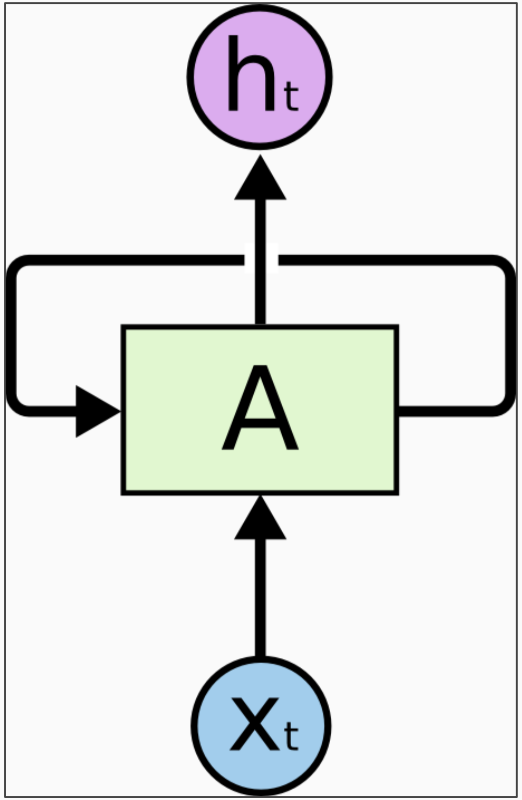
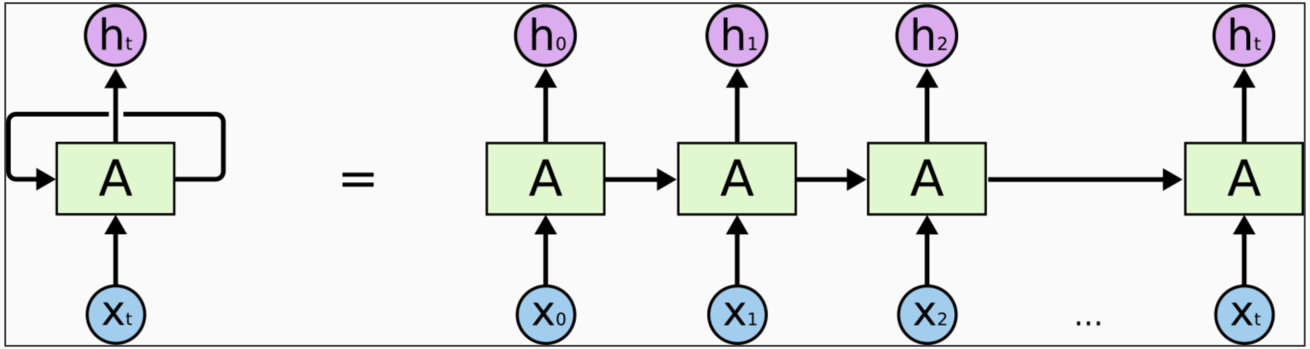
LSTM

Recurrent Neural Networks



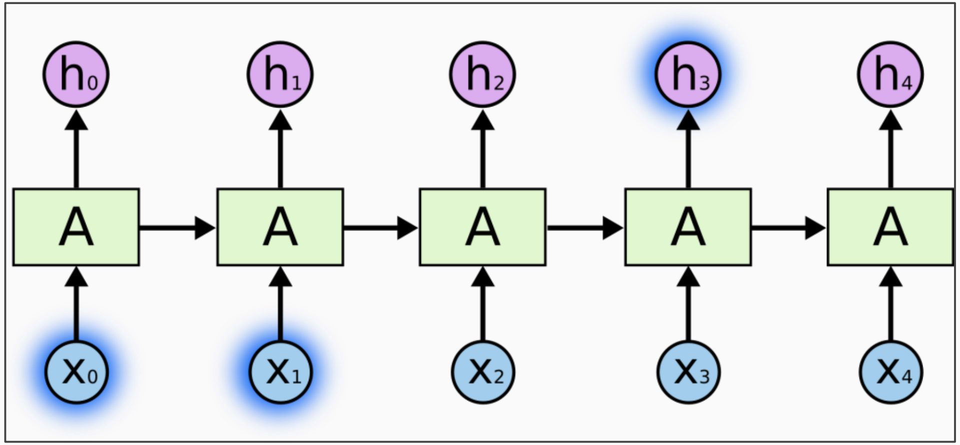
Recurrent Neural Networks have loops.

