Introduction to Recurrent Neural Networks

t.me/cvision

Abolfazl Mahdizade

Outline

- *What is RNN*
- *RNN Different Views*
- *The Rise of LSTMs*
- *LSTMs*
- *Variants on LSTMs*
- *Bidirectional RNN (LSTM)*
- *Stacked RNNs (LSTMs)*
- *TensorFlow-Examples*
- *Introduction to Deep NLP*

What is Recurrent Neural Network

- Recurrent refer to Temporal
- Time Series Data are Temporal
- Human Language are Temporal
- ...

from Wikipedia

- A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed cycle. This allows it to exhibit dynamic temporal behavior. Unlike feedforward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs.

What is Recurrent Neural Network (Continue)

- All Temporal data are **Sequence** of inputs
- **Time Step, Time Unit**
  - Each sample of input sequence data is one time step
- Is it possible to use **FeedForward** networks for **Sequence Data**?
  - Feed all time steps together
  - Use multiple feedforward nets for each time step
- RNN is Different View

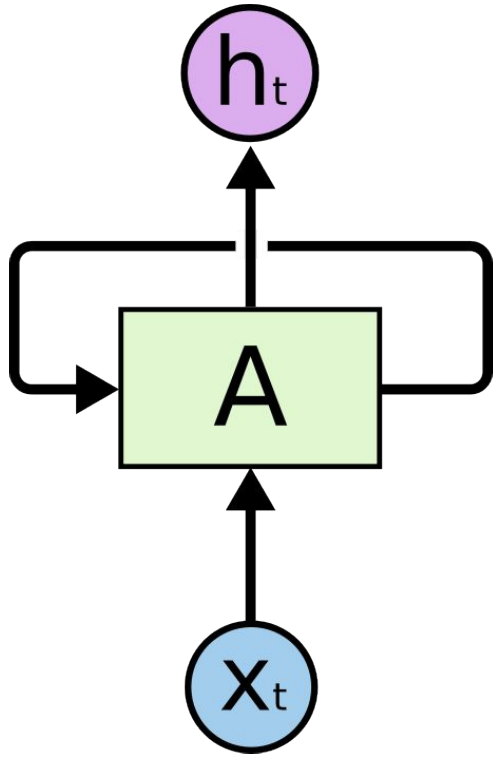
RNN Different Views

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf

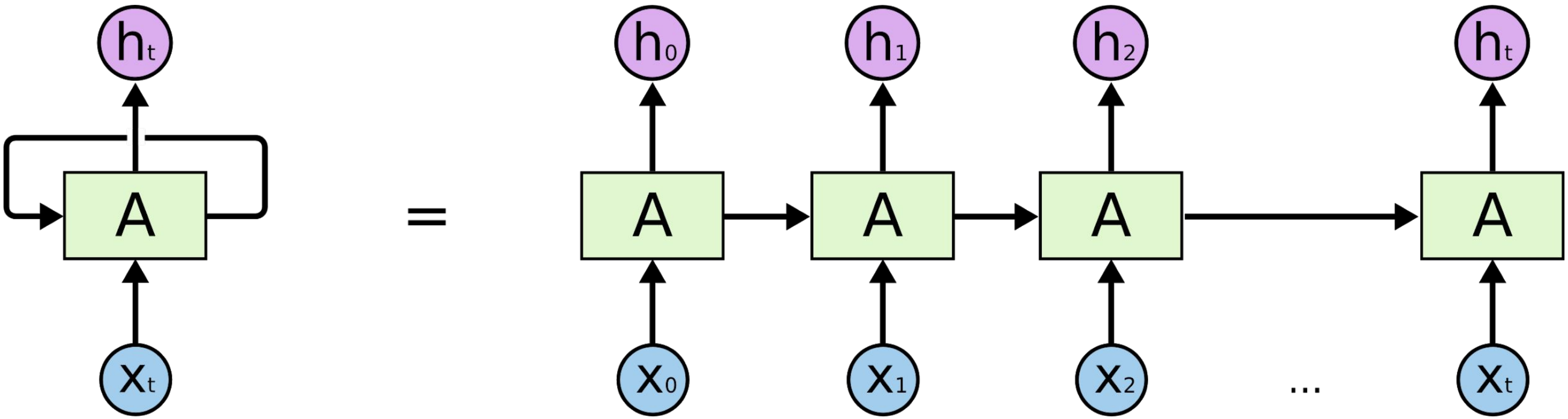
RNN Different Views

- *RNNs are networks with loops in them, allowing information to persist.*
- *RNN Rolled*



