LSTM (Long Short Term Memory)

Contents

THEORY OF LSTM

Background:

Traditional RNNs suffer from the vanishing and exploding gradient problem, which makes them less effective in learning and retaining information over long sequences. This limitation occurs because they have difficulties in capturing long-range dependencies due to the nature of their architecture.

Structure of LSTM:

LSTM networks are composed of units called cells, and each cell has several components that help in retaining and forgetting information selectively. The key components of an LSTM cell are:

1. **Cell State (Ct):** This represents the memory of the cell and runs straight down the entire chain of LSTM units, allowing information to flow through the cell unchanged.
2. **Hidden State (ht):** This is the output of the LSTM cell and contains information about the current input, previous hidden state, and cell state.
3. **Gates:** LSTMs use three gates to control the flow of information:
    - **Forget Gate (ft):** Decides what information to discard from the cell state.
    - **Input Gate (it):** Decides what new information to store in the cell state.
    - **Output Gate (ot):** Decides what information to output based on the cell state.

Operations within an LSTM cell:

1. **Forget Gate (ft):**
    - Calculates the relevance of the old cell state (Ct-1) to the new cell state.
    - Uses a sigmoid activation function to output values between 0 (completely forget) and 1 (completely keep) for each element in the cell state.
2. **Input Gate (it):**
    - Determines what new information to store in the cell state.
    - Uses a sigmoid activation function to decide which values will be updated and a tanh function to create a vector of new candidate values ($\tilde{C}t$).
3. **Update Cell State:**
    - The new cell state (Ct) is computed by removing the parts forgotten by the forget gate and adding the new candidate values scaled by how much they need to be added.
4. **Output Gate (ot):**

- Determines what parts of the cell state will be output as the hidden state.
- Uses the sigmoid function to decide which parts of the cell state to output and then applies the tanh function to the cell state to push the values to be between -1 and 1.

Advantages of LSTMs:

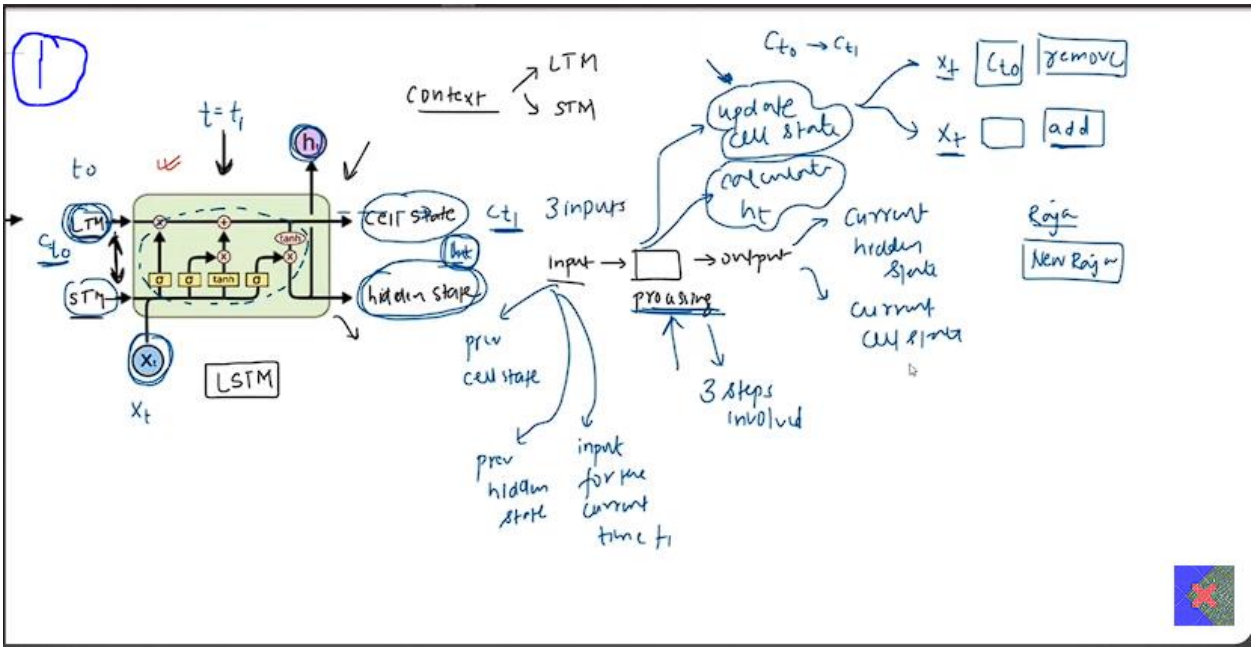
1. **Long-Term Dependencies:** LSTMs can handle long sequences and remember information over long periods, making them suitable for tasks involving time-series data, natural language processing (NLP), and more.
2. **Reduced Vanishing Gradient:** The gated structure of LSTMs helps in mitigating the vanishing gradient problem by selectively retaining and updating information.

LSTMs have been widely used in various fields such as machine translation, sentiment analysis, speech recognition, and more due to their ability to capture long-term dependencies and effectively model sequential data.