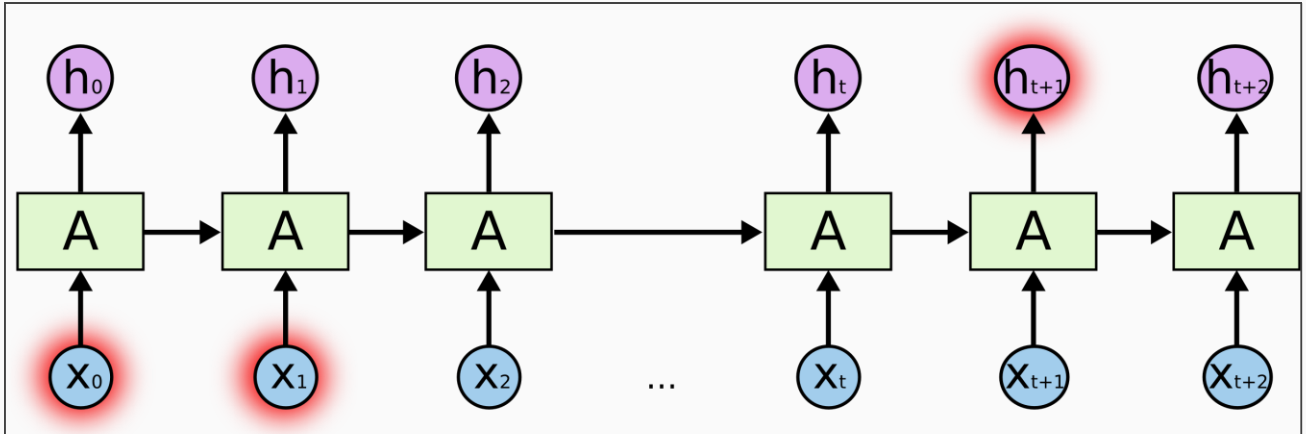


An unrolled recurrent neural network.