RNN Different Views (Continue)

- *A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor. Consider what happens if we unroll the loop:*
- *RNN Unrolled*



RNN Different Views (Continue)

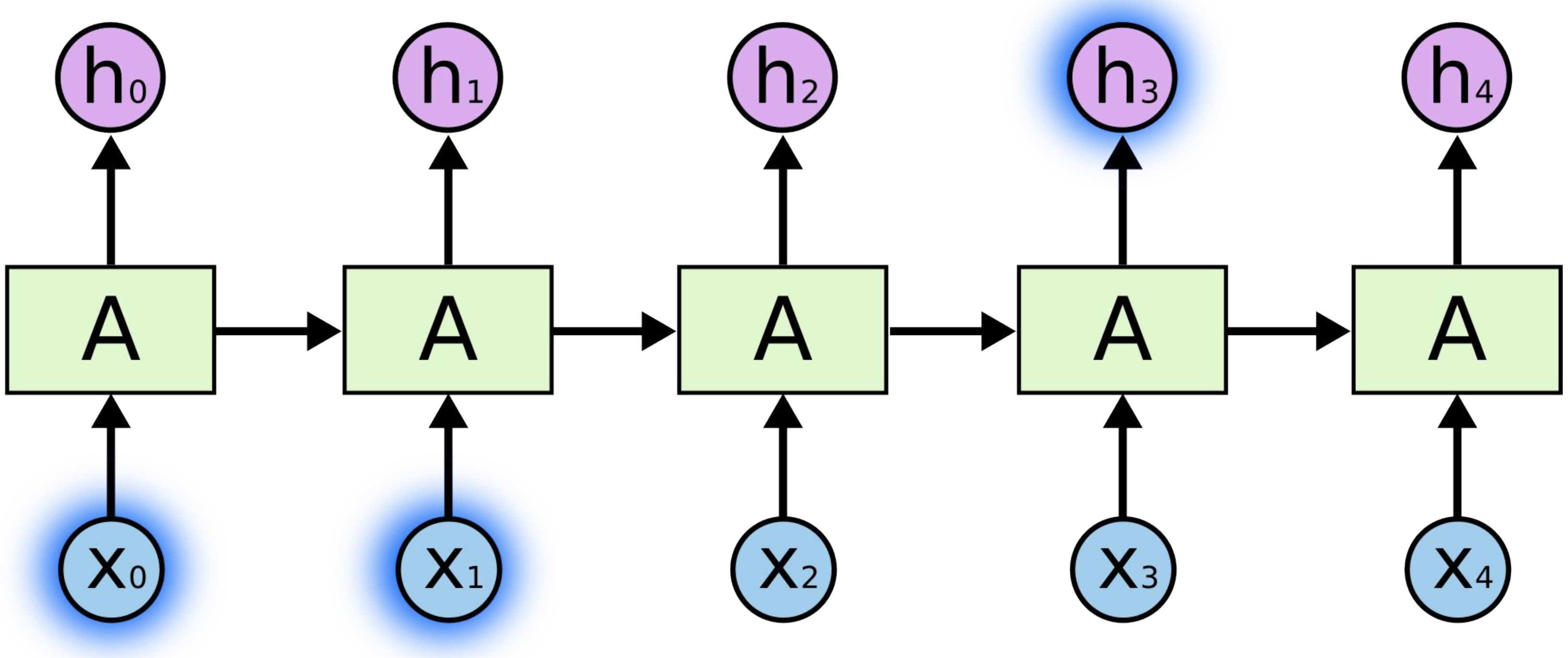
- *RNNs chain-like nature reveals that they are related to*
  - *Sequences*
  - *Time Series*
- *RNNs are the natural architecture of neural network to use for such data*
- *Incredible success applying RNNs to a variety of problems:*
  - *Automatic Speech Recognition*
  - *Language Modeling*
  - *Machine Translation*
  - *Image Captioning*
  - *...*

