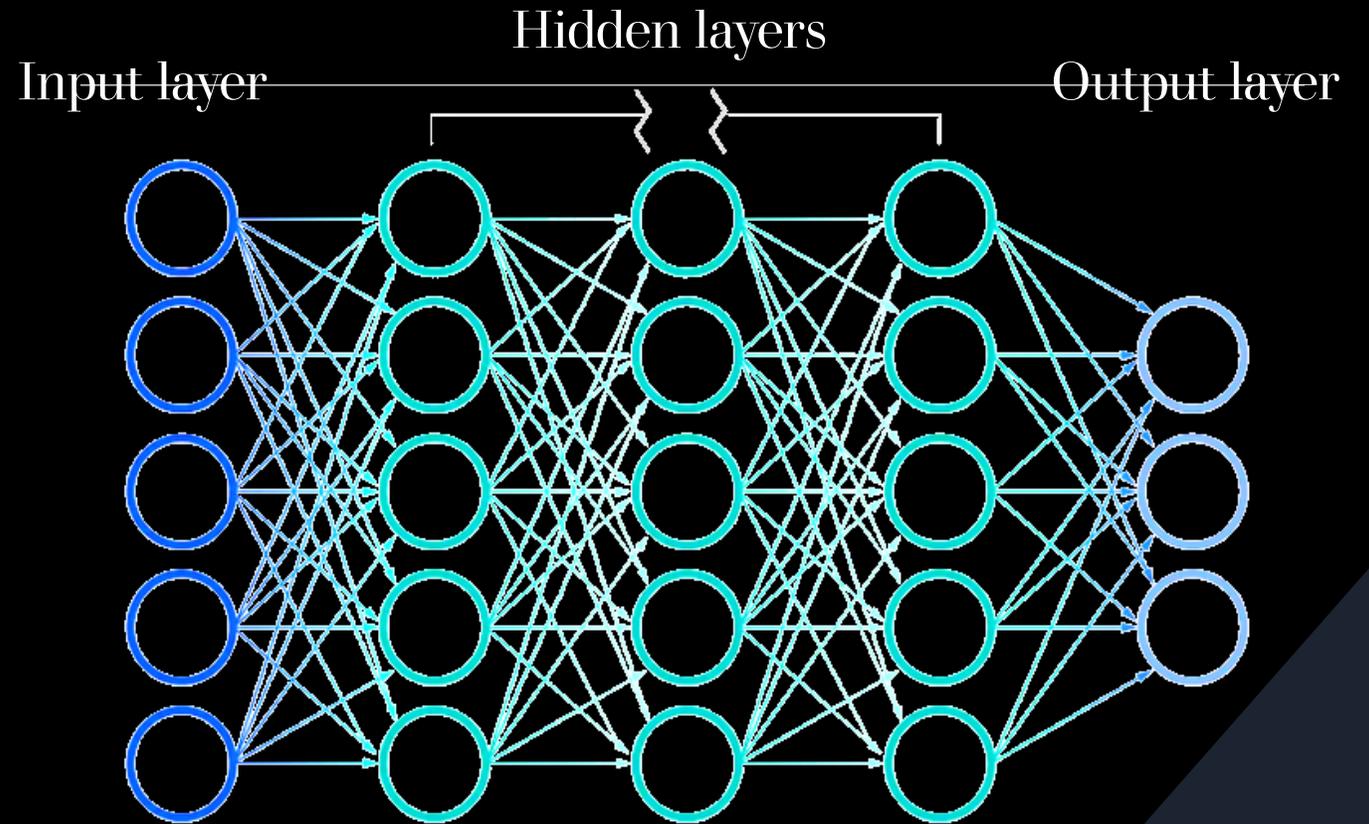


# LSTM (LONG SHORT- TERM MEMORY)

Anastasia Korenova  
Ekaterina Basharina

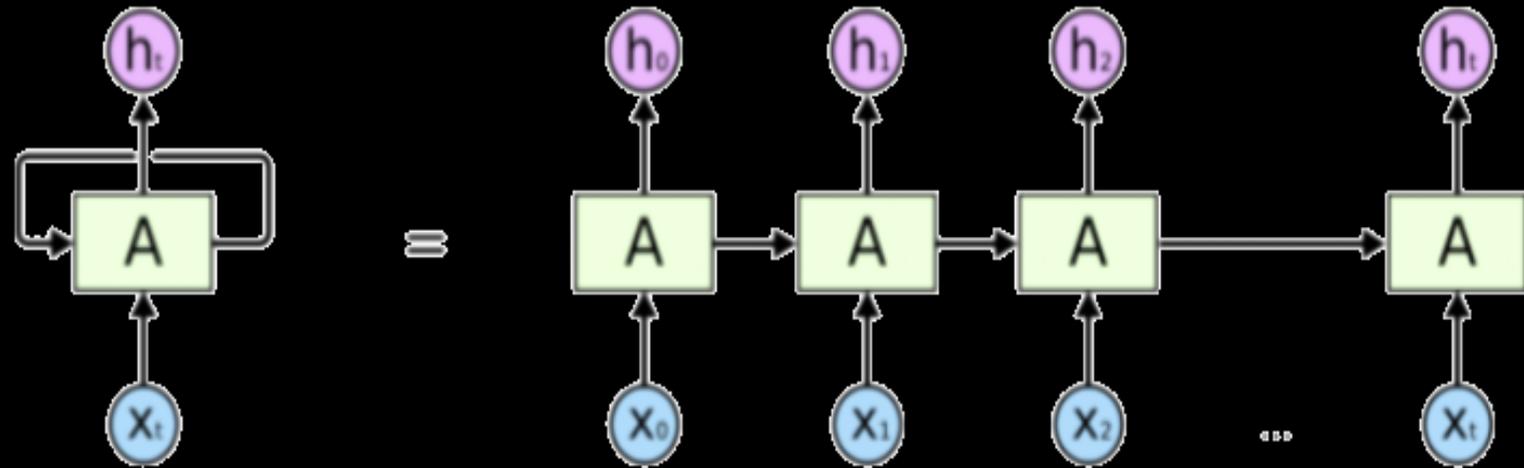
# What is a neural network?

- Node layers
- Linear regression models
- Training data to learn



# Recurrent neural networks

- Problem: persistence needed!
- Loops allow to save previous information
- Multiple copies of same network