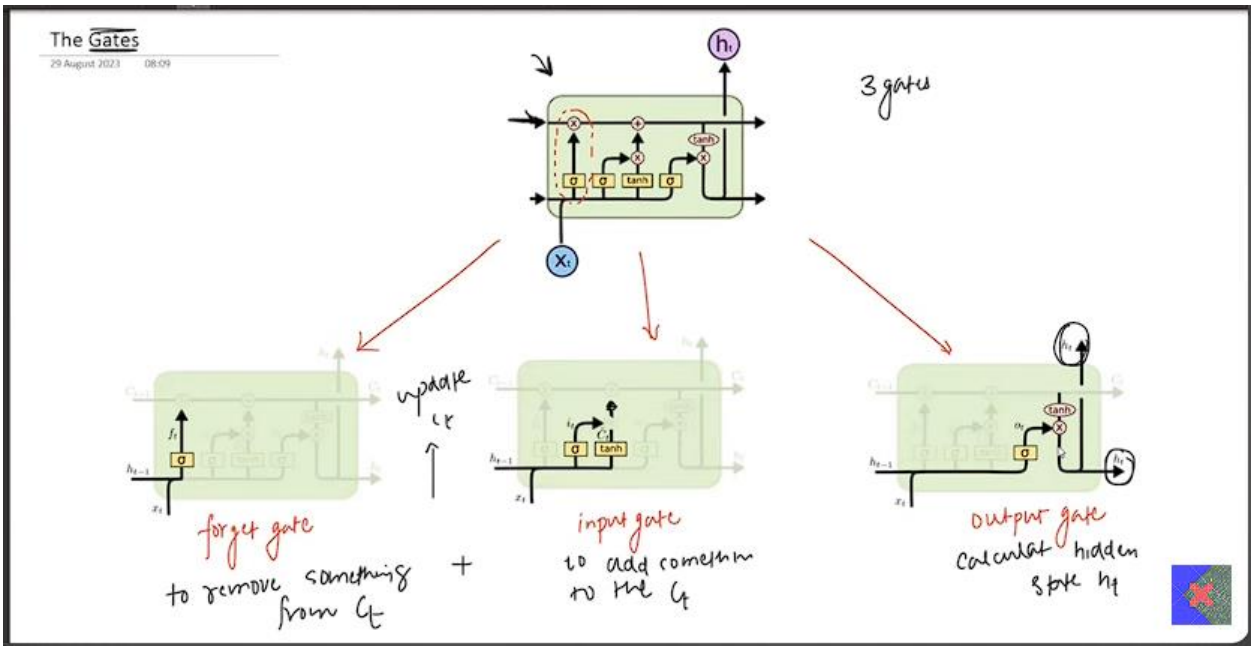
Overview

How it works overall:

3 inputs: current input (Xt), previous cell state (Ct-1 /LTM), previous hidden state (ht-1/STM)
2 outputs : current hidden state (ht/ updated STM), current (Ct/LTM).

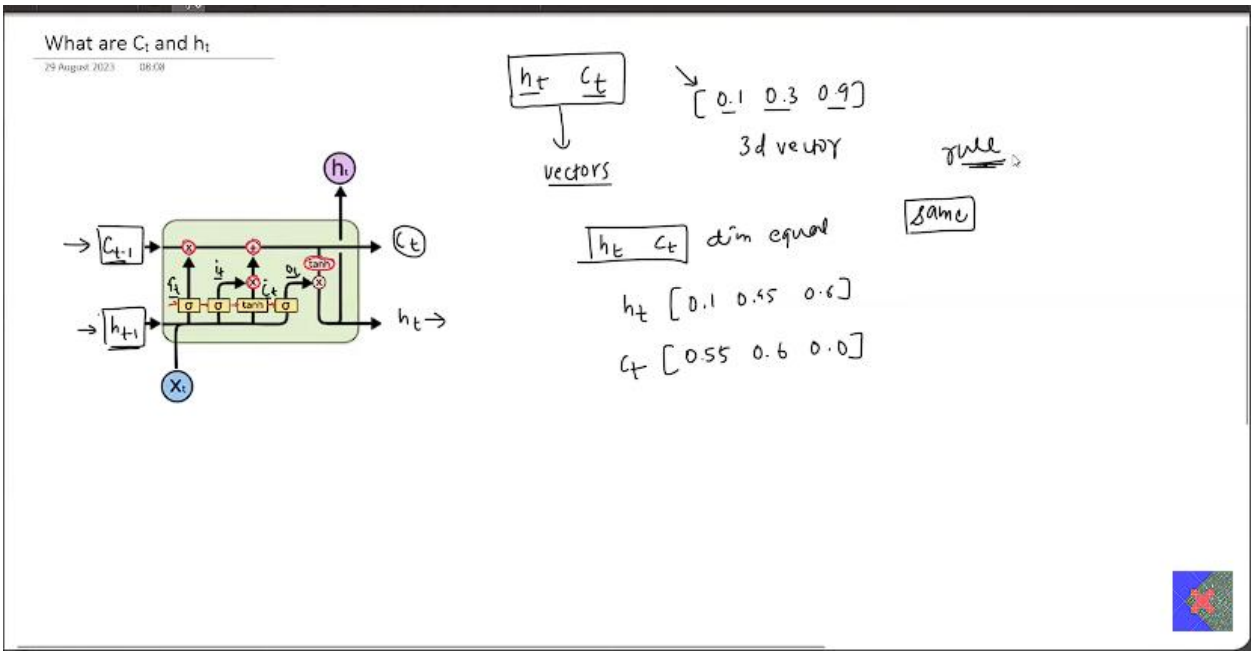


GATES:



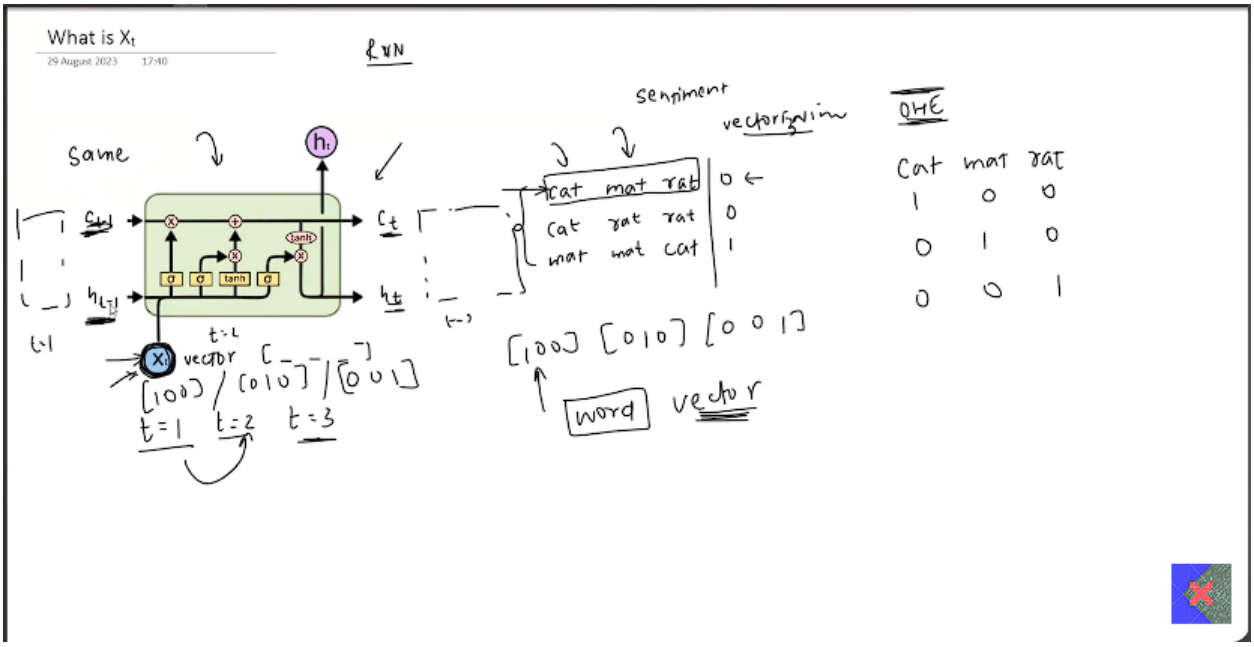
What are Ct and Ht, how they looks

These are vectors and shape of both will always be same.



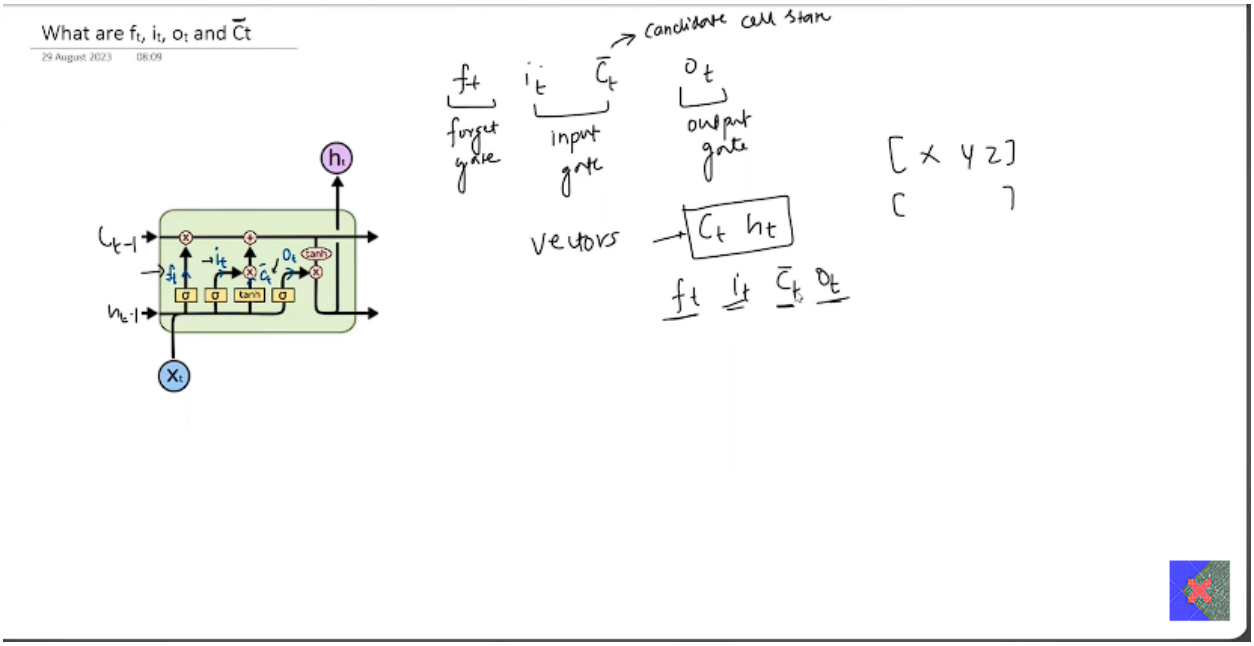
What is Xt:

Input how it is calculated?



