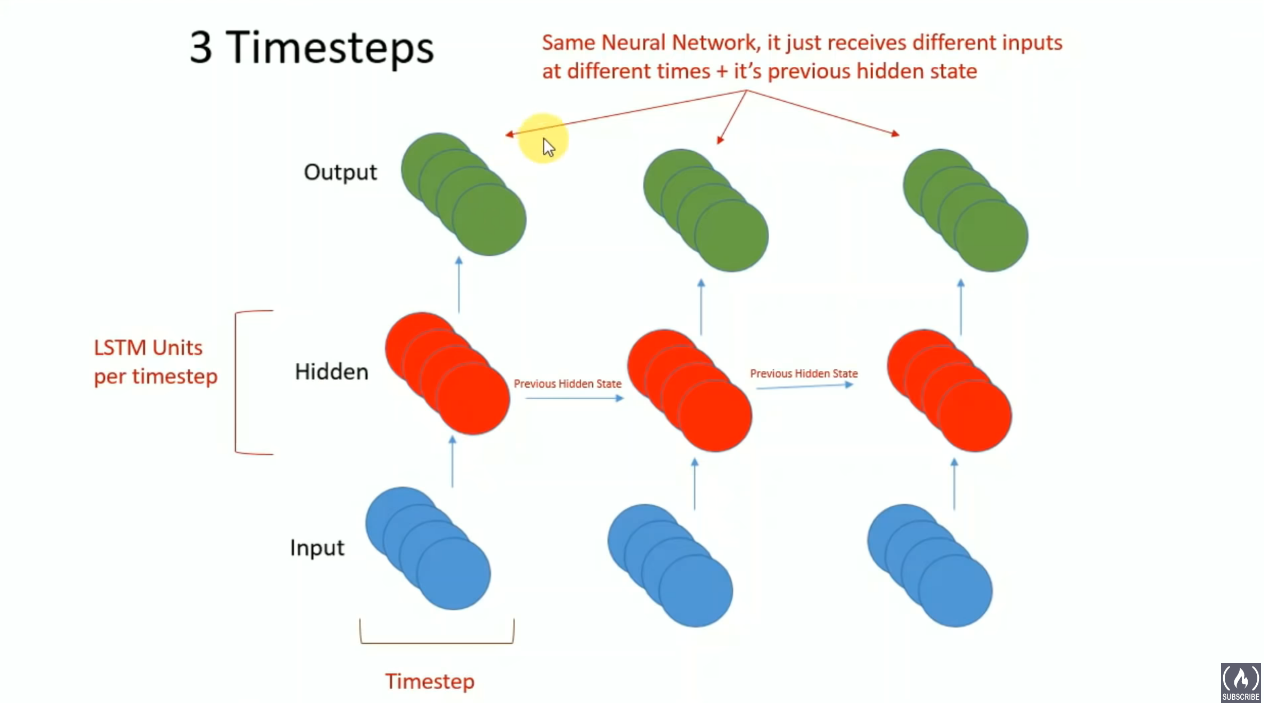
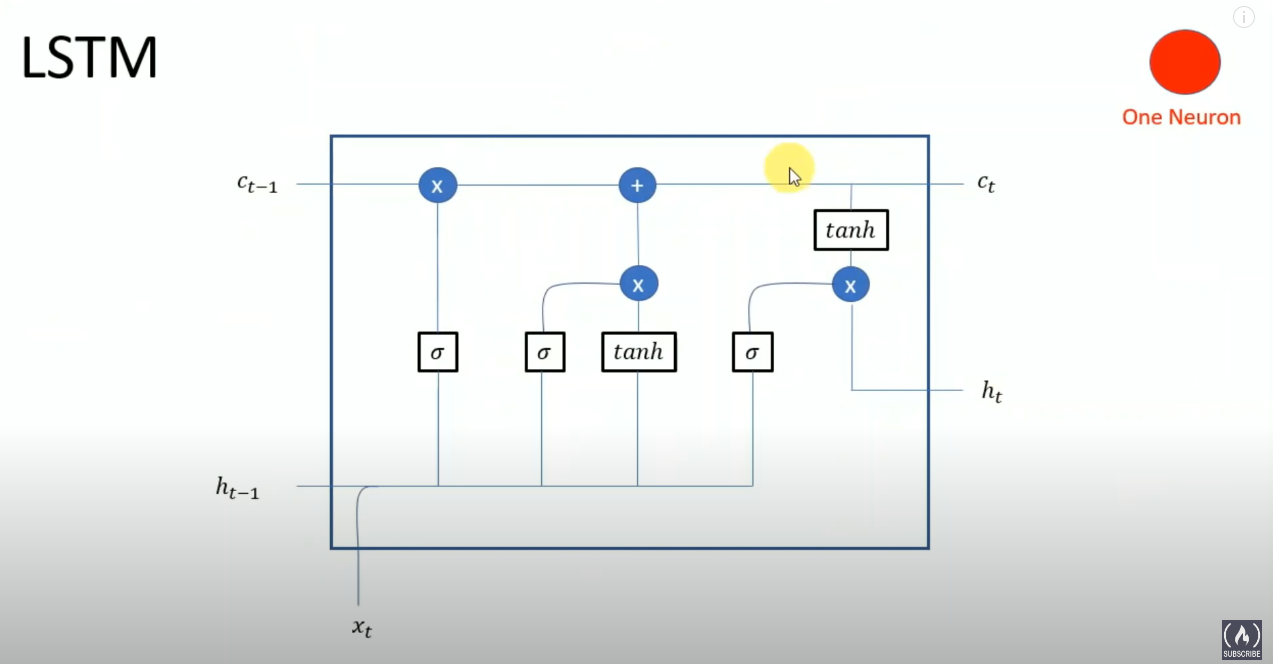
This is the entire Neural Network Architecture.