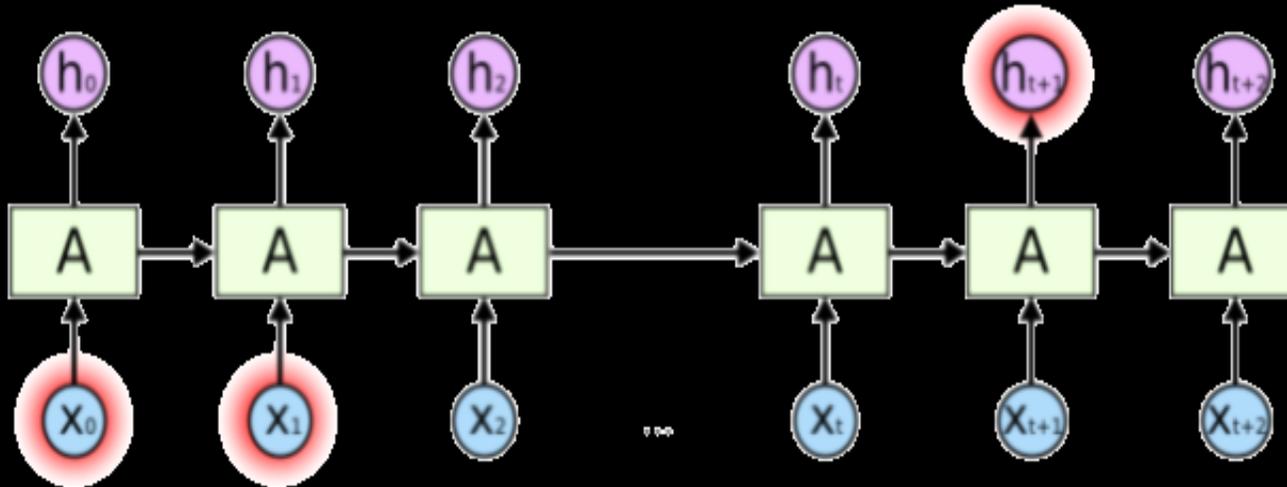


# The problem of long-term dependencies

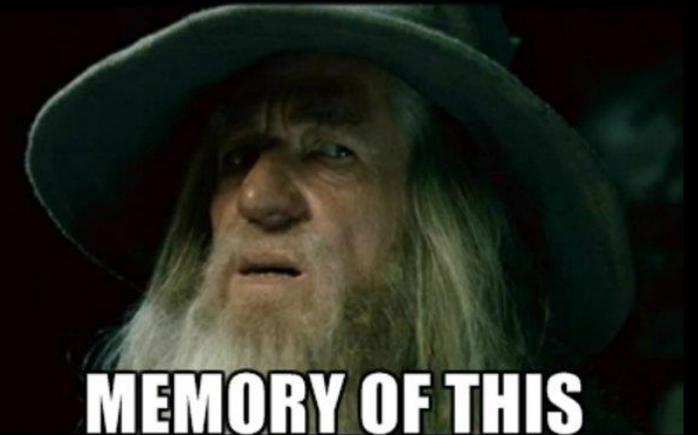
- The more gaps are, the worse our predictions become

Example:

- 1) The clouds are in the ...?
- 2) I grew up in France. Blah blah blah. I speak ...?



