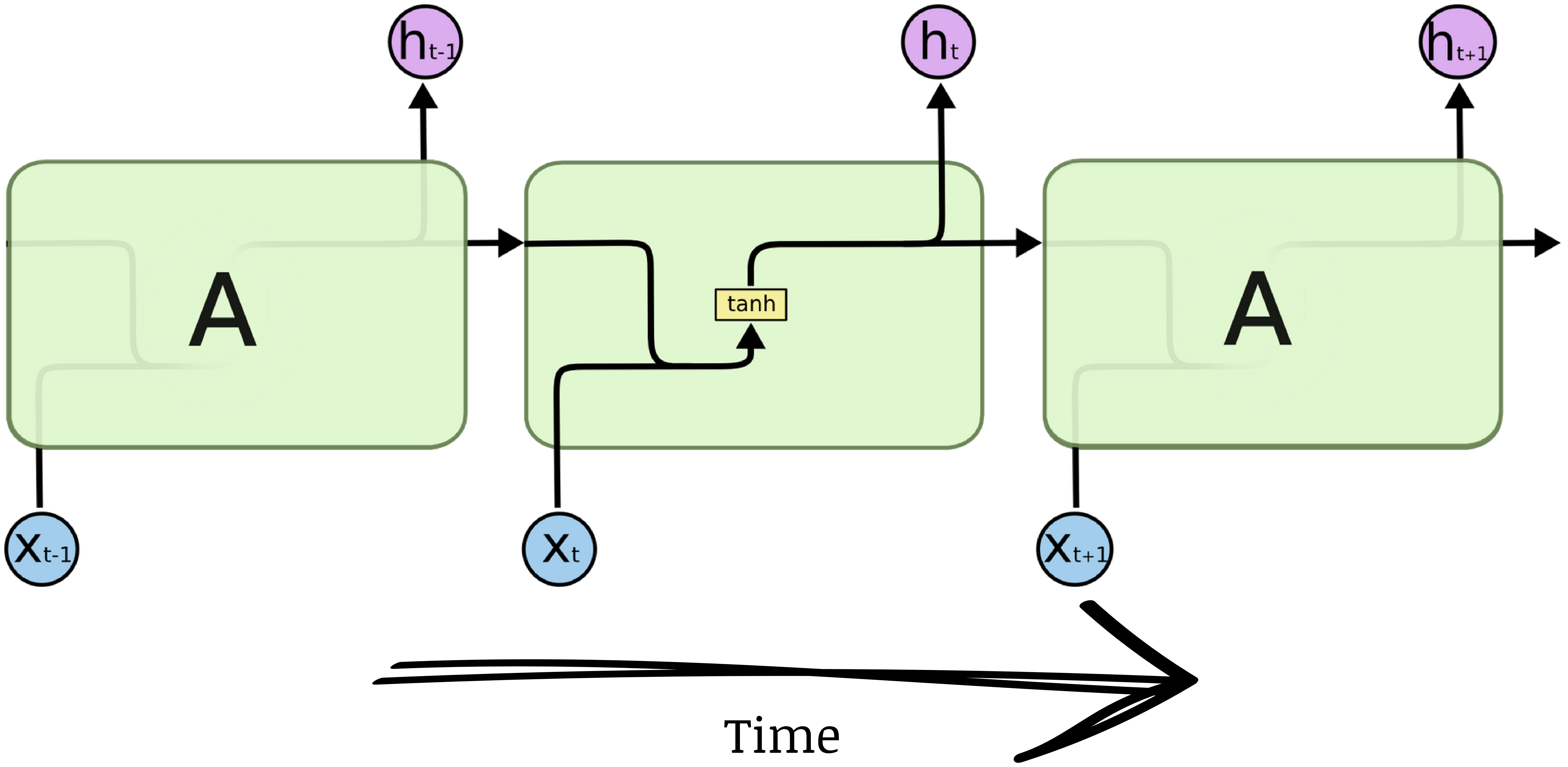
