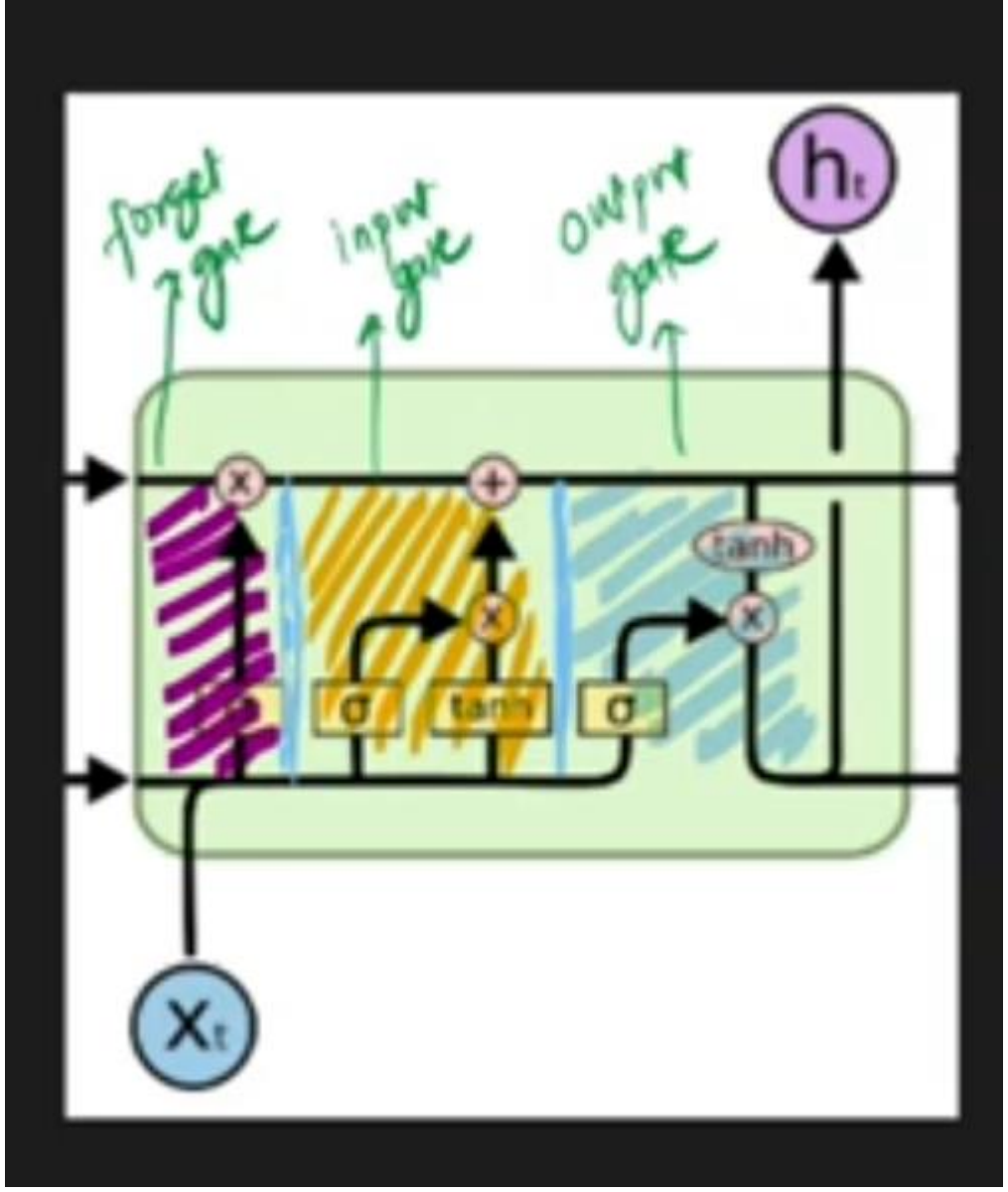
RNN


h_{t-1}
h_t
h_{t+1}
A
tanh
A
X_{t-1}
X_t
X_{t+1}

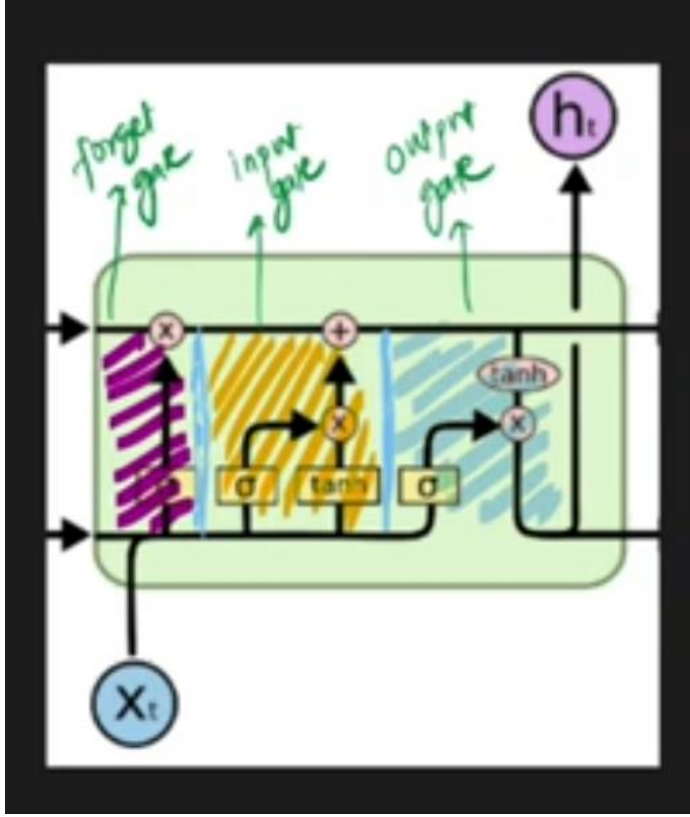
LSTM


h_{t-1}
h_t
h_{t+1}
A
tanh
σ
σ
tanh
σ
A
X_{t-1}
X_t
X_{t+1}


h_t
tanh
σ
σ
tanh
σ
X_t


forget gate
input gate
output gate
h_t
tanh
σ
tanh
σ
X_t

Gates Over View



- Based on current input +LTM,
- **Forget Gate:** which one is to remove from LTM and
- **Input Gate:** decide which is to be add in LTM.
- **Output Gate:** decide which to show as output form LTM

e.g. story is good or bad.

And clear the STM.

Summary: LSTM take three things as input:

(i)At time t a cell state ct-1 and

(ii)short tem memory ht-1 and

(iii)the current work xt

And there are two thing as output :

