Memory Networks

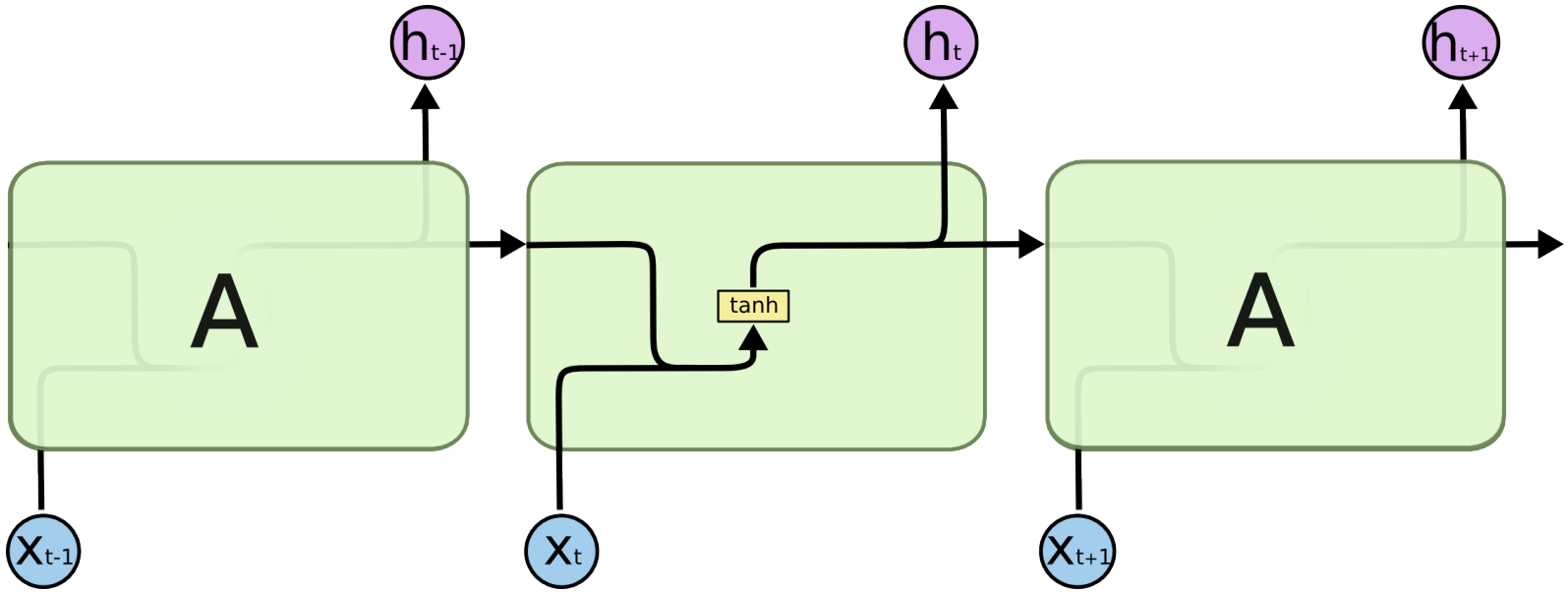
Advanced Machine Learning for NLP

Jordan Boyd-Graber