This is **one node** in the Hidden layer. If there are 4 nodes in the hidden layer, there will be four of these nodes

h(t-1) is the previous hidden state

x(t) is the current input

c(t-1) is the memory from the previous state

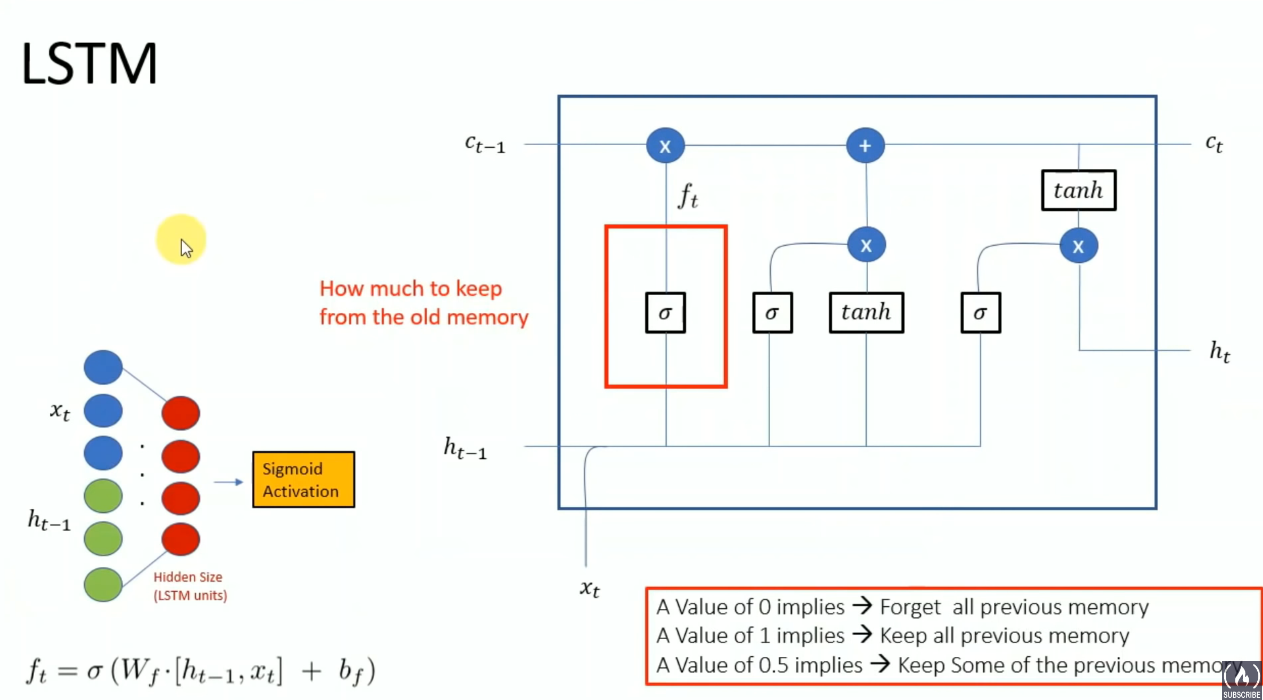
c(t) is the final memory state

h(t) is the output of the node. This is essentially the tanh activation applied to c(t)