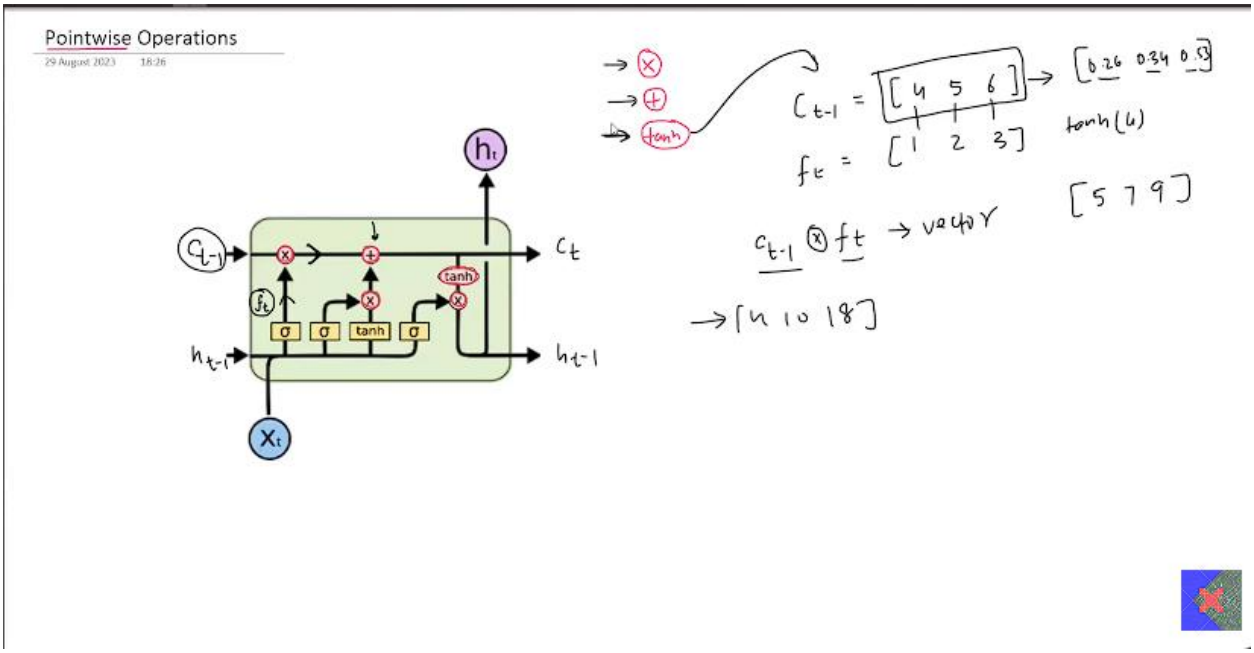
Ft, it, ot and ct:

All are vectors of same dimension/shape



Point wise Operation (+, *, tanh):

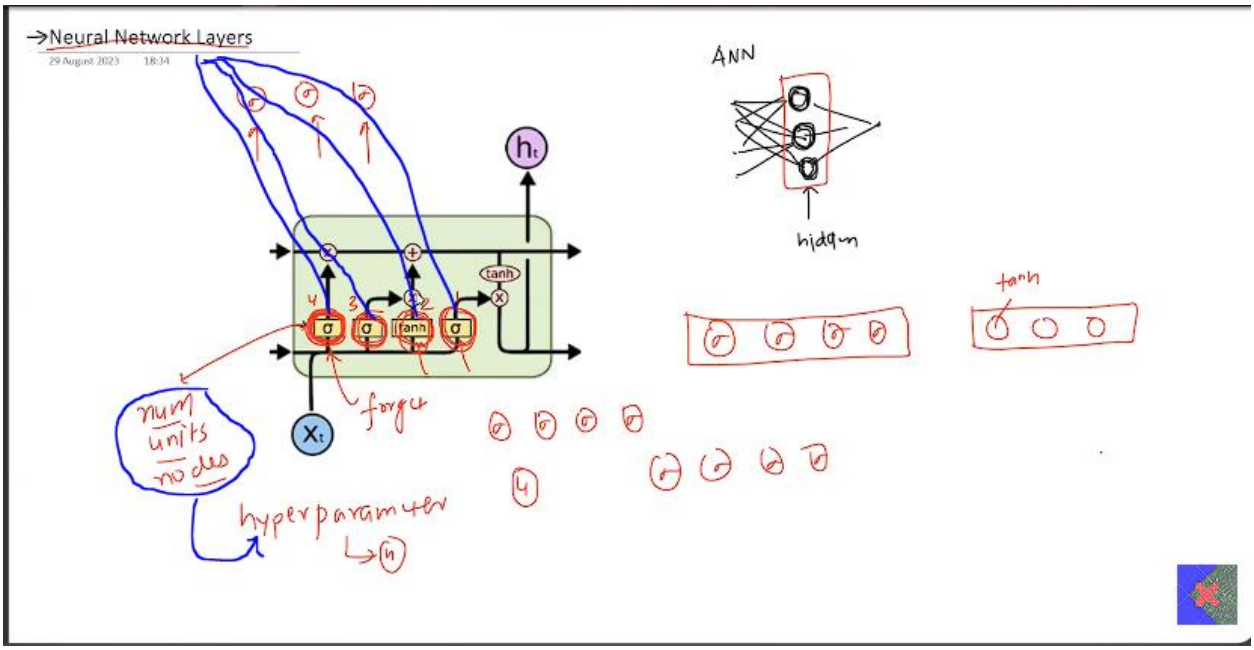
Always btw vectors of same share/dimensional.



NN Layers:

Can have many layers in each of 4 networks used in it,
three of those 4 uses (sigmoid activation function) and one is using (tanh activation function),
Number of neurons is a hyper-parameter and we can choose, but all network must have same number of neurons.

