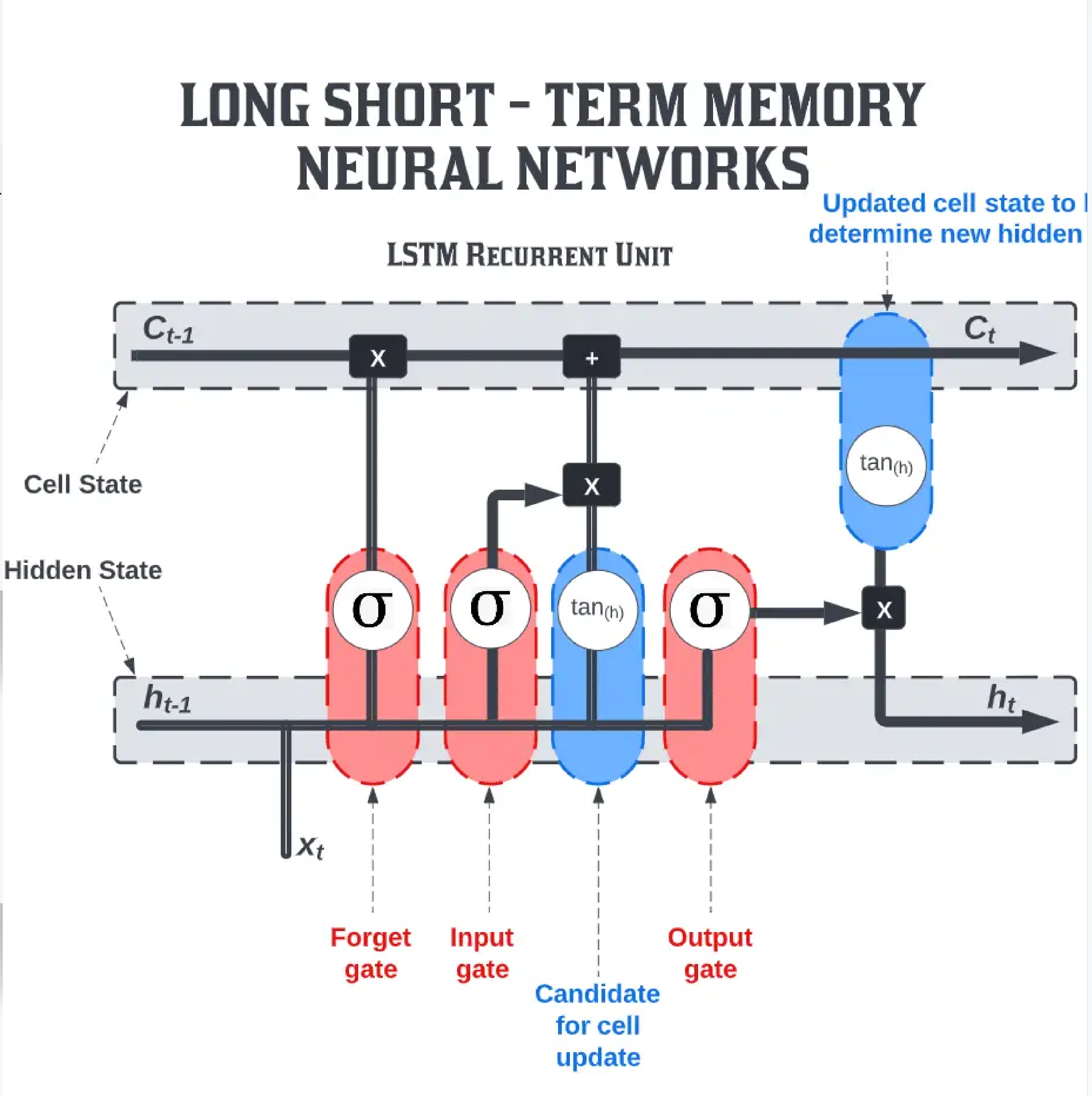
LSTM

LONG-SHORT TERM MEMORY

CONTENT

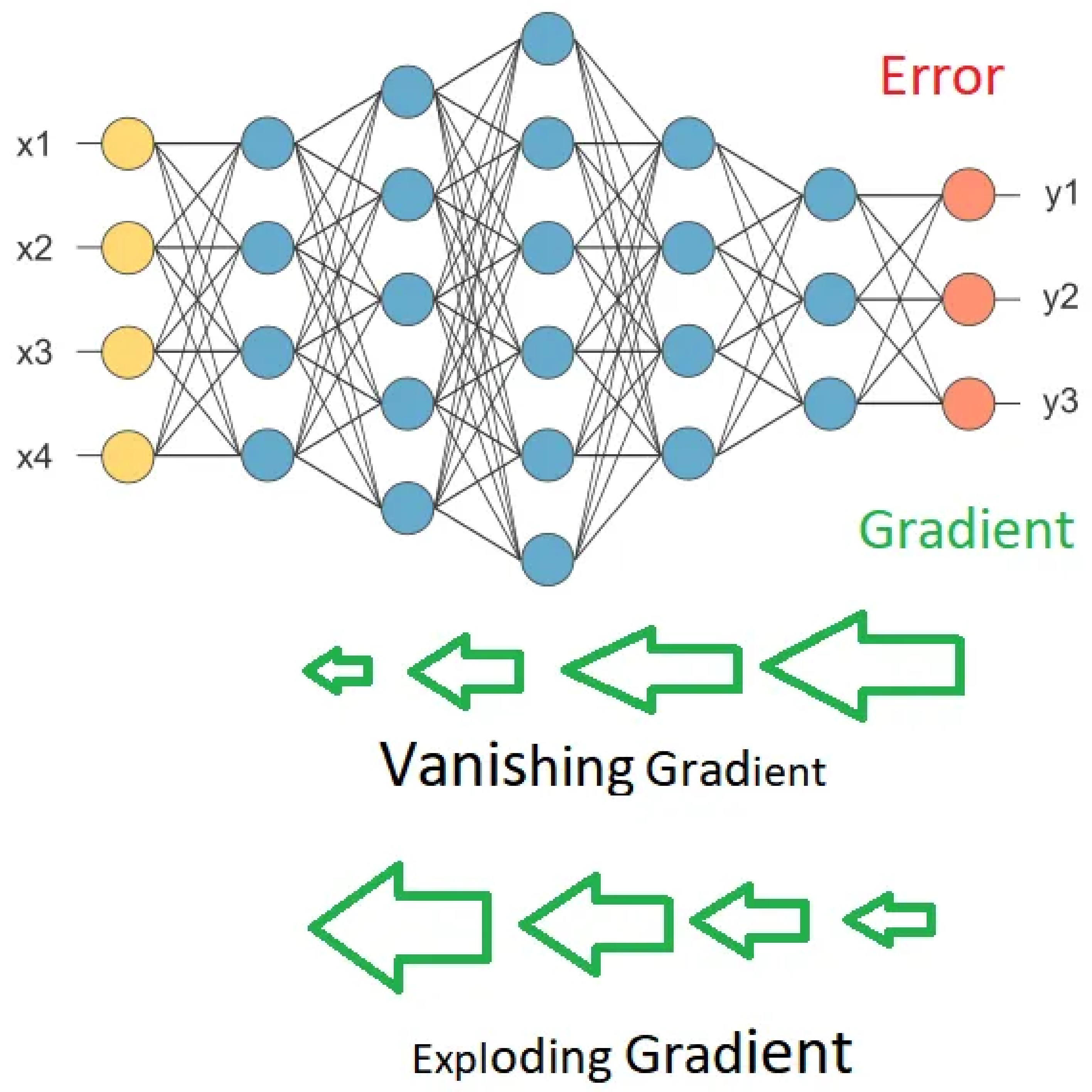
ABOUT

- LSTM is a model for the **short-term memory** which can last for a **long period of time.**
- Type of **recurrent neural network (RNN)** architecture. LSTMs are particularly well-suited for handling **sequences of data,** such as **time series data, natural language text, and speech.**
- LSTMs were developed to deal with t**he exploding and vanishing gradient problem** when training traditional RNNs.



REVISE

RNN



- "**Vanishing gradient**" problem. The problem is that the contribution of information decays geometrically over time.
- During the backward pass, the network calculates the **gradient of the loss function** with respect to the network's weights. This involves propagating the error backward from the output layer to the input layer.**Wij(k)=Wij(k)−α∂L/∂Wij(k)**
- Derivative always changes between 0-->0.25 keeps decreasing.
- **Exploding Gradient problem**
- RNNs are absolutely incapable of handling**"long-term dependencies".**