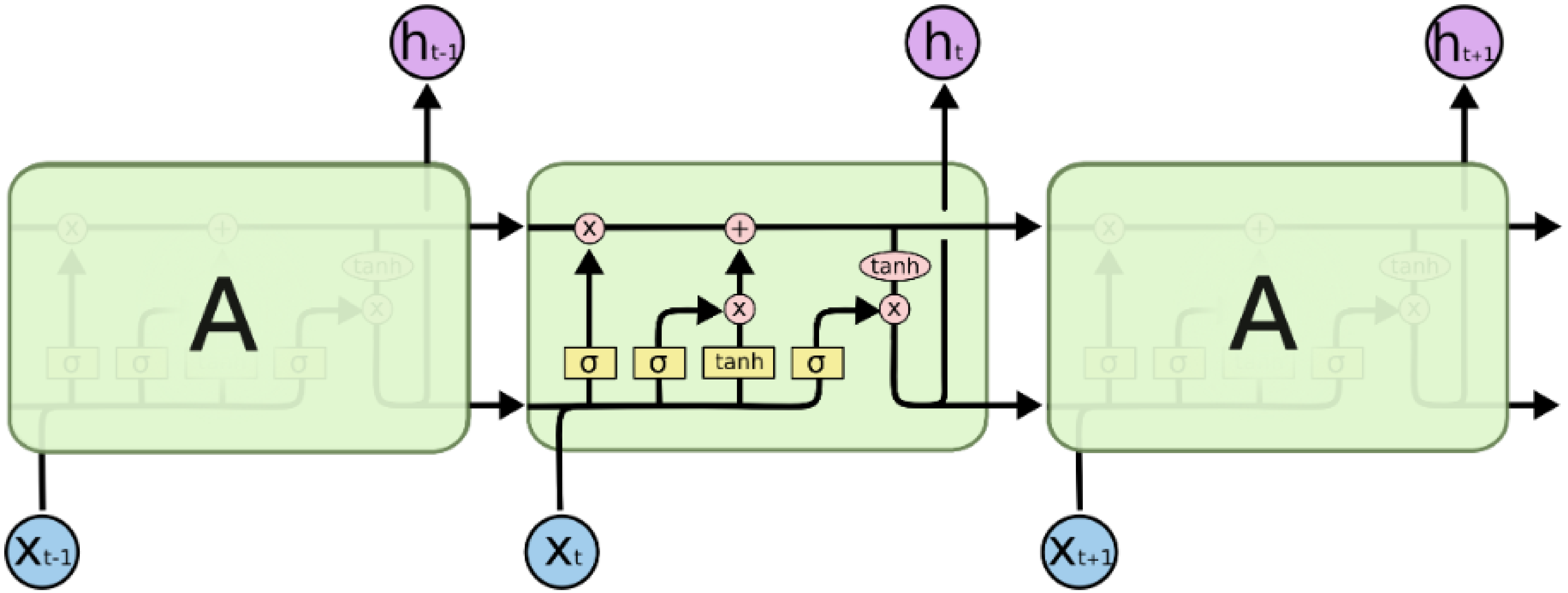
# Long Short Term Memory