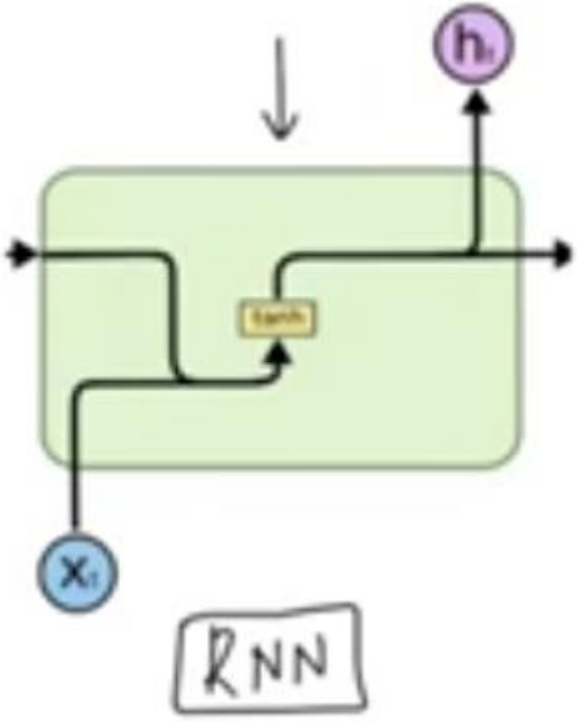
(i) long term memory ct and

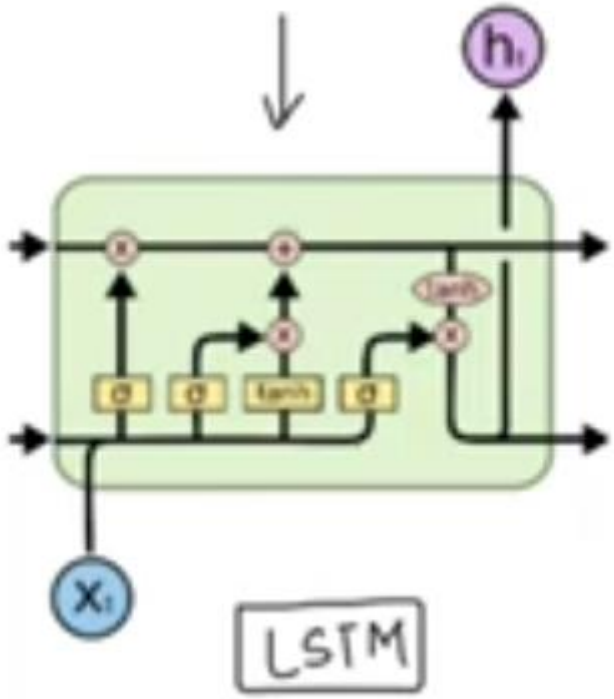
(ii) (ii) short term memory ht

- In middle you update the memory LTM and you create STM

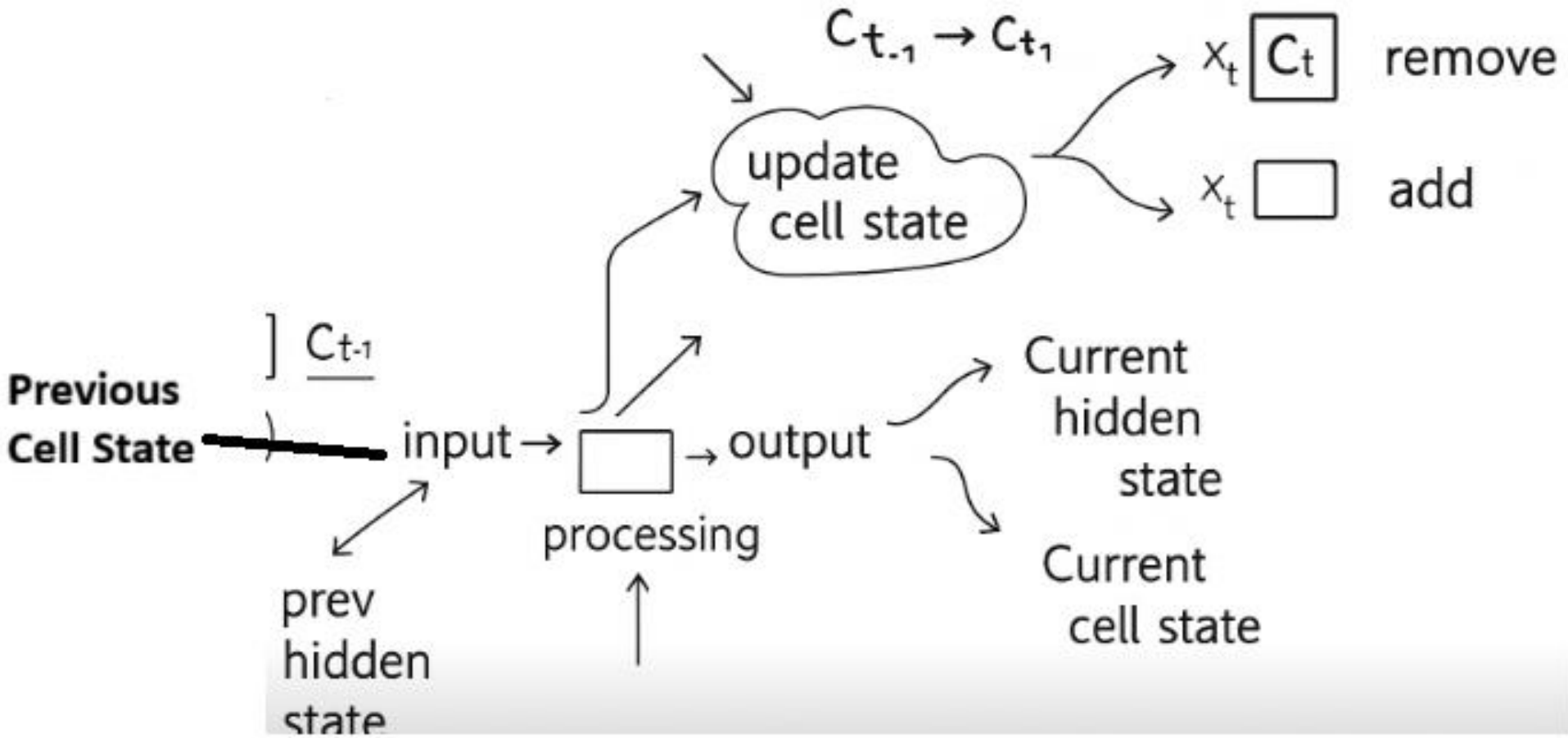
Architecture RNN vs. LSTM

- LTM= Cell state
- STM=hidden state

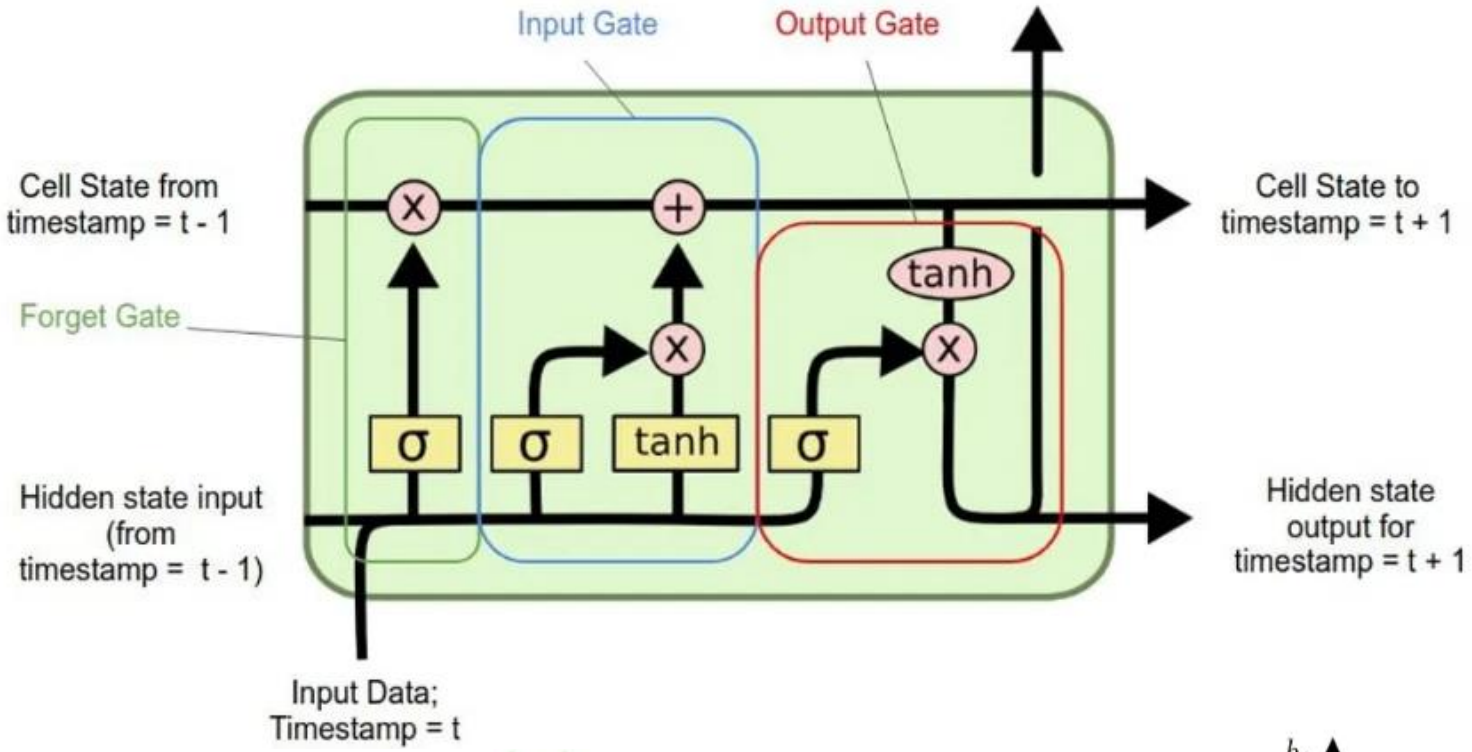


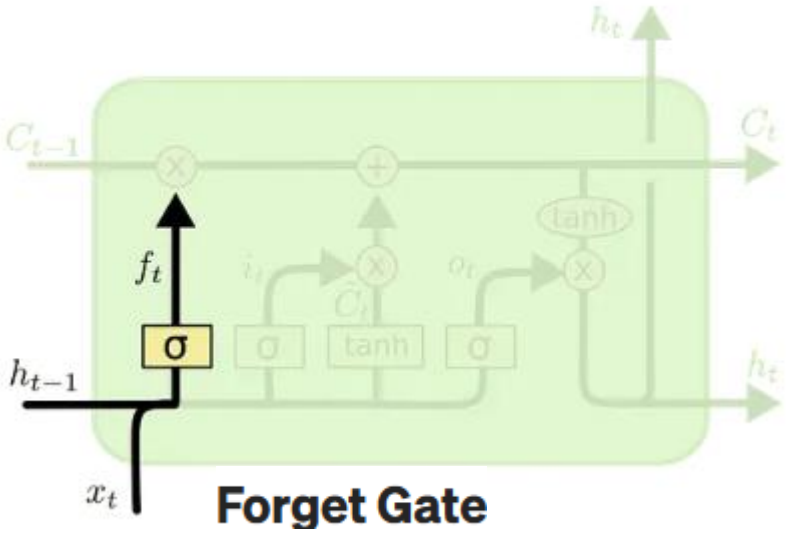


Simple Architecture

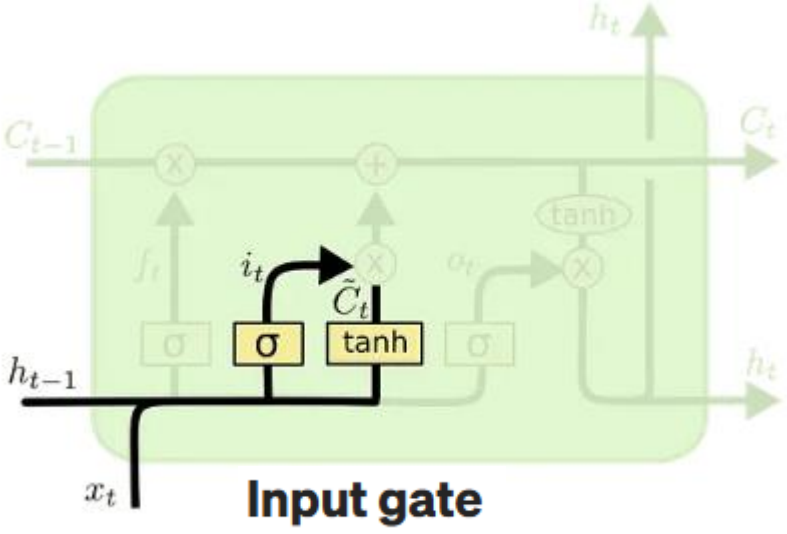