## 

## Forget Gate



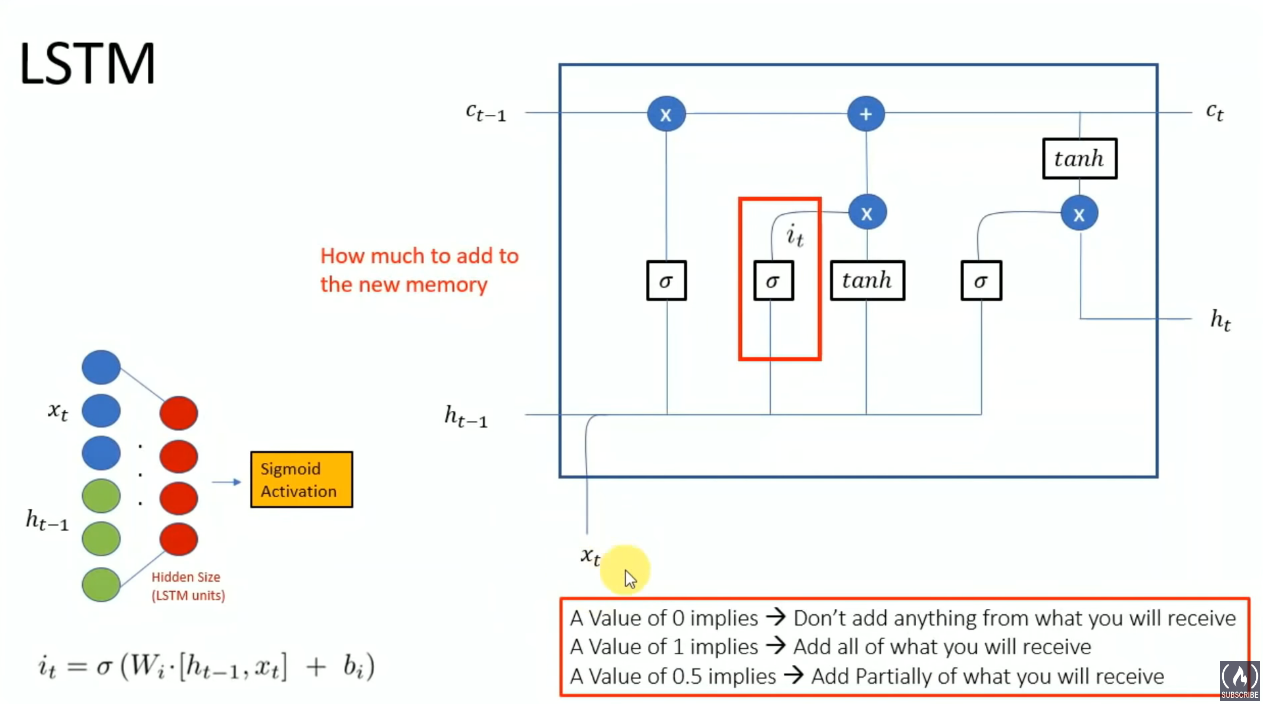


**Defines how much we want to keep from the old memory**

Has a sigmoid Activation Function → Will output a number between 0 and 1

Everytime there is a forward pass, the current input (xt) as well as the previous hidden state (ht-1) will be concatenated together and passed as input to the node.