Also the number of neurons/units and the shape of the ct and ht will be same.



THE ARCHITECTURE

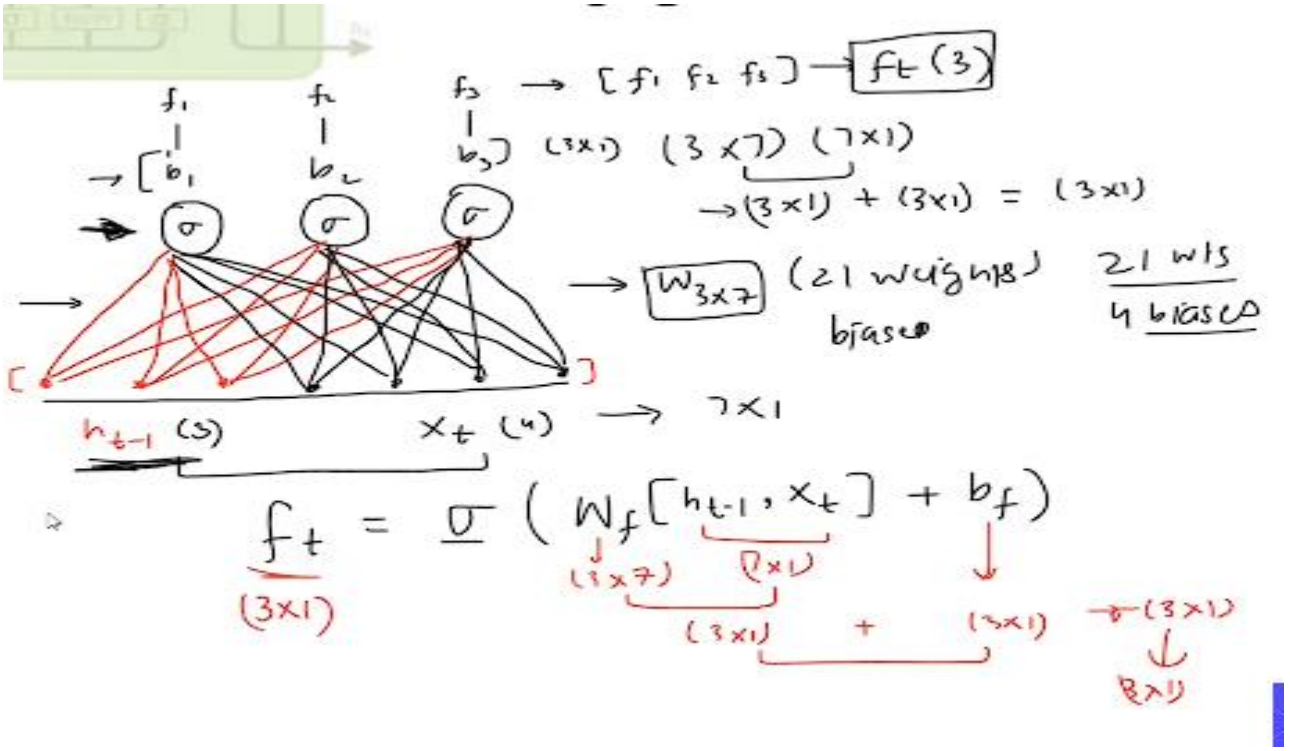
Network At Forget Gate:

The purpose of forget gate is to remove/forget the unnecessary things from Long Term Memory (Ct/cell state).

How?

Suppose we have Ct-1 and Ht-1 of shape 3*1 and current input Xt is of 4*1.

First we concat Ht-1 and Xt (current input) and pass to our forget gate's neural network, with corresponding weights Wf:



Output of this network is of shape (3*1) and it's our **ft** (output of forget gate at time t/current)**,** now we calculate **ft*ct-1,** and see what is going to be removed/forget from ct-1(LTM):

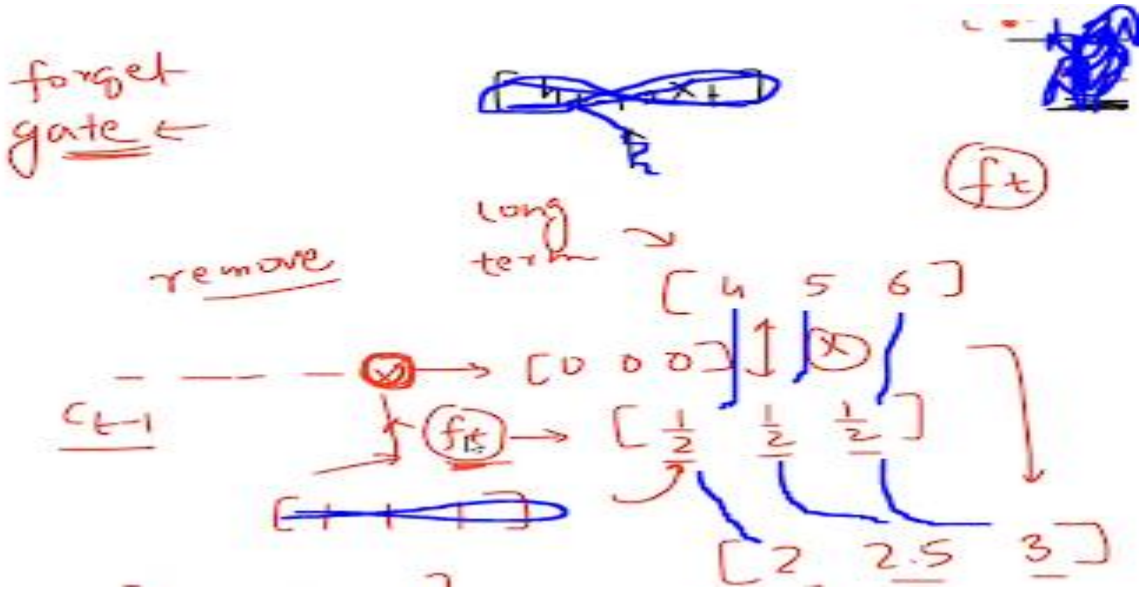
And as we know the shape of Ct-1 is also (3*1) as of the shape of ft, both are vector of same shape/dim, we can apply pointwise operator (*).