LSTM WALKTHROUGH

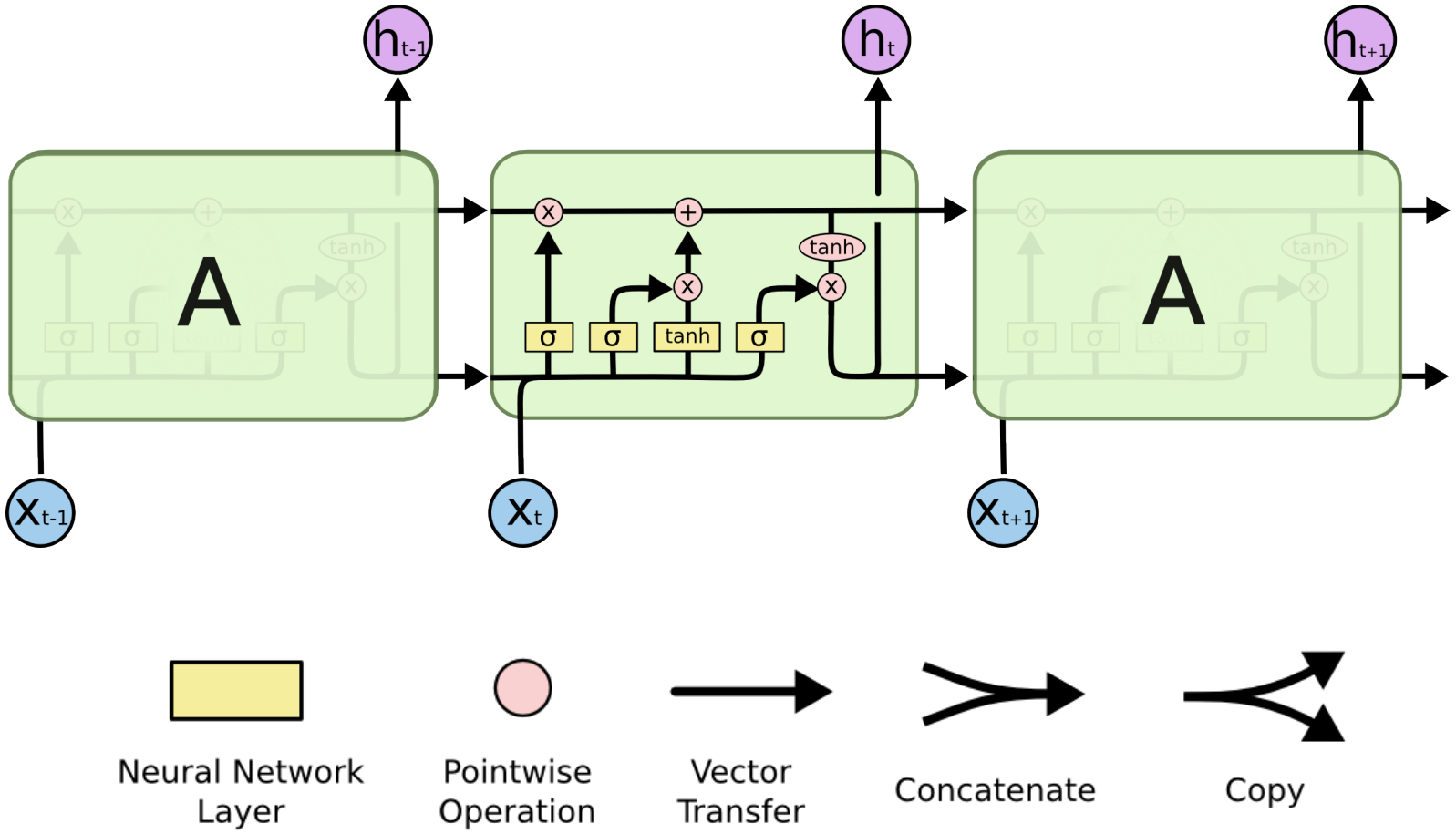
Slides adapted from Christopher Olah

