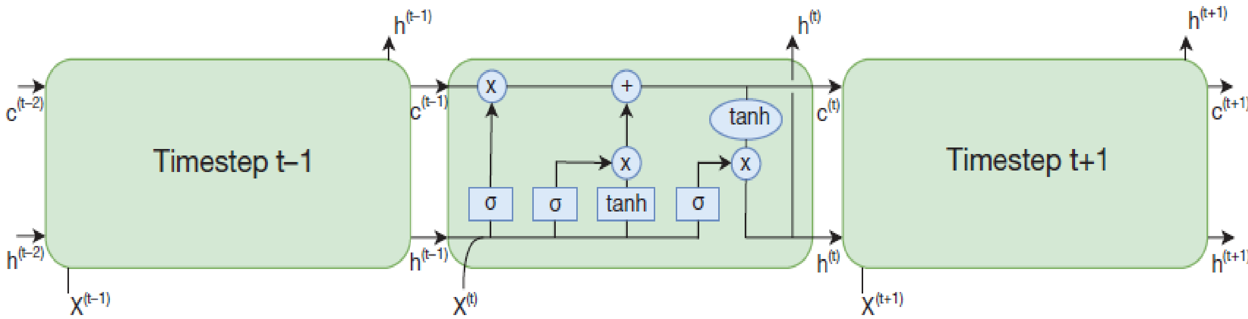
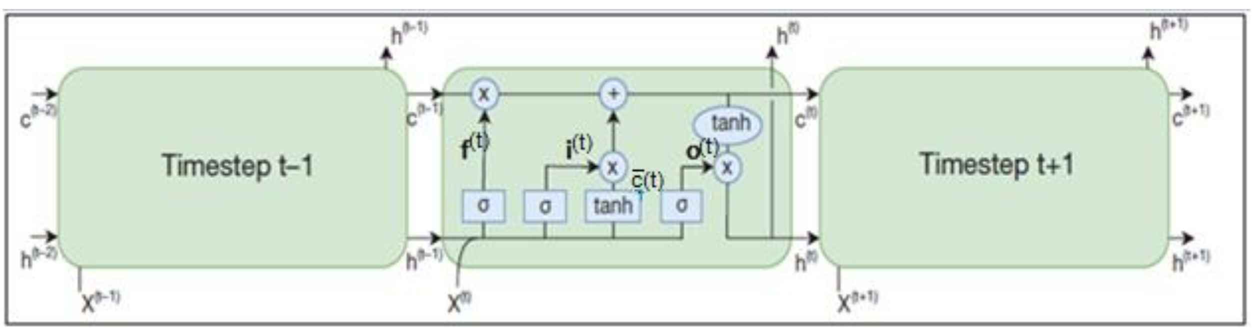
LSTM Networks (draw second diagram alone)

1. Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies.
2. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior
3. The key to LSTMs is the cell state, the horizontal line running through the top of the diagram
4. The LSTM have the ability to remove or add information to the cell state, $\tilde{C}_t$ carefully regulated by structures called gates.





$$f^{(t)} = \sigma\left(w_f\left[h^{(t-1)}, x^{(t)}\right] + b_f\right) \quad (1)$$

$$i^{(t)} = \sigma\left(w_i\left[h^{(t-1)}, x^{(t)}\right] + b_i\right) \quad (2)$$

$$\tilde{C}^{(t)} = tanh\left(w_c\left[h^{(t-1)}, x^{(t)}\right] + b_c\right) \quad (3)$$

$$C^{(t)} = f^{(t)} * C^{(t-1)} + i^{(t)} * \tilde{C}^{(t)} \quad (4)$$

$$o^{(t)} = \sigma\left(w_o\left[h^{(t-1)}, x^{(t)}\right] + b_o\right) \quad (5)$$

$$h^{(t)} = o^{(t)} * tanh\left(C^{(t)}\right) \quad (6)$$

An LSTM has three of these gates, to protect and control the cell state.

Forget gate:

- The first step in our LSTM is to decide what information needs to be removed from the cell state.
- This decision is made by a sigmoid layer called the "forget gate layer **f**." It looks at $\mathbf{h_{t-1}}$ and $\mathbf{x_t}$, and outputs a number between 0 and 1 for each number in the cell state $\mathbf{C_{t-1}}$.
- A 1 represents "completely keep this" while a 0 represents "completely get rid of this".

Update gate:

- The next step is to decide what new information needs to be store in the cell state.
- This has two parts. First, a sigmoid layer called the "input gate layer **i**" decides which values need to update.
- Next, a tanh layer creates a vector of candidate values, $\tilde{C}_t$, that could be added to the state. In the next step, **i** add $\tilde{C}_t$ are combined to create an update to the state
- It's now time to update the old cell state, $\mathbf{C_{t-1}}$, into the new cell state $\mathbf{C_t}$.
- Multiplying the old state by **f**, forgetting the information that decided to forget earlier. Then we add $\mathbf{i}*\tilde{C}_t$. This is the new $\mathbf{C_t}$ values, scaled by how much we decided to update each state value

Output gate:

- Finally, the output will be based on our cell state, but will be a filtered version.
- First, sigmoid layer decides which parts of the cell state that are going to output. Then, put the cell state through tanh (to push the

values to be between −1 and 1) and multiply it by the output of the sigmoid gate, so that the output is obtained.