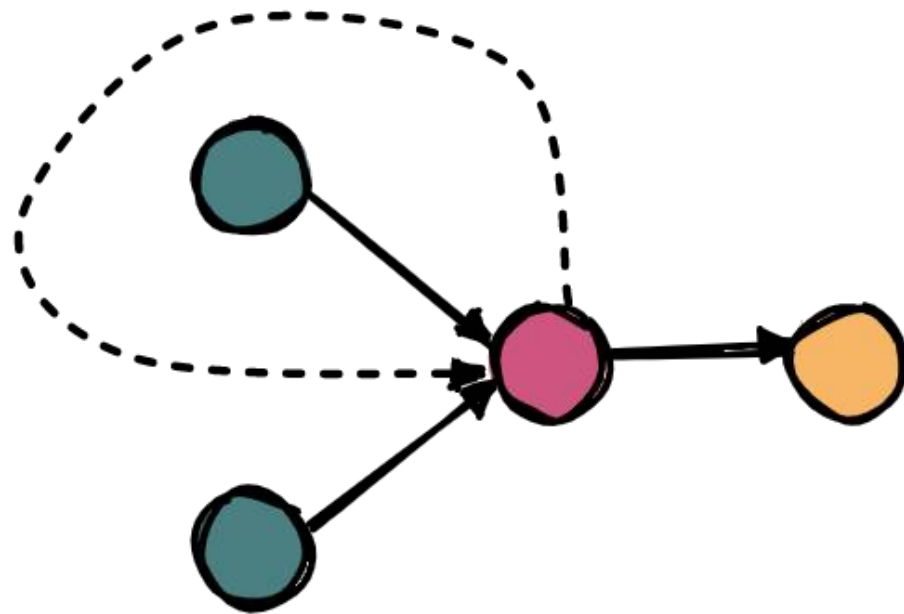


Recurrent neural networks

LSTM, GRU



Week 19

Middlesex University Dubai; CST4050 Fall21;
Instructor: Dr. Ivan Reznikov

Long-term dependencies

Solution for resolving vanishing gradient problem:
Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU).

LSTMs and GRUs are like regular RNNs, but they're capable of understanding long-term dependencies. They achieve it by using "gates", that are responsible for learning what information to add or remove to the hidden state.



**Aniruddho
Chakraborty**
Film Companion



To its credit, No Way Home weaves together stories, characters, moments and icons of more than two decades, to deliver applause and cheers that reverberate just as loud in half-empty, post-pandemic cinema halls.

[Full Review](#)