RNN transforms Input into Hidden

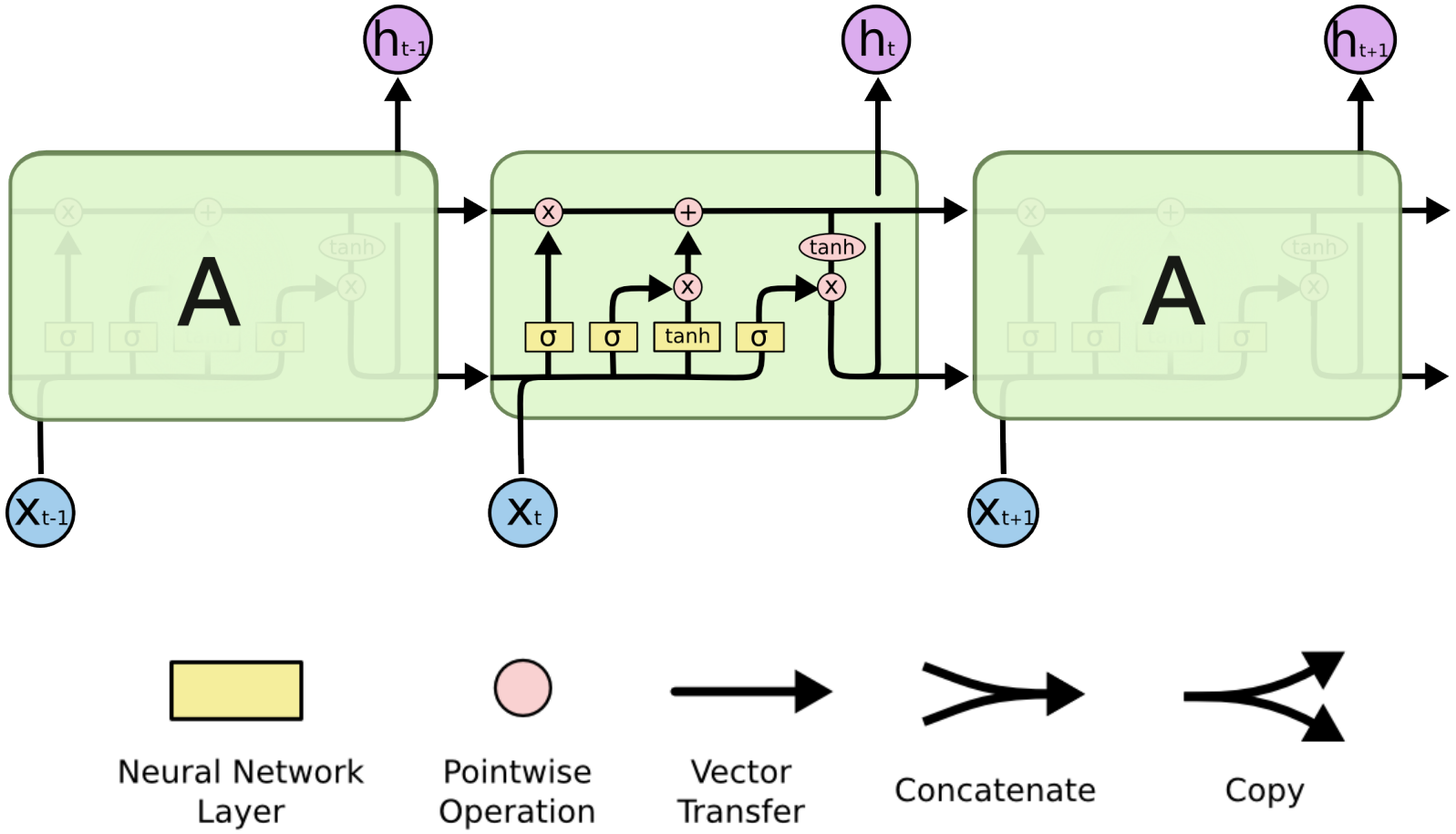


(Can be other nonlinearities)