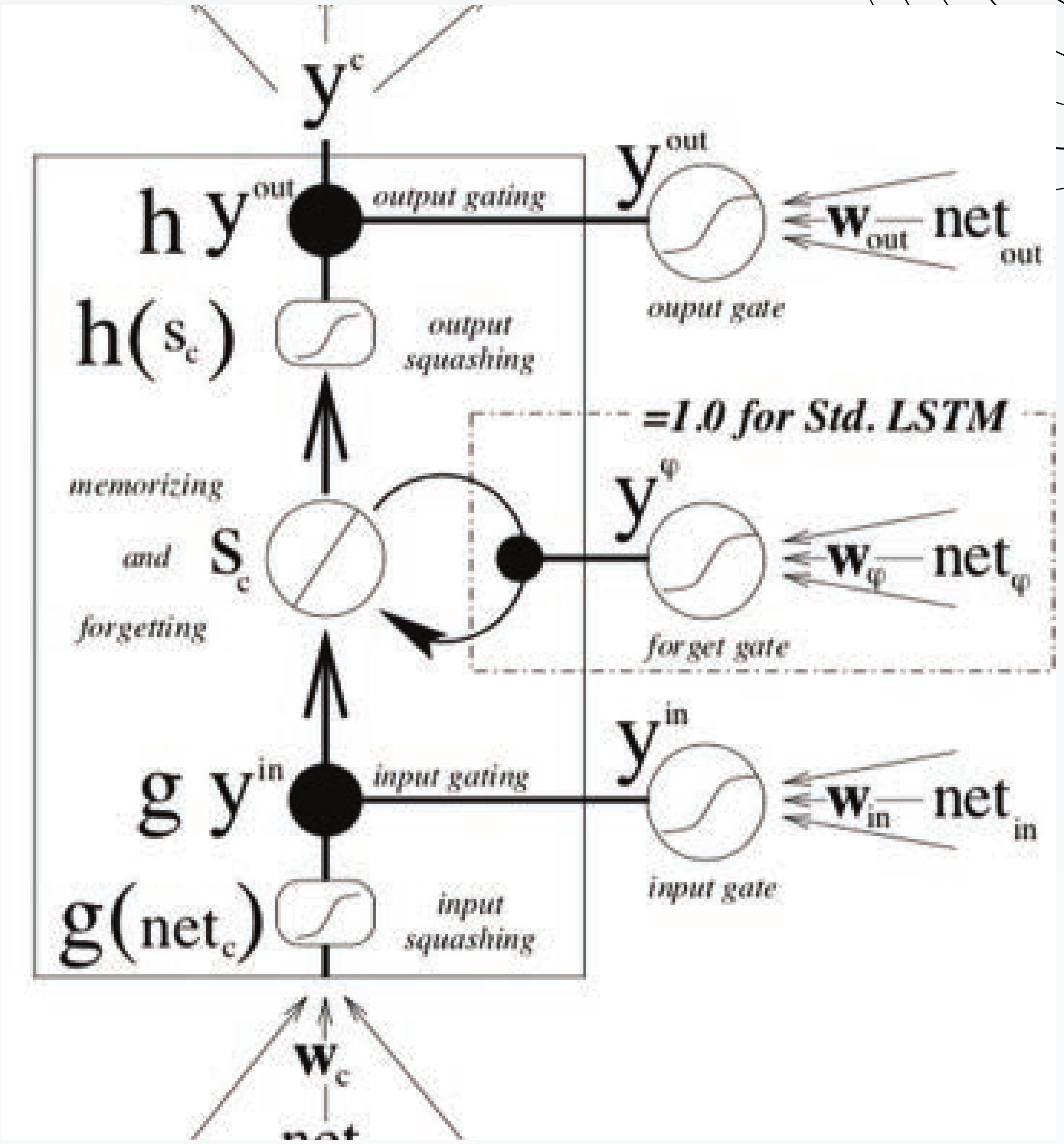
LSTM

Proposed by

Sepp Hochreiter and Jurgen Schmidhuber in 1997.

LSTM

- LSTM is an example of **a gated unit.**
- It consists of logistic sigmoid functions known as **gates** in addition to traditional activation functions.