# RNN Revisited



## LSTM

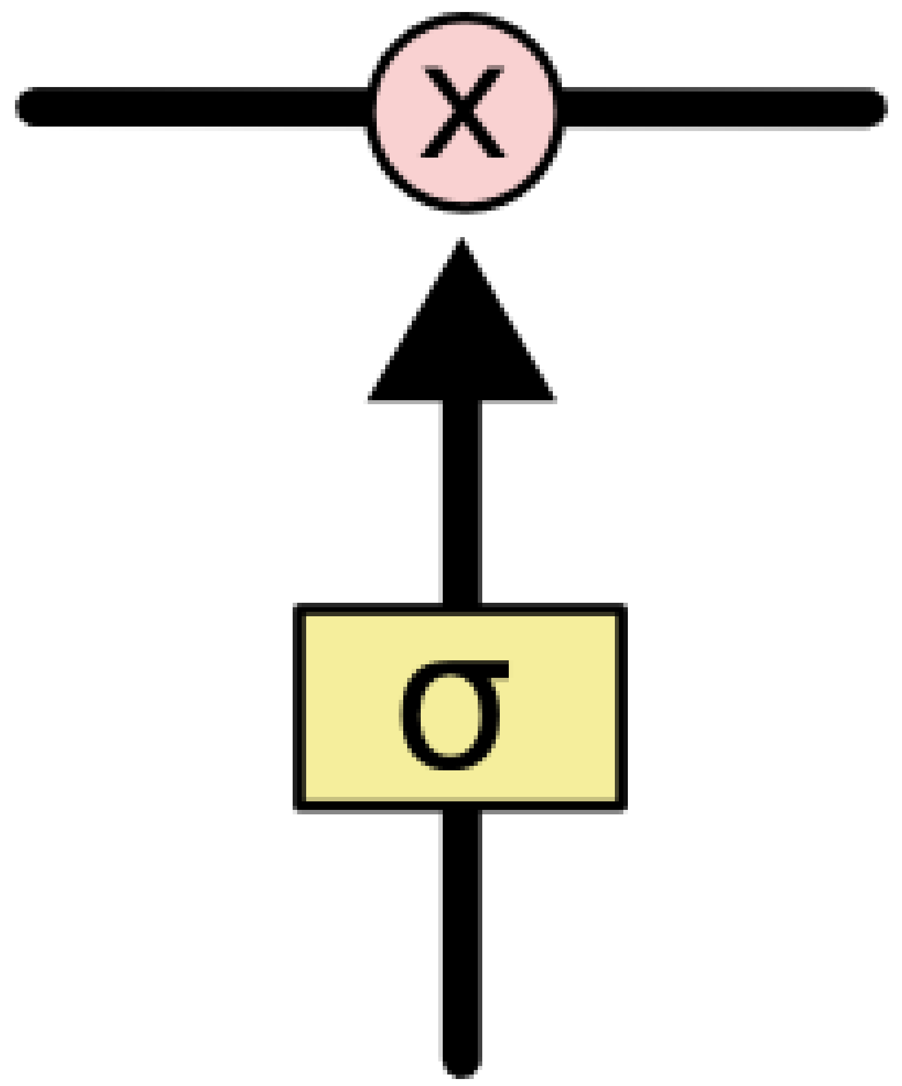


The repeating module in an LSTM contains four interacting layers.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

The key to LSTMs is the cell state, the horizontal line running through the top of the diagram. The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only minor linear interactions. It’s very easy for information to just flow along it unchanged.