The Gates

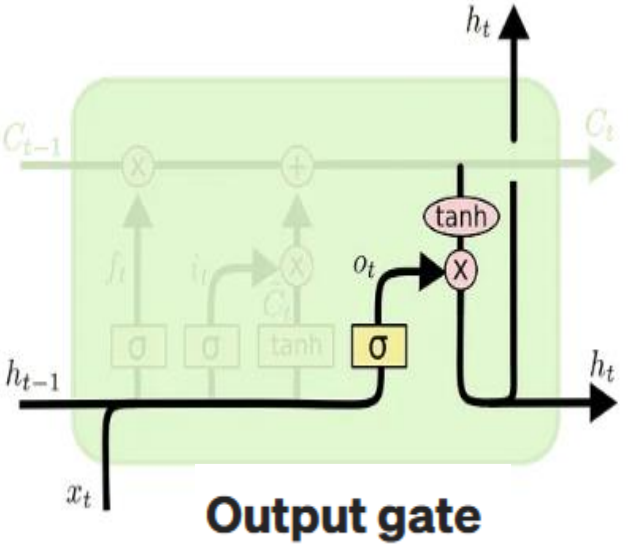




Remove something form Ct



To add something in Ct



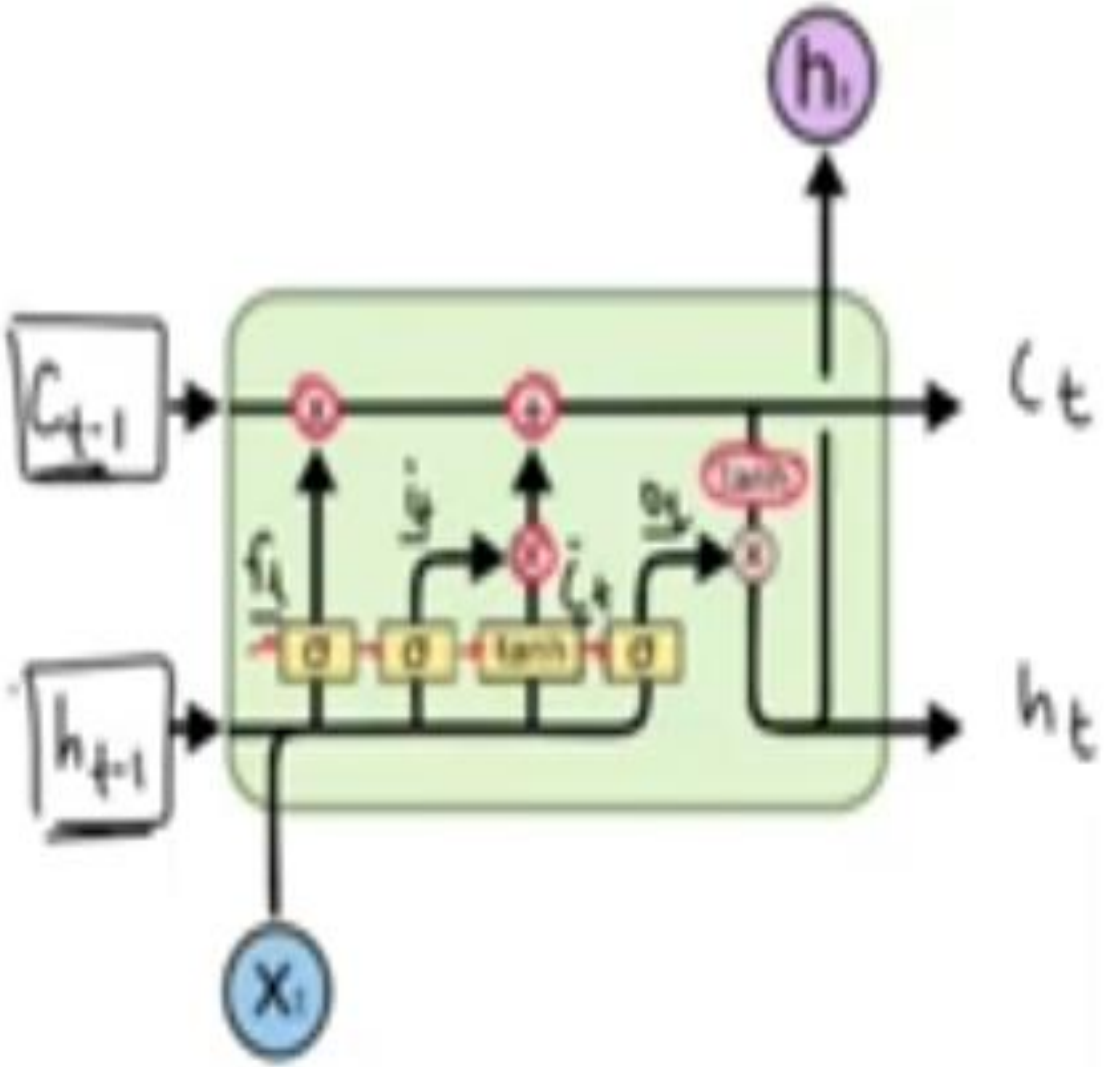
To calculate something in ht

What are Ct and ht

- Ct and ht are vectors
- And their dimension are same
- Xt can be of any lenght

F1	F2	F3	ouptut
Cat	mat	rat	1
rat	rat	mat	1
mat	mat	cat	0

t=1 t=2 t=3



What are ft, It, Ot, Ct

- Three gates:
- Forget gate: ft
- Inptut gate it