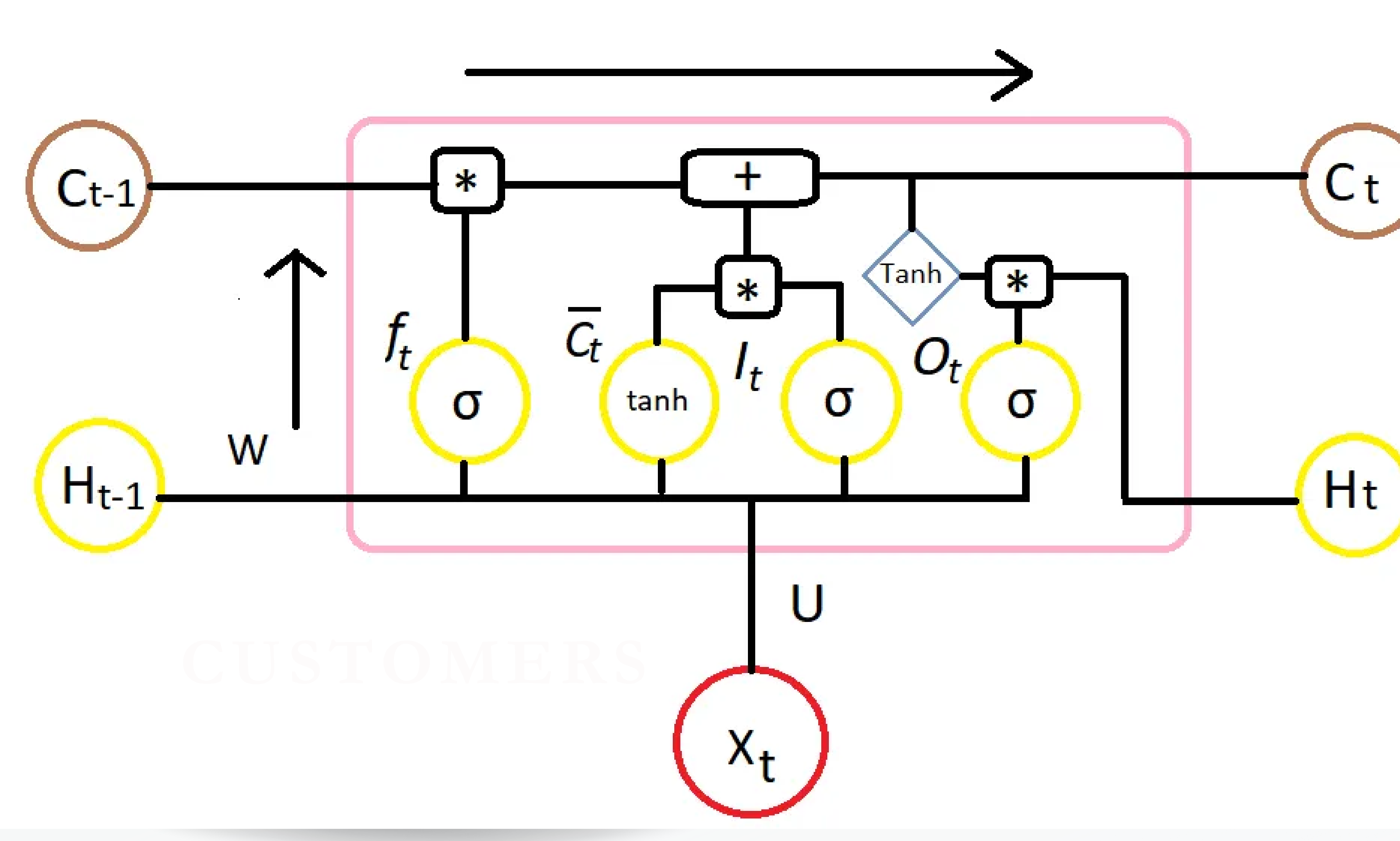


ARCHITECTURE

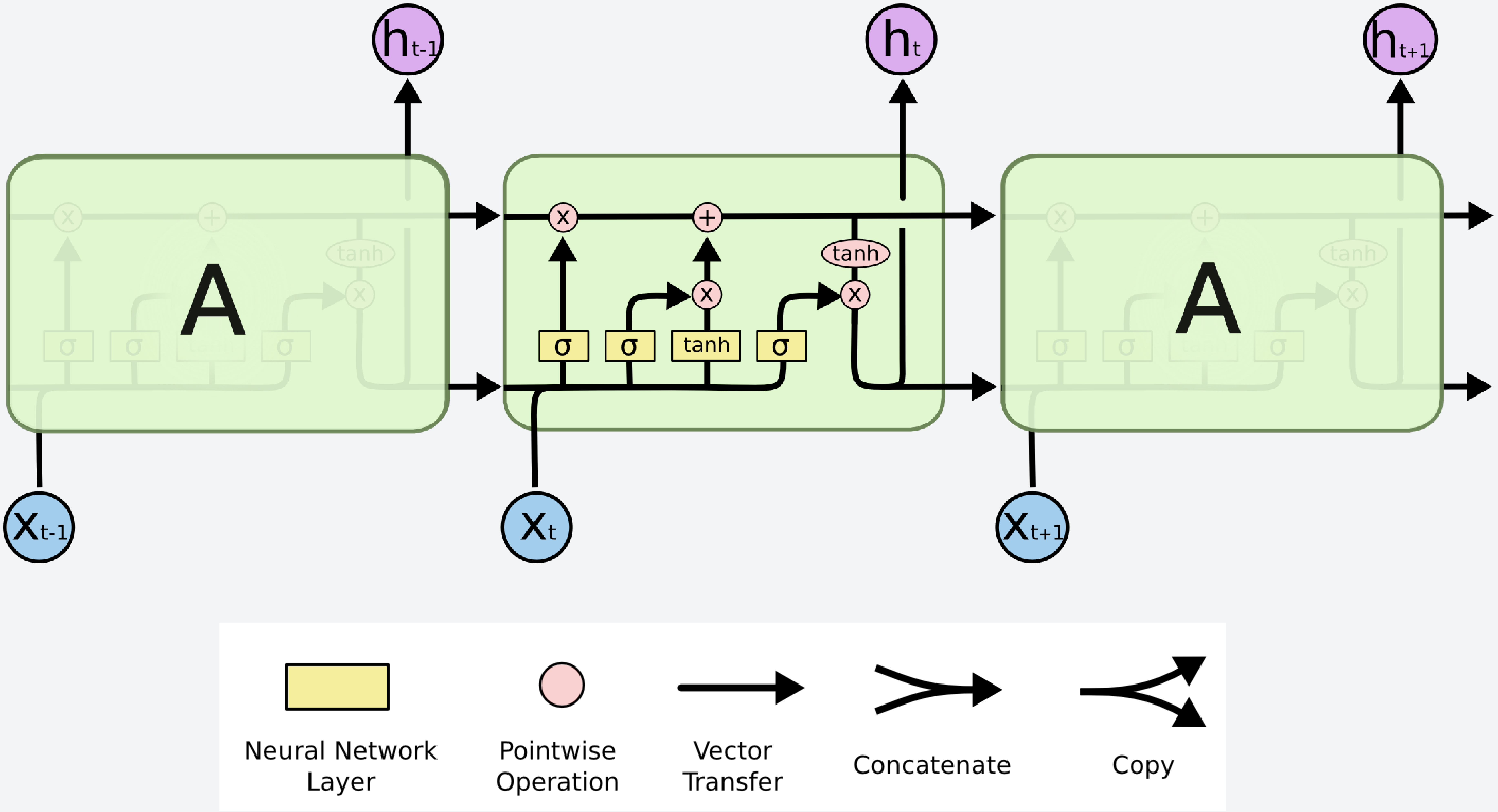
- A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate.
- The cell is responsible for "remembering" values over arbitrary time intervals; hence the word "memory" in LSTM.
- There are connections between these gates and the cell.

COMPONENTS

- Candidate layer“C`"(a NN with Tanh)
- Forget Gate “f” (a neural network with sigmoid
- Input Gate “I” (a NN with sigmoid)
- Output Gate “O”(a NN with sigmoid)
- Hidden state“H”(a vector)
- Memory state“C” (a vector)



- Inputs to the LSTM cell at any step are
- X (current input) ,
- H(previous hidden state)
- C(previous memory state)