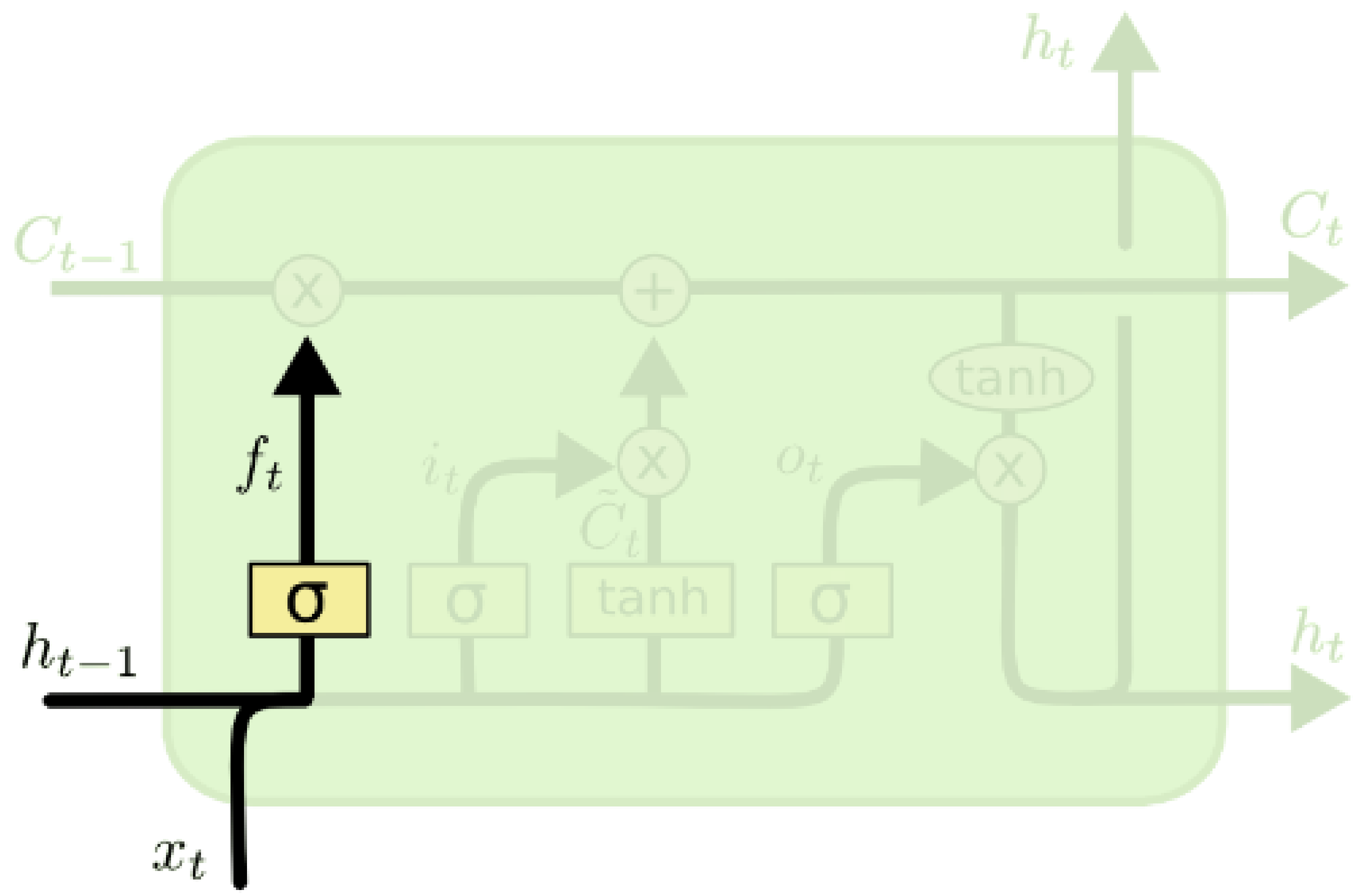
The LSTM can remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to let information through optionally. They comprise a sigmoid neural net layer and a pointwise multiplication operation.



The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means “let nothing through,” while a value of one means “let everything through!” An LSTM has three of these gates, to protect and control the cell state.

Say we have a sentence "*The boy ran through the park. The girl twirled in her pretty dress.* ***She*** *stood proudly on the stage.*"

In this example of a language model trying to predict the next word based on all the previous ones. In such a problem, the cell state might include the gender of the present subject, so that the correct pronouns can be used. When we see a new subject, we want to forget the gender of the old subject.