The Rise of LSTMs

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

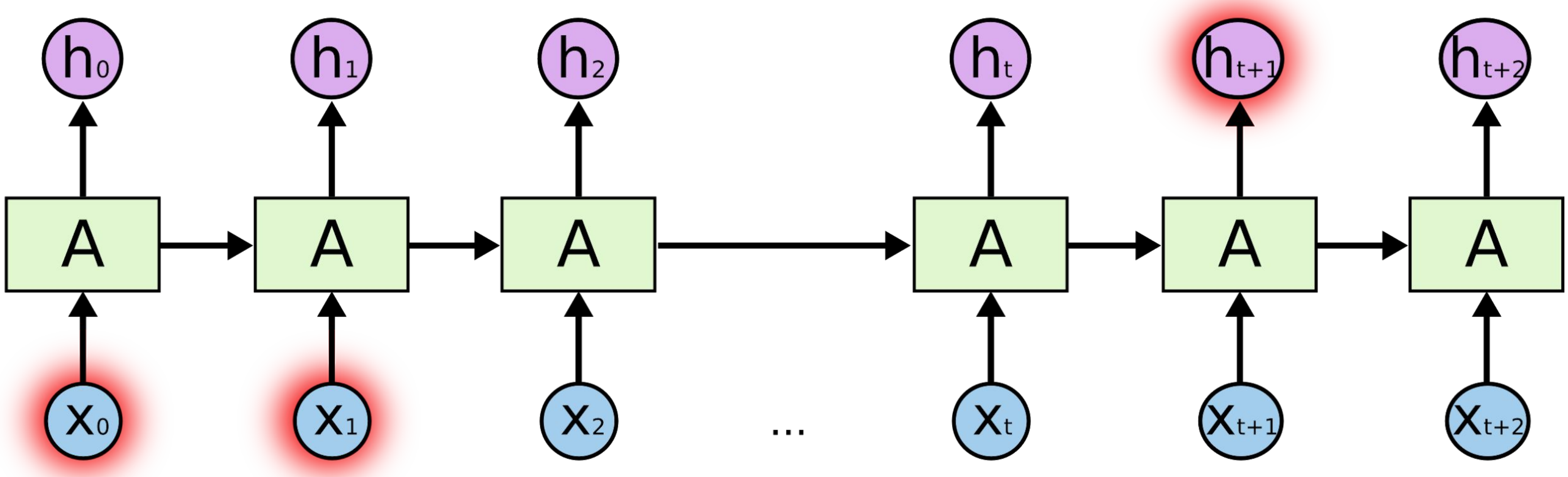
The Rise of LSTMs

- *RNNs should learn to use past information (memory)*
- *Simple RNNs can't memories long informations*



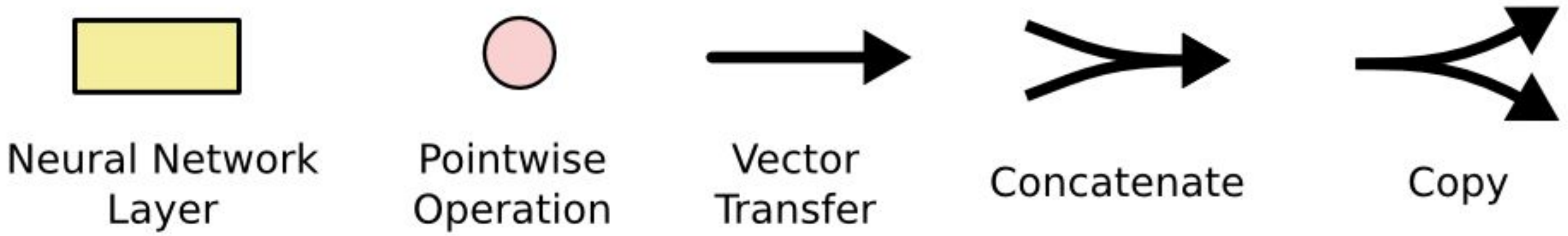
The Rise of LSTMs (Continue)

- *It's entirely possible for the gap between the relevant information and the point where it is needed to become very large*
- *LSTMs are explicitly designed to avoid the long-term dependency problem. (maybe!)*