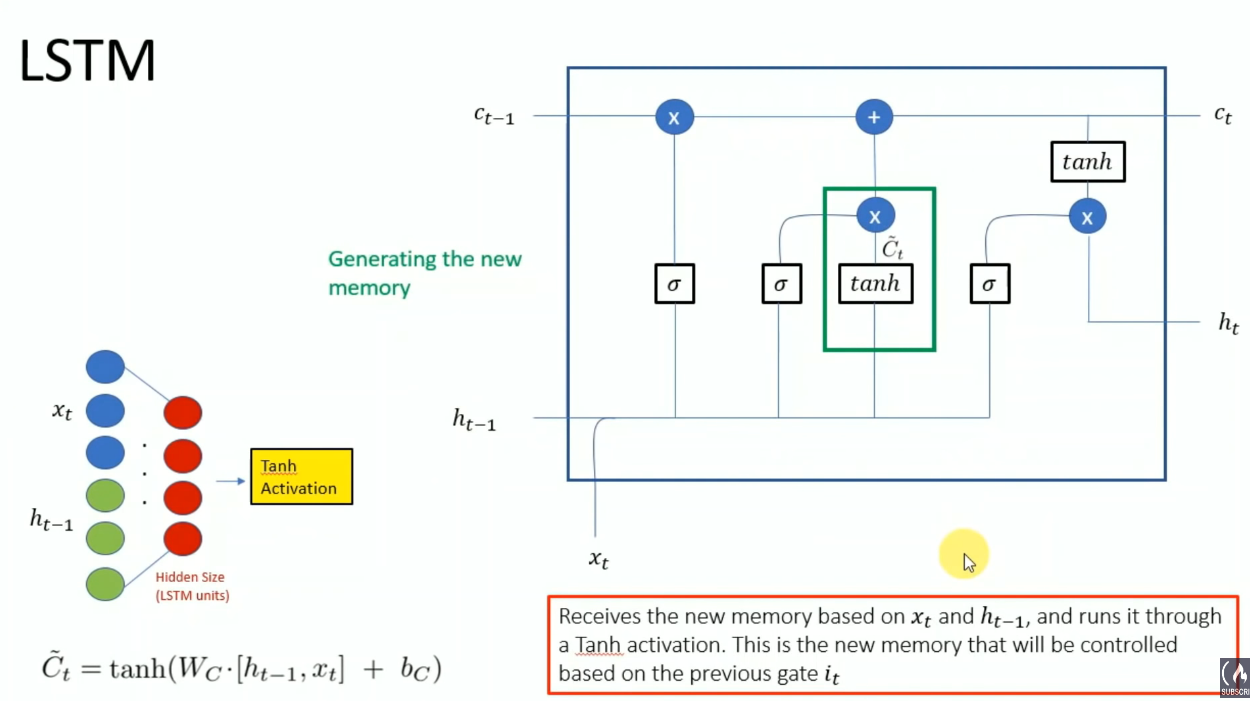
## Input Gate



**Defines how much to add to new memory**

Uses a sigmoid Activation Function

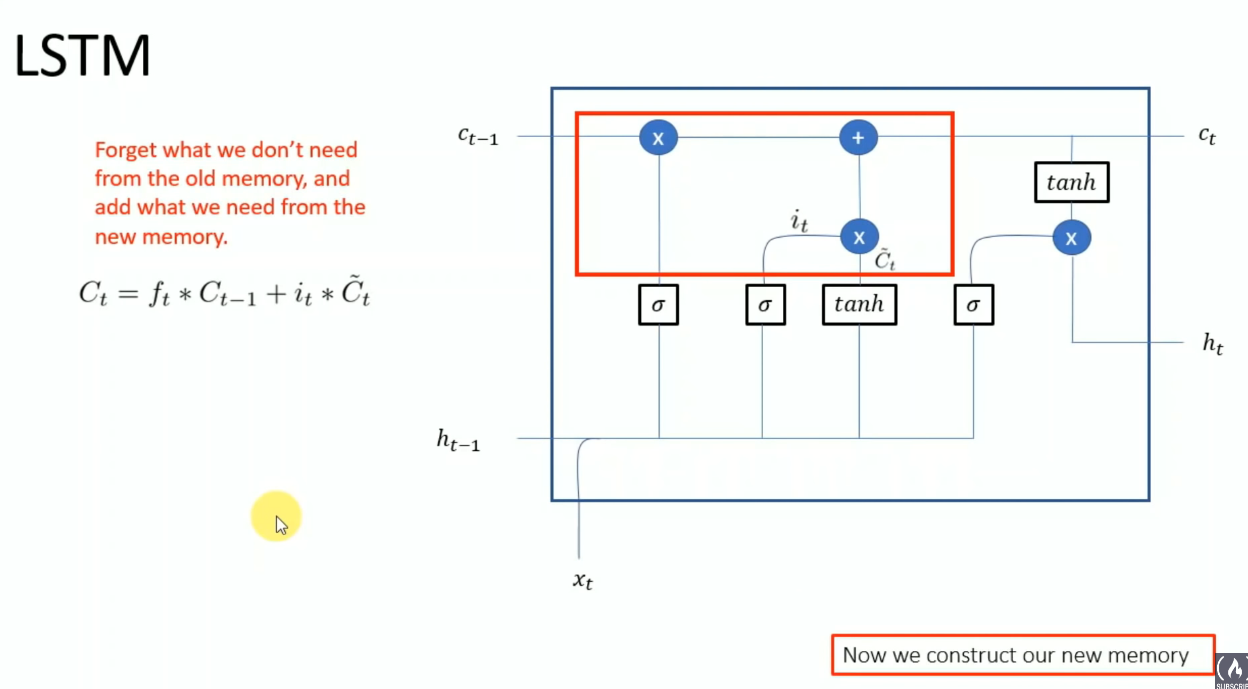
## Generating new memory



New memory is the tanh activation function applied over the inputs (ht-1 & xt) multiplied by their weights + bias