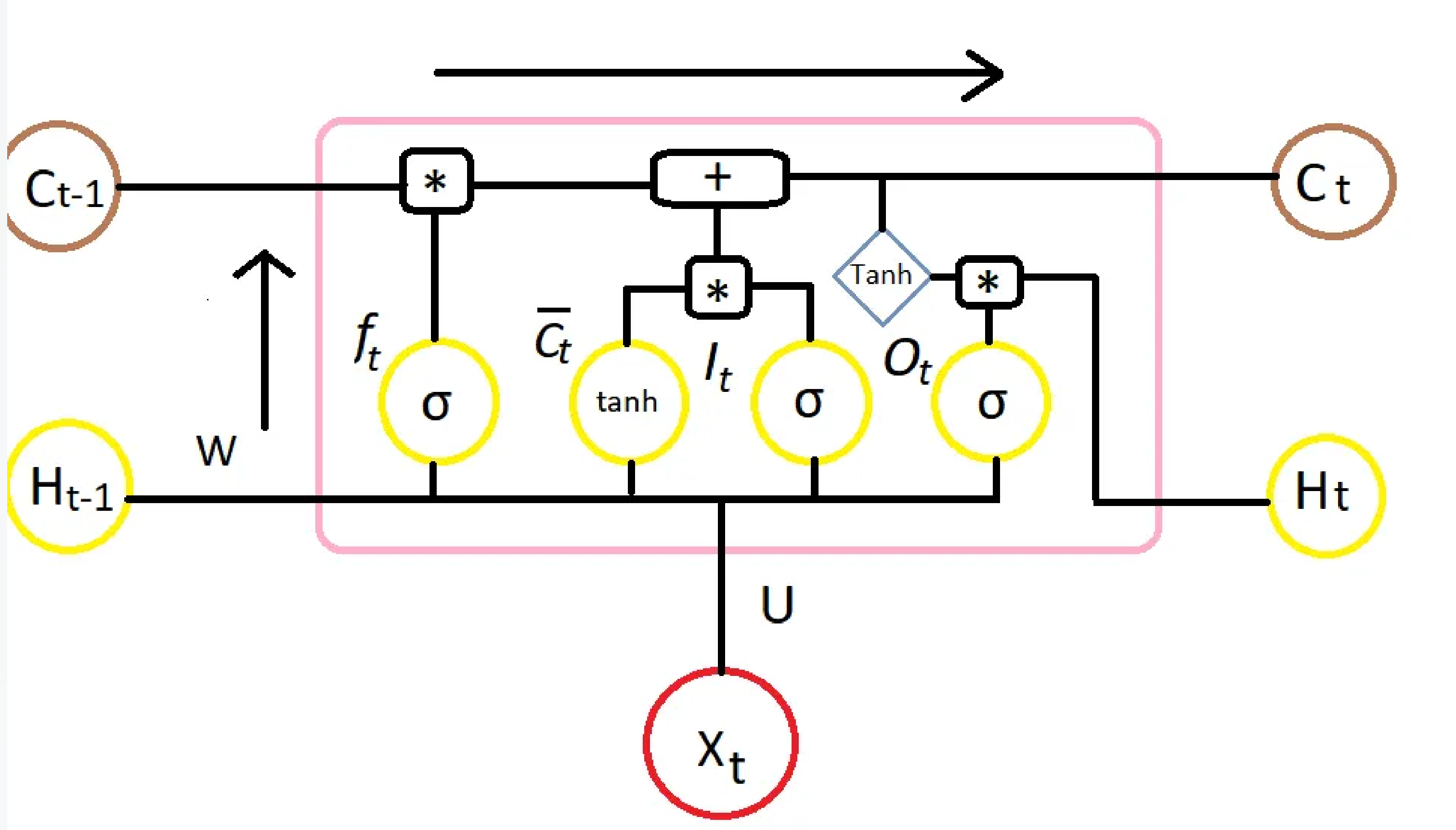
h_{t-1}
h_t
h_{t+1}
A
σ
σ
tanh
σ
tanh
A
x_{t-1}
x_t
x_{t+1}
Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy

IN ONE TIMESTEP

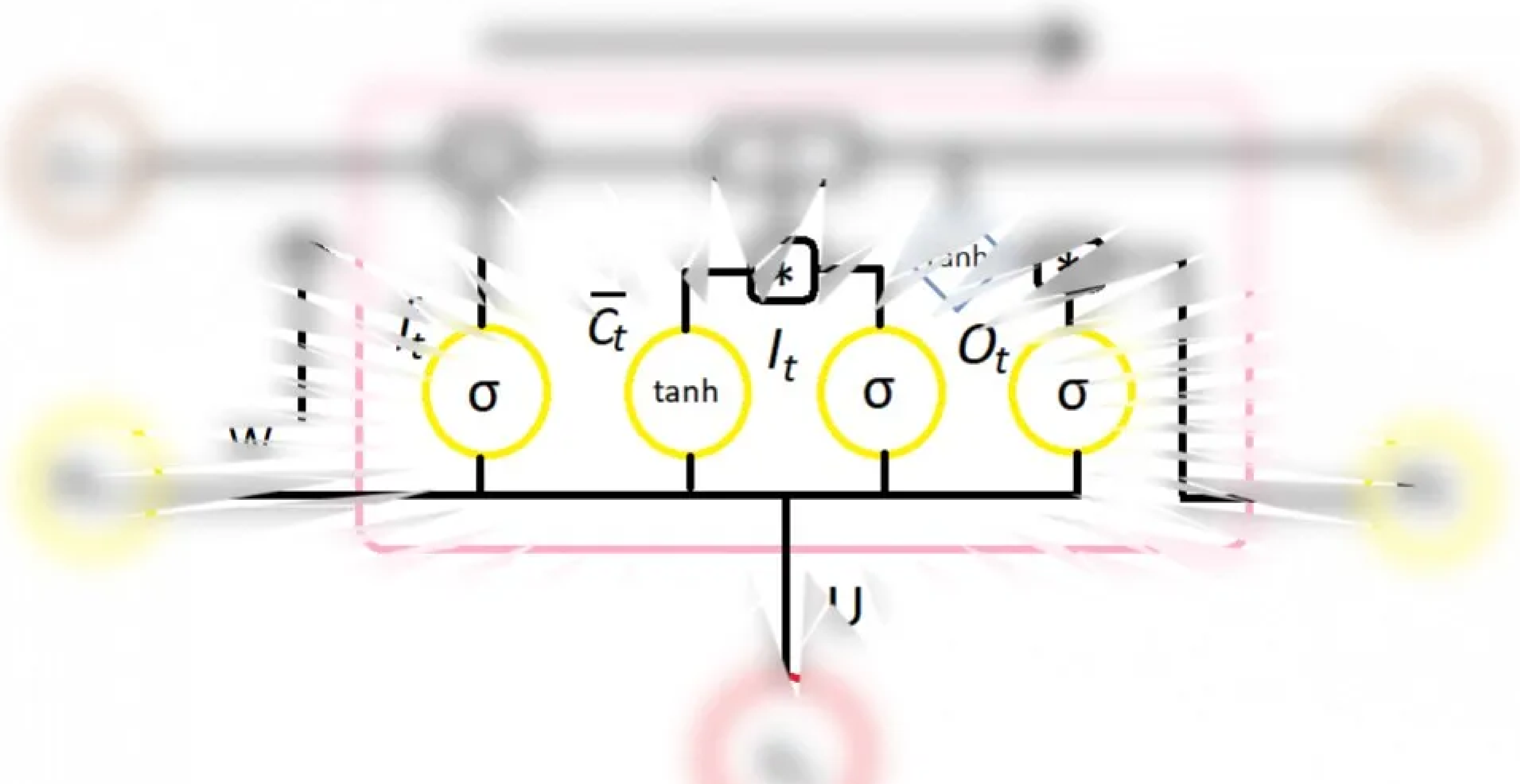
- Inputs to the LSTM cell at any step are X (current input) ,H(previous hidden state) and C(previous memory state)
- Outputs from the LSTM cell are H (current hidden state) and C(current memory state)