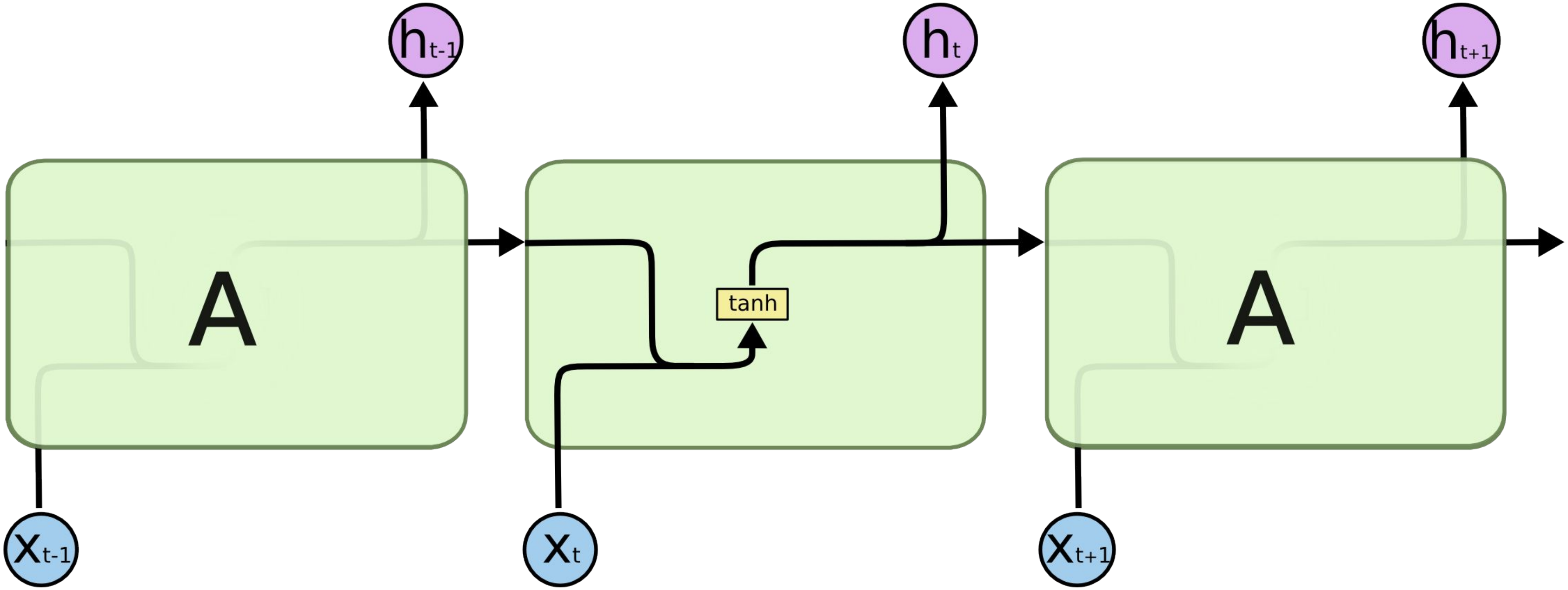
LSTMs

- *LSTMs = Long Short Term Memory networks*
  - *Hochreiter & Schmidhuber (1997)*
- *LSTMs are a special kind of RNN, capable of learning long-term dependencies*
- *Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!*