Dec 29, 2021

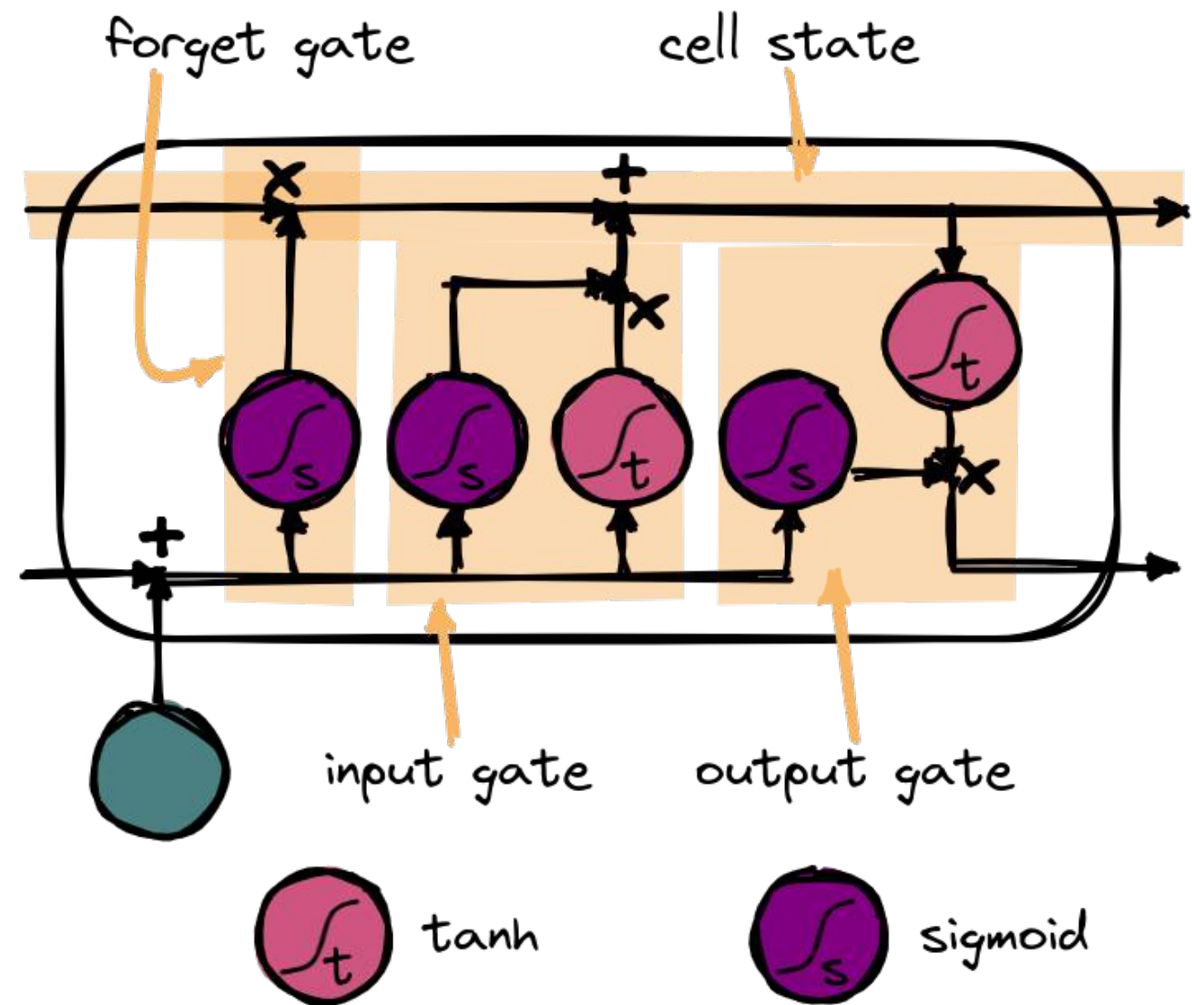
attention

Long-short term memory (LSTM)

One of the solution to overcome the vanishing gradient problem is using LSTM cell, that is shown on the right.

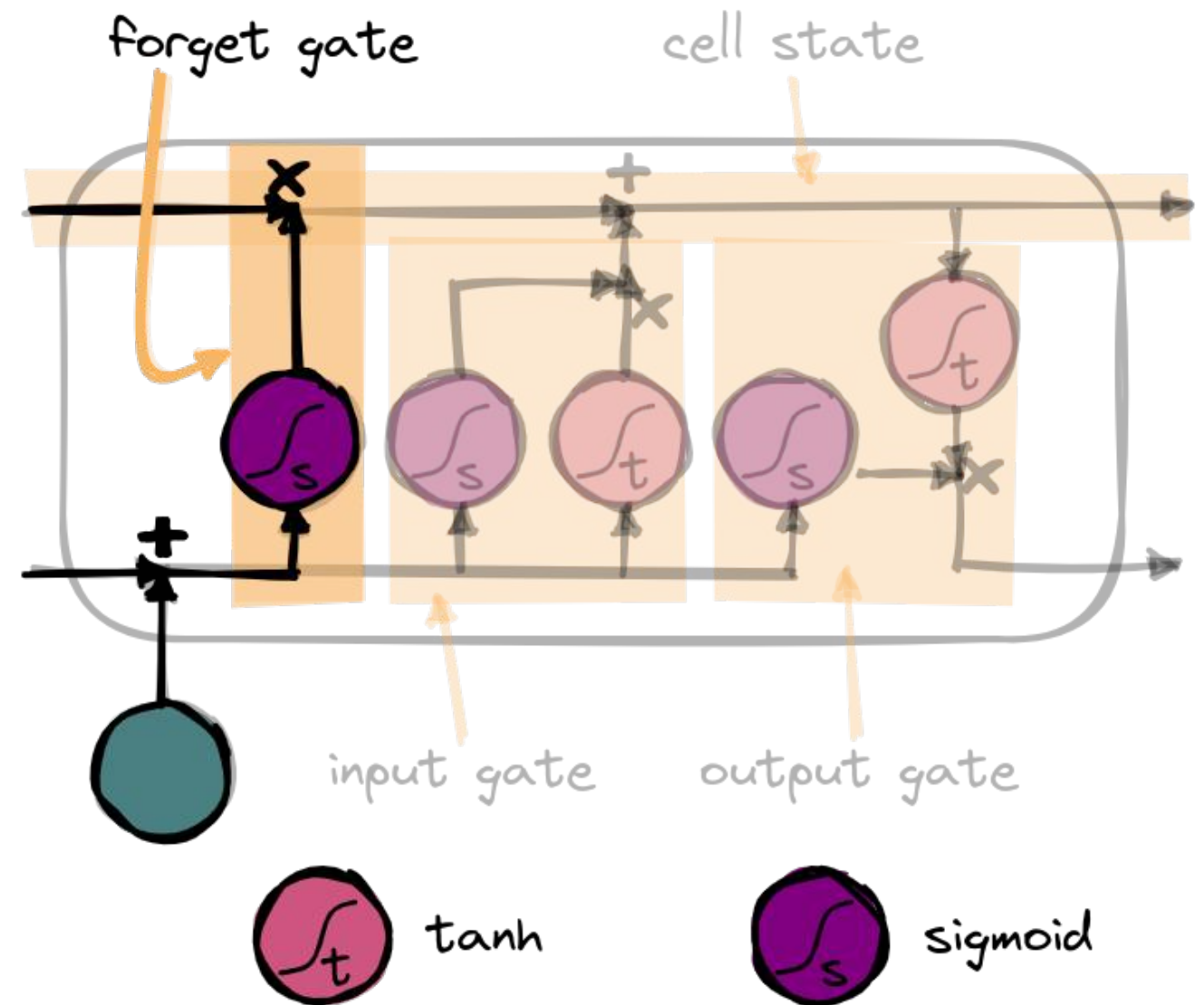
The LSTM contains of several components:

- forget gate
- input gate
- output gate
- cell state



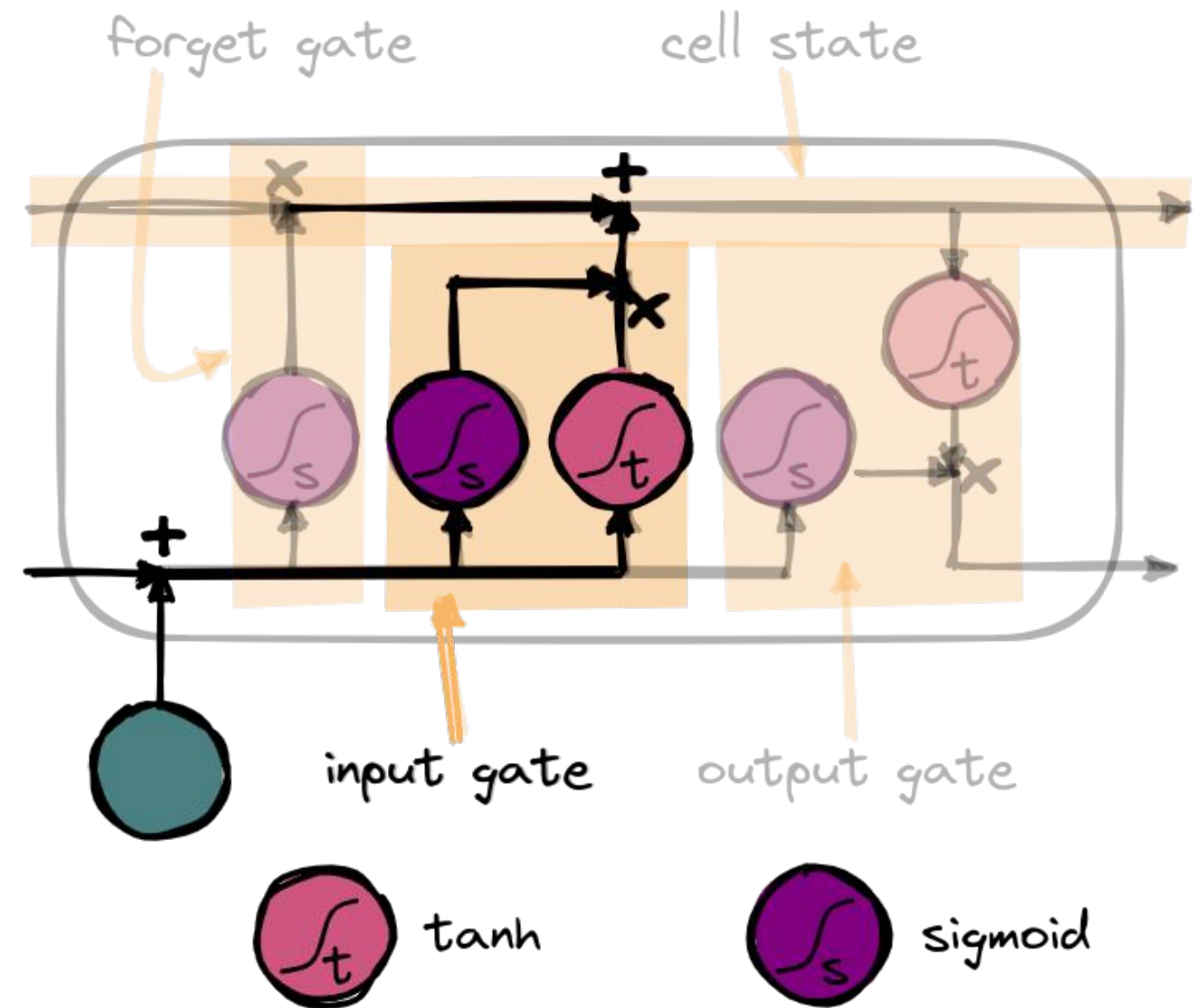
LSTM: Forget gate

This gate decides what pieces of information should be thrown away or kept. Data from the previous hidden state and information from the current input is passed through the sigmoid function. Values come out between 0 and 1 — the closer to 0 means to forget, and the closer to 1 means to keep.



