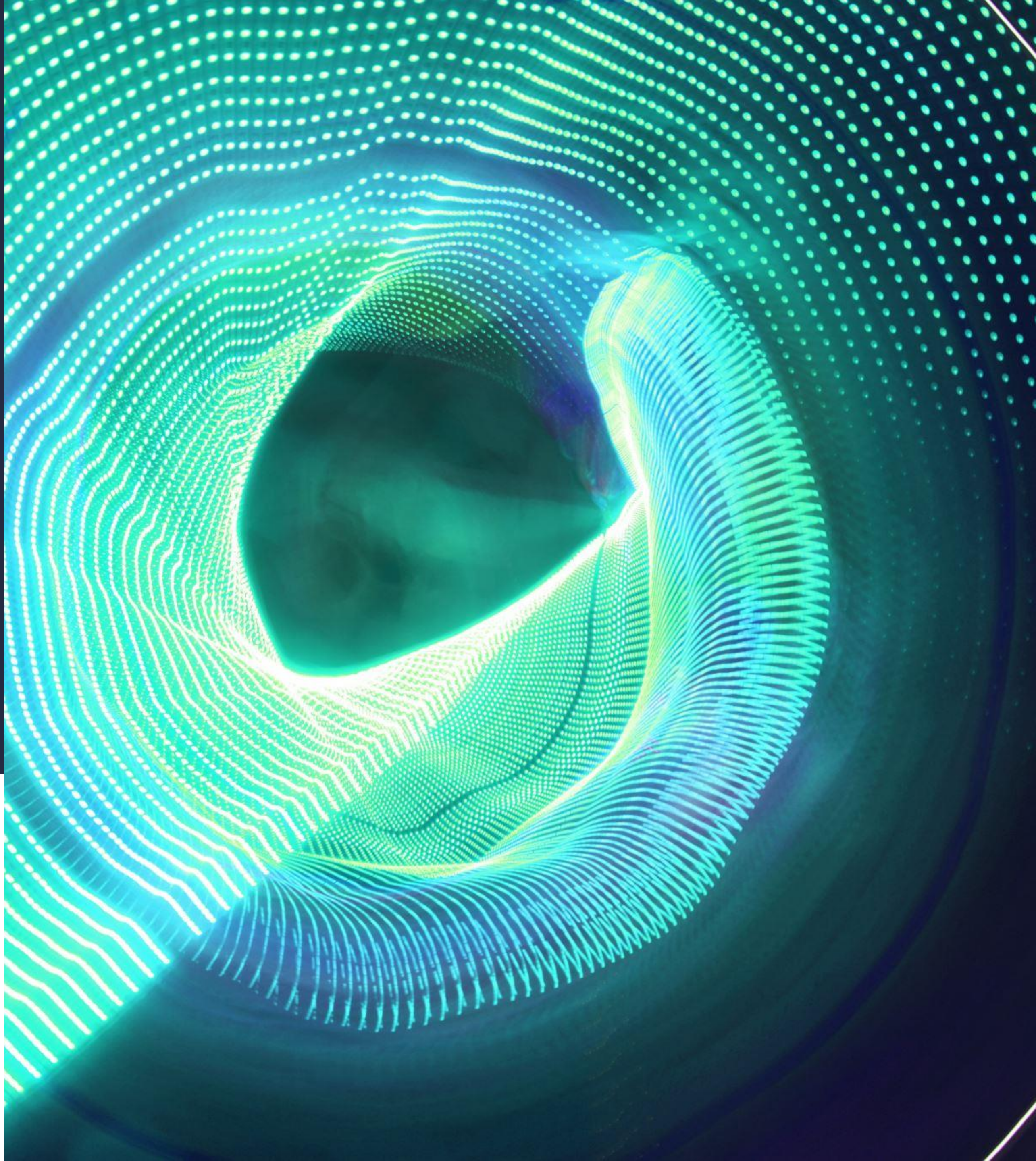
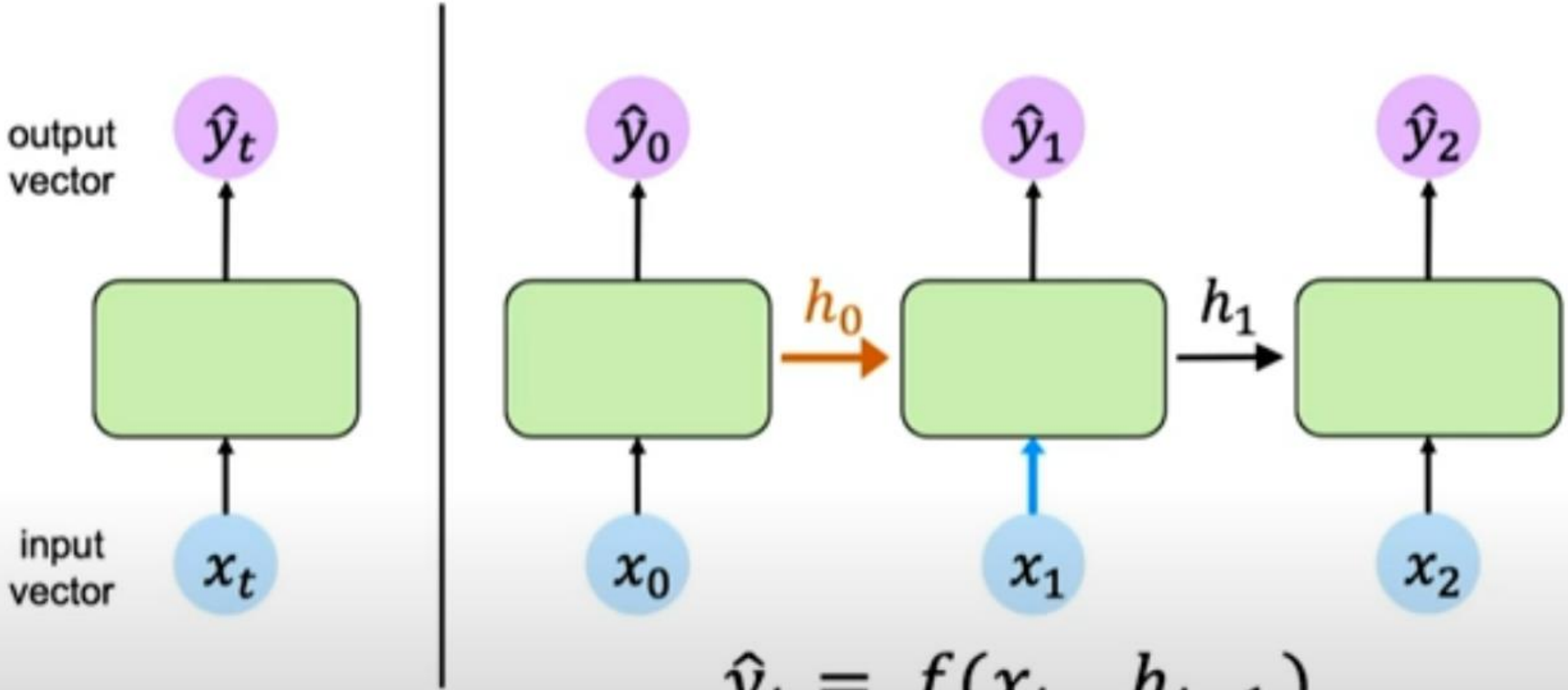
Long Short-Term Memory Network (LSTM)