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

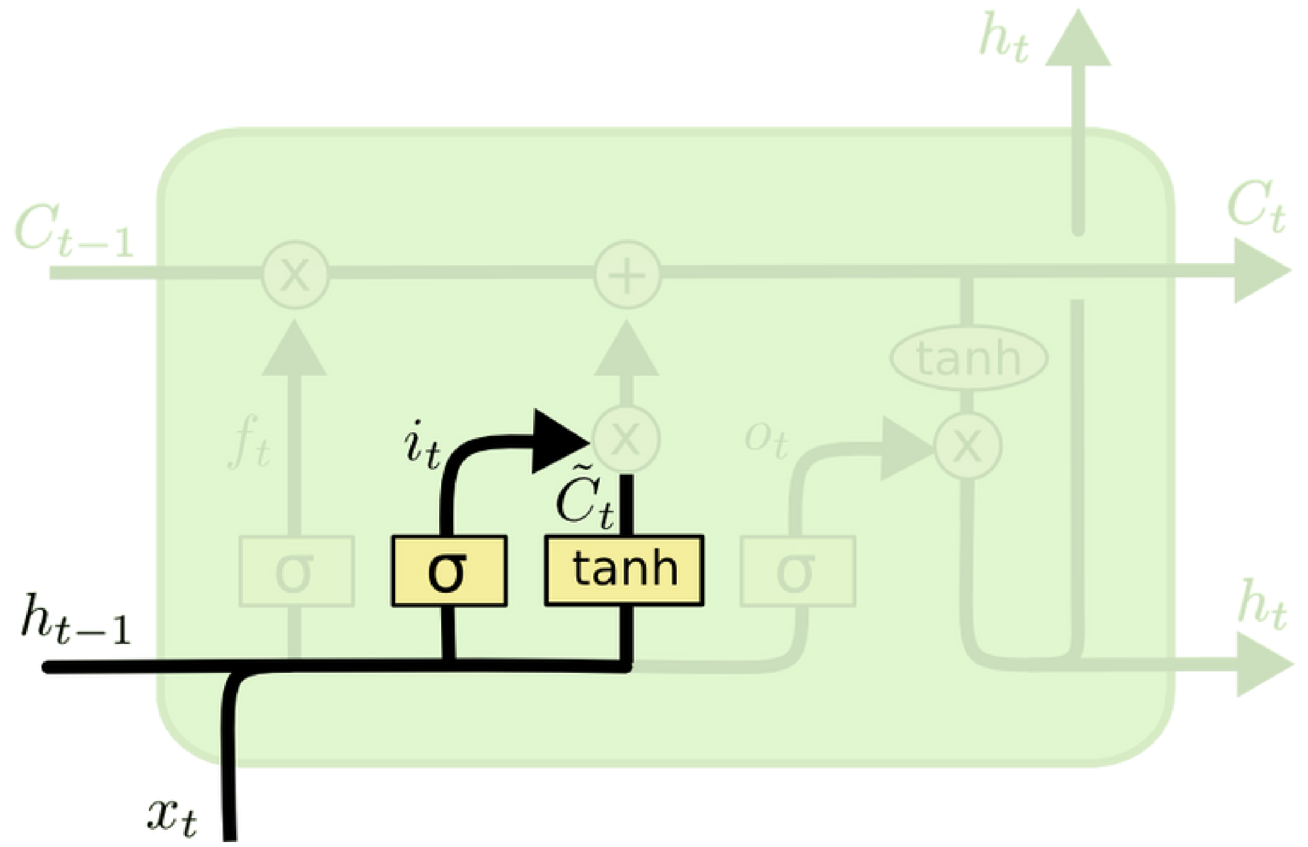
$$\begin{bmatrix}1\\2\\3\\4\\5\\6\end{bmatrix} \otimes \begin{bmatrix}1\\1\\0\\0\\1\\0\end{bmatrix} \Longrightarrow \begin{bmatrix}1\\2\\0\\0\\5\\0\end{bmatrix}$$