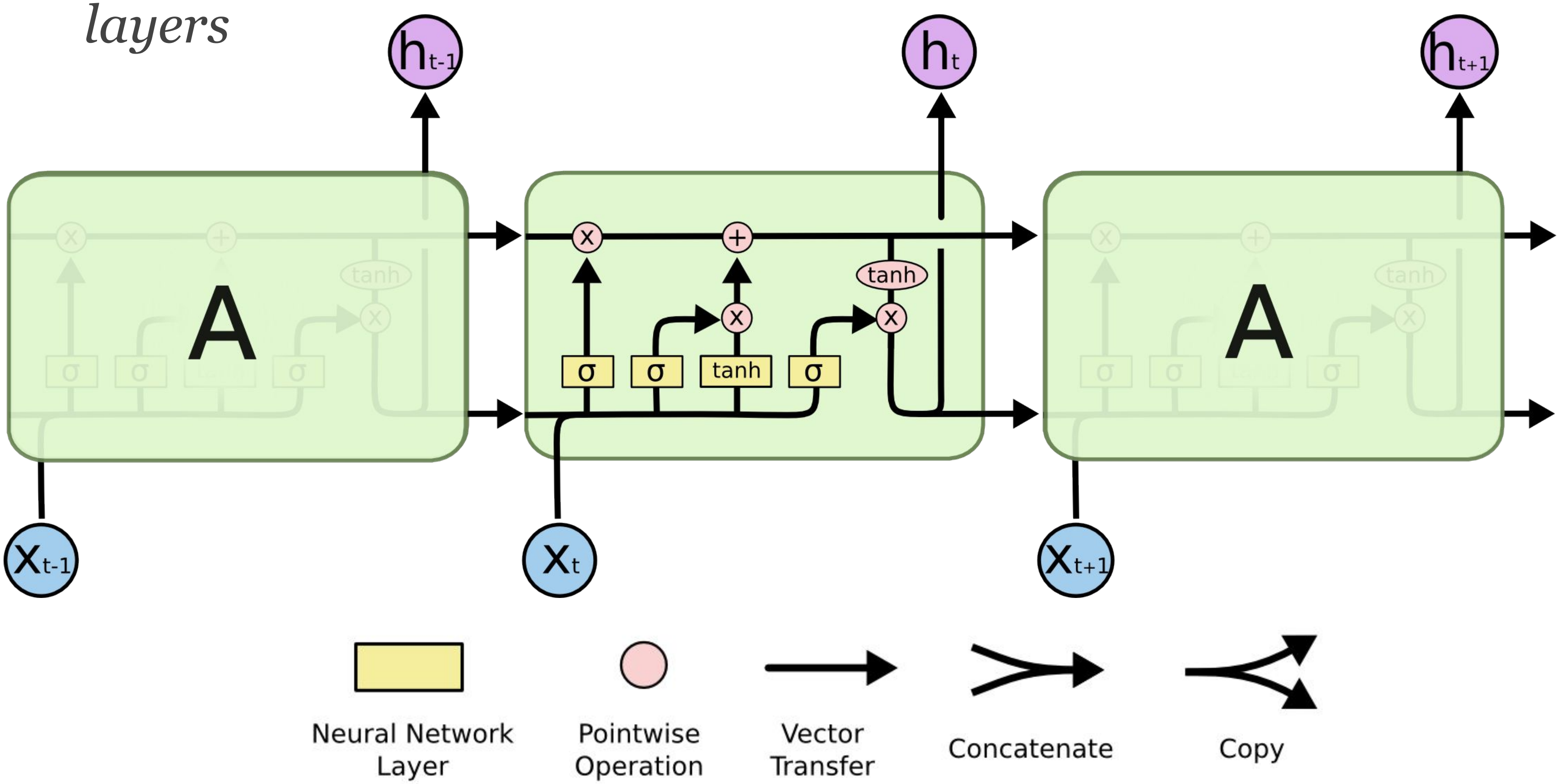
LSTMs (Continue)

- *The repeating module in a standard RNN contains a single layer*



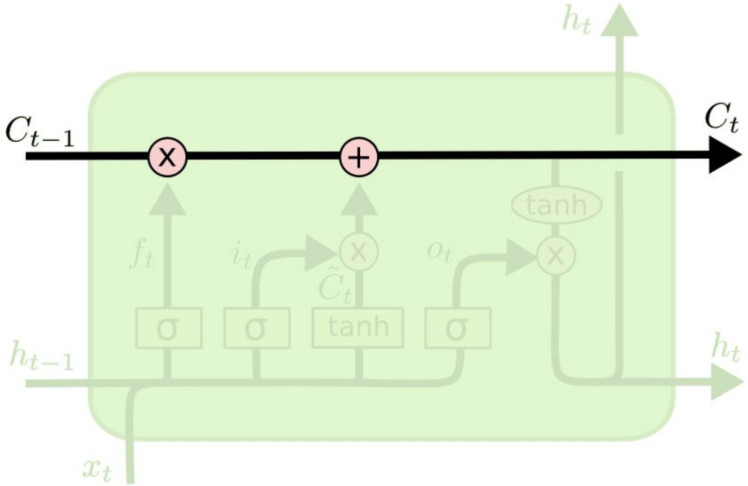
LSTMs (Continue)

- *The repeating module in an LSTM contains four interacting layers*



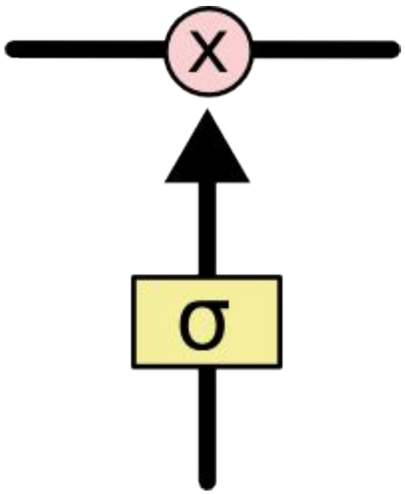
LSTMs (Continue)

- *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 some minor linear interactions. It's very easy for information to just flow along it unchanged*