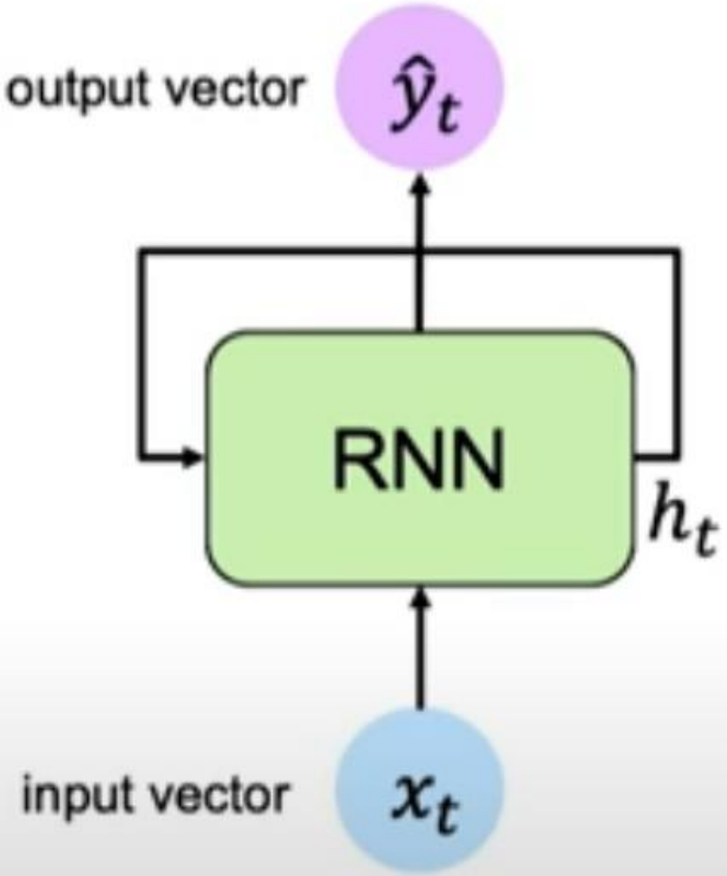
Pr Samira Douzi

Neurons with Recurrence



$$\underbrace{\hat{y}_t}_{\text{output}} = f(\underbrace{x_t}_{\text{input}},\ \underbrace{h_{t-1}}_{\text{past memory}})$$

RNN State Update and Output



Output Vector

$$\hat{y}_t = \boldsymbol{W}_{hy}^{T} h_t$$

Update Hidden State

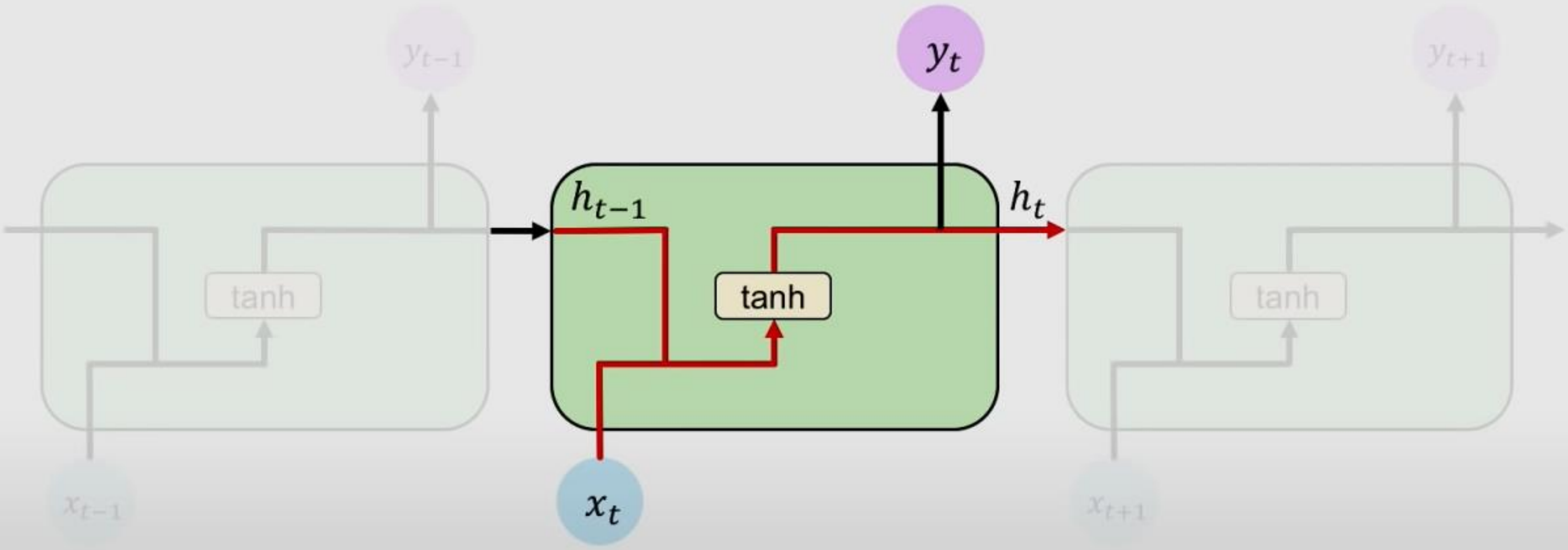
$$h_t = \tanh(\boldsymbol{W}_{hh}^{T} h_{t-1} + \boldsymbol{W}_{xh}^{T} x_t)$$