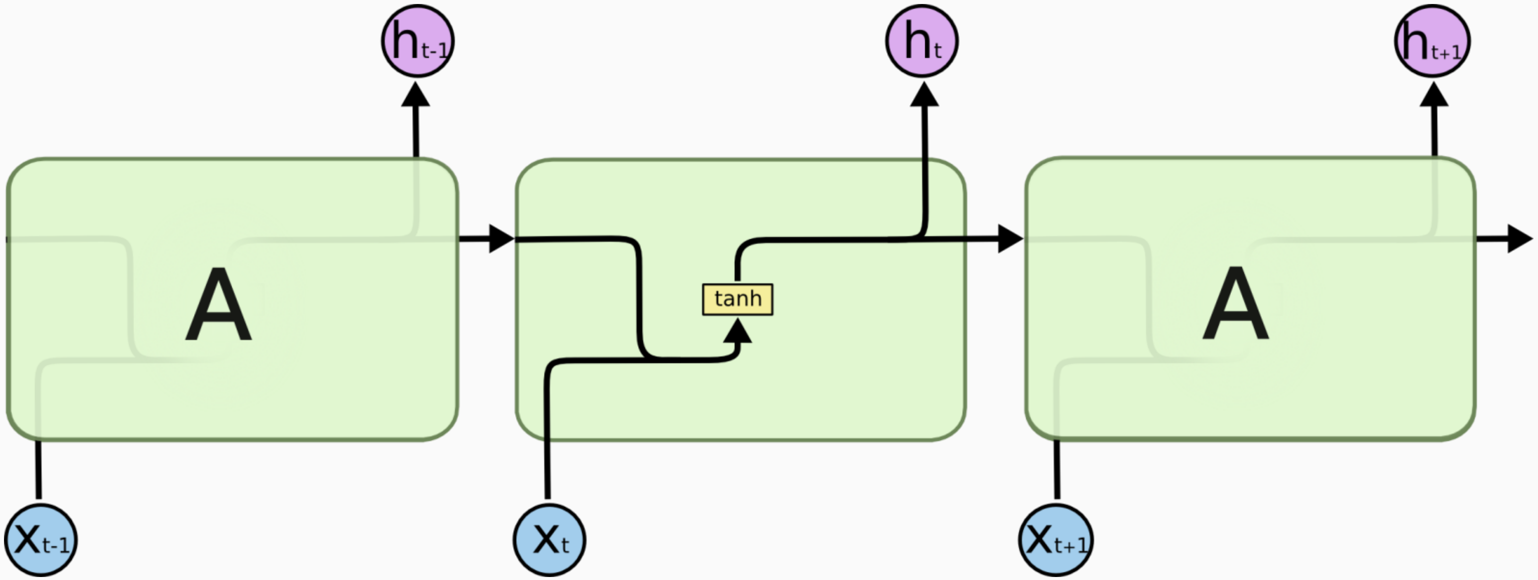
LSTM

The Problem of Long-Term Dependencies



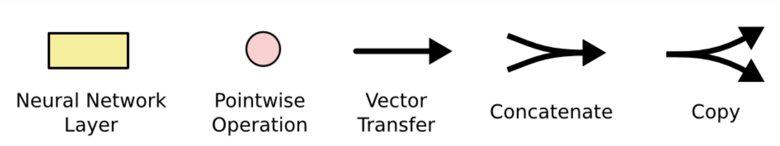


RNN Networks



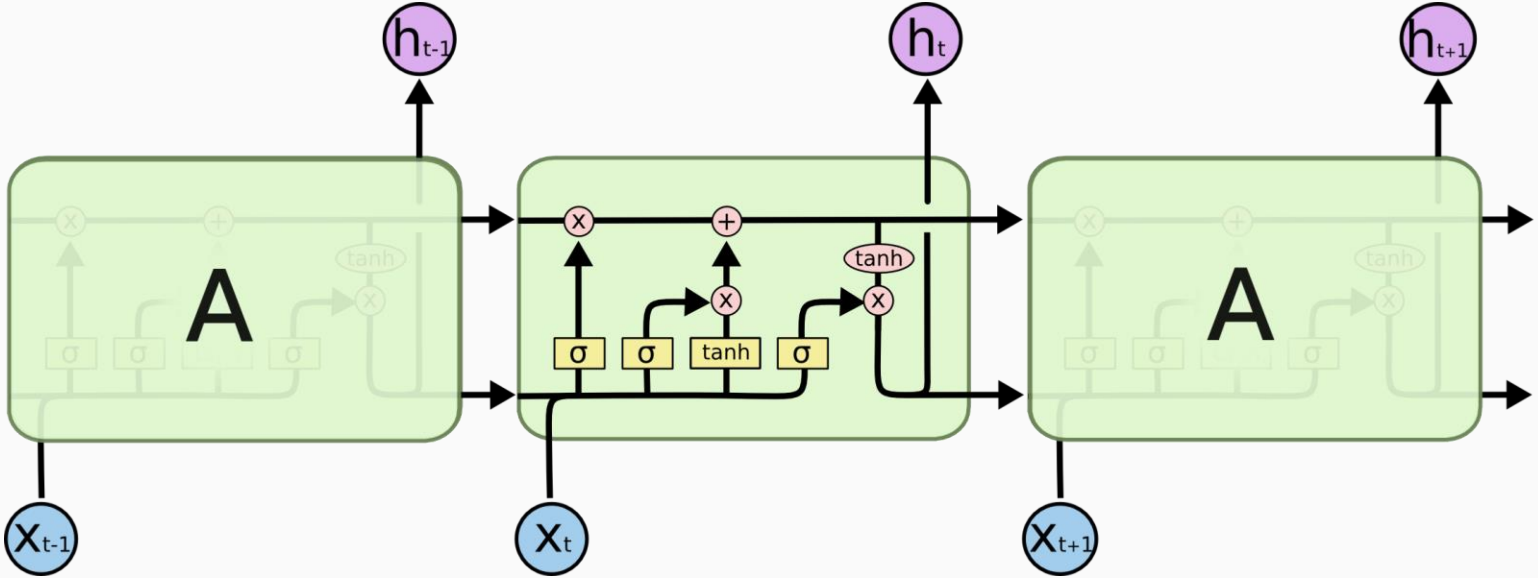
The repeating module in a standard RNN contains a single layer.