- Candidate cell state: Ct
- Output gate : Ot

Forget gate

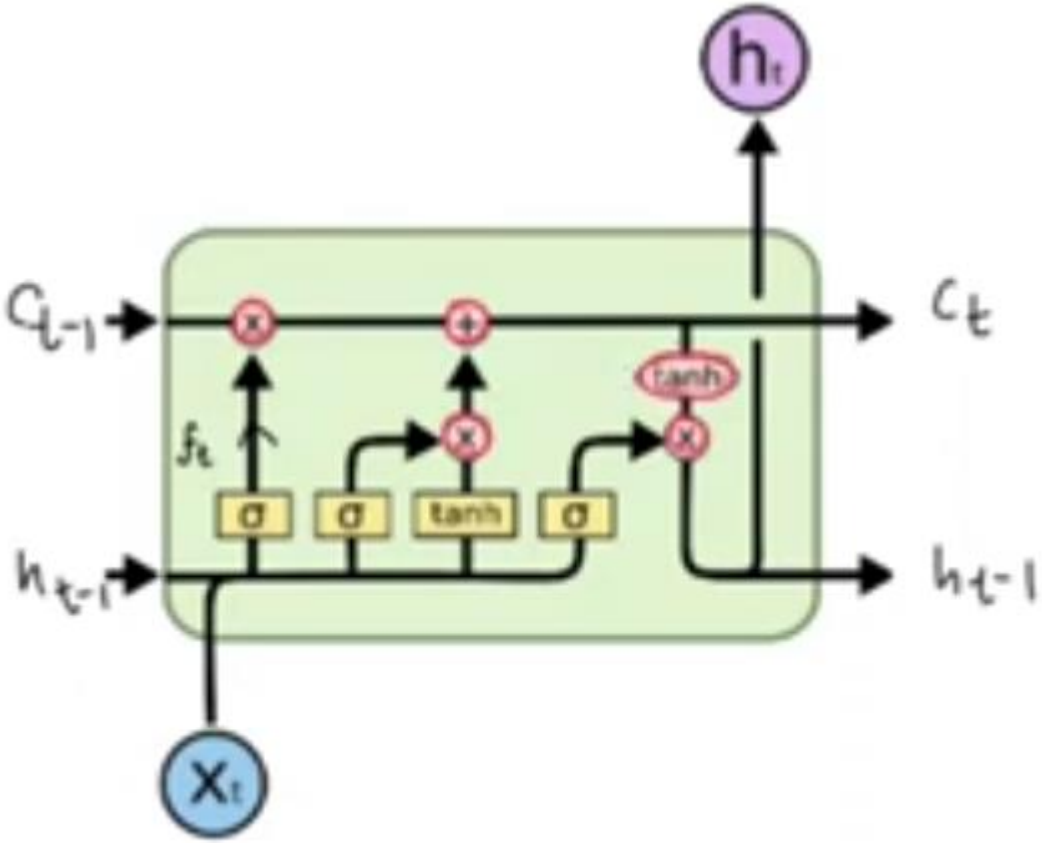
Input gate

Output gate

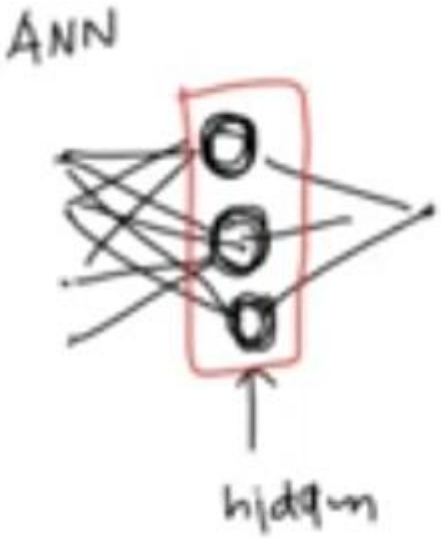
- Vector dimensions Ct, ht,Ft, It, $\check{C}$t, Ot are same

Point wise Operations

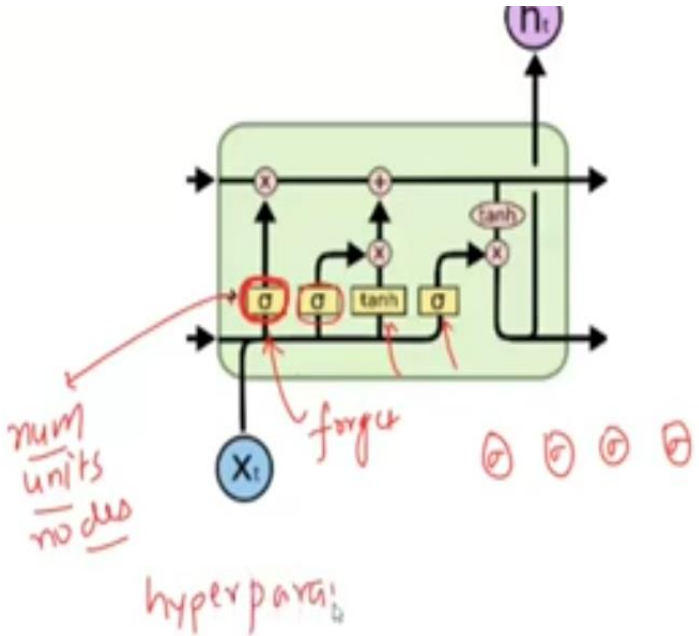
- ⊗ element wise multiplication
- Ct=[4,5,6]
- Ft=[1,2,3]
- Ct ⊗ ft =[4, 10, 18]
- Similarly point element wise addition
- tanh([4,5,6])
- =0.25, .31, .4

