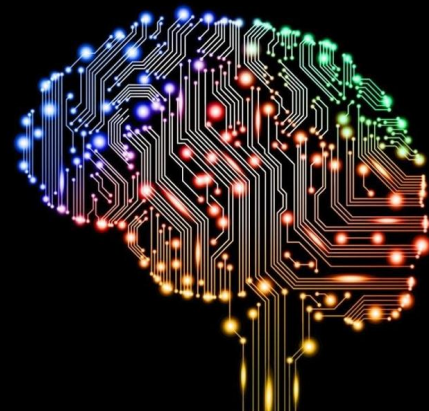


JAN 17 & 18, 2018



# Recurrent Neural Nets & Long Short Term Memory



**Prasanna Balaprakash**  
**Mathematics and Computer Science Division &**  
**Leadership Computing Facility**  
**Argonne National Laboratory**

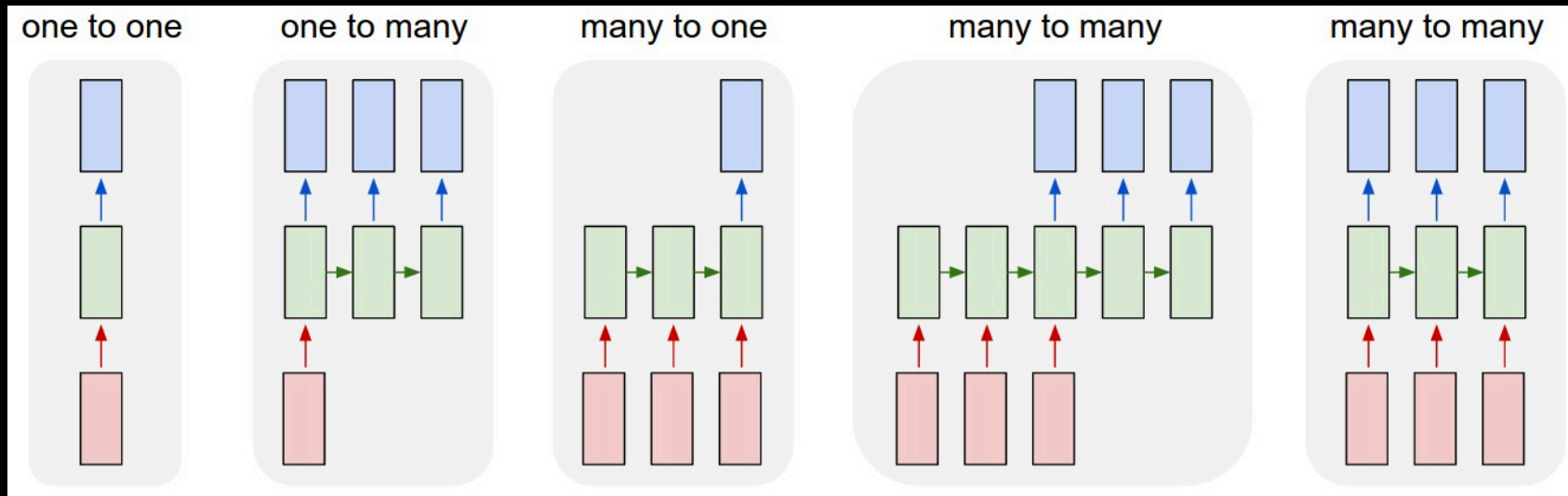
# Predict the next number?

- 1
- 1
- 2
- 3
- 5
- 8
- 13
- 21
- 34
- 55

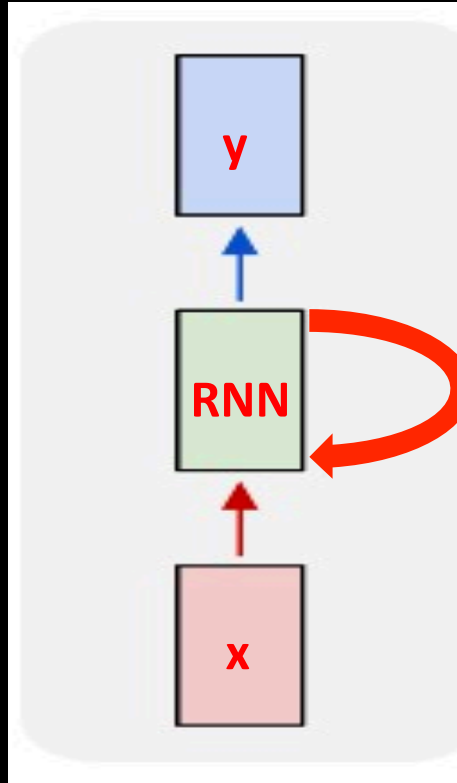
# Predict the next letter?