$C_{t-1}$ $\qquad f_t$ $\qquad C_{t-1} \otimes$

$\hookrightarrow \sigma(W_f [h_{t-1}, x_t] + b_f)$

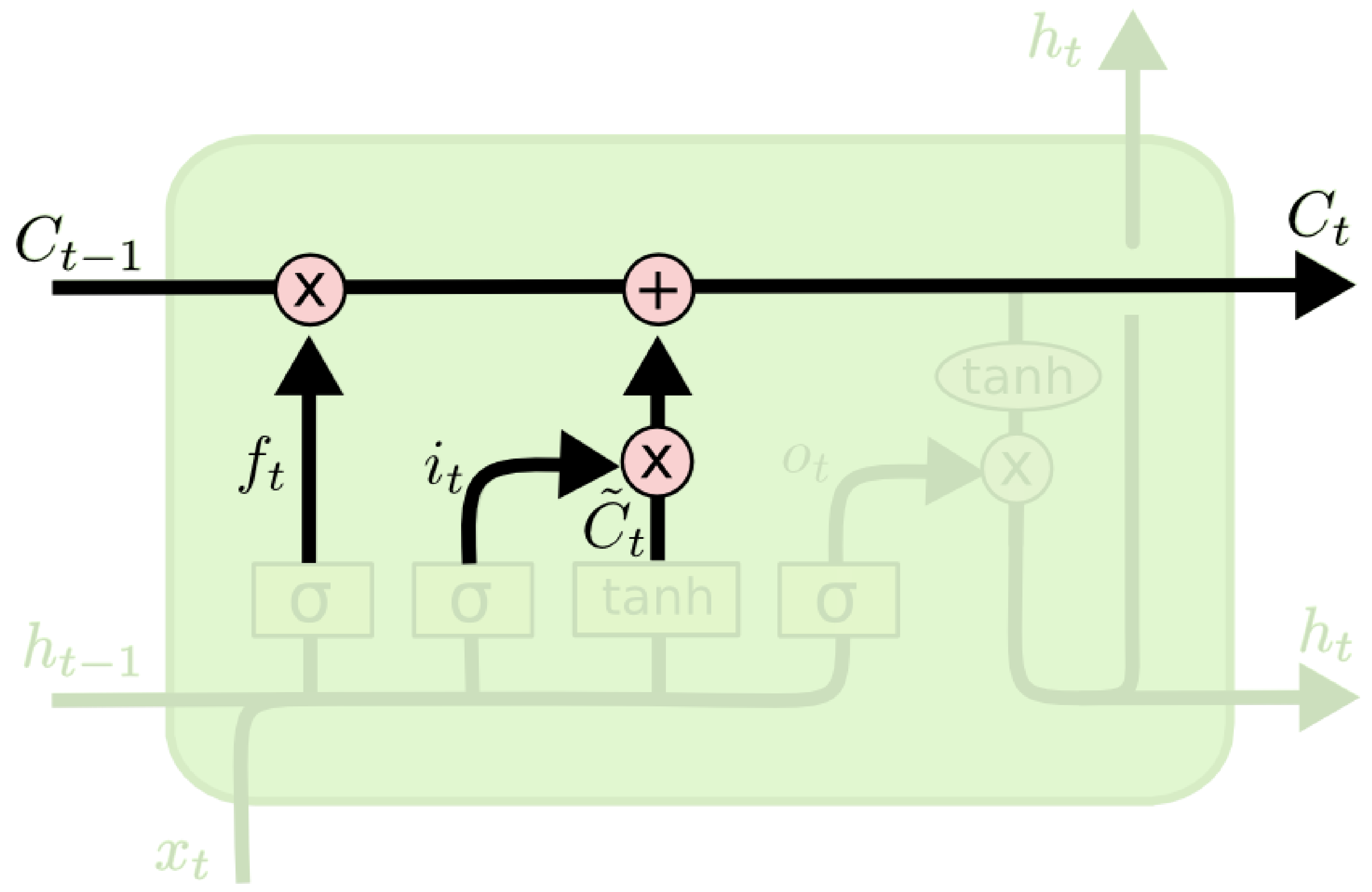
The next step is to decide what new information we will store in the cell state. This has two parts. First, a sigmoid layer called the “input gate layer” decides which values we’ll update. Next, a tanh layer creates a vector of new candidate values, C~t, that could be added to the state. In the next step, we’ll combine these two to update the state.

In the example of our language model, we’d want to add the gender of the new subject to the cell state, to replace the old one we’re forgetting.