The final memory will be based on this new memory as well as the Input Gate (Which determines how much of the new memory to use for the final memory)

## Constructing final memory

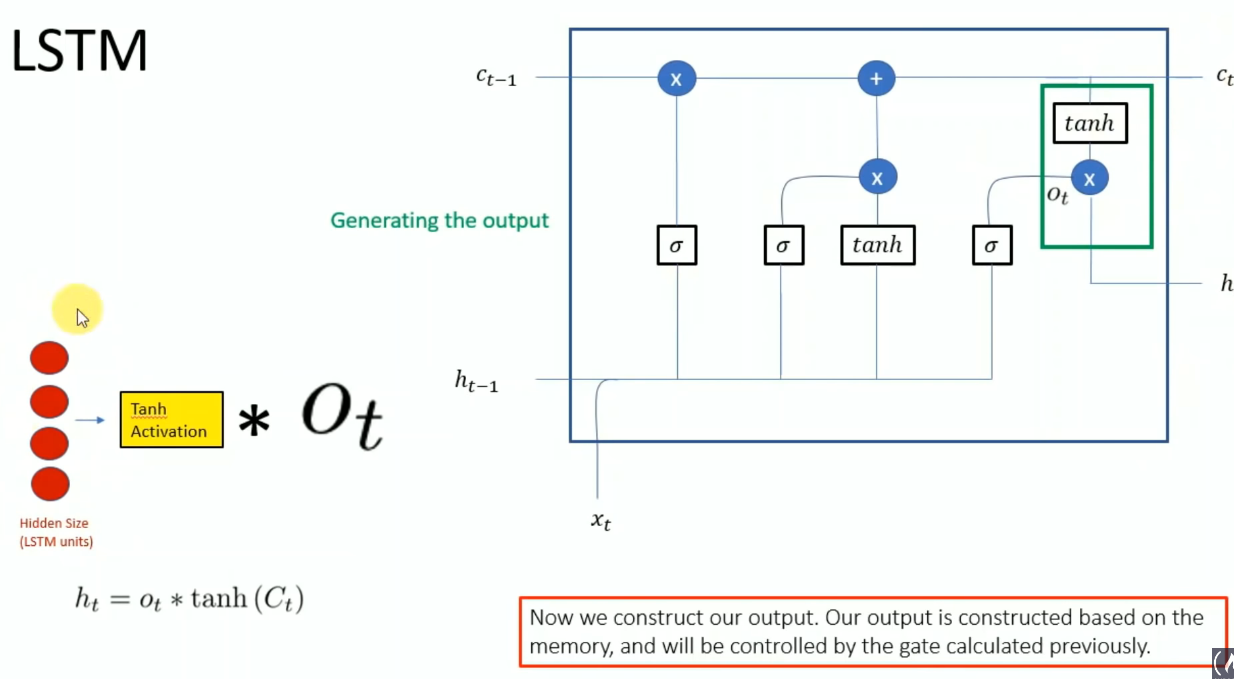


Ct is the final memory that this node will generate (And will pass down to the next node)