- th\_
- an\_
- en\_
- in\_
- io\_

# Recurrent neural nets

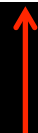
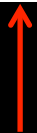


# Recurrent neural net



# Recurrent neural net

$$h_t = f_w(h_{t-1}, x_t)$$

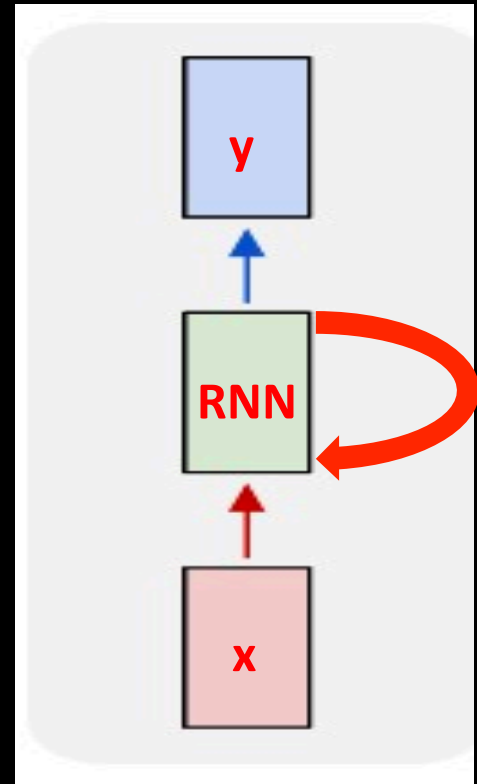


*new state*

*function with  
parameters w*

*old state*

*input vector*

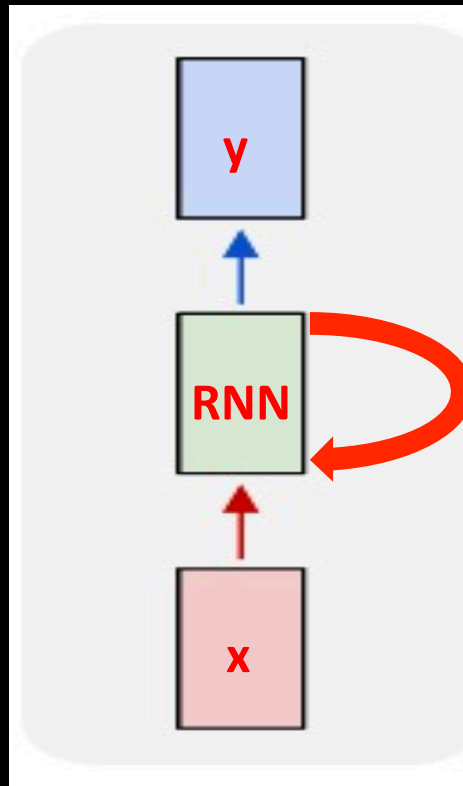


# Recurrent neural net

$$h_t = f_w(h_{t-1}, x_t)$$

$$h_t = \tanh(W_{hh}h_{t-1}, W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$