Forget gate(f) , Candiadte layer(C`), Input gate(I), Output Gate(O) are single layered neural networks with **sigmoid activation function** except **candidate layer(tanh function).**

These gates first take input vector.dot(U) and previous hidden state.dot(W) then concatenate them and apply activation function



finally these gate produce vectors (between 0 and 1 for Sigmoid, -1 to 1 for Tanh) so we get four vectors f, C`, I, O for every time step.



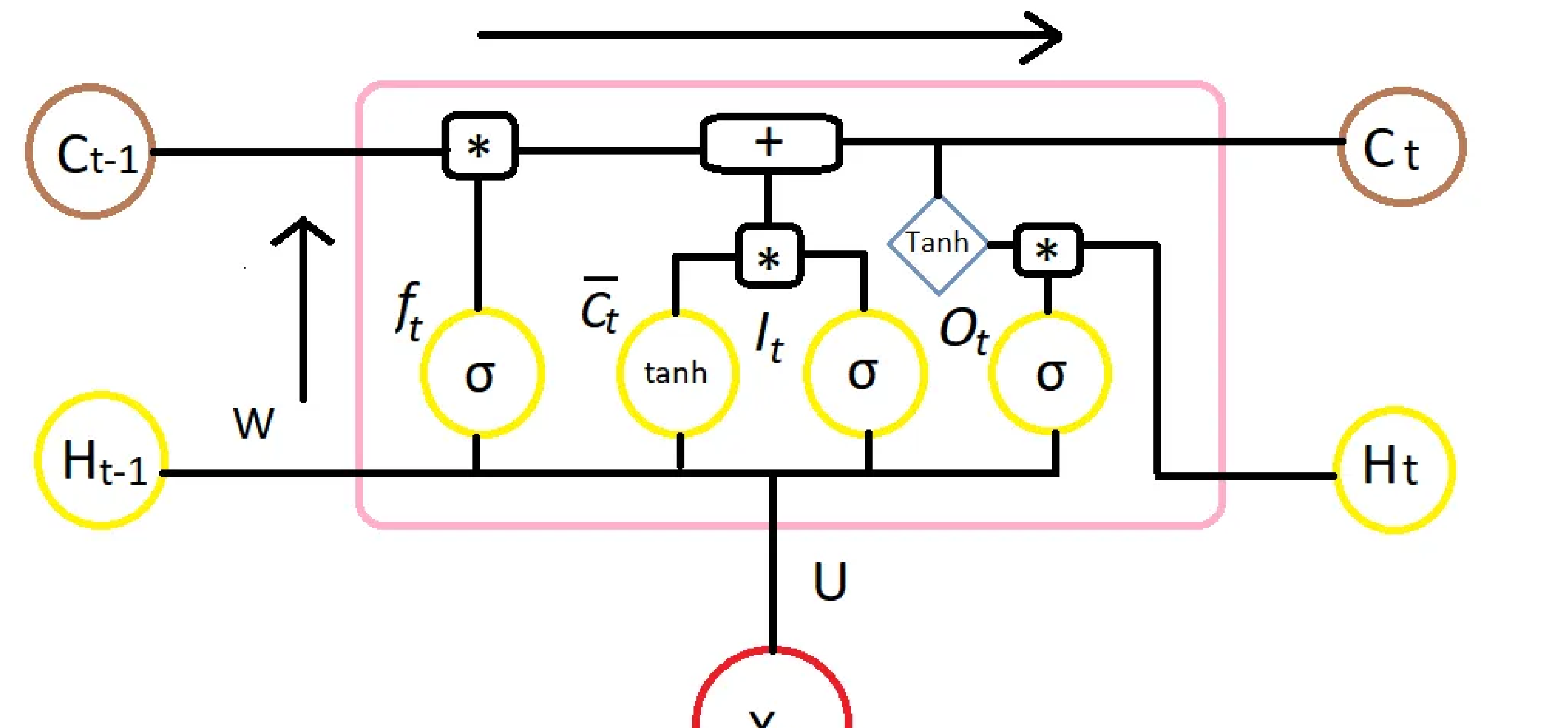
$\bar{C}_t$
I_t
O_t
σ
tanh
σ
σ

MEMORY STATE

- LSTM cell takes the previous memory state Ct-1 and does element wise multiplication with forget gate (f)