LSTM has more complicated innards

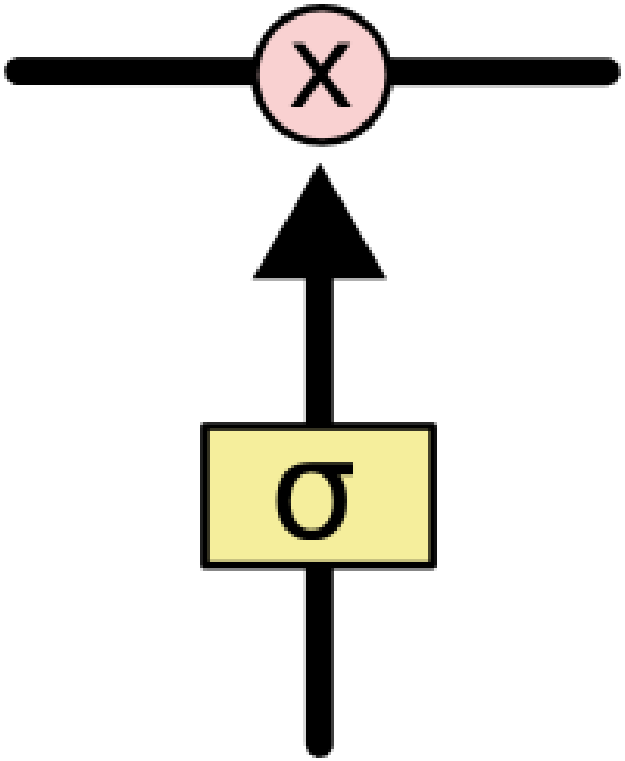


LSTM has more complicated innards



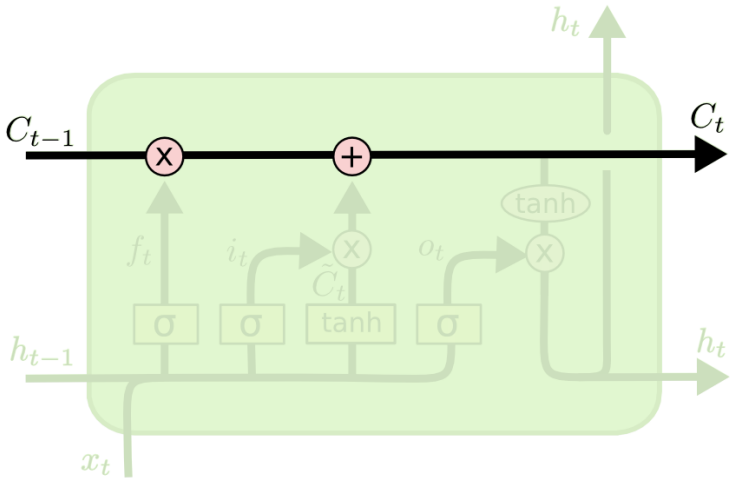
Built on gates!

Gates



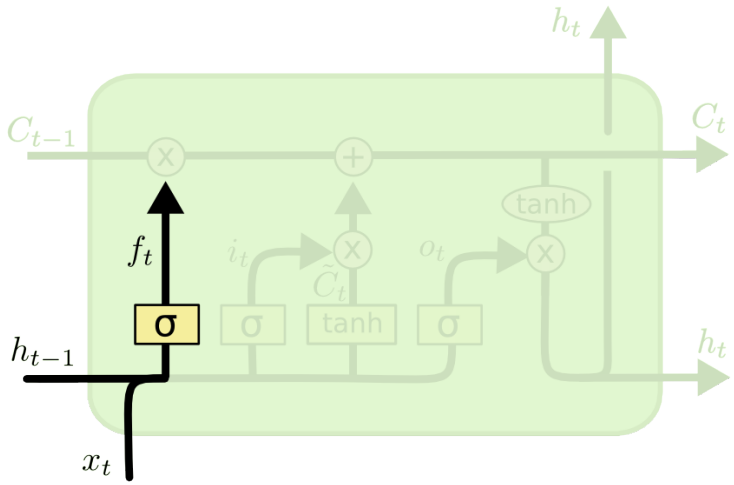
- Multiply vector dimension by value in $[0, 1]$
- Zero means: forget everything
- One means: carry through unchanged
- LSTM has three different gates

Cell State



Can pass through (memory)