Let see how it forgets the information from Ct-1 (previous cell state/LTM):

Suppose ct-1 = [4, 5, 6] and ft = [½ , ½ , ½]

Ct-1 * ft = [4,5,6] * [½ , ½ , ½]

Ct-1 * ft = [2, 2.5 ,3]



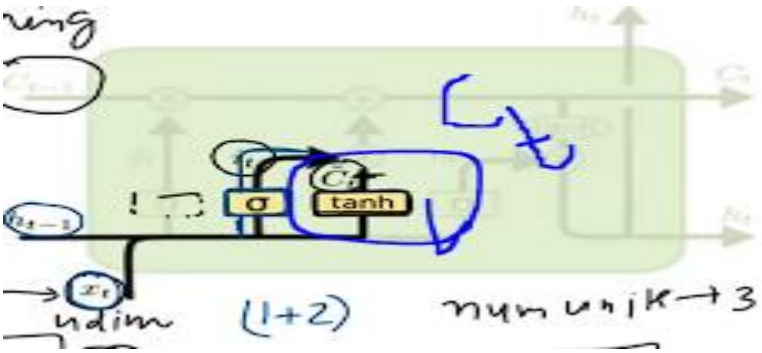
INPUT GATE:

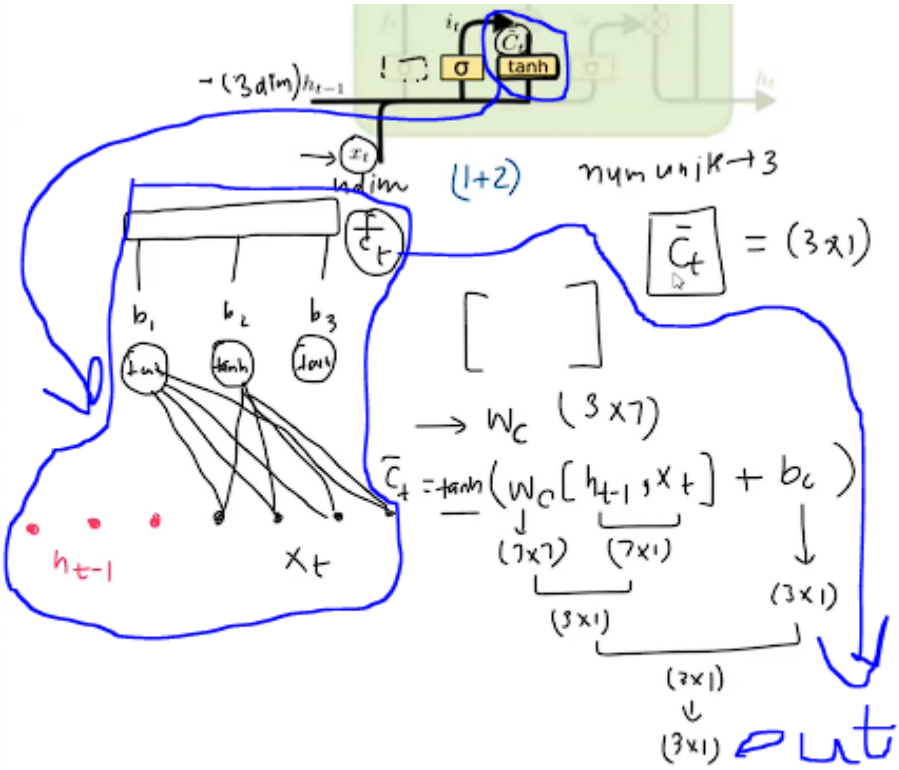
It consists of two neural networks (one is of sigmoid activation function and one have a tanh activation function),

At let say first one a (It) and second (Ct also called candidate cell state),

Input for both network is same and that is (concatenation of ht-1 and xt) .

Candidate cell (potential important info):

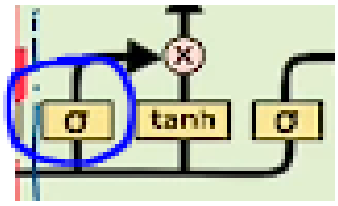




Output of candidate function is also a vector of (3*1), now we calculate (**It**) by using sigmoid activation function in (It) network,