LSTM



LSTM Networks

Long Short Term Memory networks– are a special kind of RNN

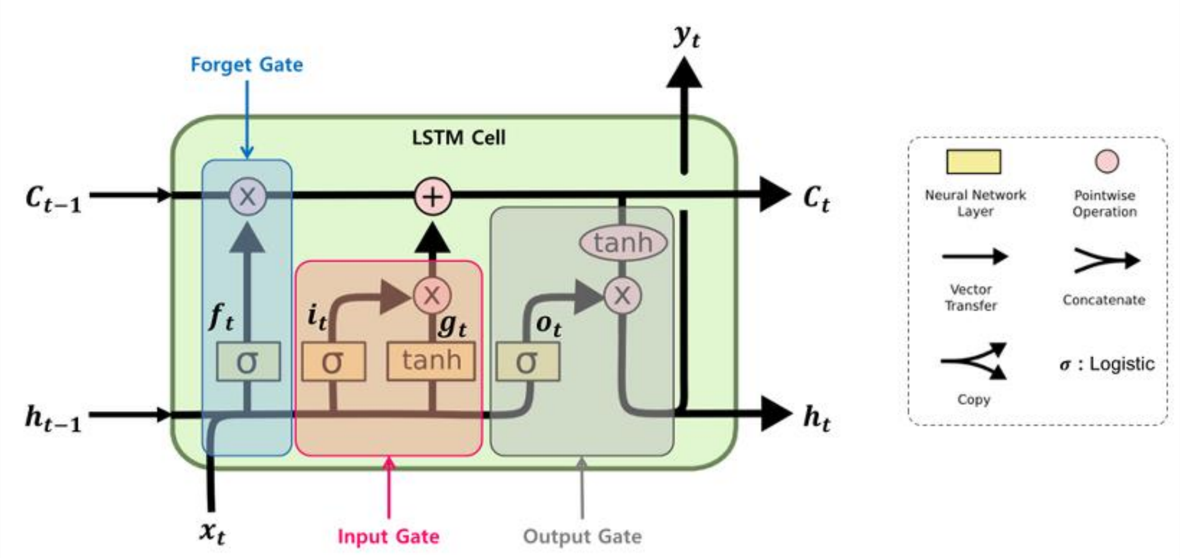


The repeating module in an LSTM contains four interacting layers.

LSTM

LSTM Networks

Long Short Term Memory networks– are a special kind of RNN



- **Forget gate** : $\mathbf{f}_t$에 의해 제어되며 장기 상태 $\mathbf{c}_t$의 어느 부분을 삭제할지 제어한다.
- **Input gate** : $\mathbf{i}_t$에 의해 제어되며 $\mathbf{g}_t$의 어느 부분이 장기 상태 $\mathbf{c}_t$에 더해져야 하는지 제어한다.
- **Output gate** : $\mathbf{o}_t$는 장기 상태 $\mathbf{c}_t$의 어느 부분을 읽어서 $\mathbf{h}_t$ 와 $\mathbf{y}_t$로 출력해야 하는지 제어한다.