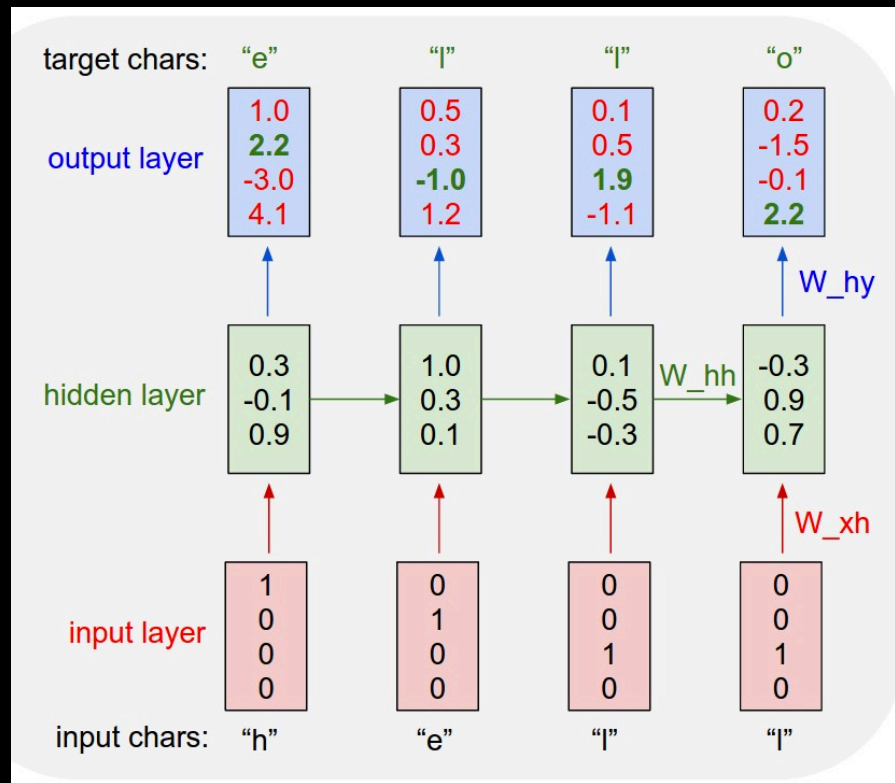
# Recurrent neural net

Vocabulary:  
[h, e, l, o]

Training sequence:  
“hello”

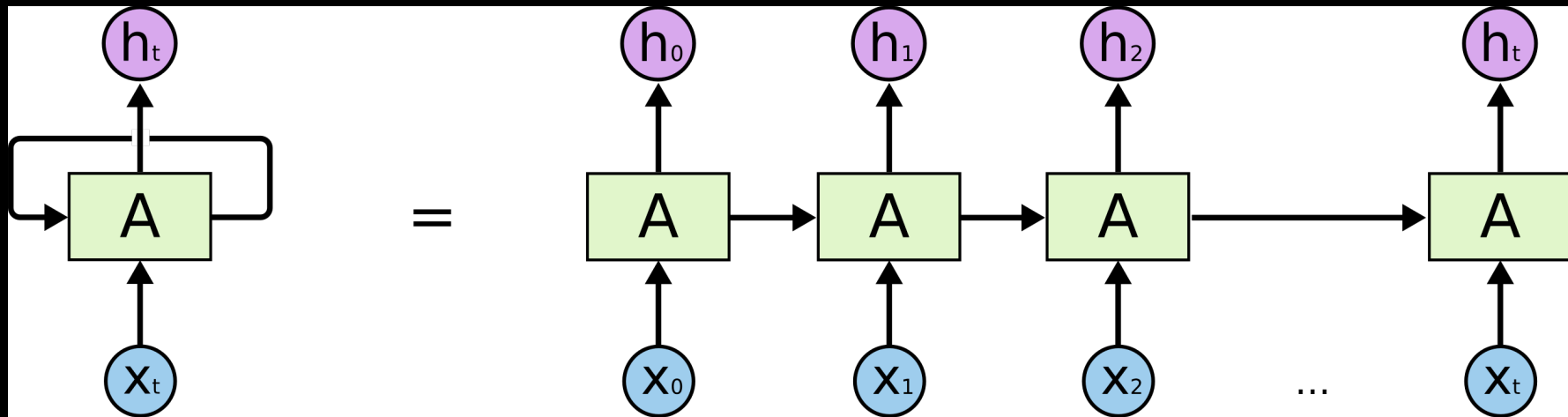
$$h_t = \tanh(W_{hh}h_{t-1}, W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