Input Vector

$$x_t$$

Standard RNN

In a standard RNN, repeating modules contain a **simple computation node**

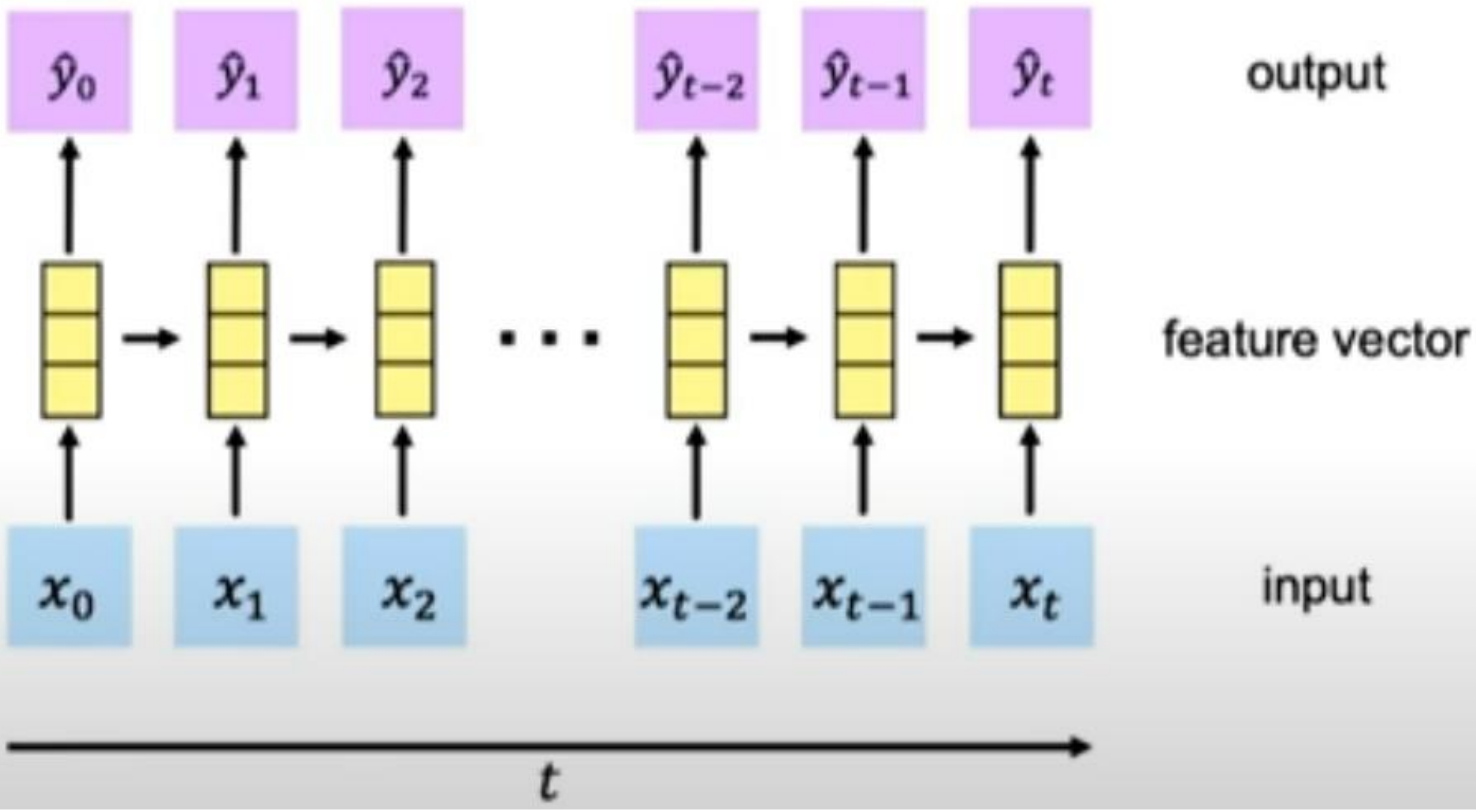


Goal of Sequence Modeling

RNNs: recurrence to model sequence dependencies

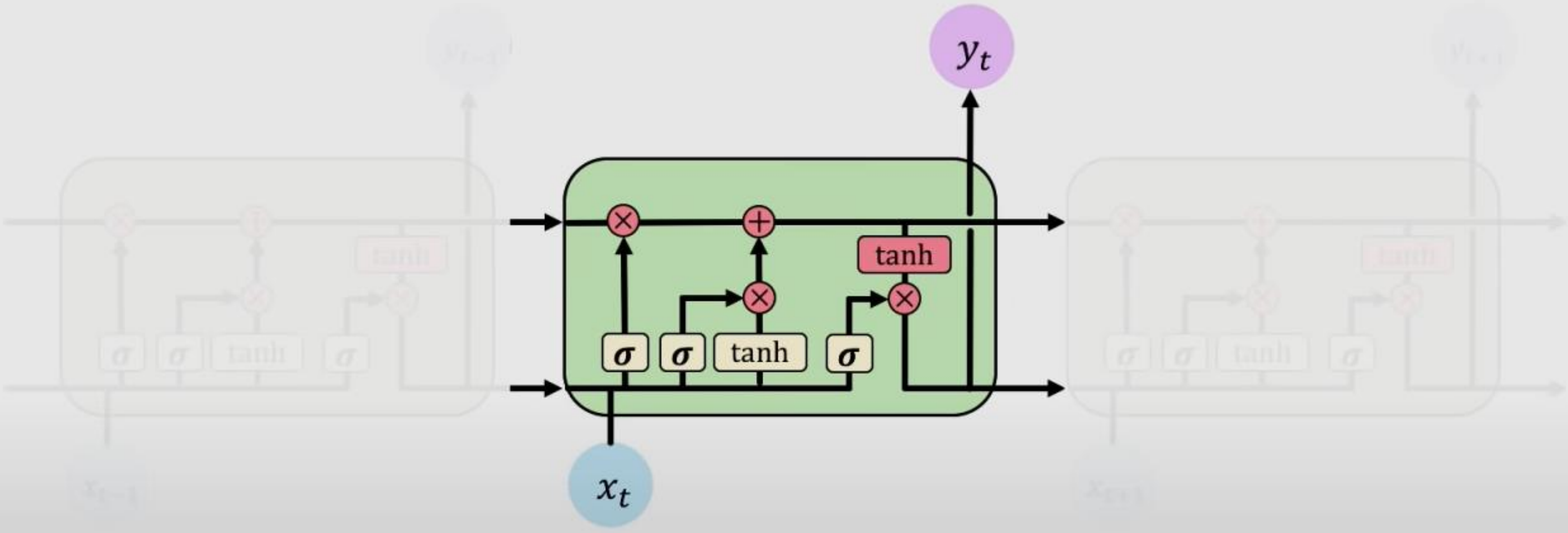
Limitations of RNNs

- Encoding bottleneck
- Slow, no parallelization
- Not long memory



Long Short Term Memory (LSTMs)