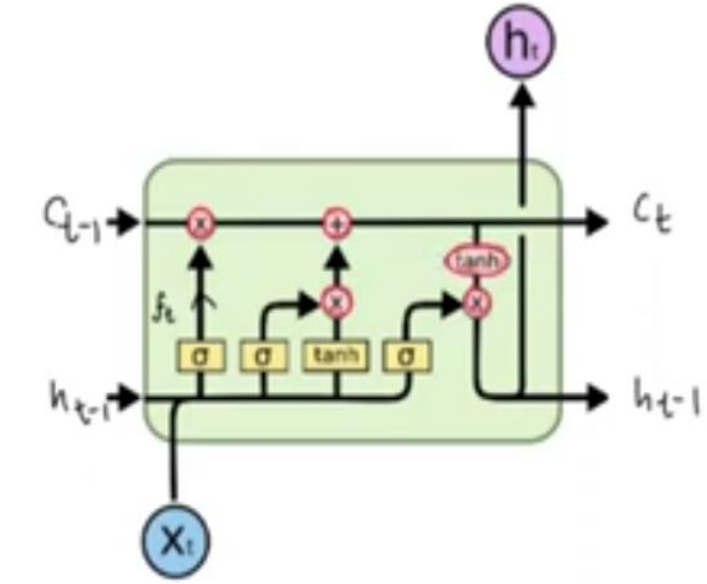


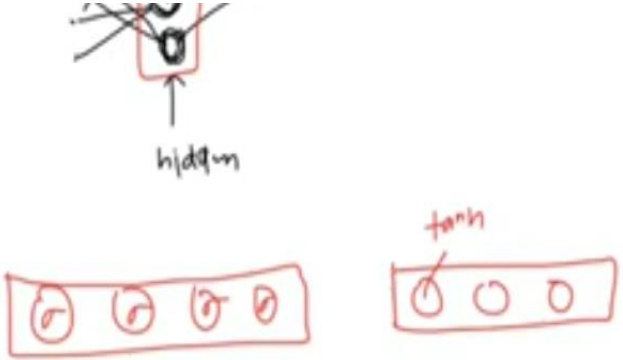
Yellow boxes



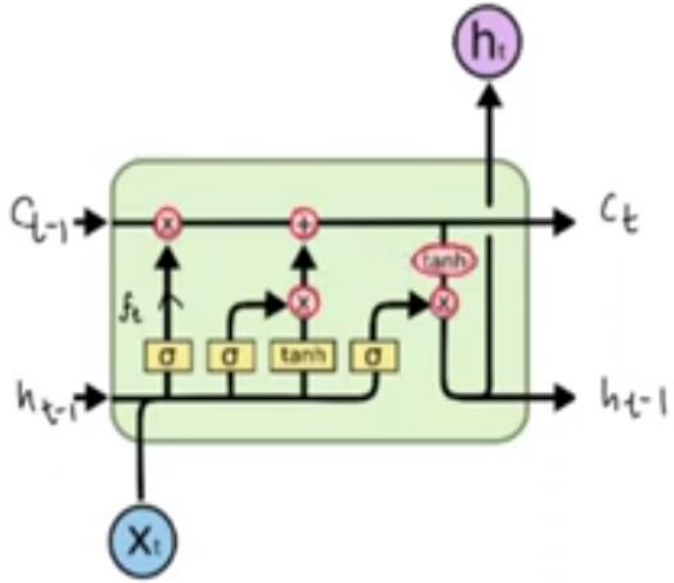
- # of Neurons will be
- Same in all.