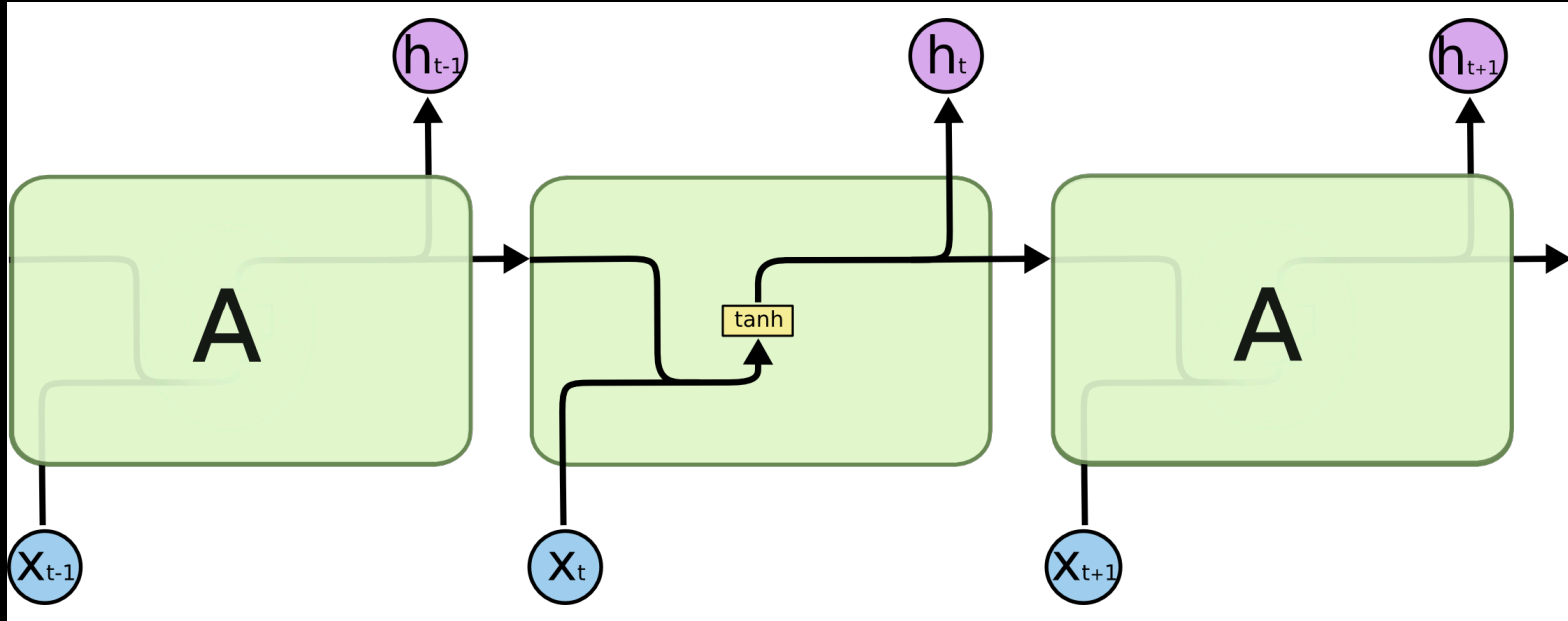




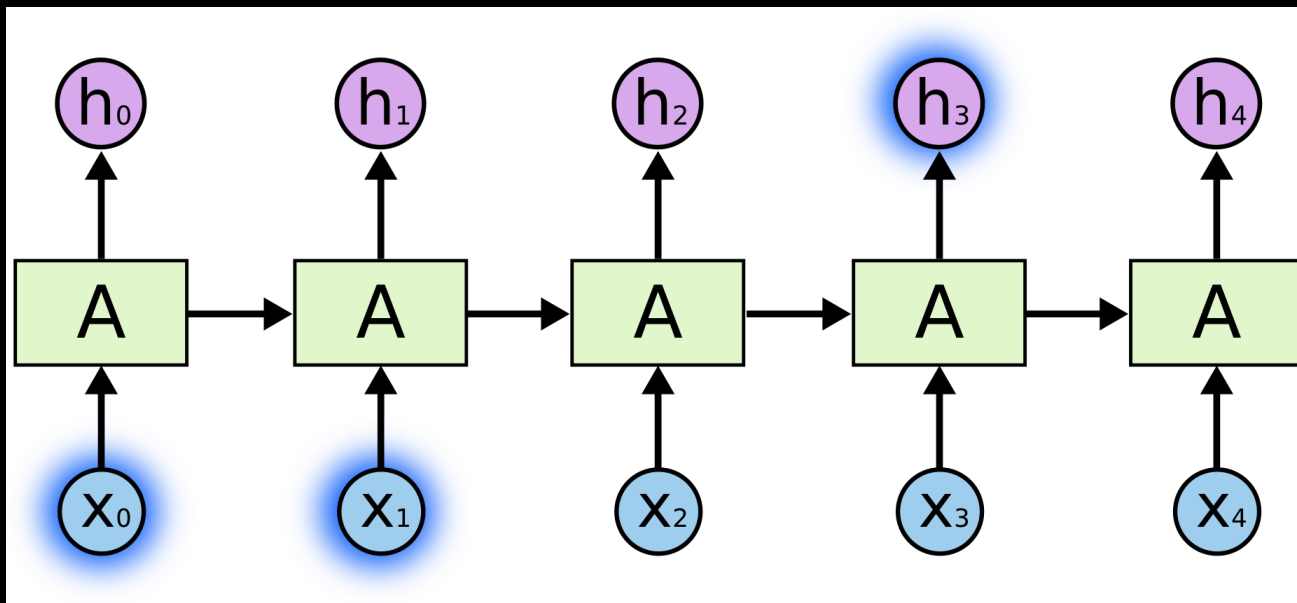