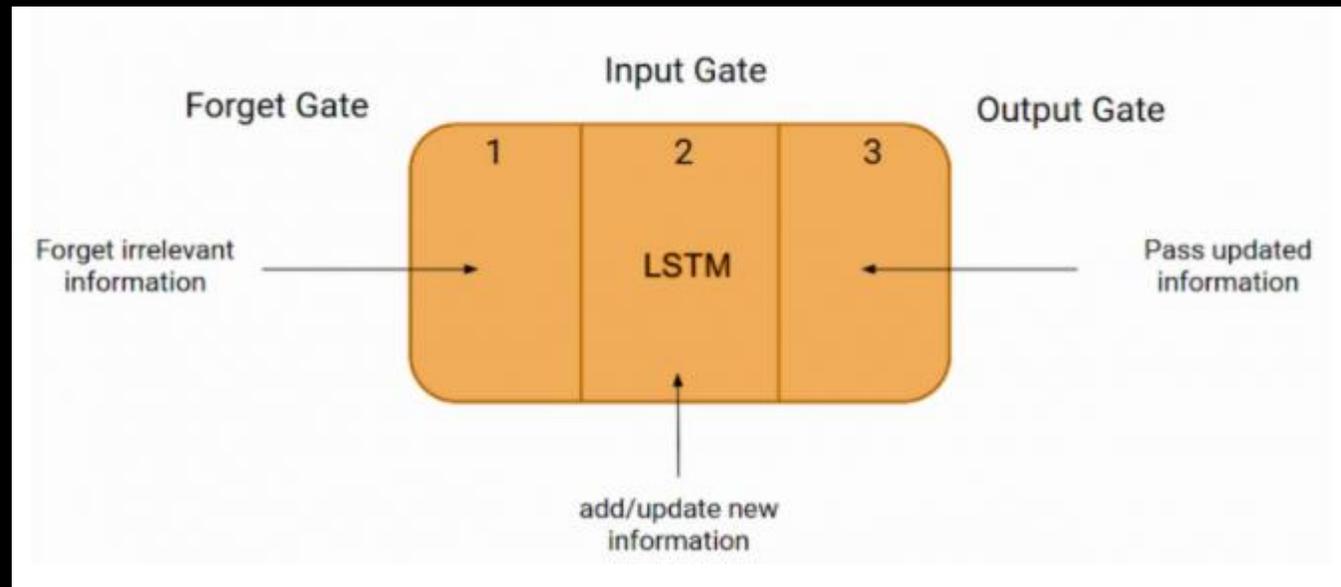
**I HAVE NO**



**MEMORY OF THIS  
PLACE**

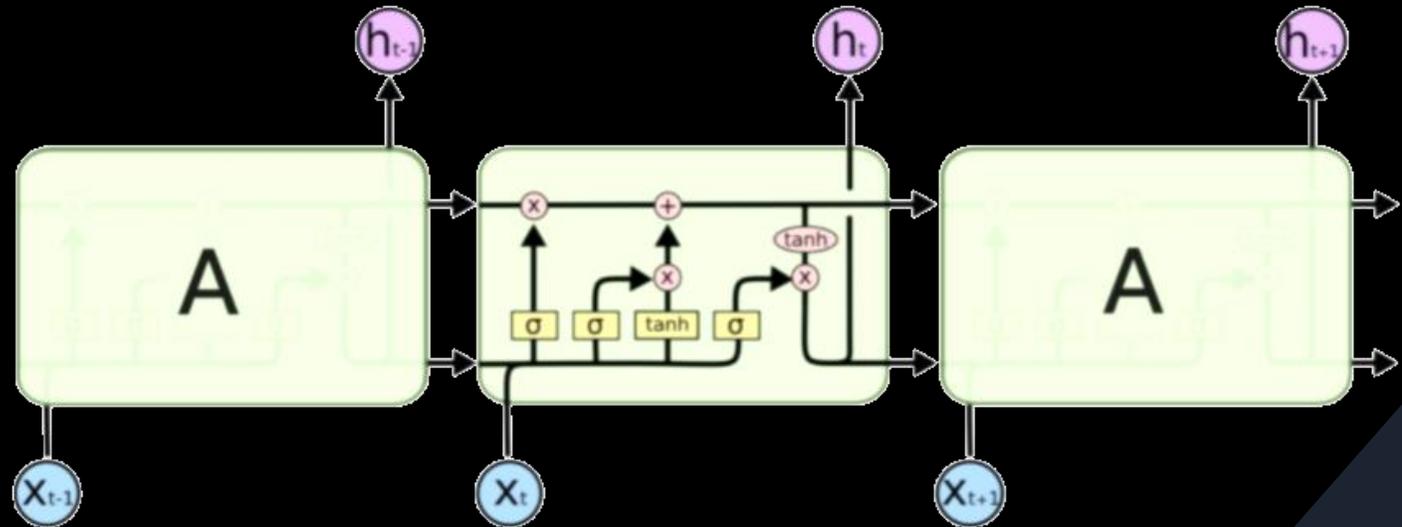
# Long short-term memory networks

- Learning long term dependencies!
- Forget gate, Input gate, Output gate



# LSTM: core idea

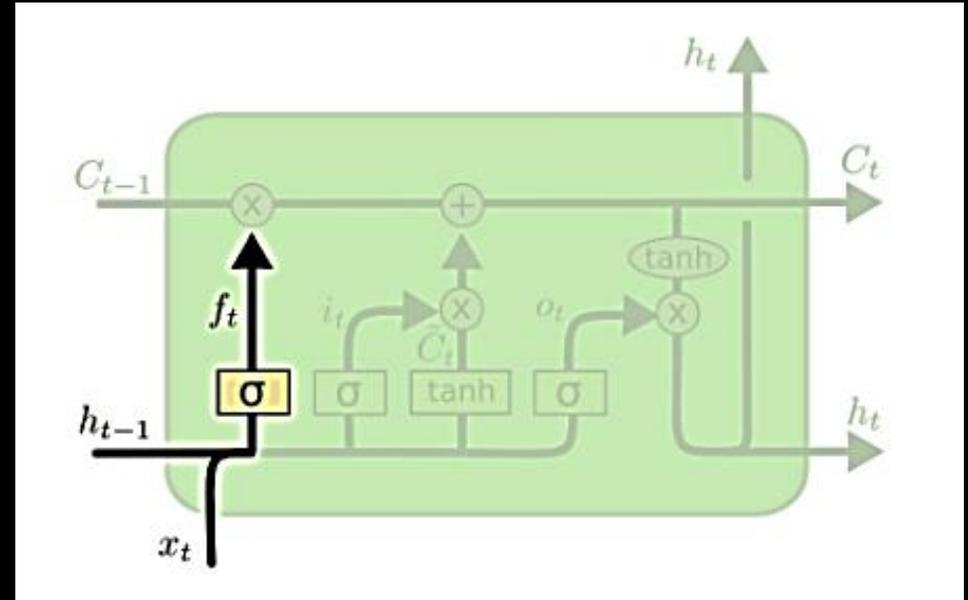