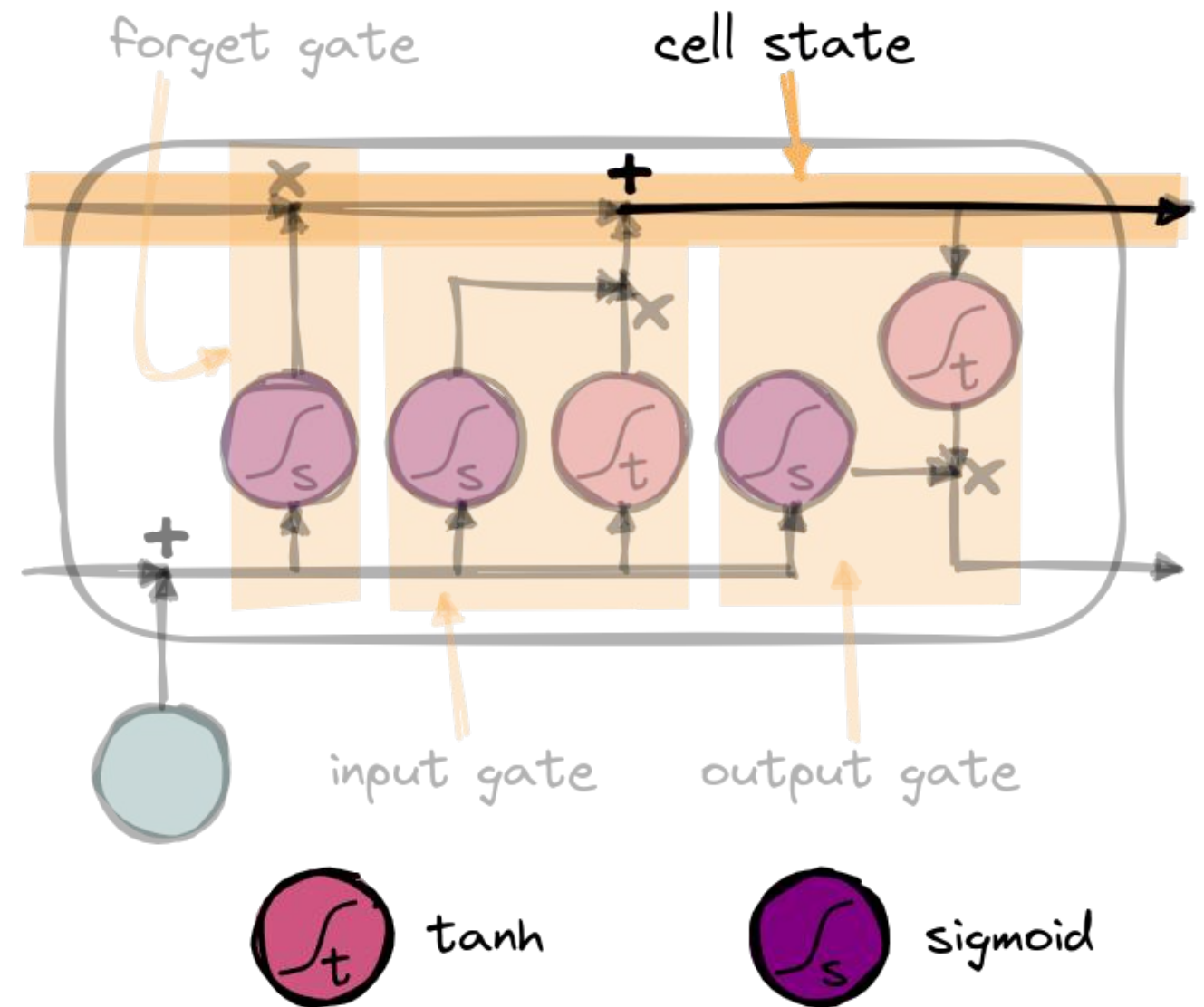
LSTM: Input gate

To update the cell state, we have the input gate. First, pass the previous hidden state and current input into a sigmoid function. That decides which values will be updated by transforming the values between 0 (not important) and 1 (important). Also, pass the hidden state and current input into the tanh function to squish values between -1 and 1 to help regulate the network. You multiply the tanh output with the sigmoid output. The sigmoid output will decide which information to keep from the tanh output.