LSTM modules contain **computational blocks** that **control information flow**

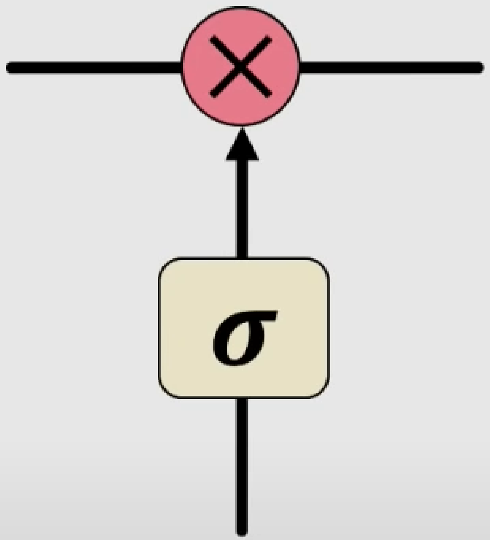


LSTM cells are able to track information throughout many timesteps

```
tf.keras.layers.LSTM(num_units)
```

Long Short Term Memory (LSTMs)

Information is **added** or **removed** through structures called **gates**

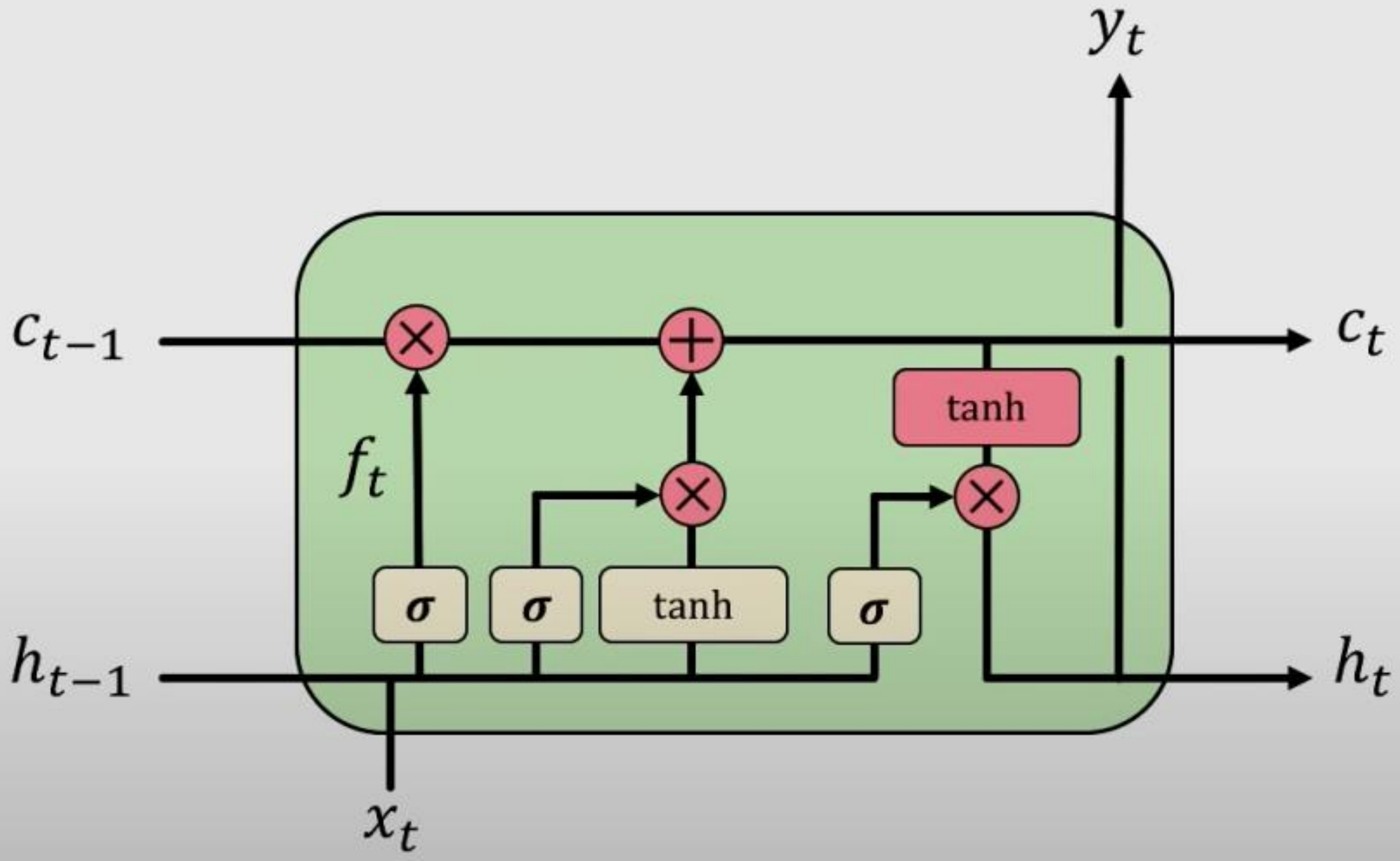


Gates optionally let information through, for example via a sigmoid neural net layer and pointwise multiplication

Long Short Term Memory (LSTMs)

How do LSTMs work?