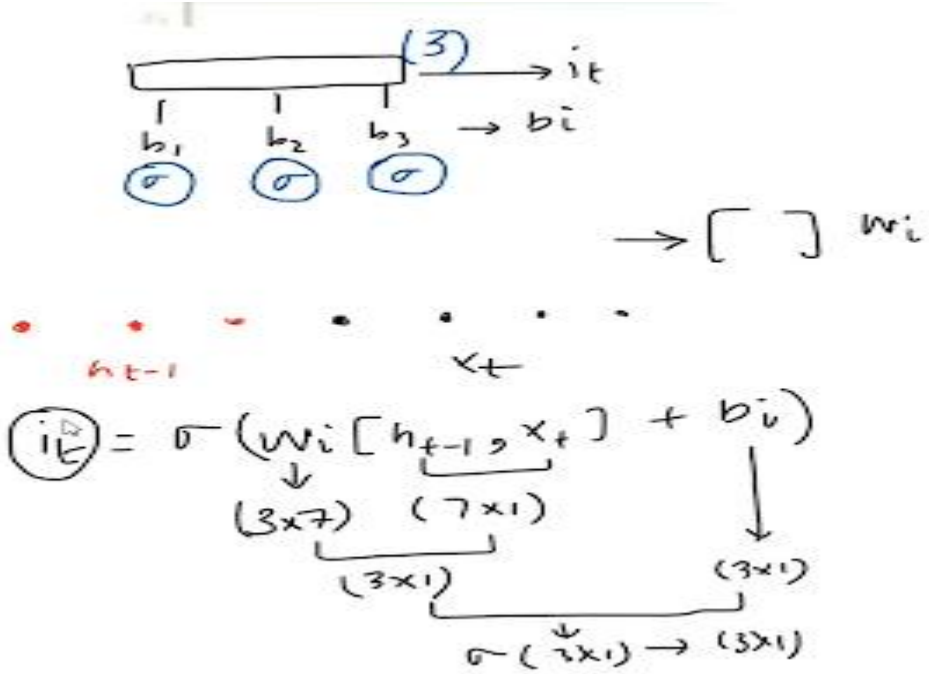
It network will also have same input which is concat of Xt and ht-1:

This (It) will filter that what information from current input will be pass to the current cell state,



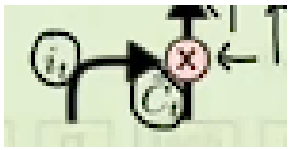
It is out this network with sigmoid activation function.

How It(network with sigmoid function in Input Gate) works?

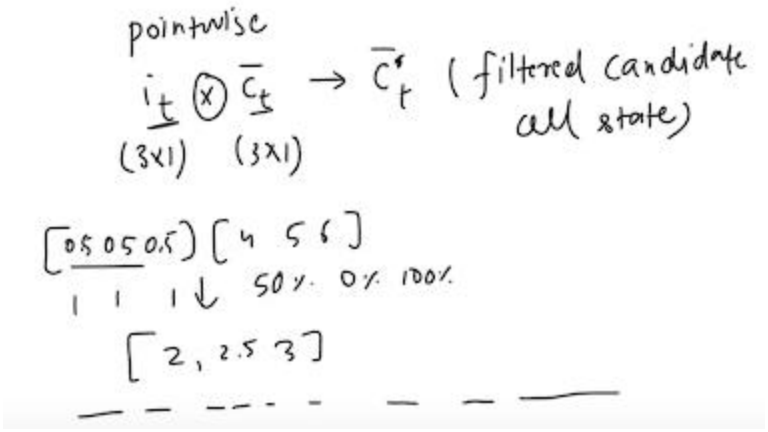


Now we have output of both (It and candidate cell Ct) now we apply (*) pointwise operator on this output of (3*1) dimension and get out current cell state (Long Term Memory) to pass next.

Pointwise of Ct and It:

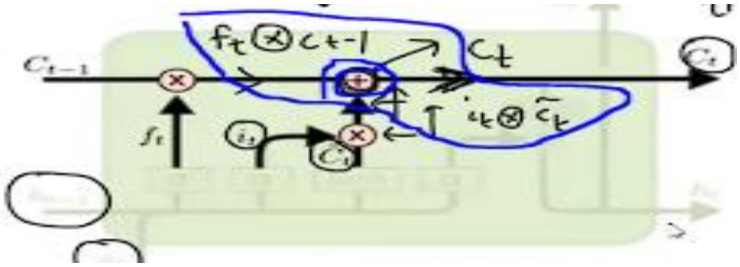


How to calculate Pointwise(*) of Ct and It?