# Recurrent neural nets



# Recurrent neural nets

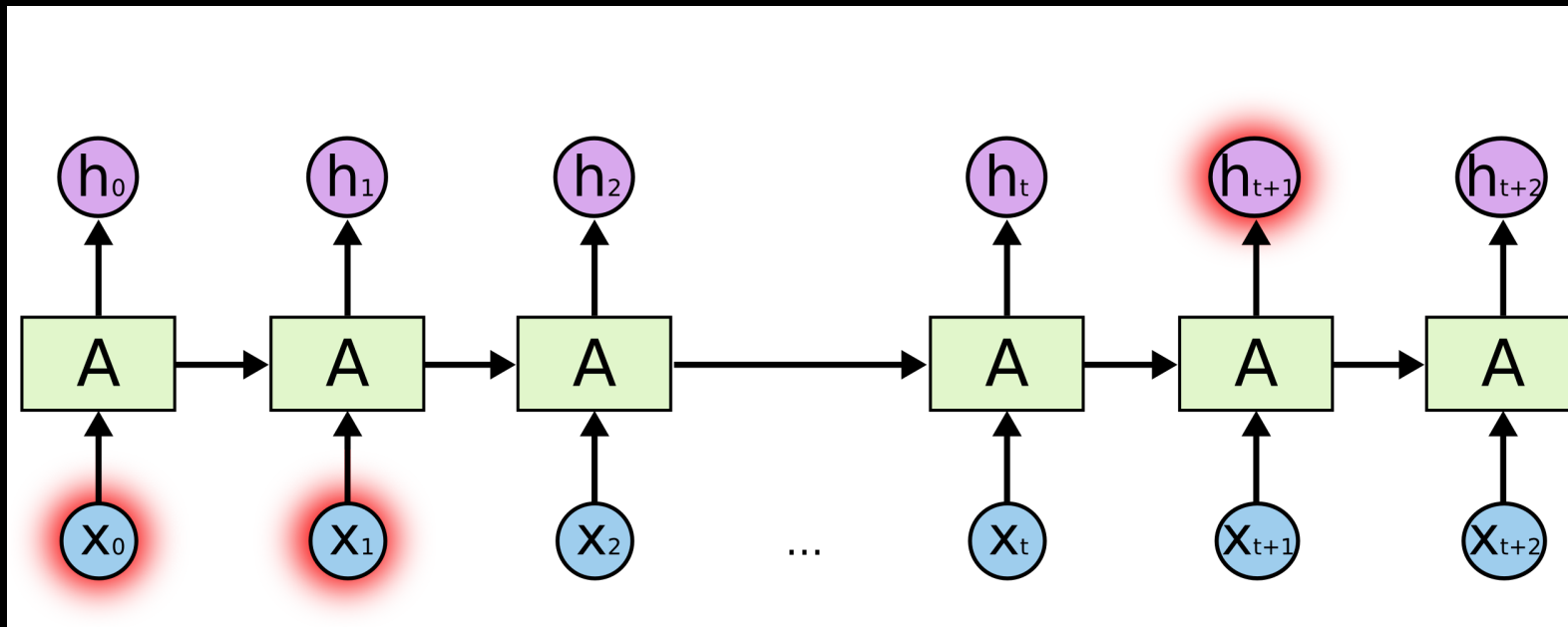