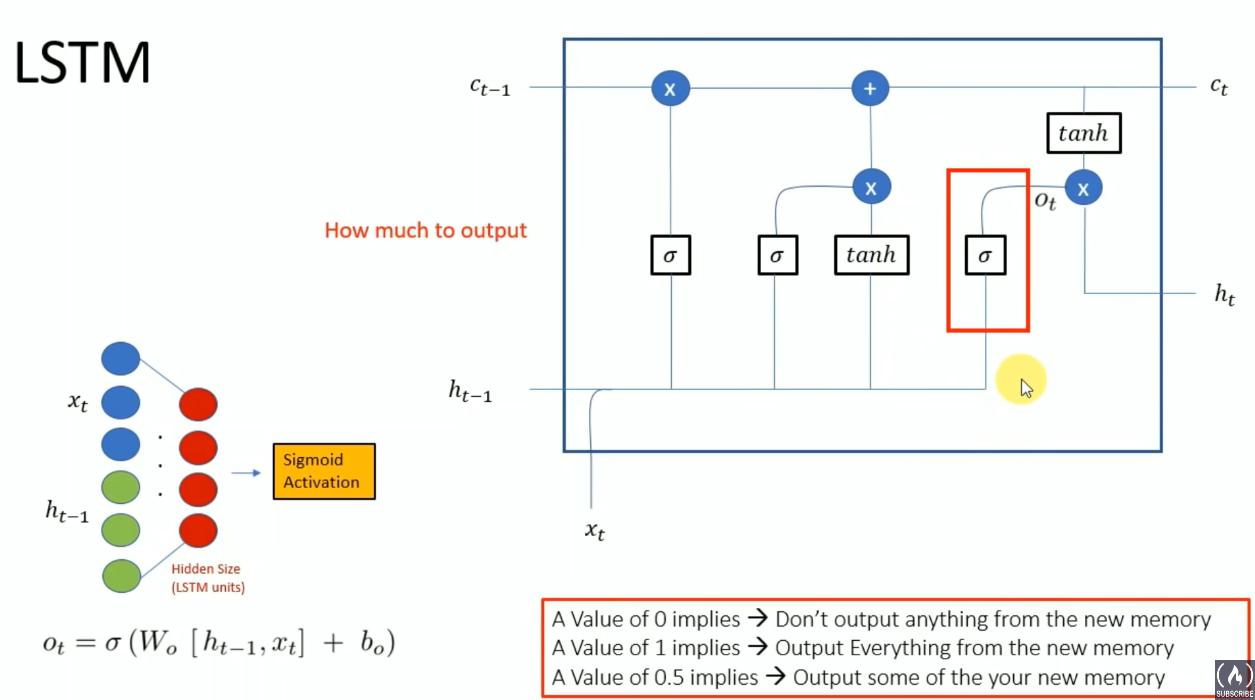
Ct is formed by:

* Using the Forget Gate to determine how much of the previous memory (ct-1) to forget
* Using the Input Gate and the new memory to determine how much of the new memory to actually use for the final memory

## Creating Output

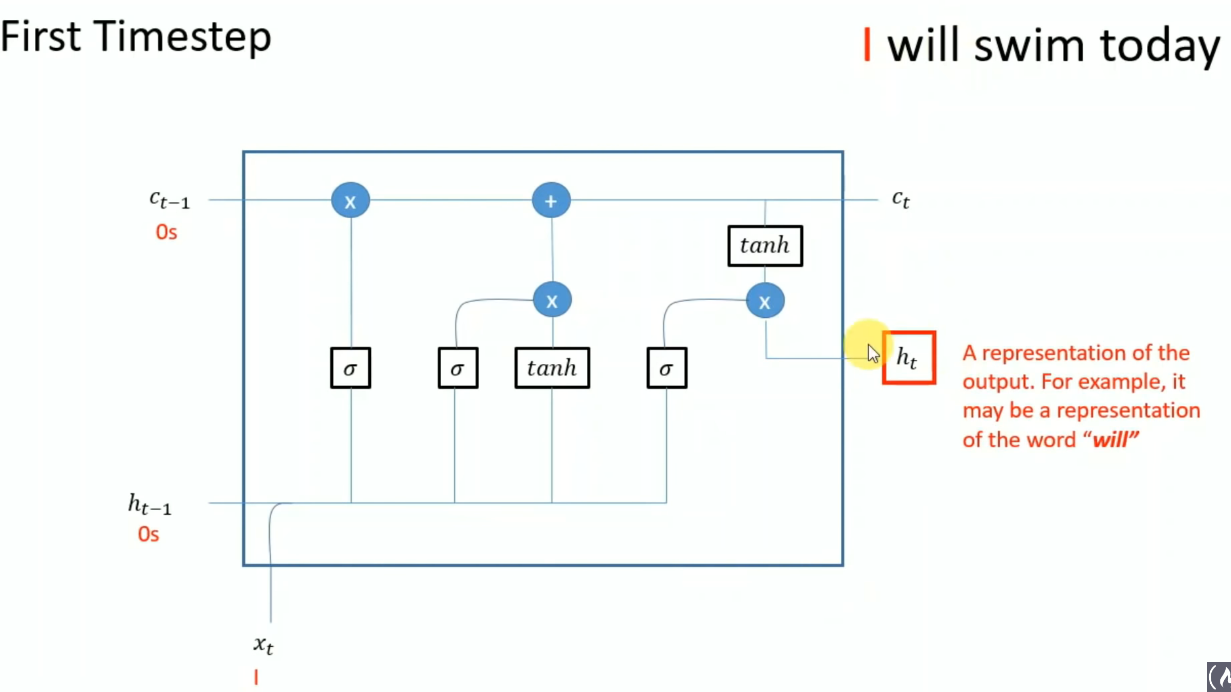


The output h of the LSTM Node (h) will be the tanh activation applied to the final memory ct (Which will also be passed to the next node)