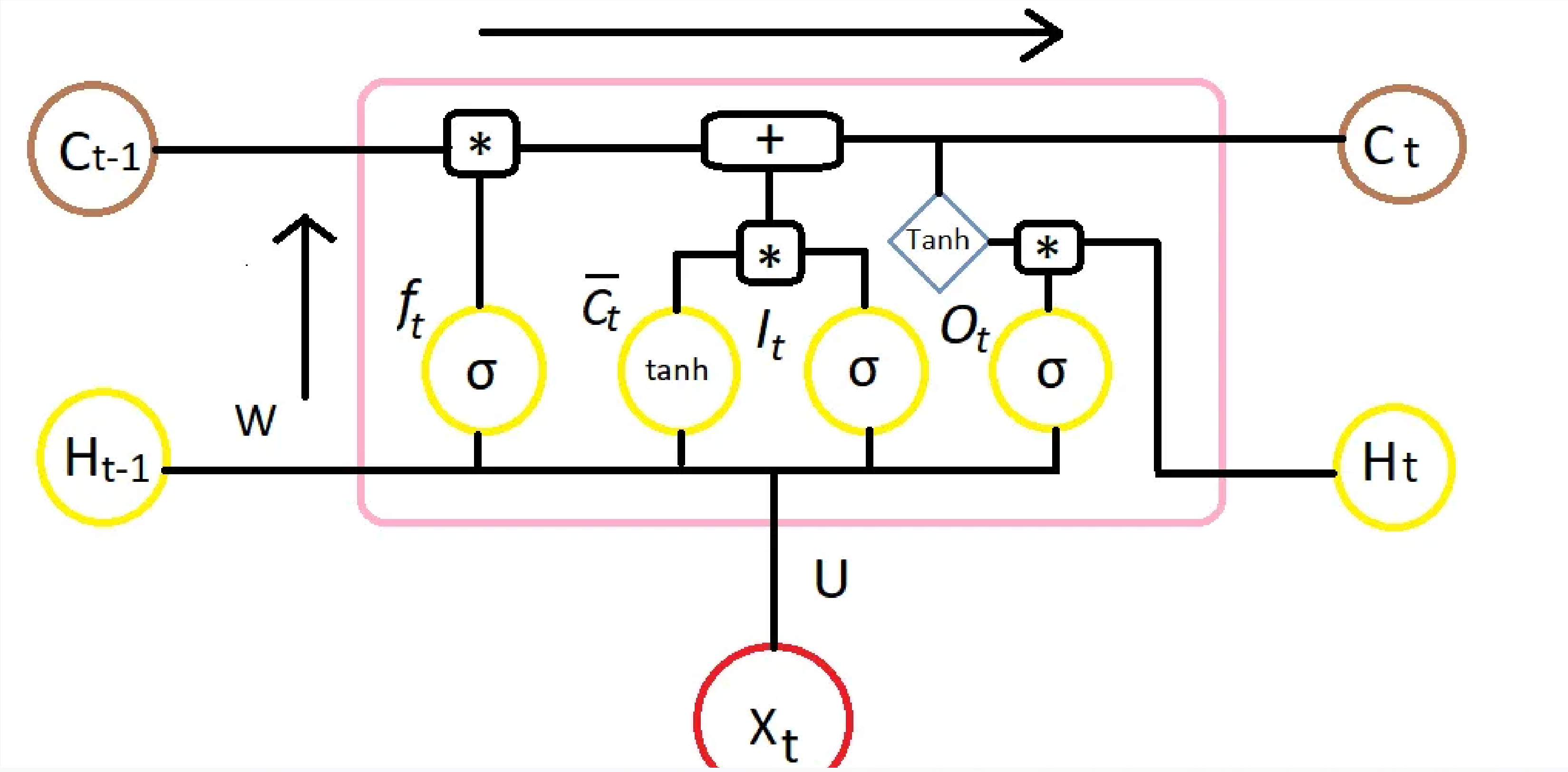
$$C_t = C_{t-1} * f_t$$

- If forget gate value is 0 then **previous memory state is completely forgotten.**
- If forget gate value is 1 then **previous memory state is completely passed to the cell.**
- **current memory state Ct-** calculate new memory state from input state and C layer.



OUTPUT

- Output will be based on our cell state Ct but will be a filtered version.
- Apply Tanh to Ct then we do element wise multiplication with the output gate O, that will be current hidden state Ht.
- Ht = Tanh(Ct)
- Pass these two Ct and Ht to the next time step and repeat the same process.



MEMORY STATE

This is the state where the memory (context) of input is stored-C

Ex : Mady walks in to the room, Monica also walks in to the room. Mady Said "hi" to ____??