# Recurrent neural nets



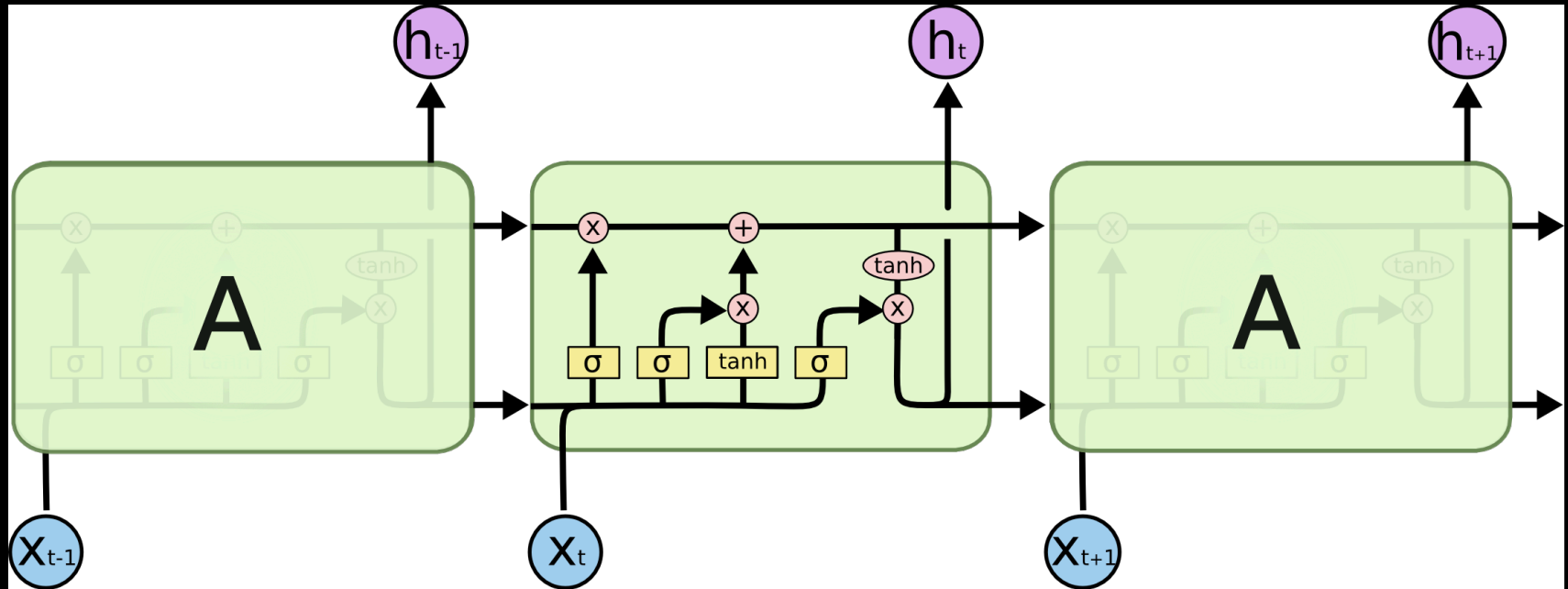
“The grass is green”