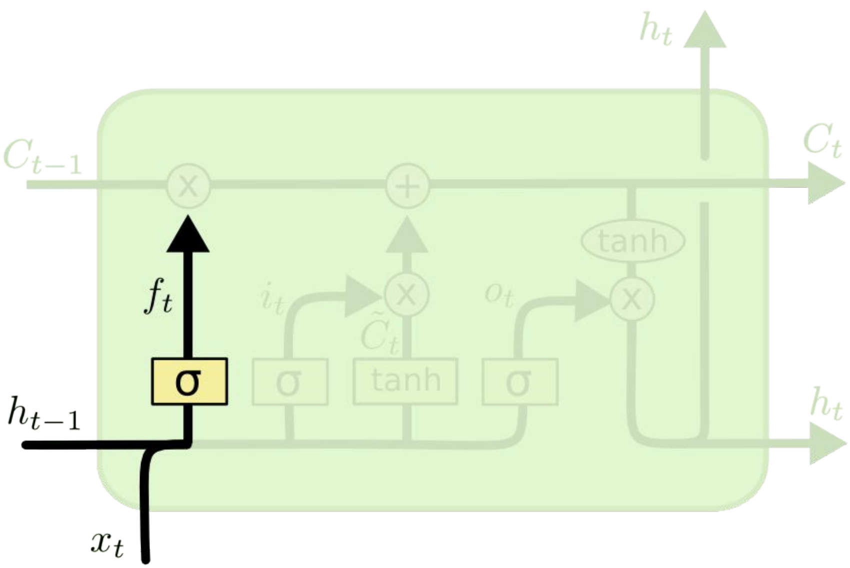
LSTMs (Continue)

- *Gates*
  - *The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates*
- *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*



LSTMs (Continue)

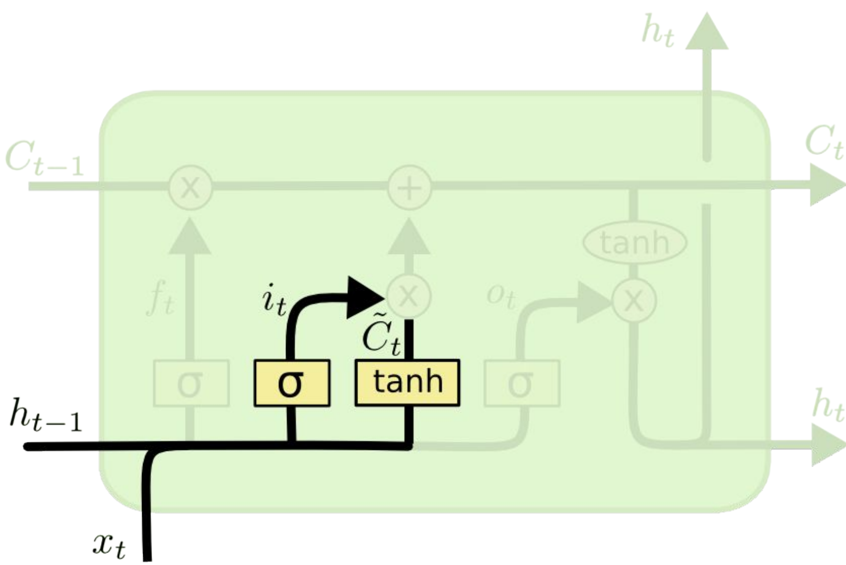
- *Forget gate:*
  - *1 represents "completely keep this" while 0 represents "completely get rid of this."*



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

LSTMs (Continue)

- *Input gate:*
  - *sigmoid layer*
  - *decides which values update*
- *Tanh layer*
  - *creates a vector of new candidate values that could be added to the state*

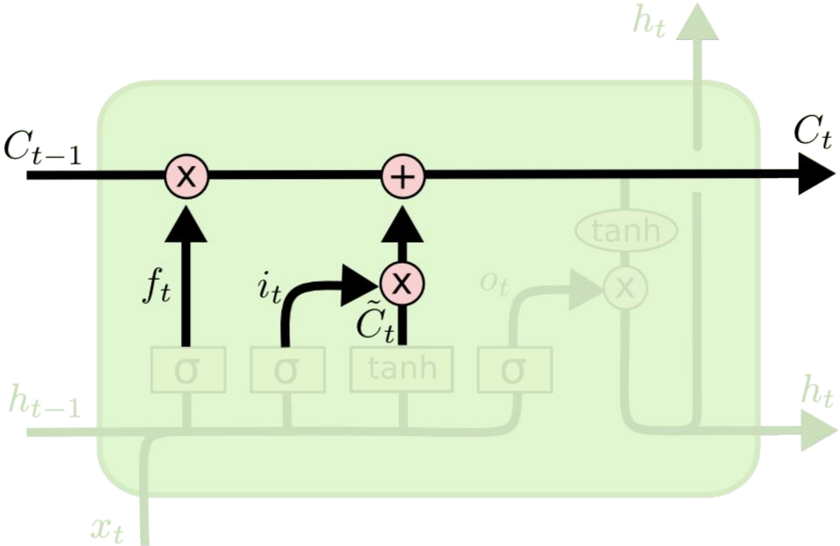


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

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

LSTMs (Continue)