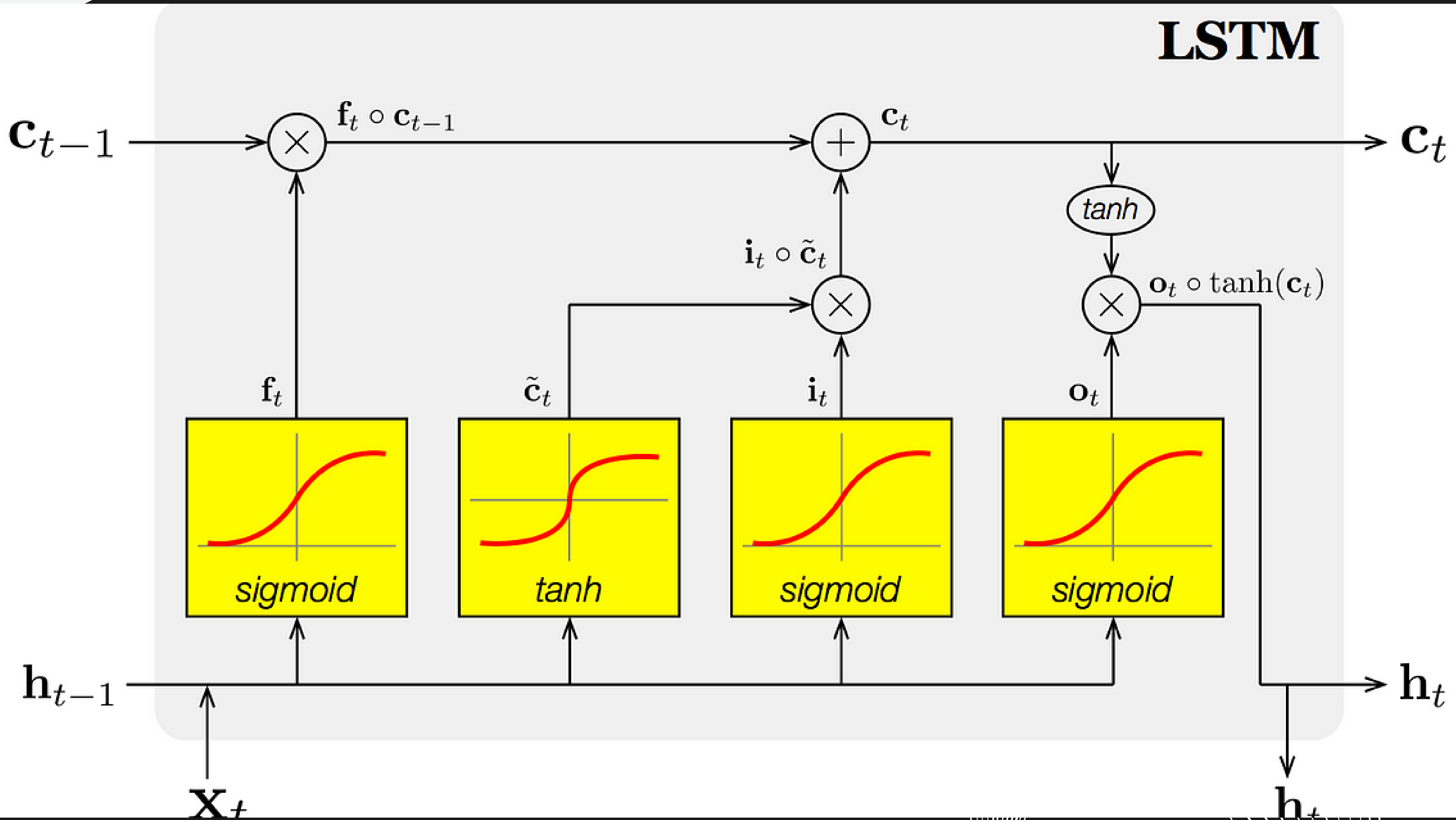
Inorder to predict correctly Here it stores "Monica" into memory C.

This state can be modified.

LSTM cell can add /remove the information.

Ex : Mady and Monica walk in to the room together , later Richard walks in to the room. Mady Said "hi" to ____??

The assumption making is memory might change from Monica to Richard.



Gating variables

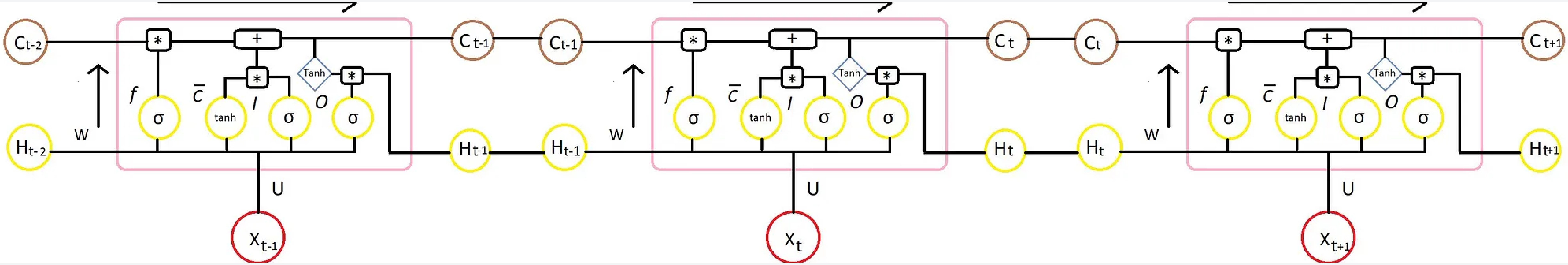
$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_t\right)$$
$$\mathbf{i}_t = \sigma\left(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i\right)$$
$$\mathbf{o}_t = \sigma\left(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o\right)$$

Candidate (memory) cell state

$$\tilde{\mathbf{c}}_t = \tanh\left(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}\right.$$

Cell & Hidden state

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t$$
$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t)$$


Ct-2
Ct-1
Ct-1
Ct
Ct
Ct+1
Ht-2
Ht-1
Ht-1
Ht
Ht
Ht+1
W
U
Xt-1
Xt
Xt+1
f
σ
tanh
Tanh
I
O

CONCLUSION

- The LSTM can delete or add information to the cell state, which is carefully controlled by structures called **gates.**
- **The cell state, represented by the horizontal line** at the top of the diagram, is crucial to LSTMs.
- An LSTM's repeating module is made up of four layers that interact with one another:
- Gates are a mechanism to selectively allow information to pass through. **A sigmoid neural net layer plus a pointwise multiplication operation** make them up.
- Using the forget gate, information to be forgotten is identified from a prior time step.
- Using input gate and tanh, new information is sought for updating cell state.
- The information from the two gates above is used to update the cell state.
- The output gate and the squashing operation provide useful information.

Applications

- Image captioning
- Machine translation
- Language modelling
- Handwriting generation
- Question answering chatbots