- Four interacting layers
- Cell state - runs straight down the entire chain
- Gates



# Step by step: forget gate

- What should we throw away?

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

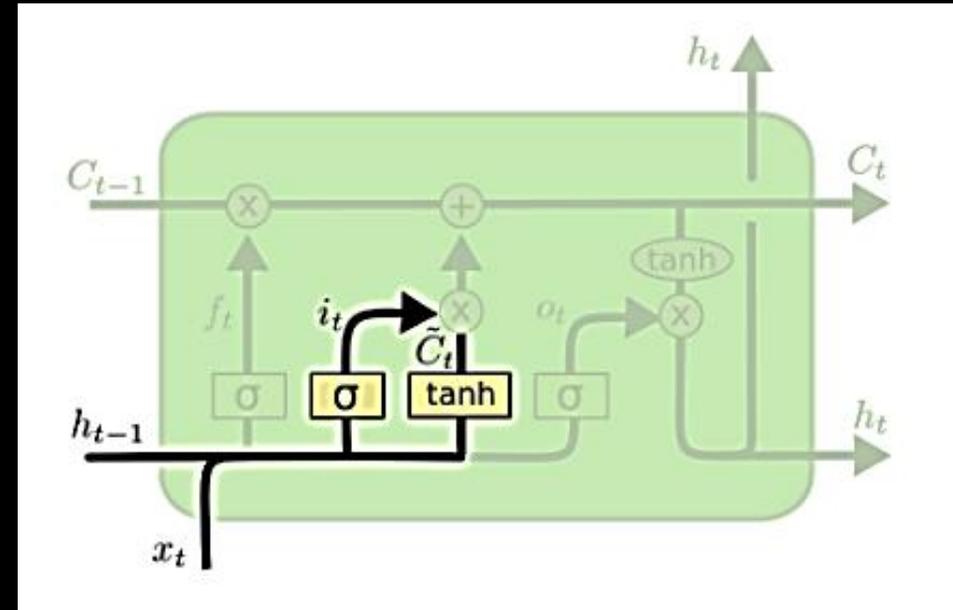


# Step by step: input gate

- What should we store in the cell state?