This gate controls how much of the final memory to use for the output

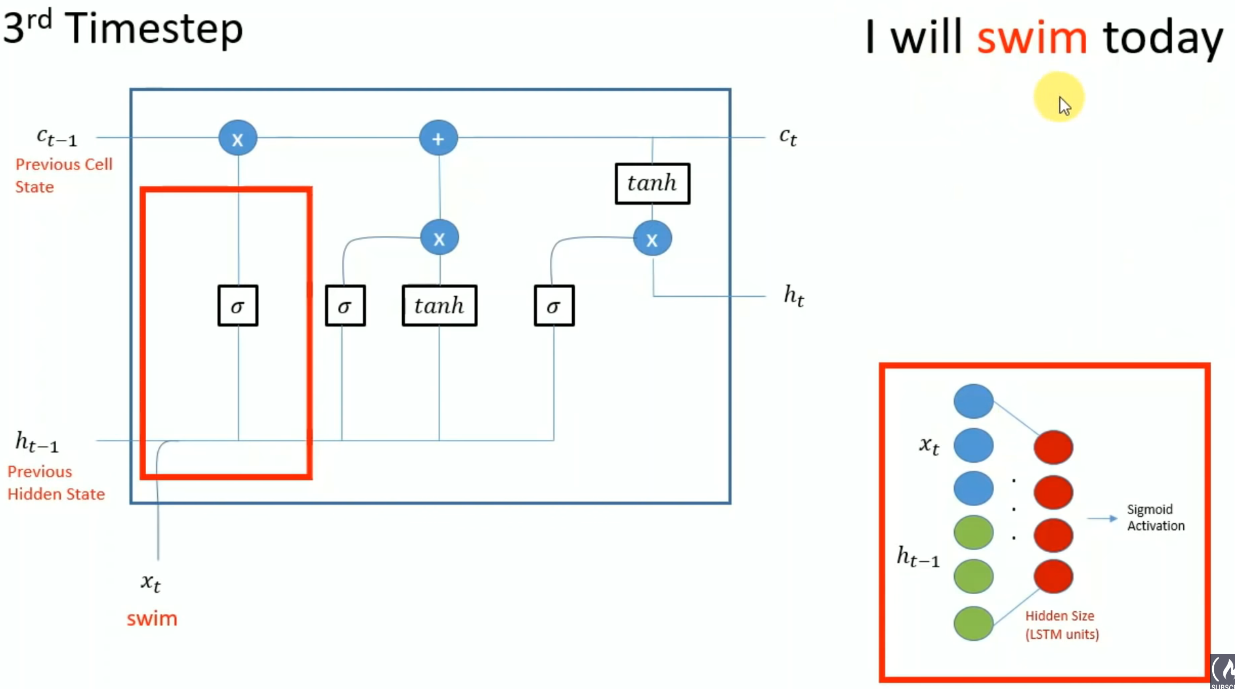
## Example



h(t-1) and c(t-1) will be zeros, as there is no previous hidden state and previous memory state

h(t) will be the output of timestep=1, which is a representation of the prediction (‘Will’ in this case)