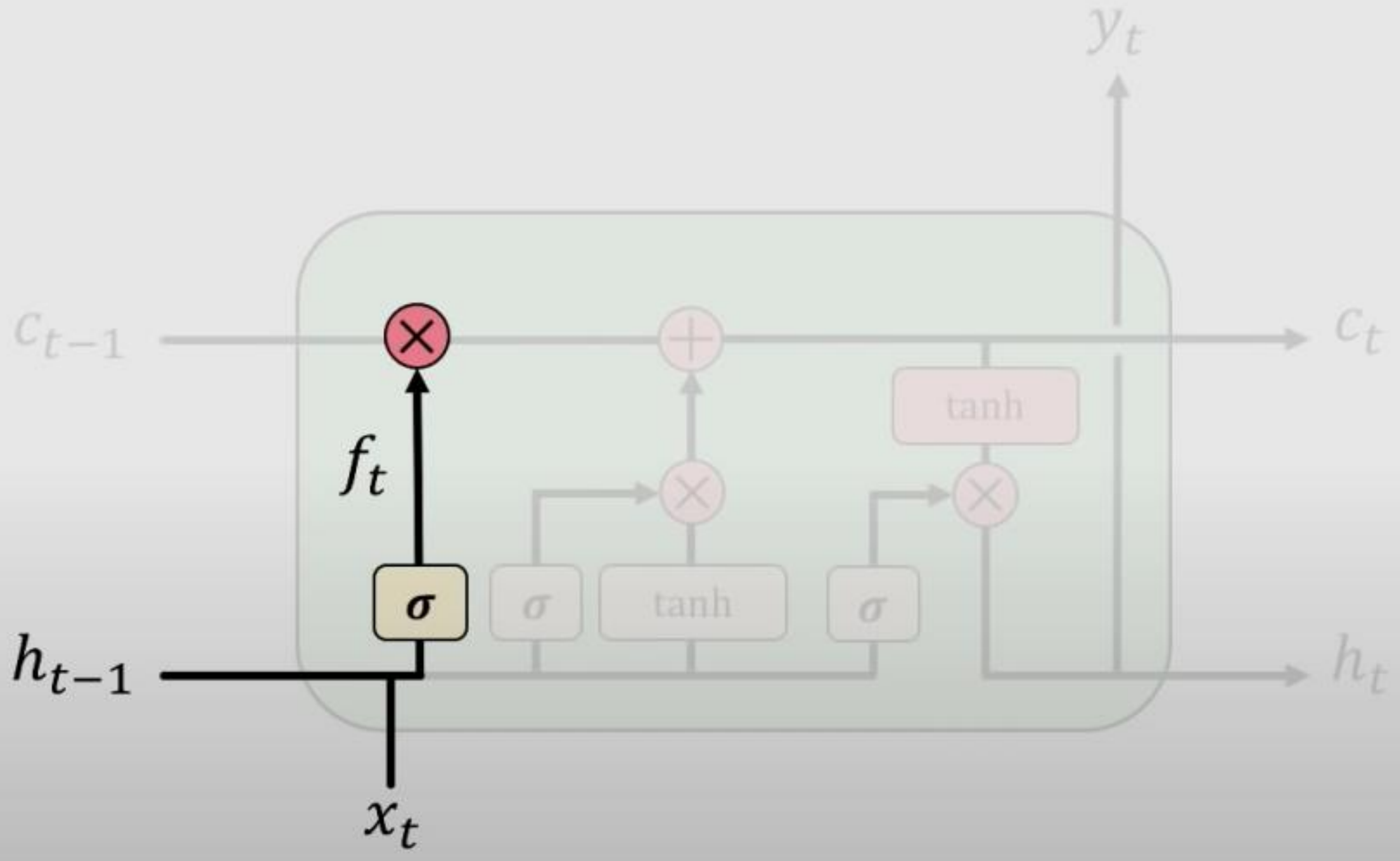
1) Forget   2) Store   3) Update   4) Output



Long Short Term Memory (LSTMs)

1) Forget 2) Store 3) Update 4) Output

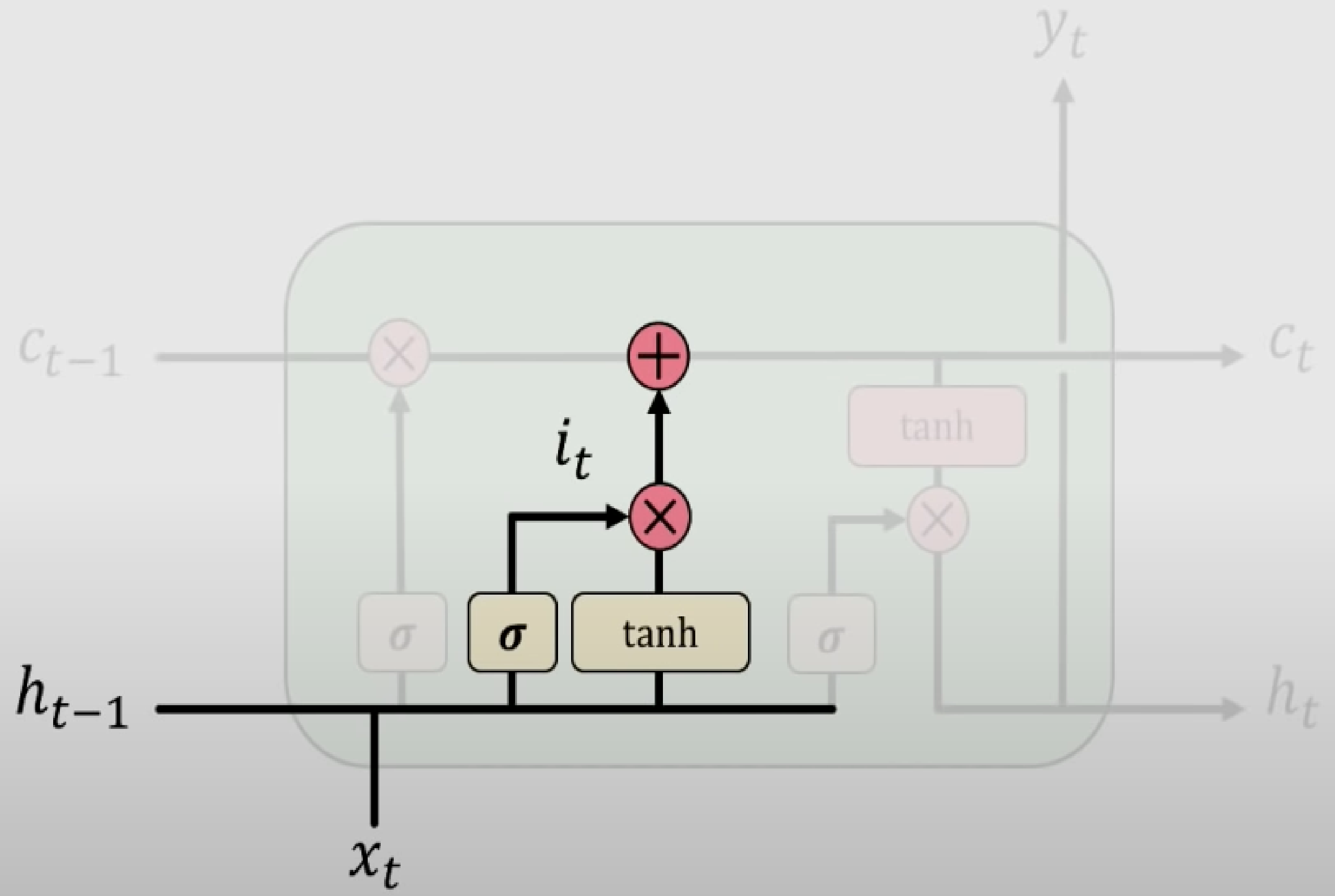
LSTMs **forget irrelevant** parts of the previous state