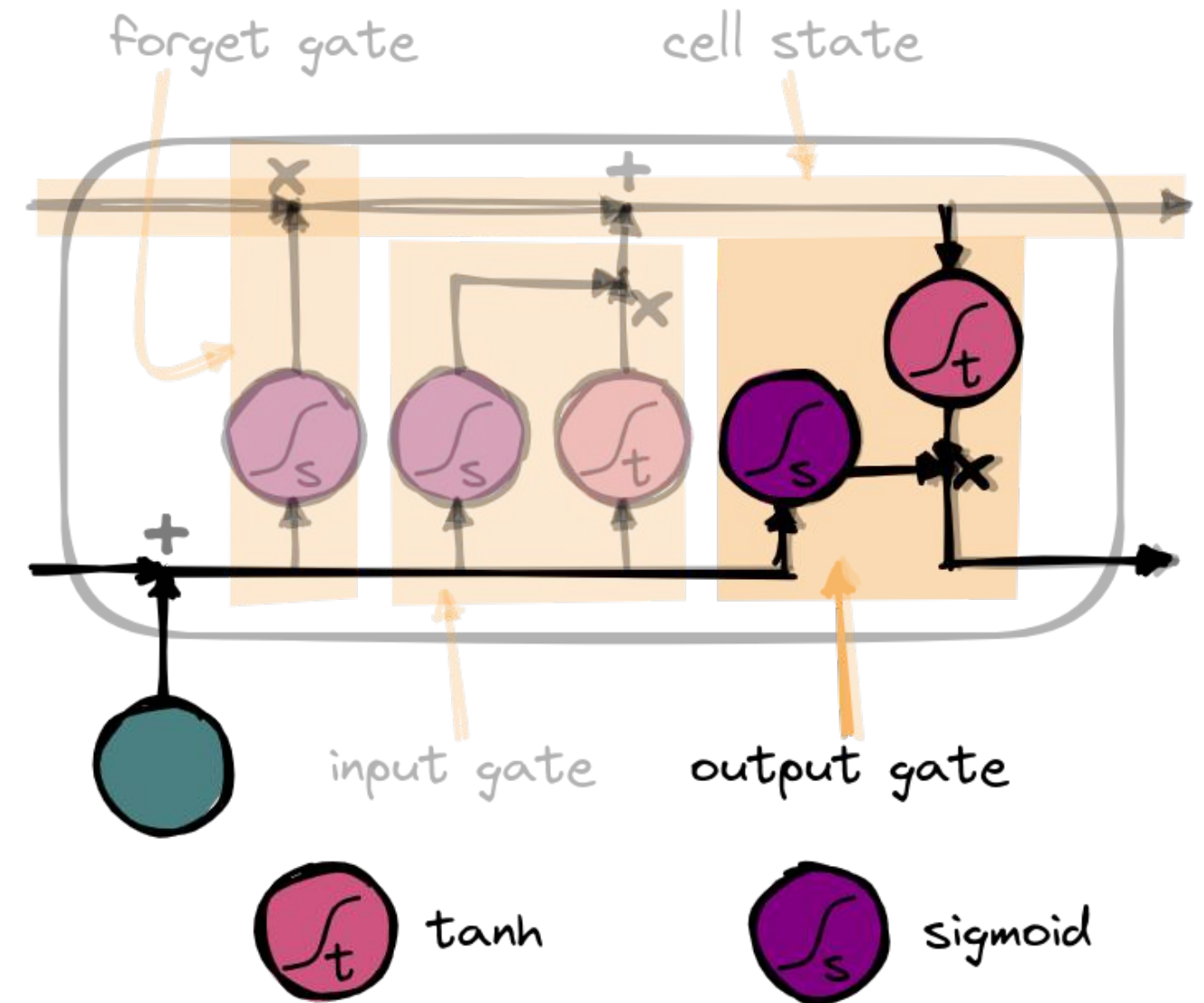
LSTM: Cell state

We now have enough information to calculate the cell state. First, the cell state is pointwise multiplied with the forget gate output. This drops values in the cell state if it is multiplied by values near 0. Further, we pointwise add the result with the input gate. The update is the new cell state that represents relevant information. That gives us our new cell state.



LSTM: Output gate

Last we have the output gate. The output gate decides what the next hidden state should be. The hidden state contains information on previous inputs and is used for predictions. First, pass the last hidden state and the current input into a sigmoid function. Then pass the newly modified cell state to the tanh function. We multiply the tanh output with the sigmoid output to decide the hidden state's information. The result is the new hidden state. The new cell state and the new hidden are then carried over to the next step.