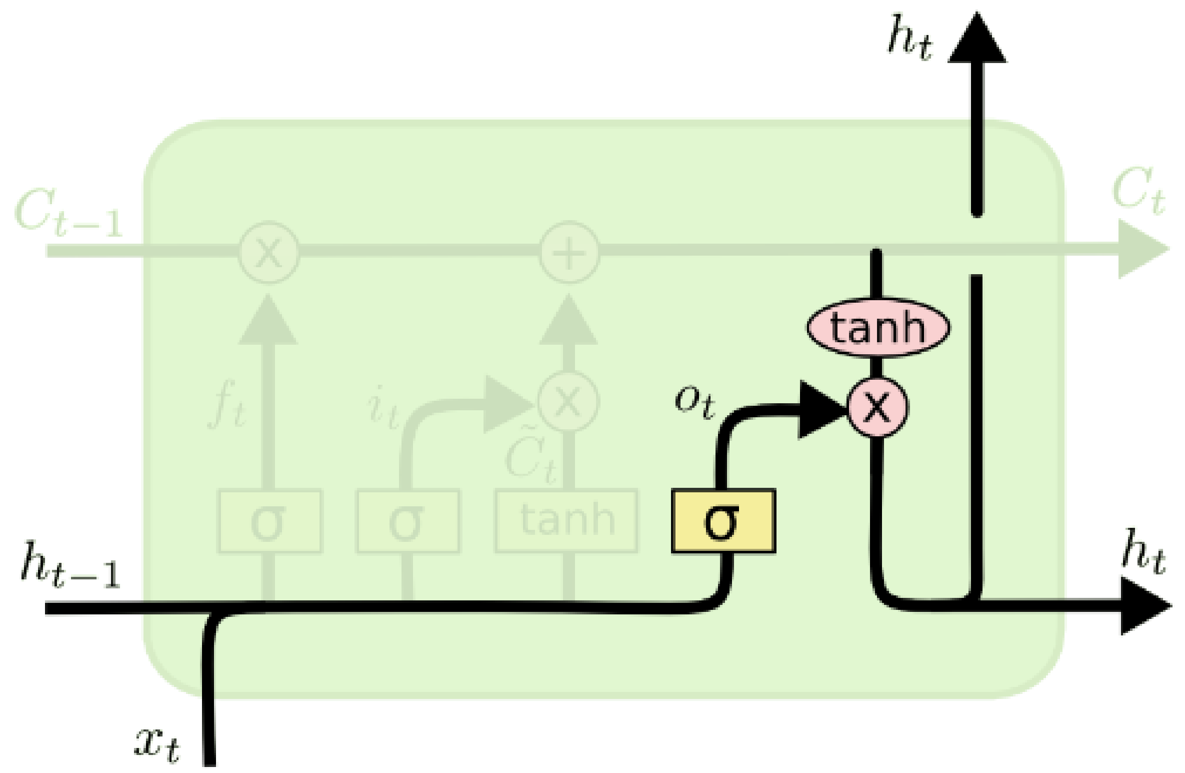
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

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



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

Finally, we need to decide what we're going to output. This output will be based on our cell state but will be filtered. First, we run a sigmoid layer which decides what parts of the cell state we will output. Then, we 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 we only output the parts we decided to.



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

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

For the language model example, since it just saw a subject, it might want to output information relevant to a verb, in case that’s what is coming next. For example, it might output whether the subject is singular or plural so that we know what form a verb should be conjugated into if that’s what follows next.