# Recurrent neural nets

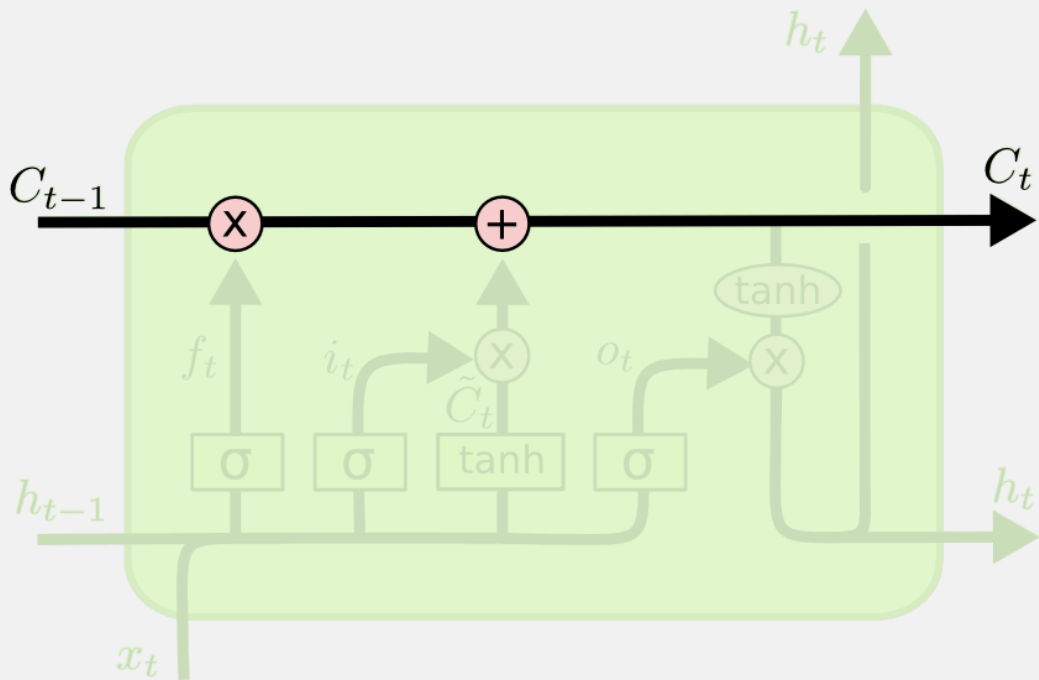


“I am **French**. (2000 words later). I speak fluent **French**”