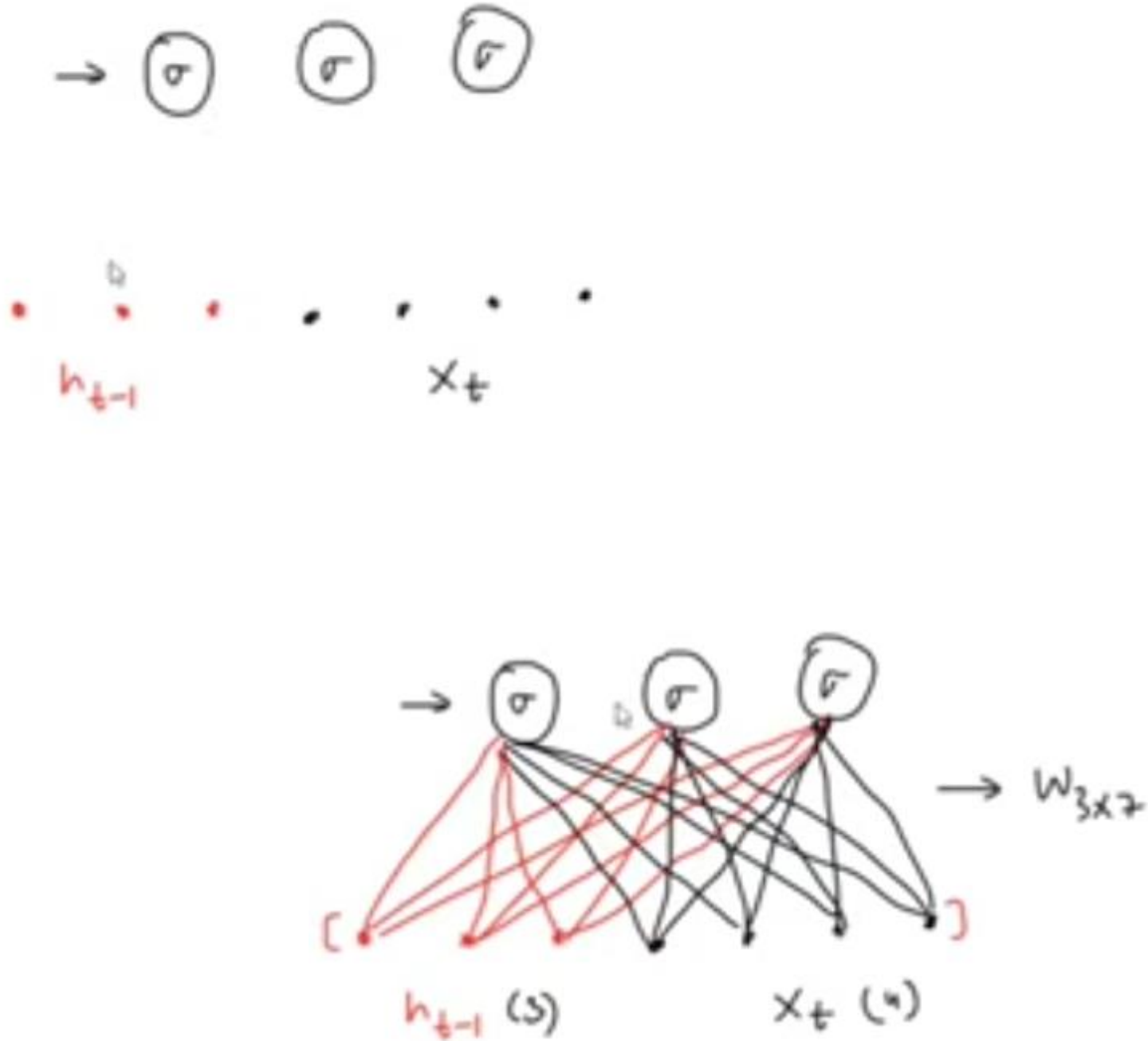




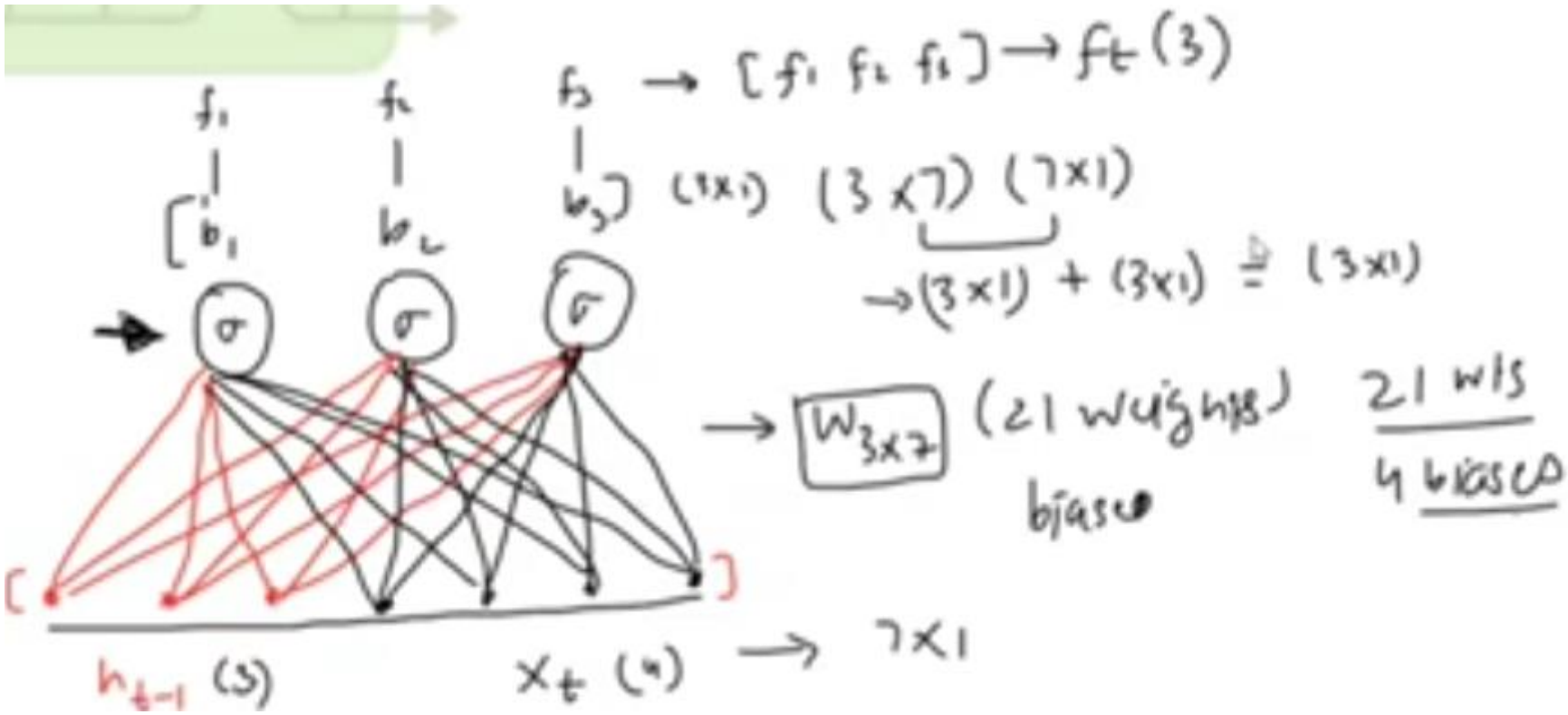
The Forget Gate



- Assume there are 3 neurons in yellow box.
- That why Ct-1 and ht-1 will be of 3 dimensions
- Reason is that they must of the same dimensions
- Lets, Xt is input and 4 dimensions
- X=[xi1, xi2, xi3, xi4]
- To calculate the Forget Gate , following steps are taken:
- 1) Calculate ft
- 2) calculate ft*Ct-1


σ
σ
σ
h_{t-1}
x_t
σ
σ
σ
W_{3x2}
h_{t-1} (3)
x_t (2)

Forget Gate


f_1 f_2 f_3
b_1 b_2 b_3
σ σ σ
h_{t-1} (3)
x_t (4)
$[f_1\ f_2\ f_3] \rightarrow f_t$ (3)
(3x1) (3x7) (7x1)
(3x1) + (3x1) ≈ (3x1)
W_{3x7} (21 weights)
bias
21 wts
4 biases
7x1

Forget Gate

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

f_t: 3x1

W_f: (1x7)

$[h_{t-1}, x_t]$: 7x1

$W_f[h_{t-1}, x_t]$: (1x7) · 7x1 → (3x1)

b_f: (3x1)

(3x1) + (3x1) → (3x1)

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

$f_t \otimes C_{t-1} \rightarrow (3x1)$

Resultant is removed Ct-1

Ct-1=[4,5,6]

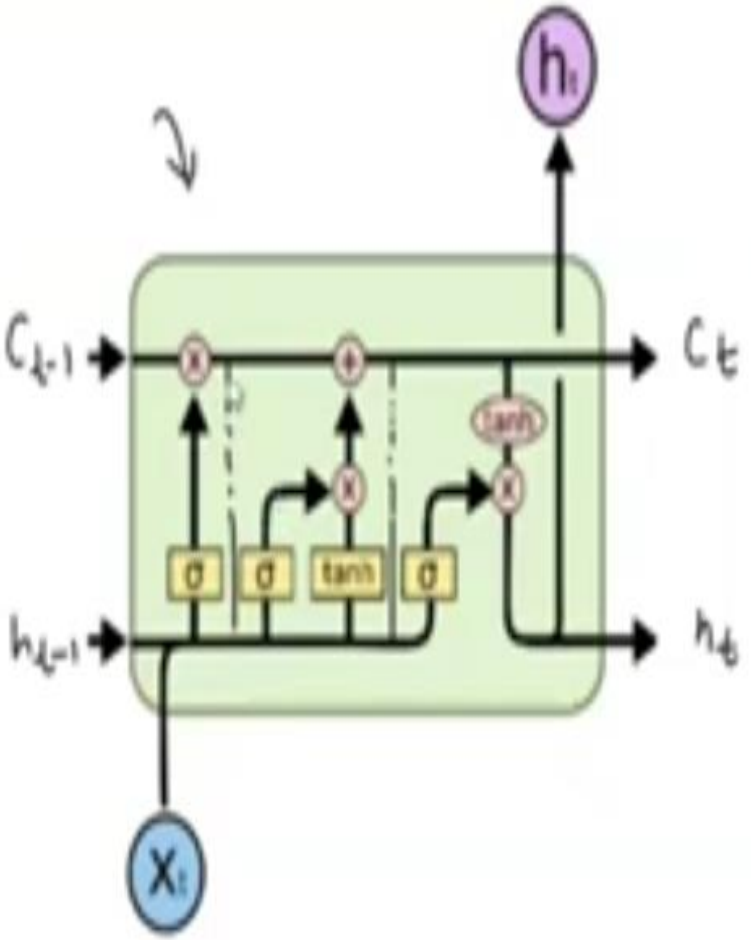
Ft=[1,1,1]

Ft=[0,0,0]

Ft=[.5, .5, .5]