Long Short Term Memory (LSTMs)

1) Forget **2) Store** 3) Update 4) Output

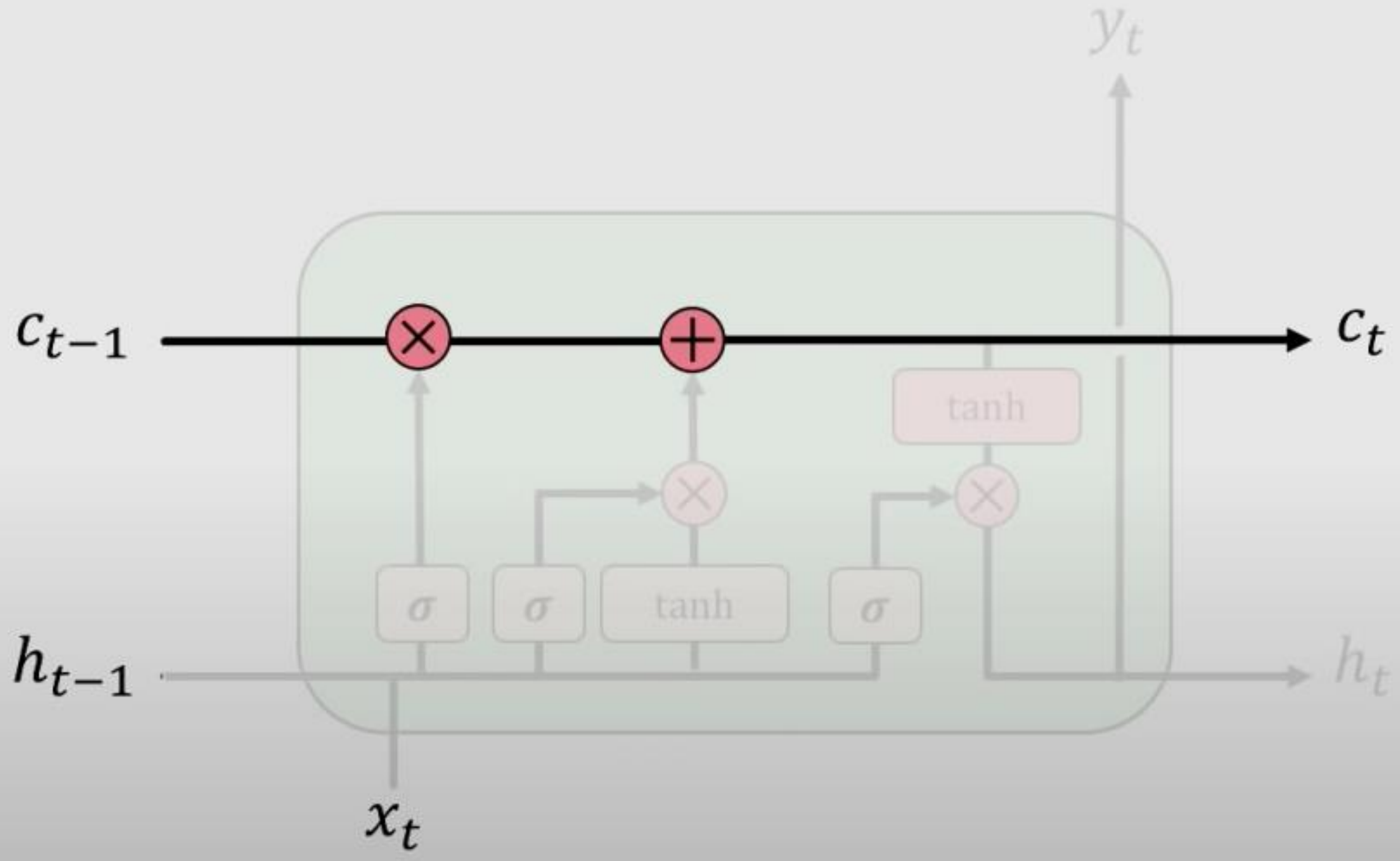
LSTMs **store relevant** new information into the cell state



Long Short Term Memory (LSTMs)

1) Forget 2) Store **3) Update** 4) Output

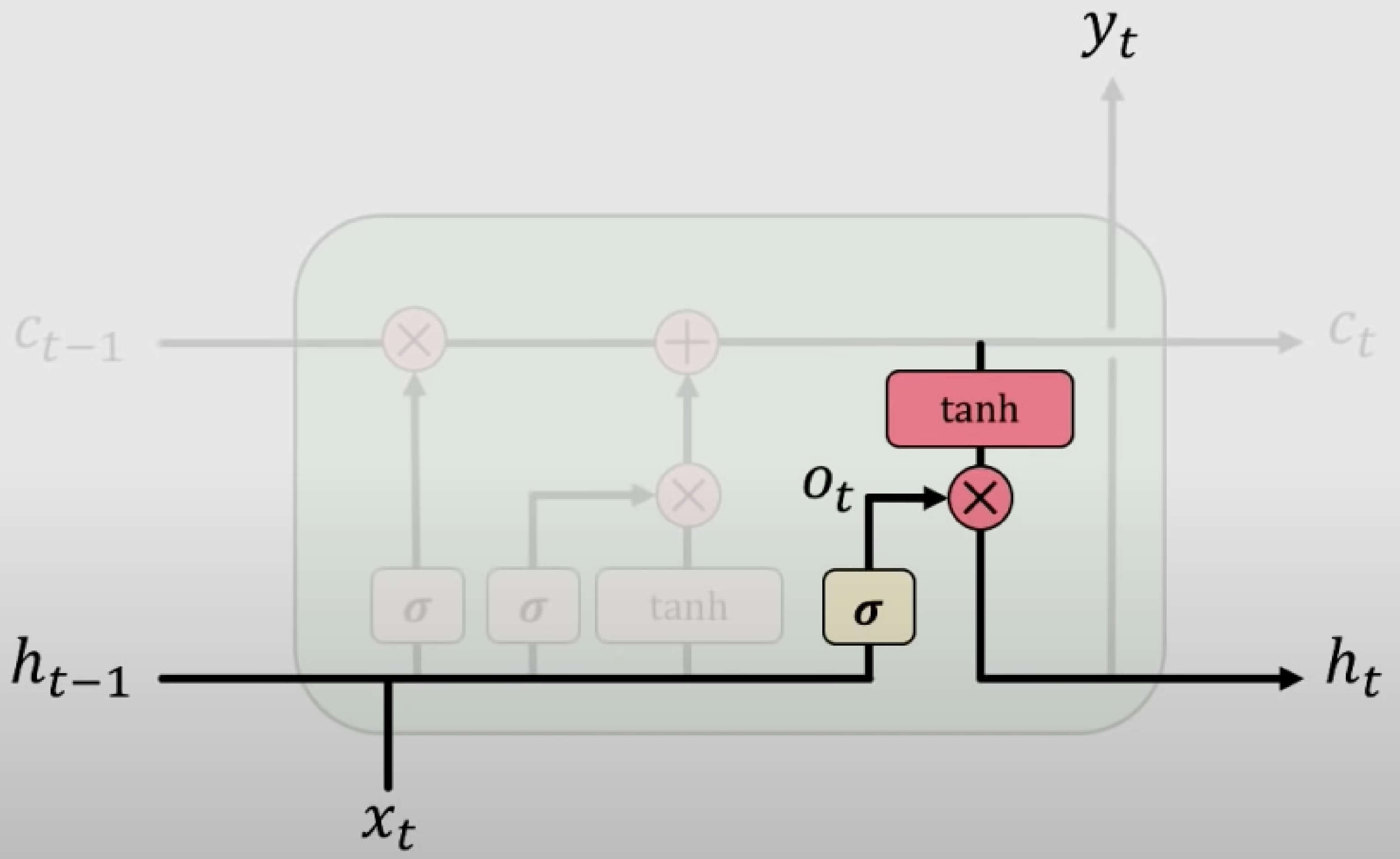
LSTMs **selectively update** cell state values