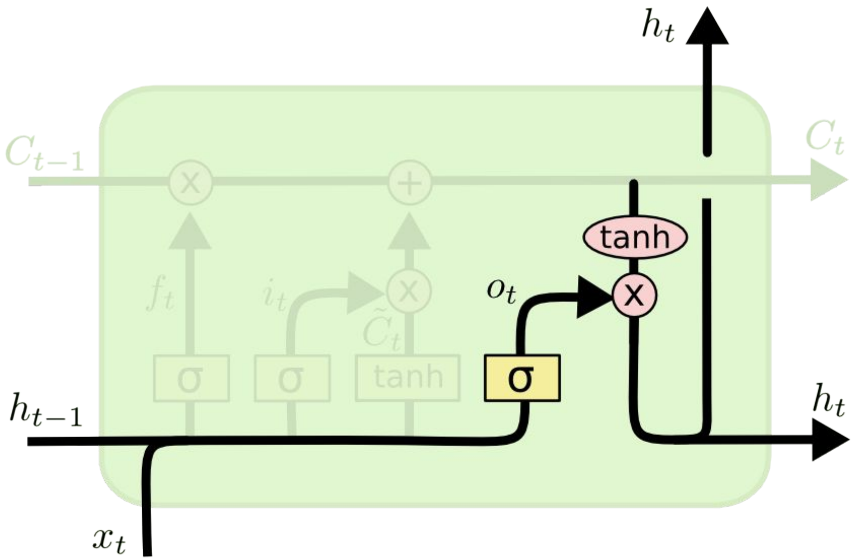
- *Update the state*



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

LSTMs (Continue)

- *Output gate:*
  - *Output will be based on cell state*

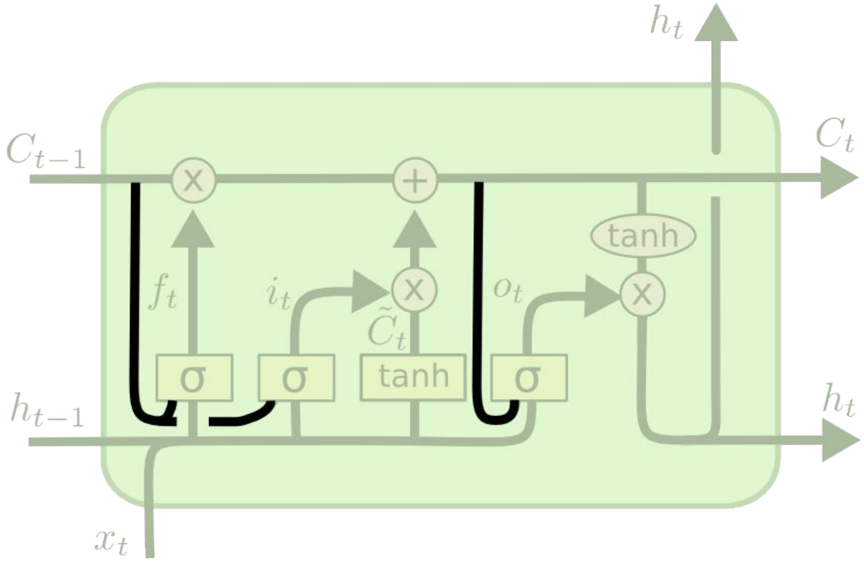


$$o_t = \sigma\left(W_o\ [\,h_{t-1}, x_t]\ +\ b_o\right)$$

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

Variants on LSTMs

- *LSTM with peephole connections*
  - *Let the gate layers look at the cell state*