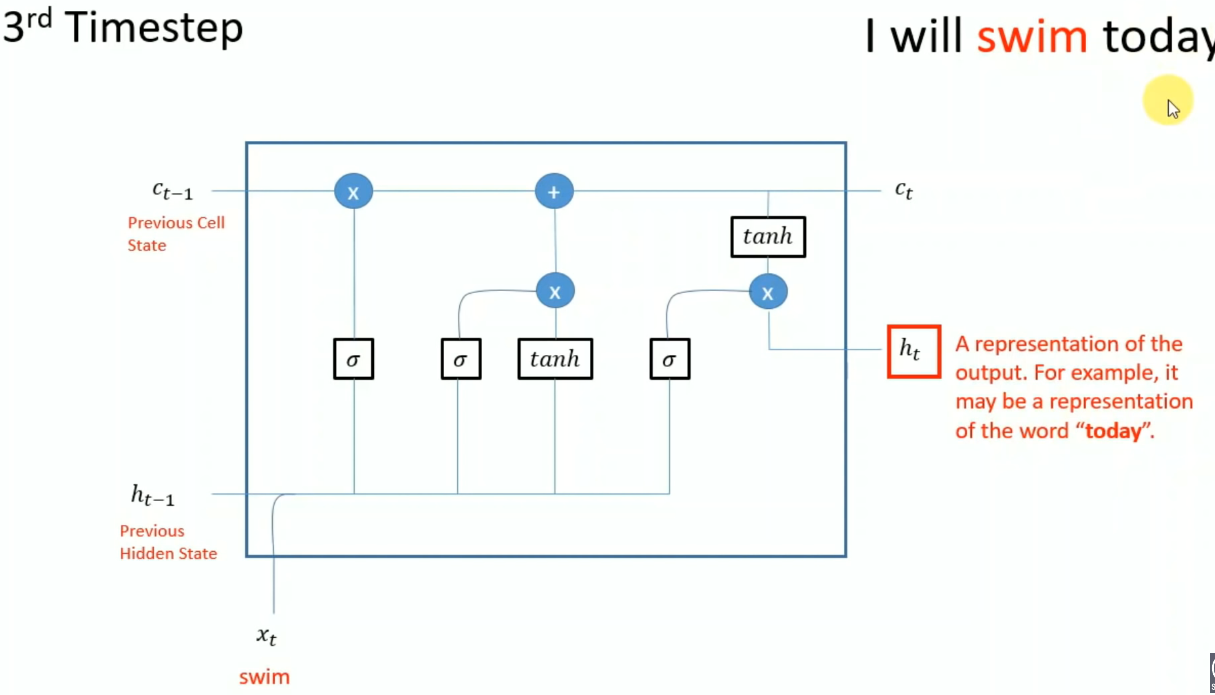
c(t) will be the final memory that is passed on to the next timestep (t=2). c(t0 will be a representation of what the LSTM knows about the sentence so far (Currently, it only knows ‘I will’)



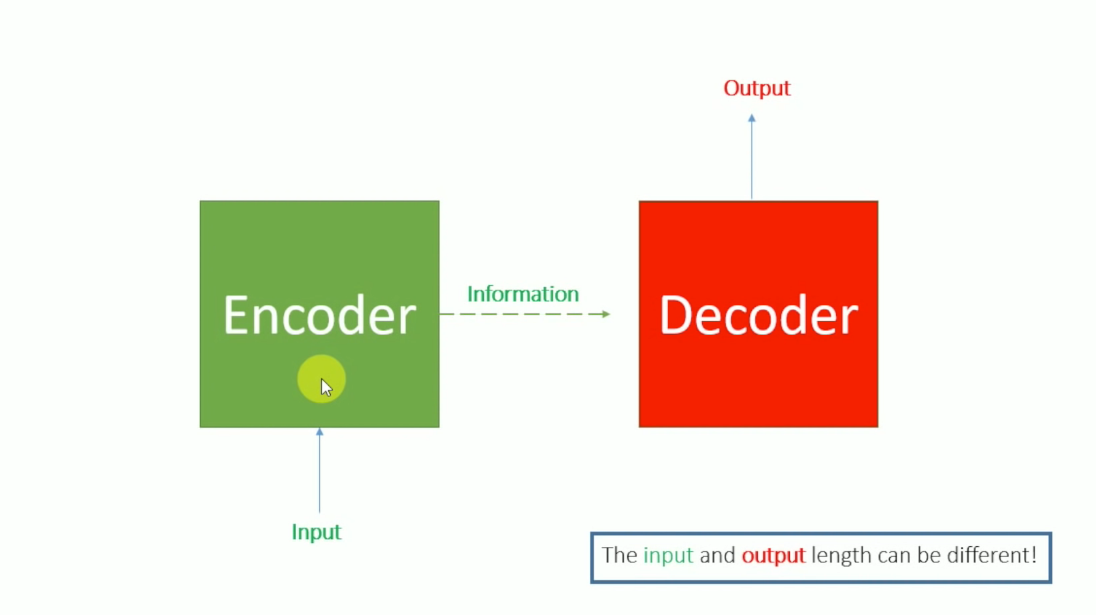
Same thing for timesteps=2 and 3….

Previous Memory state (ct-1) of time=3 will be the final Memory state (ct) of time=2

Previous Hidden state (ht-1) of time=3 will be the output (h) of time=2. Remember that the output is simply the tanh activation applied to the final memory of time=2 (With an extra gate taken into account to see how much of the final memory state will be used for the output)



## Seq2Seq Model



Input is passed to the Encoder and a Feature vector is produced (A representation of the Input)

This Feature vector will be passed to the decoder to produce the final output.