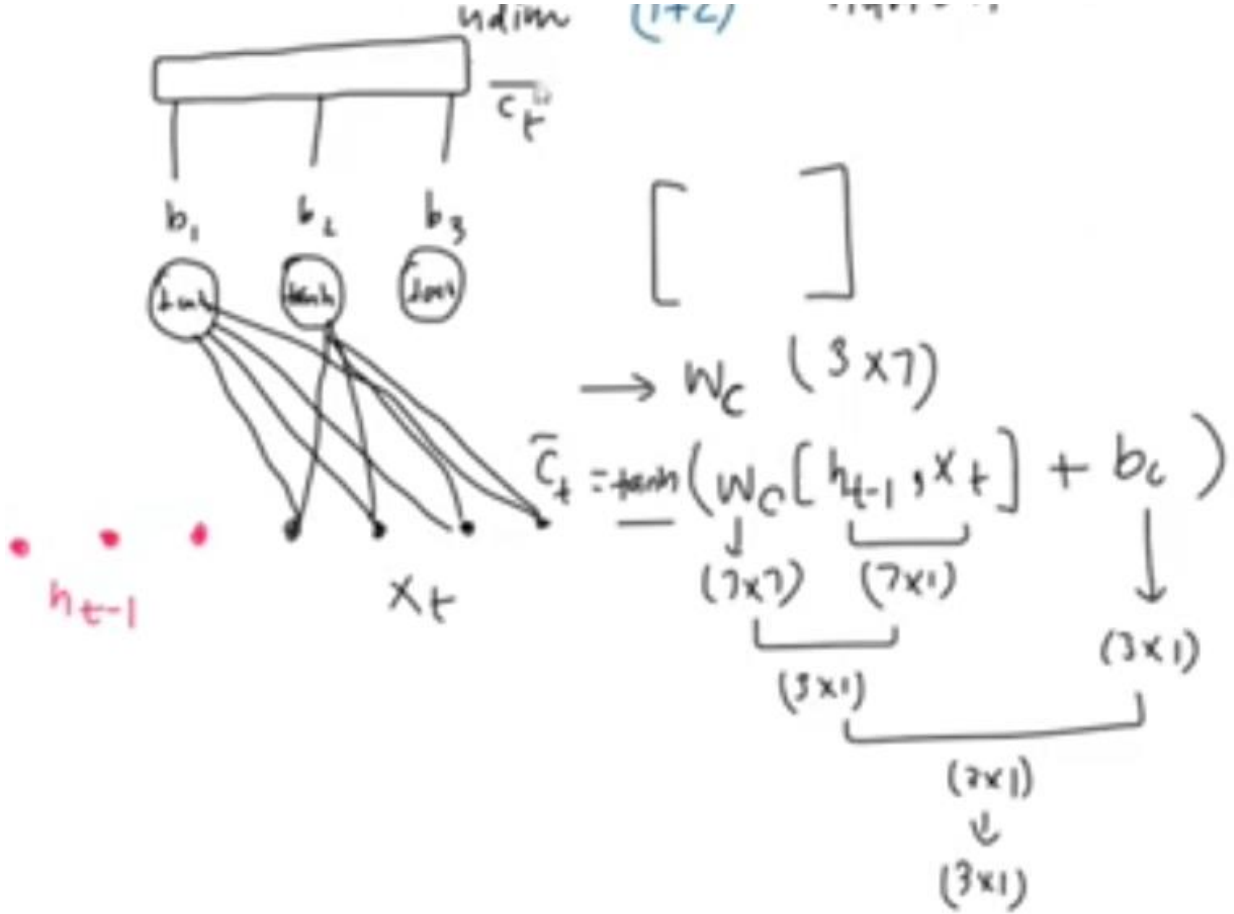
InPut Gate:

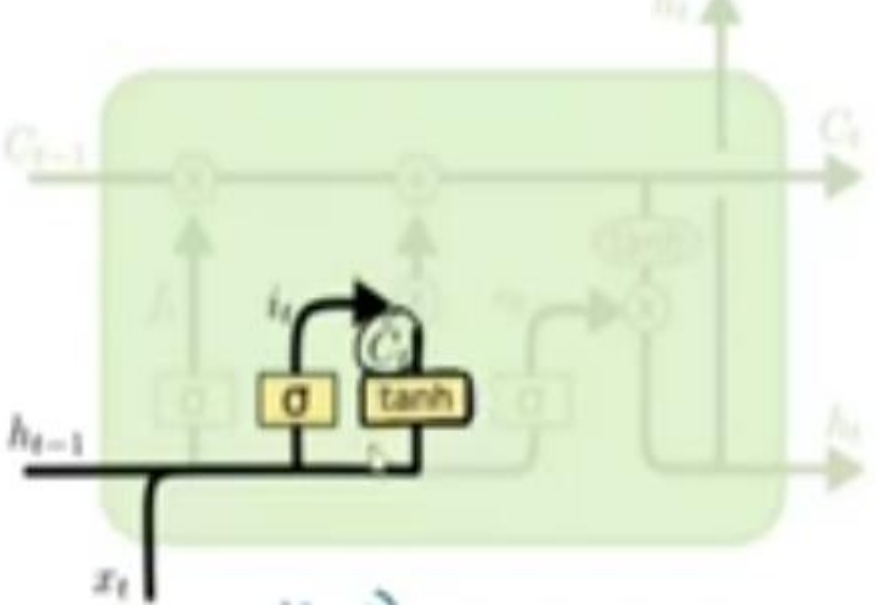
- Purpose:
  to add some new information into the cell state Ct.
- Works in Three stages:
  - Candidate cell state , $\check{C}t$
  - Calculate It. That further filter the info from $\check{C}t$.
  - Calculate Cell state Ct



Calculate Candidate Cell State

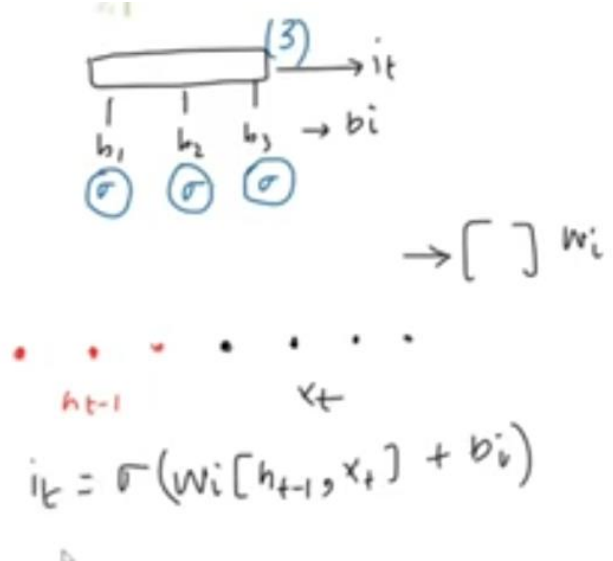
- Outr xt= 4 dim and ht-1 is 3 dim





Candidate cell state hold the potential important information.

It→ is filter: further , filter information from the Ct and will add that info in context.