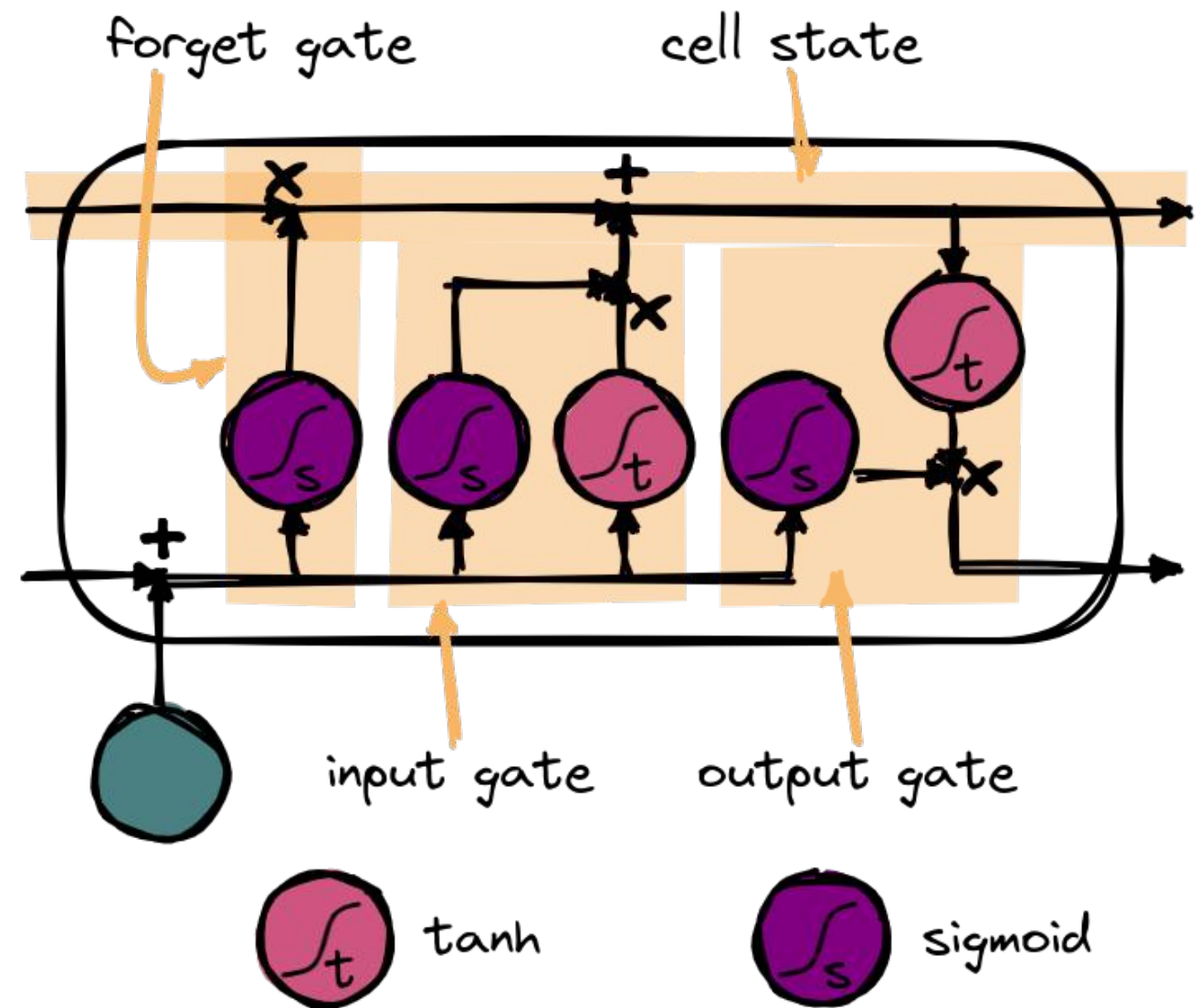
LSTM: Overall

The forget gate decides what is relevant to keep from previous steps.

The input gate decides what information is relevant to add from the current step.

The output gate determines what the next hidden state should be.

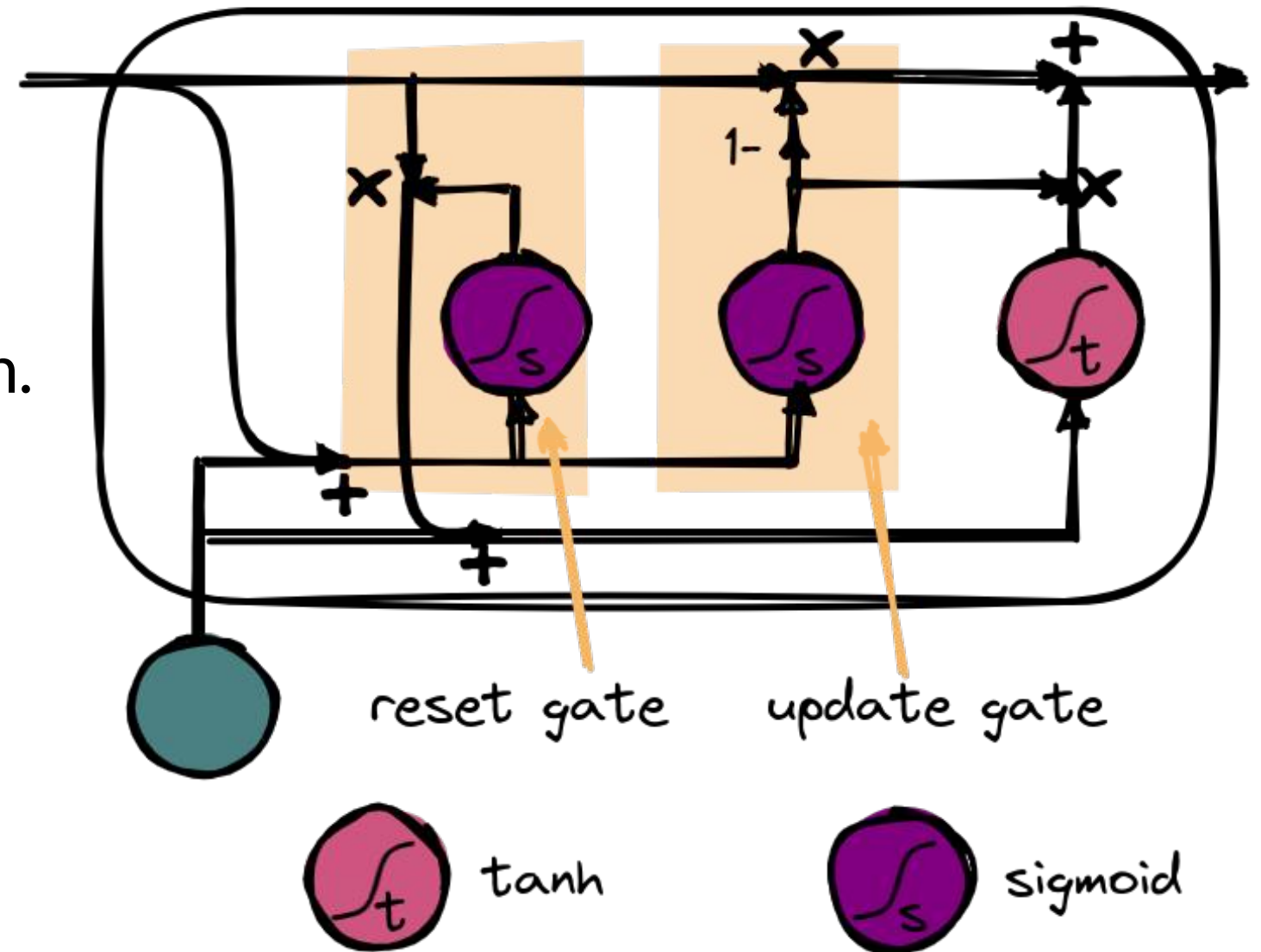
The cell state carries "important memory".