LSTM

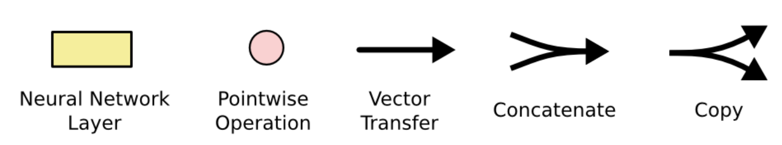
LSTM Networks

Long Short Term Memory networks– are a special kind of RNN

- **Forget gate** : $\mathbf{f}_t$에 의해 제어되며 장기 상태 $\mathbf{c}_t$의 어느 부분을 삭제할지 제어한다.
- **Input gate** : $\mathbf{i}_t$에 의해 제어되며 $\mathbf{g}_t$의 어느 부분이 장기 상태 $\mathbf{c}_t$에 더해져야 하는지 제어한다.
- **Output gate** : $\mathbf{o}_t$는 장기 상태 $\mathbf{c}_t$의 어느 부분을 읽어서 $\mathbf{h}_t$ 와 $\mathbf{y}_t$로 출력해야 하는지 제어한다.

$$
\begin{aligned}
\mathbf{f}_t &= \sigma\left(\mathbf{W}_{xf}^T \cdot \mathbf{x}_t + \mathbf{W}_{hf}^T \cdot \mathbf{h}_{t-1} + \mathbf{b}_f\right) \\
\mathbf{i}_t &= \sigma\left(\mathbf{W}_{xi}^T \cdot \mathbf{x}_t + \mathbf{W}_{hi}^T \cdot \mathbf{h}_{t-1} + \mathbf{b}_i\right) \\
\mathbf{o}_t &= \sigma\left(\mathbf{W}_{xo}^T \cdot \mathbf{x}_t + \mathbf{W}_{ho}^T \cdot \mathbf{h}_{t-1} + \mathbf{b}_o\right) \\
\mathbf{g}_t &= \tanh\left(\mathbf{W}_{xg}^T \cdot \mathbf{x}_t + \mathbf{W}_{hg}^T \cdot \mathbf{h}_{t-1} + \mathbf{b}_g\right) \\
\mathbf{c}_t &= \mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes \mathbf{g}_t \\
\mathbf{y}_t, \mathbf{h}_t &= \mathbf{o}_t \otimes \tanh(\mathbf{c}_t)
\end{aligned}
$$

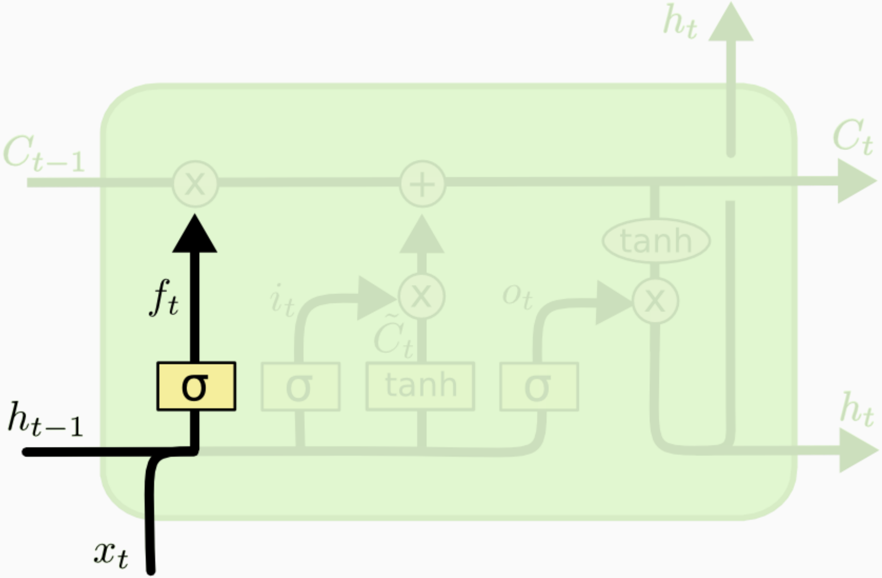
- $\mathbf{W}_{xf}, \mathbf{W}_{xi}, \mathbf{W}_{xo}, \mathbf{W}_{xg}$: 입력 벡터 $\mathbf{x}_t$에 연결된 네 개의 레이어에 대한 가중치 행렬
- $\mathbf{W}_{hf}, \mathbf{W}_{hi}, \mathbf{W}_{ho}, \mathbf{W}_{hg}$: 이전 타임스텝의 단기 상태 $\mathbf{h}_{t-1}$에 연결된 네 개의 레이어에 대한 가중치 행렬
- $\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_o, \mathbf{b}_g$: 네 개의 레이어에 대한 편향(bias), 텐서플로(TensorFlow)에서는 $\mathbf{b}_f$ 를 1 로 초기화하여 학습 시작시에 모든것을 잃어버리는 것을 방지한다.