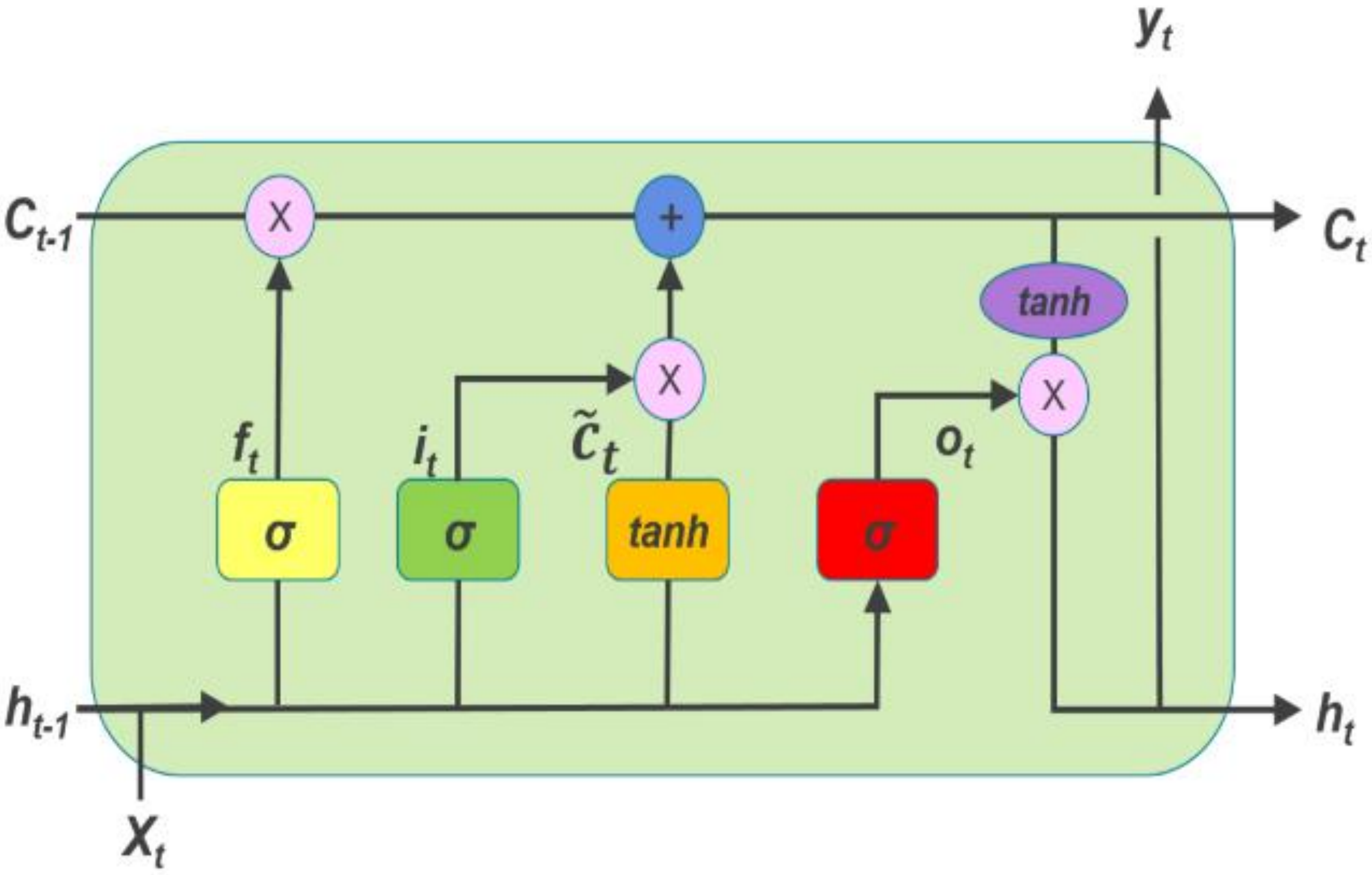


Long Short Term Memory (LSTMs)