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

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

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

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

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

$$h_t = o_t * tanh(C_t),$$

## When should you use RNNs and LSTMs?

*PERFORMANCE EVALUATION OF DEEP NEURAL NETWORKS APPLIED TO SPEECH RECOGNITION: RNN, LSTM AND GRU*

| Model | WER (%) | Loss | Mean edit distance |
|---|---|---|---|
| RNN | 78.66 | 164.60 | 0.3991 |
| LSTM | 65.04 | 134.35 | 0.3222 |
| GRU | 67.42 | 136.89 | 0.3308 |

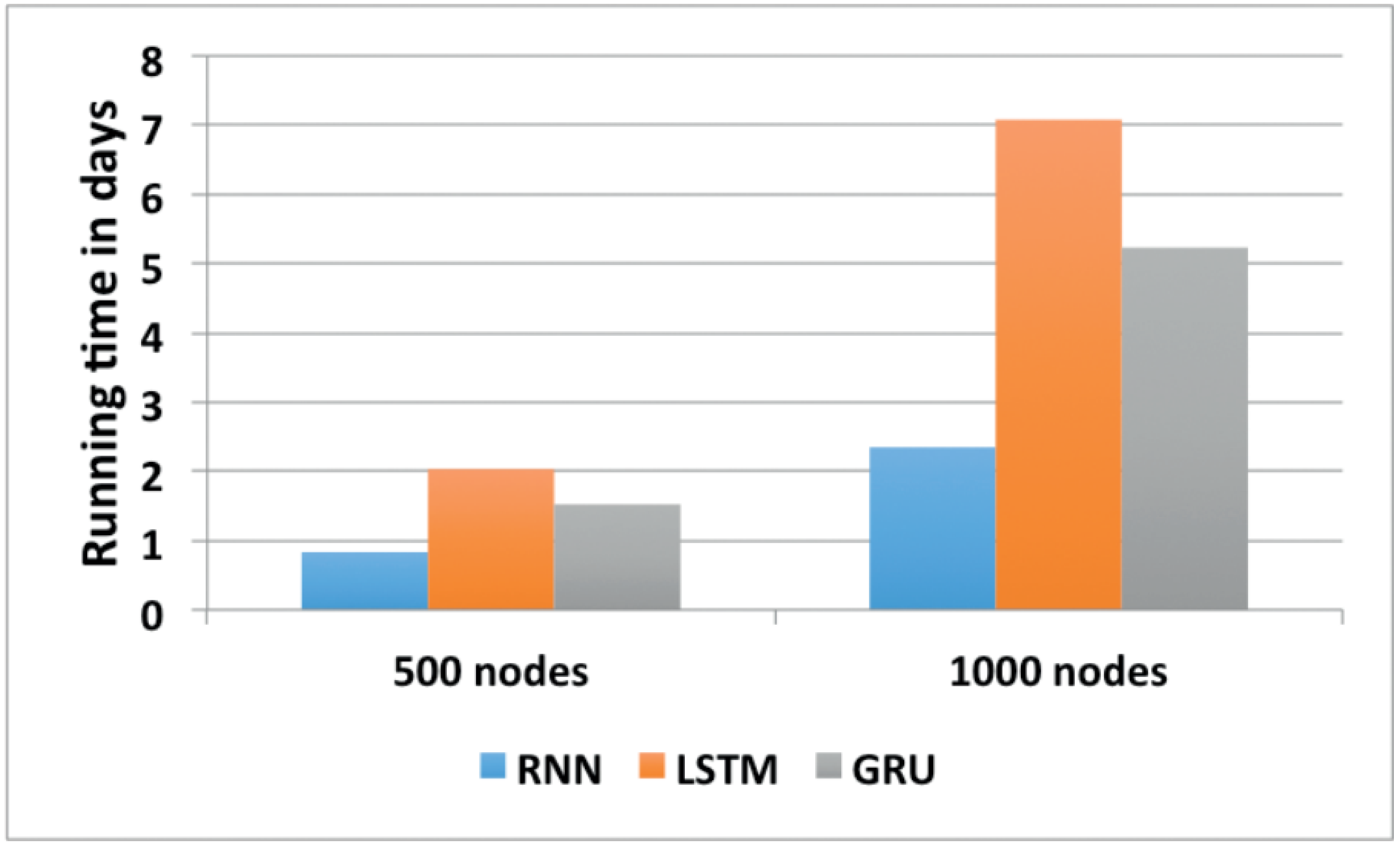


**Figure 4**. Running time in days