Step-by-Step LSTM Walk Through

Long Short Term Memory networks– are a special kind of RNN

1. Decide how much information we're going to throw away from the cell state.



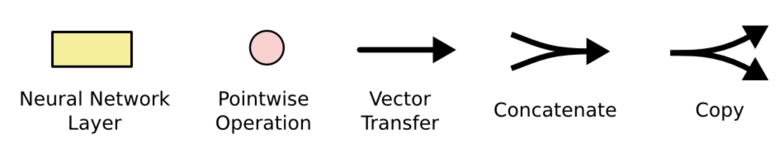
Forget gate

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

LSTM

Step-by-Step LSTM Walk Through

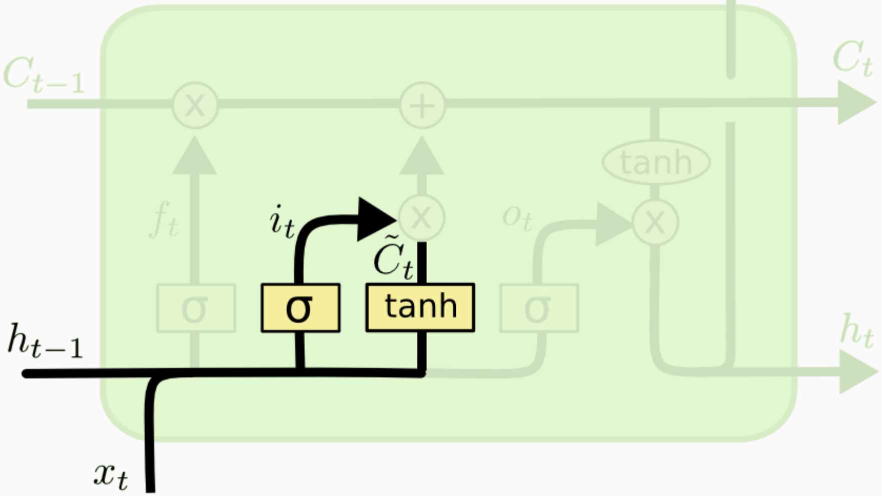
Long Short Term Memory networks– are a special kind of RNN