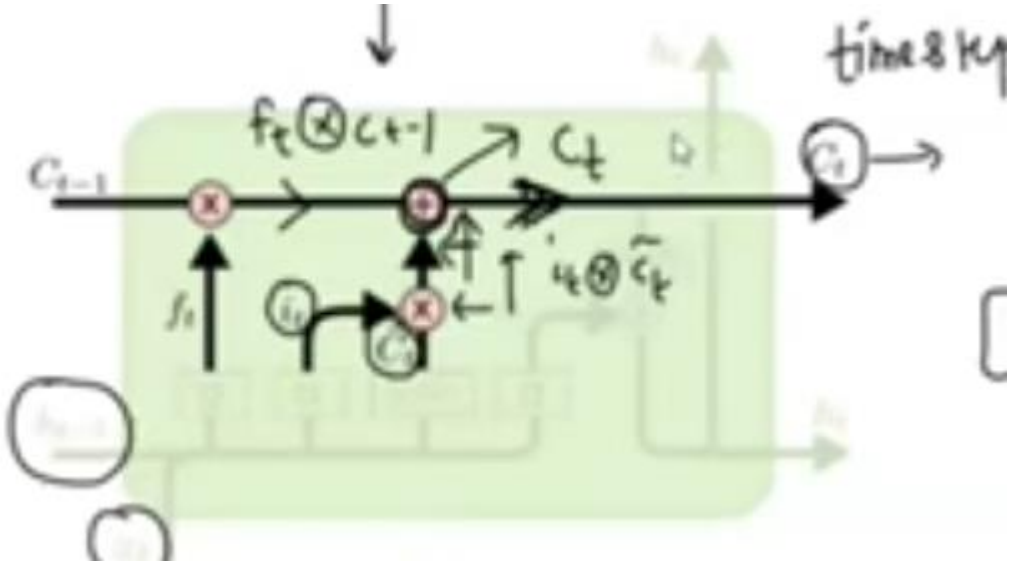
Calculate Candidate Cell State

- Point wise operation: $i_t \otimes \bar{C}_t$

pointwise

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

(3x1) (3x1)



[0.5 0.5 0.5] [4 5 6]

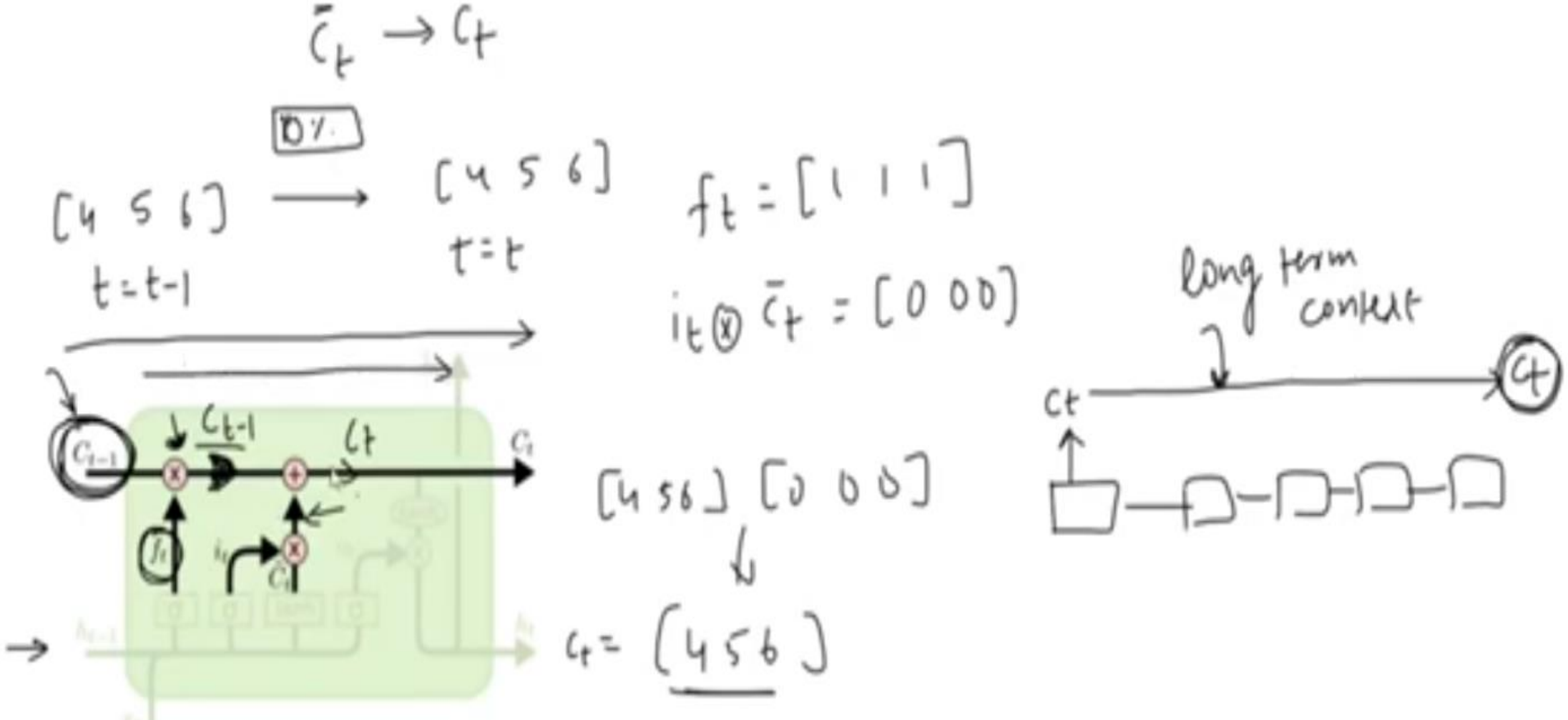
1 1 1 ↓ 50% 0% 100%

[2, 2.5 3]

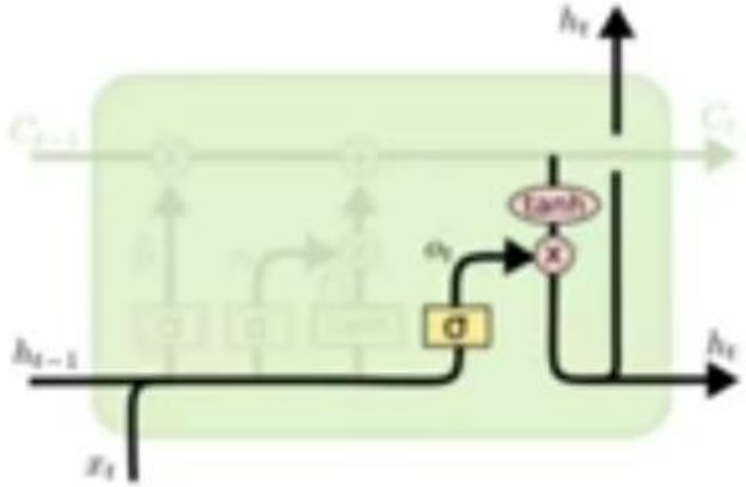
New Ct:

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

Long Term Context



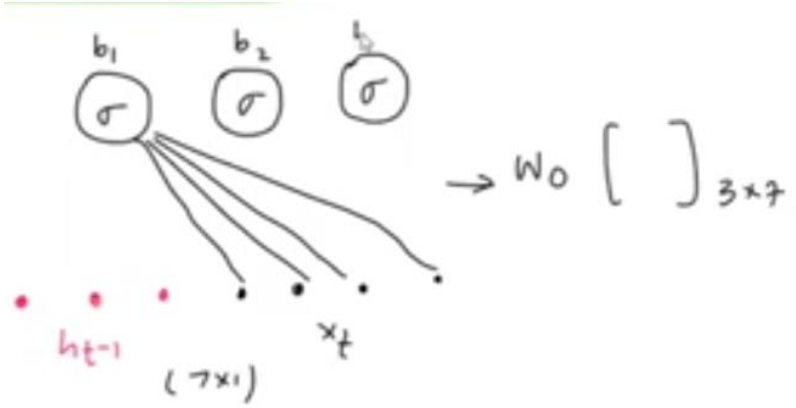
Output Gate



Calculate the next hidden state: ht
You applied some mathematical operation on Ct and extract ht
There are two steps:
Calcualte tanh(Ct) → all the value in -1, 1.
Calcuate Ot
The dot point wise

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

Ot has tw input ht-1, xt



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

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

https://www.reddit.com/r/learnmachinelearning/comments/smindi/lstm_visualized/