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

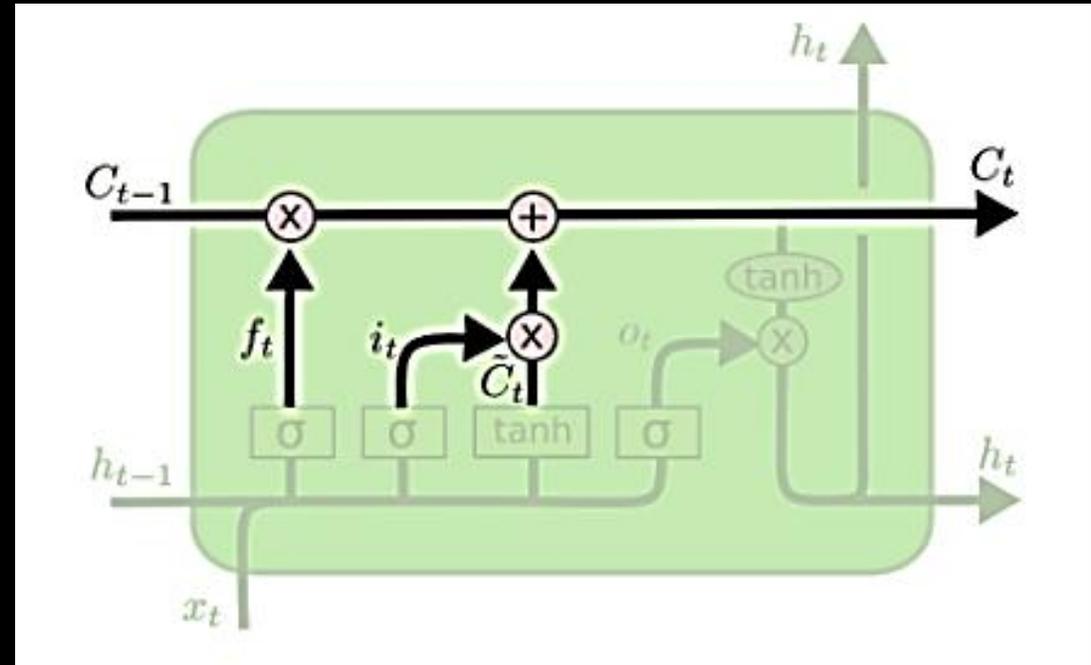
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$



# Step by step: update!

- It's now time to update the old cell state

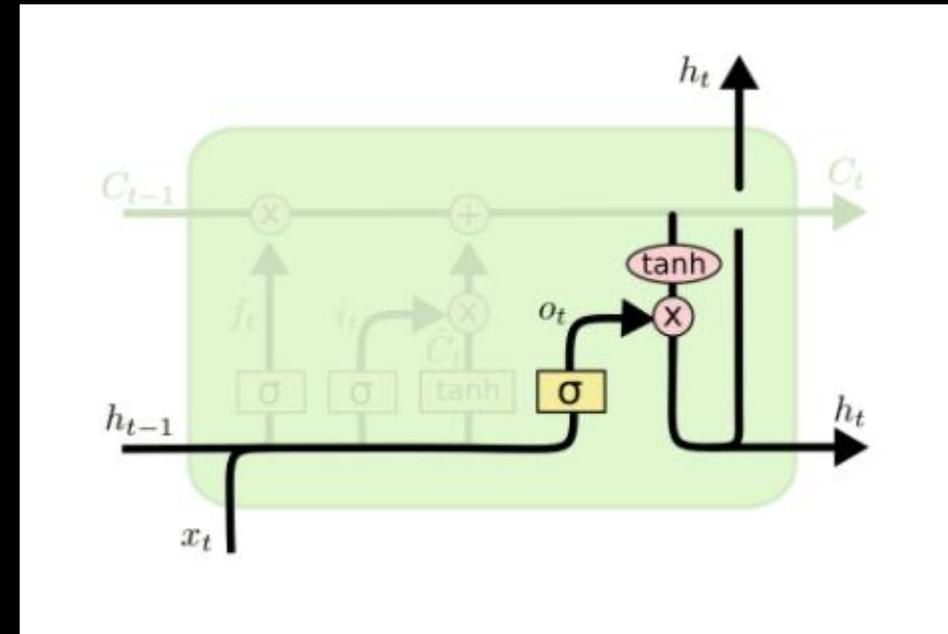
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



# Step by step: output gate

- What are we going to output?

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



Thanks for your attention!