1) Forget 2) Store 3) Update **4) Output**

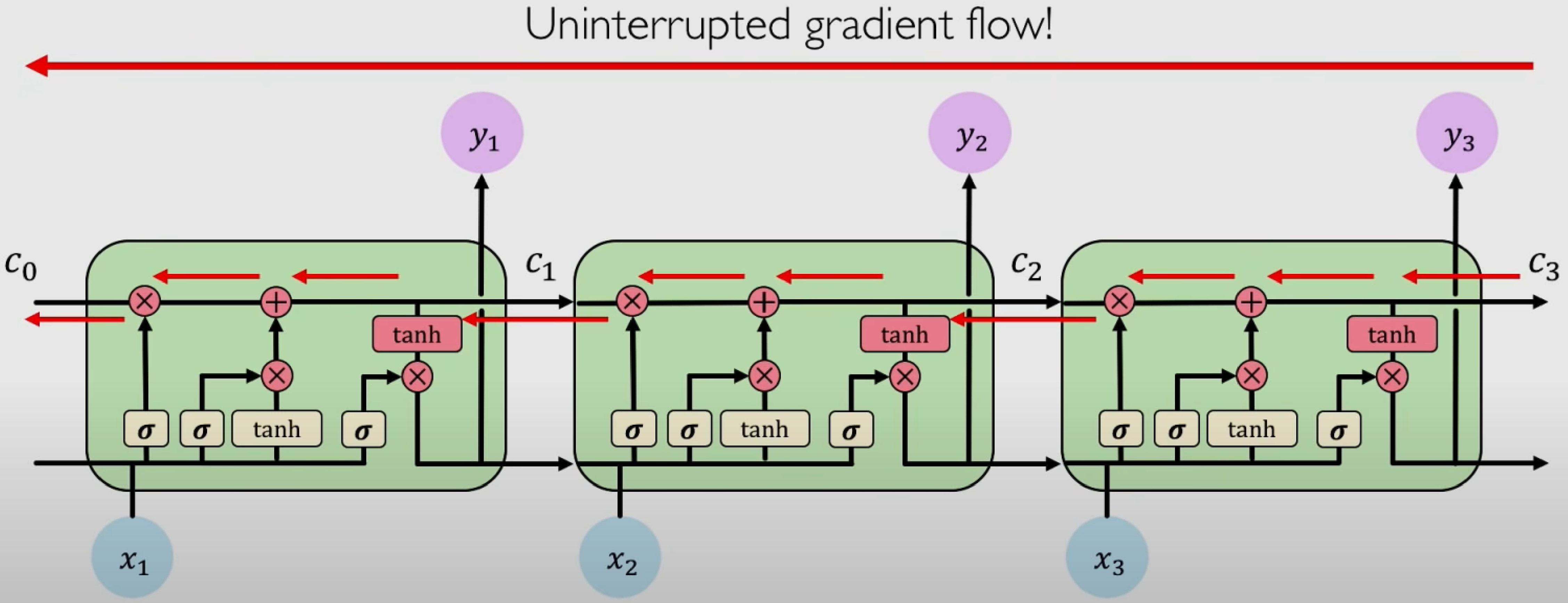
The **output gate** controls what information is sent to the next time step





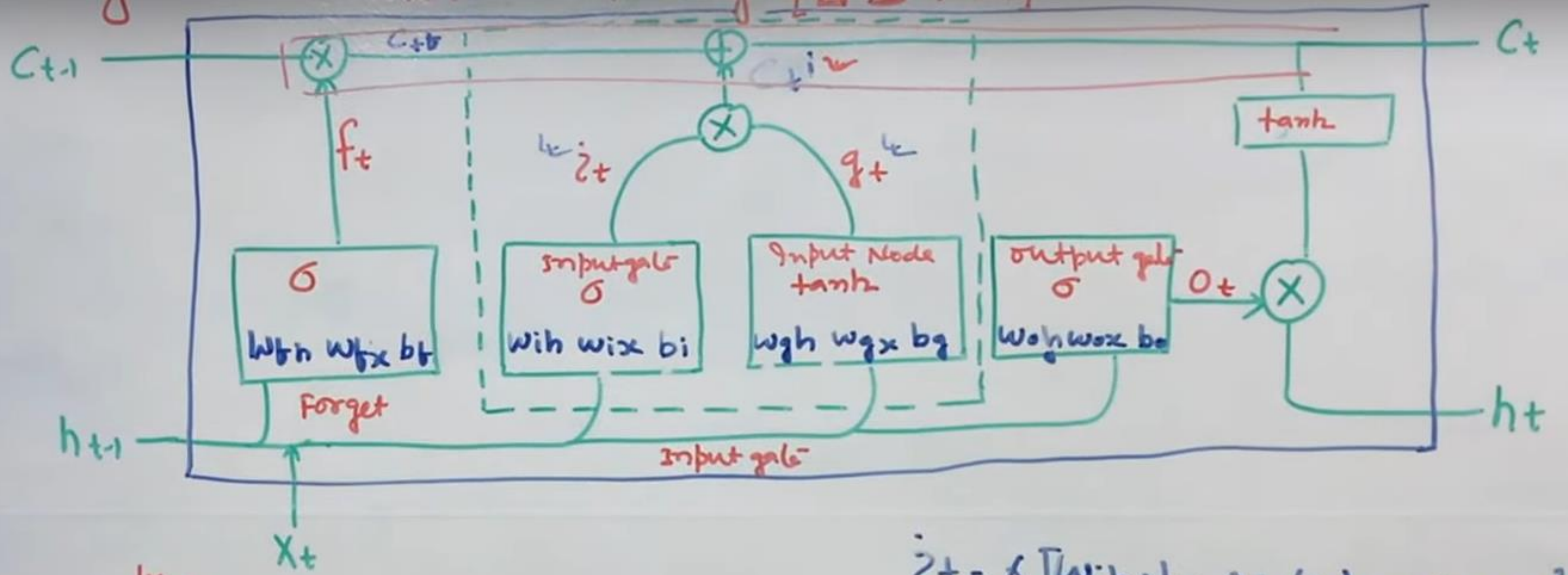
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * tanh(C_t)$$

LSTM Gradient Flow



LSTMs: Key Concepts

1. Maintain a **separate cell state** from what is outputted
2. Use **gates** to control the **flow of information**
   - **Forget** gate gets rid of irrelevant information
   - **Store** relevant information from current input
   - Selectively **update** cell state
   - **Output** gate returns a filtered version of the cell state
3. Backpropagation through time with **uninterrupted gradient flow**



l State

t+1

h_{t+1}

uation

State

$$f_t = \sigma\left[(W_{fh} * h_{t-1}) + (W_{fx} * x_t) + b_f\right]$$

$$C_t^f = C_{t-1} * f_t$$ same dimensionality

$$i_t = \sigma\left[(W_{ih} * h_{t-1}) + (W_{ix} * X_t) + b\right.$$

$$g_t = \tanh\left[(W_{gh} * h_{t-1}) + (W_{gx} * X_t) +\right.$$

$$C_t^i = i_t * g_t$$

$$\boxed{C_t = C_t^i + C_t^f}$$

$$O_t = \sigma\left[(W_{oh} * h_{t-1}) + (W_{ox} * x_t) + b_o\right]$$

n Memory

$$\boxed{h_t = \tanh(C_t) * O_t}$$