2. To decide what new information we're going to store in the cell state.

(1) Decide which values we'll update

(2) Combine these 2 to create an update to the state



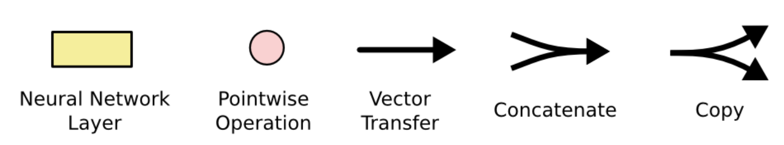
Input gate

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

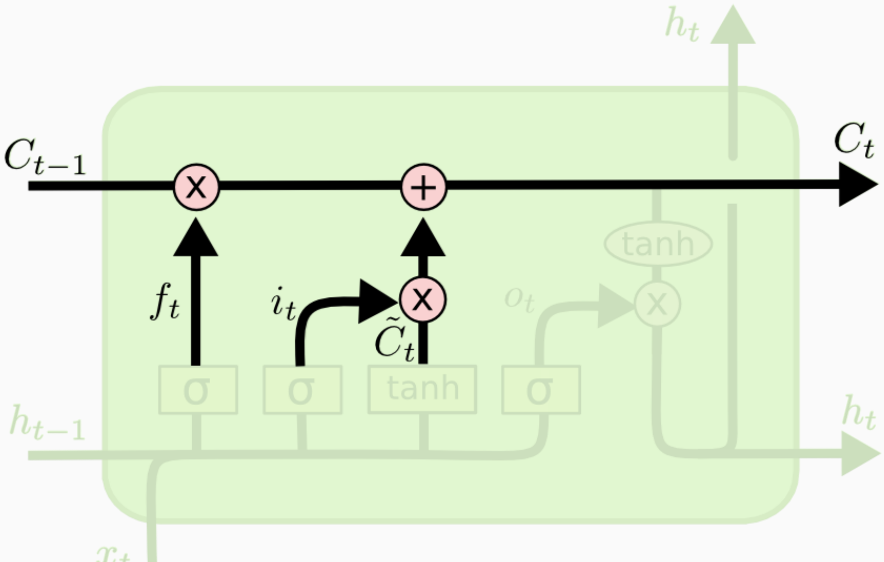
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Step-by-Step LSTM Walk Through

Long Short Term Memory networks– are a special kind of RNN



3. Time to update the old cell state, C_{t-1}, into the new cell state C_t

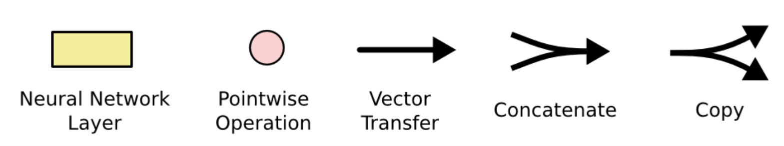


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

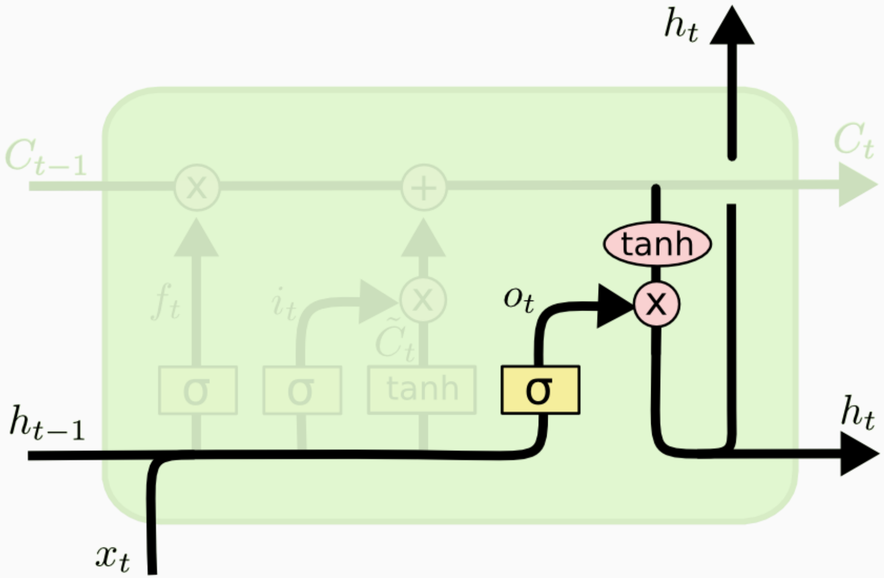
LSTM

Step-by-Step LSTM Walk Through

Long Short Term Memory networks– are a special kind of RNN



4. We need to decide what we're going to output.



Output gate

$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$