Gated Recurrent Units (GRU)

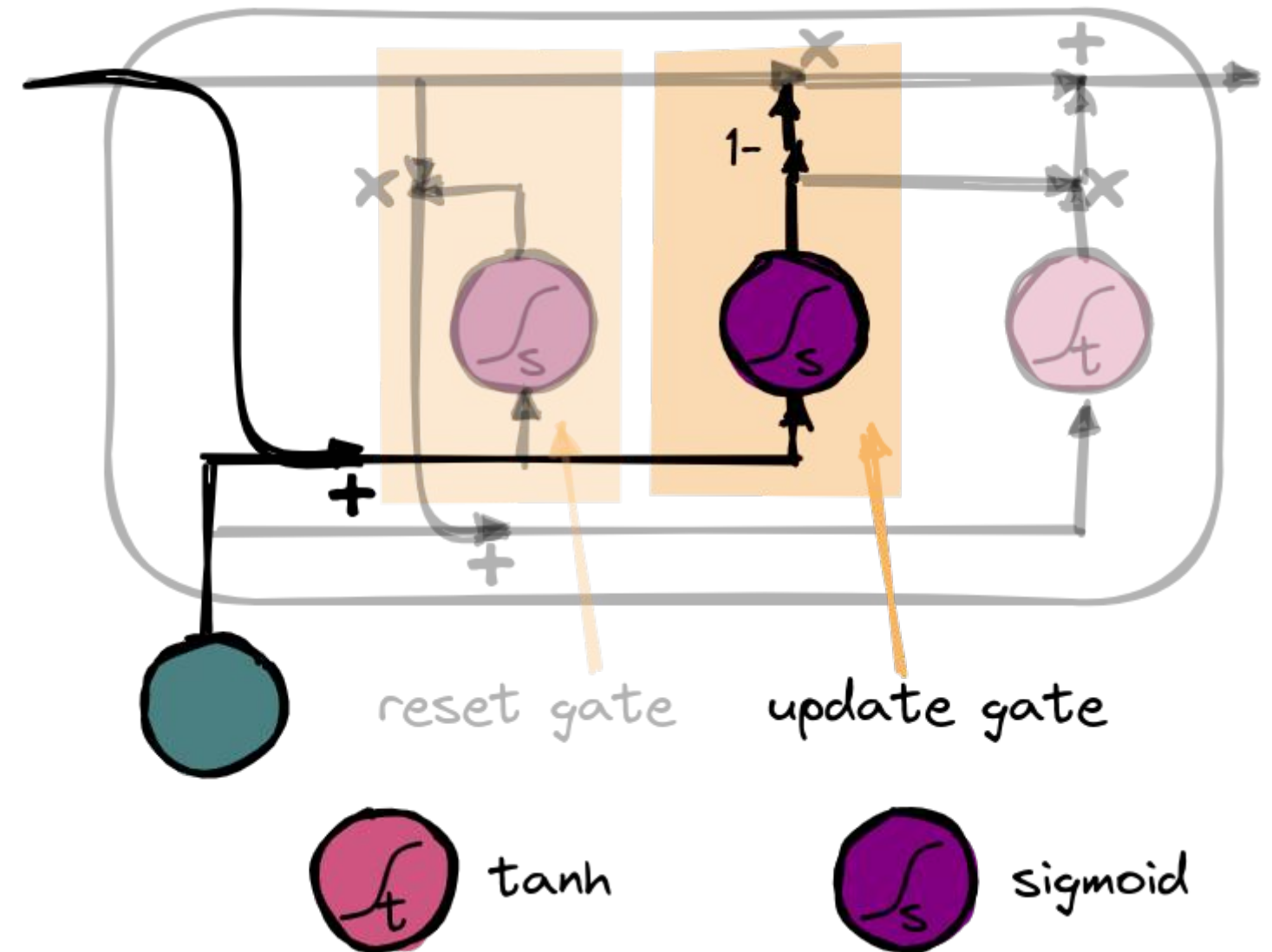
GRU is the newer generation of RNNs and is pretty similar to an LSTM.

GRUs got rid of the cell state and used the hidden state to transfer information. It also has two gates, a reset, and an update gate.