# Long Short Term Memory Networks (LSTMs)

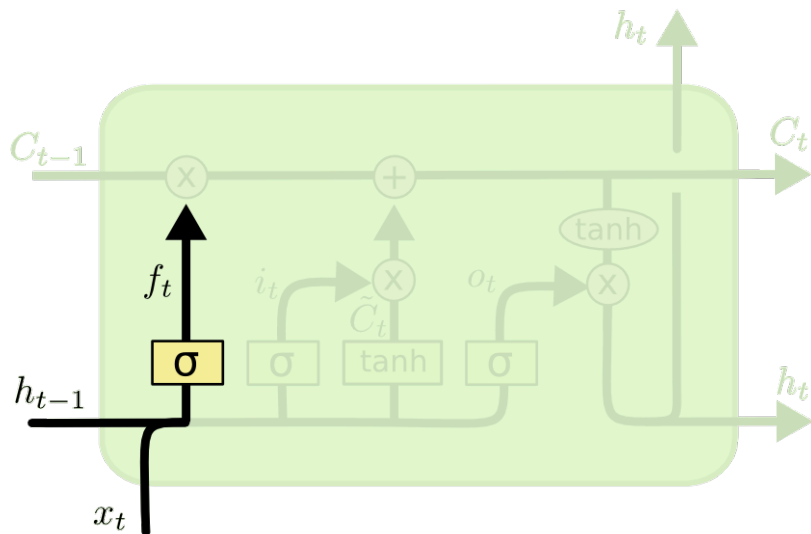


# Cell state



long-term memory

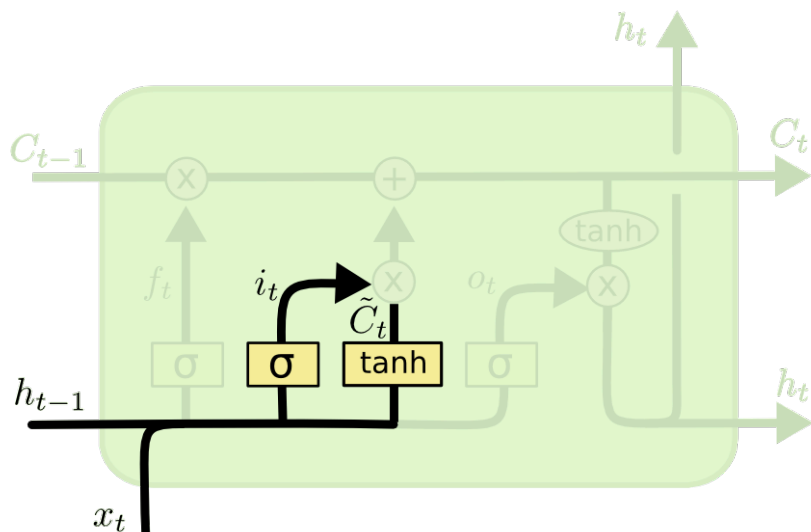
# Forget gate



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

what to keep/forget?

# Input gate



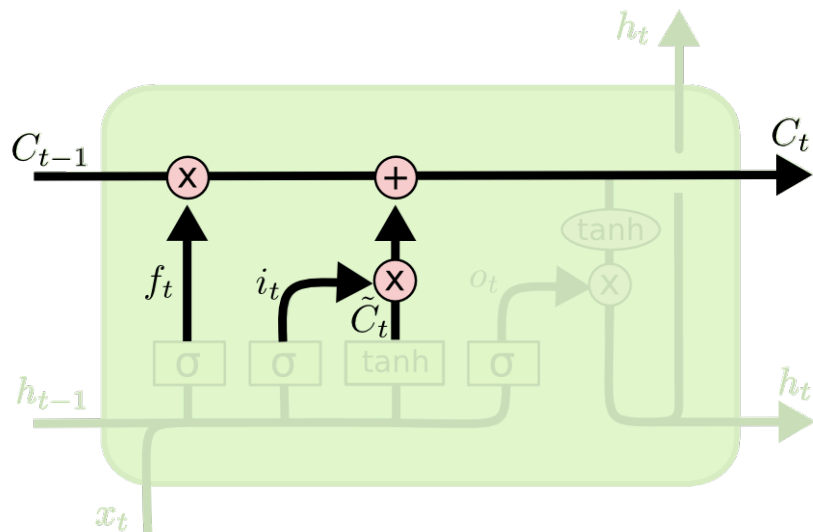
$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i)$$

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

how much of the input to let into the cell state?

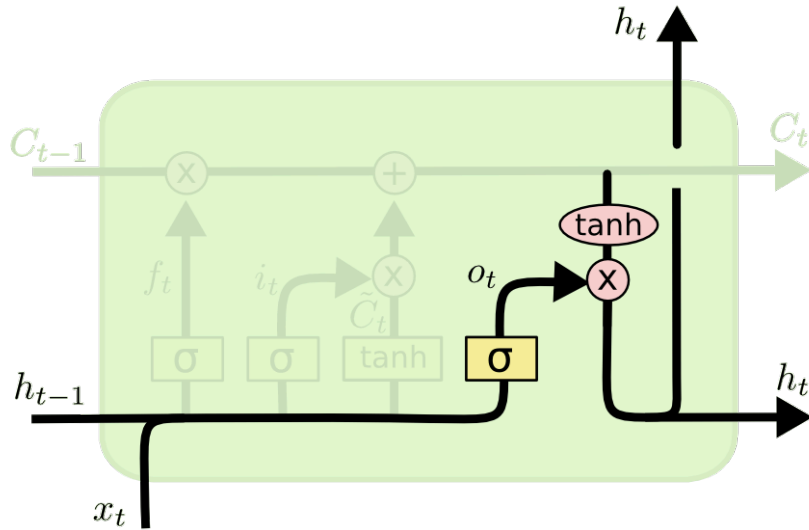


# Updating cell state



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

# Output gate



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

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

hidden state = working memory