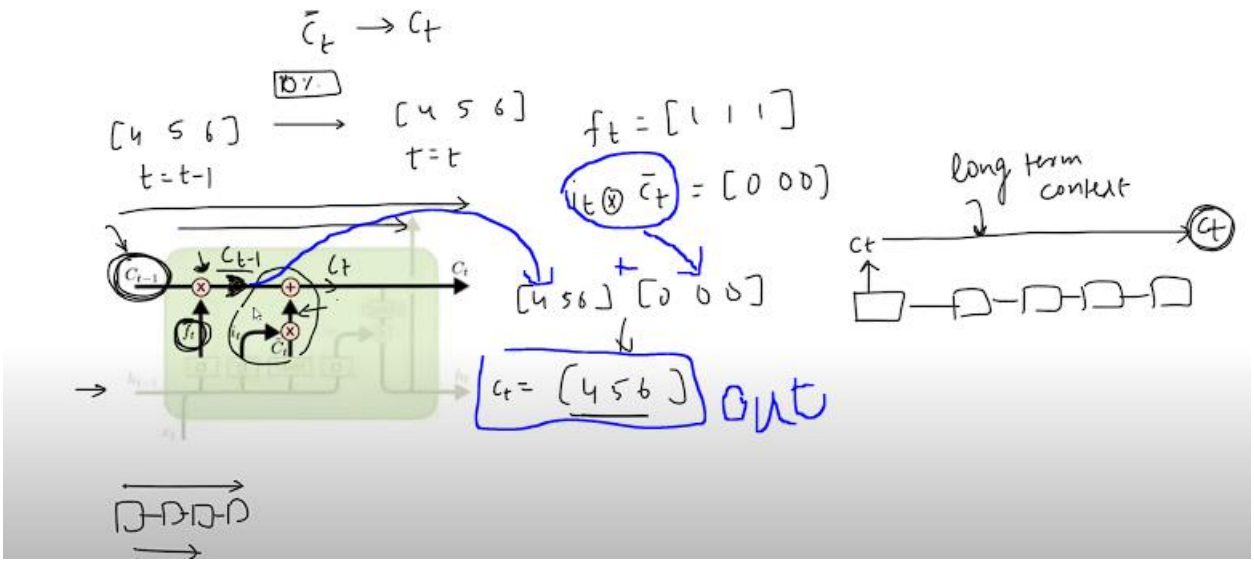


Now the final step to get our current Ct is the addition of two vectors we get from two operation:

i- Ct-1 * ft(forget gate)
ii- It*Ct(candidate gate)

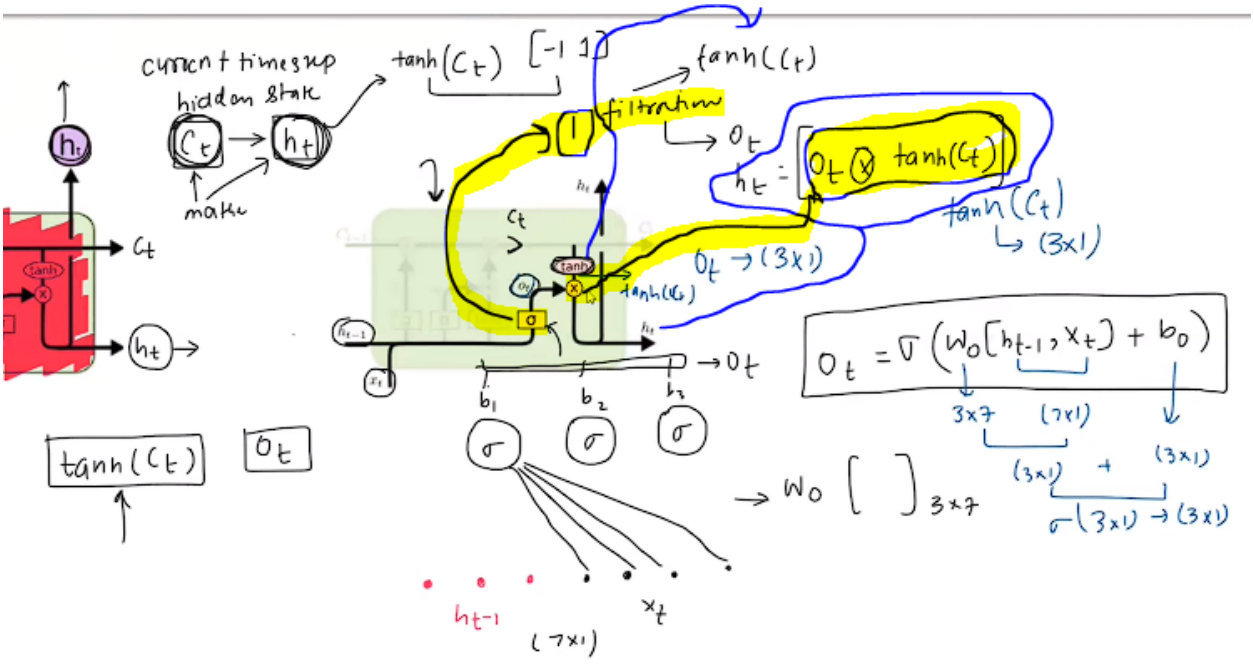


And this will give us our final Ct(current Cell state/LTM):

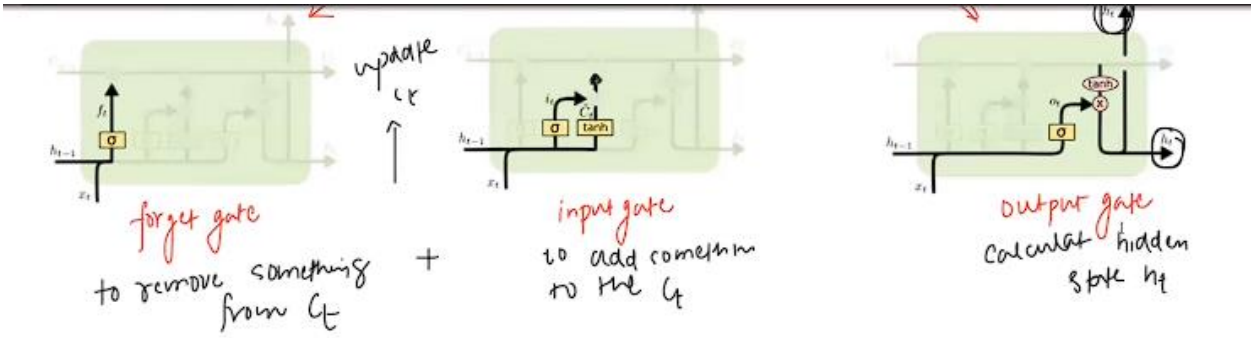


Output Gate:

This is to calculate output of current state which is ht.



CONCLUSION:



Feel free to watch this video if you have any confusion in this, it's the best video for understanding LSTM.

VIDEO REFERENCE:

https://youtu.be/Akv3poqqwI4?list=PLKnIA16_RmvYuZauWaPlRTC54KxSNLtNn

Chennel name: CampusX.