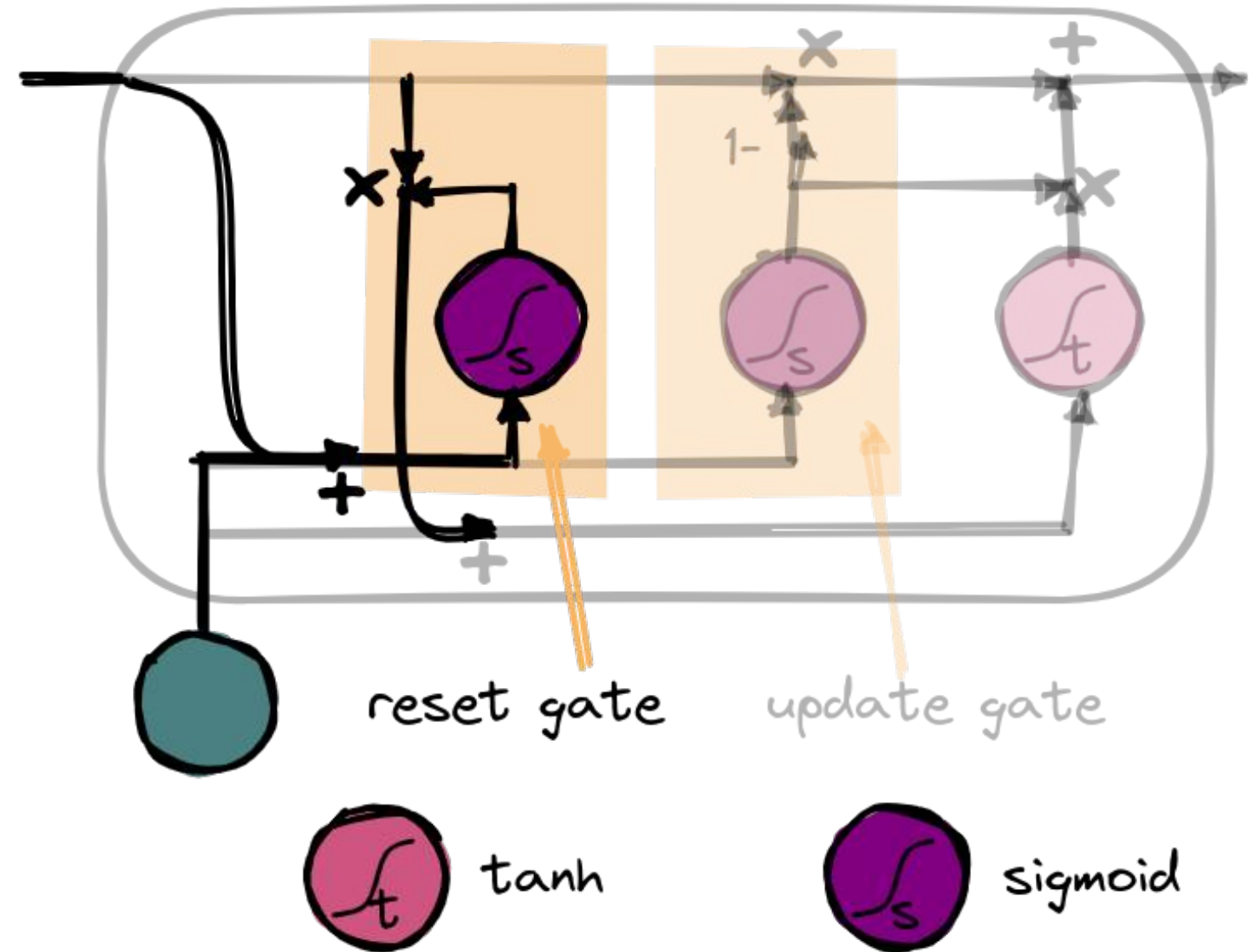
GRU: Update gate

Update gate acts similar to the forget and input gate of an LSTM. It decides what information to throw away and what new information to add.



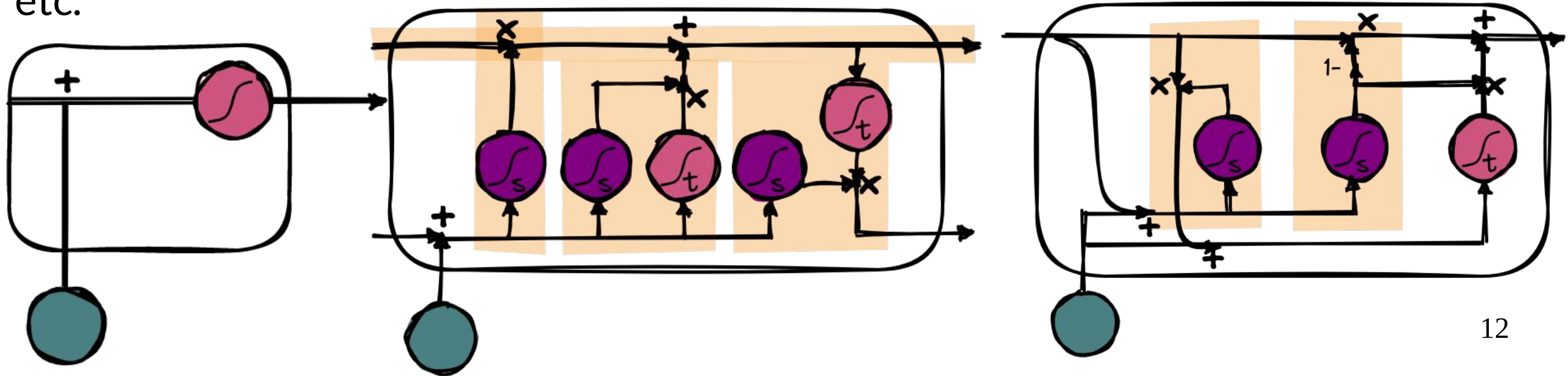
GRU: Reset gate

The reset gate is another gate used to decide how much past information to forget.



RNN: Summary

RNNs are suitable for processing sequence data for predictions but suffer from short-term memory. LSTMs and GRUs were created to mitigate short-term memory using mechanisms called gates. Gates are just neural networks that regulate the flow of information flowing through the sequence chain. LSTMs and GRUs are used in the state of the art deep learning applications like speech recognition, speech synthesis, natural language understanding, etc.



RNN vs HMM

- RNNs have greater representational power
- RNNs do not make the Markov assumption and so, in theory, take into account long-term dependencies when modeling sequences
- RNNs may require an embedding layer for certain vector representations of their inputs (e.g., words encoded as one-hot vectors)