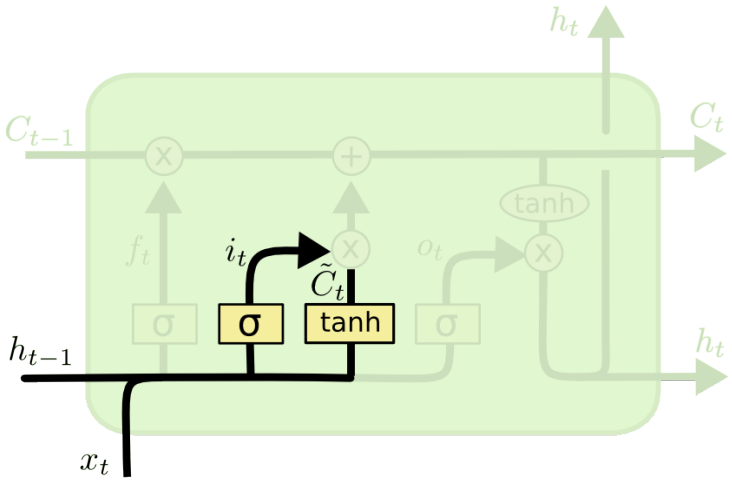
Deciding When to Forget



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

Based on previous hidden state h_{t-1}, can decide to forget past cell state

Updating representation

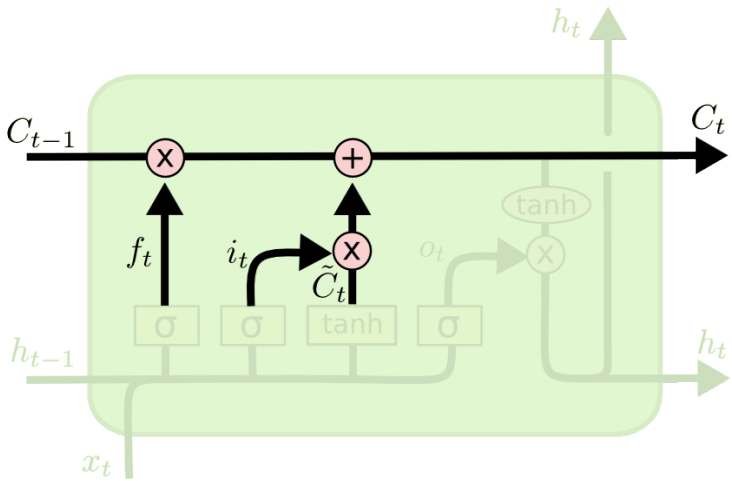


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

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

Compute new contribution to cell state based on hidden state h_{t-1} and input x_t. Strength of contribution is i_t

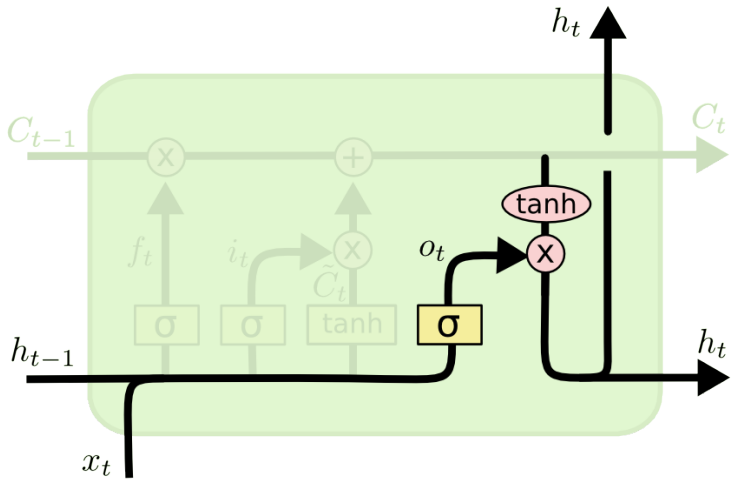
Updating representation



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

Interpolate new cell value

Output hidden



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

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

Hidden layer is function of cell C_t, not h_{t-1}

Loss function

- To create a complete model, need to make a prediction
- Usually function of hidden layer(s)
  - Hinge loss of sentence label based on last word
  - Softmax of tag for each word