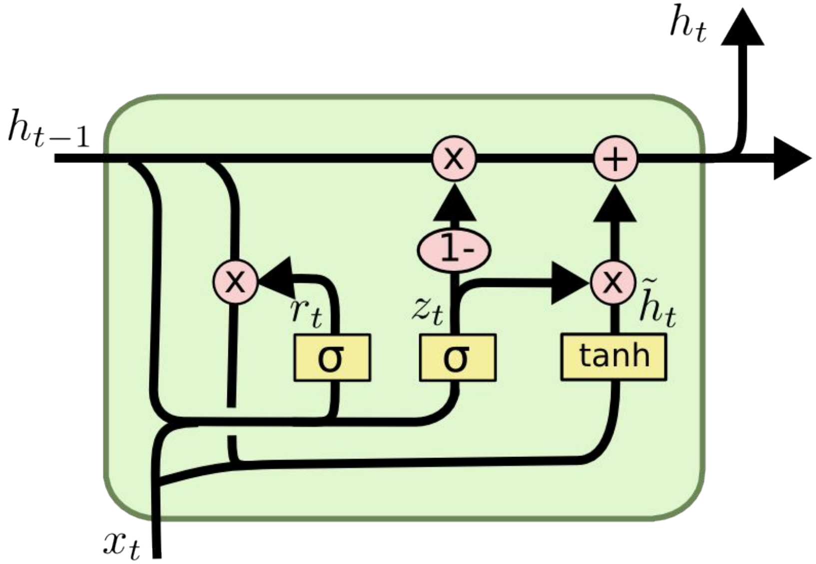
$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] + b_f\right)$$

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

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

Variants on LSTMs

- *GRU: Gated Recurrent Unit*



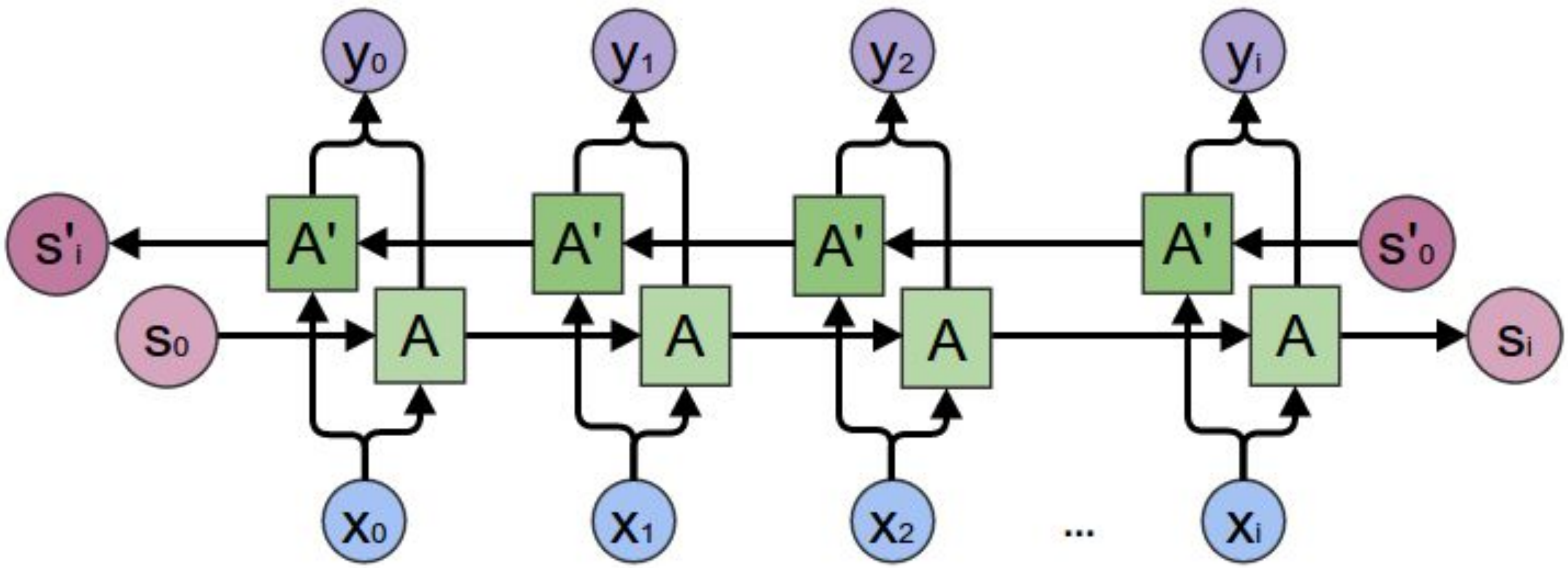
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

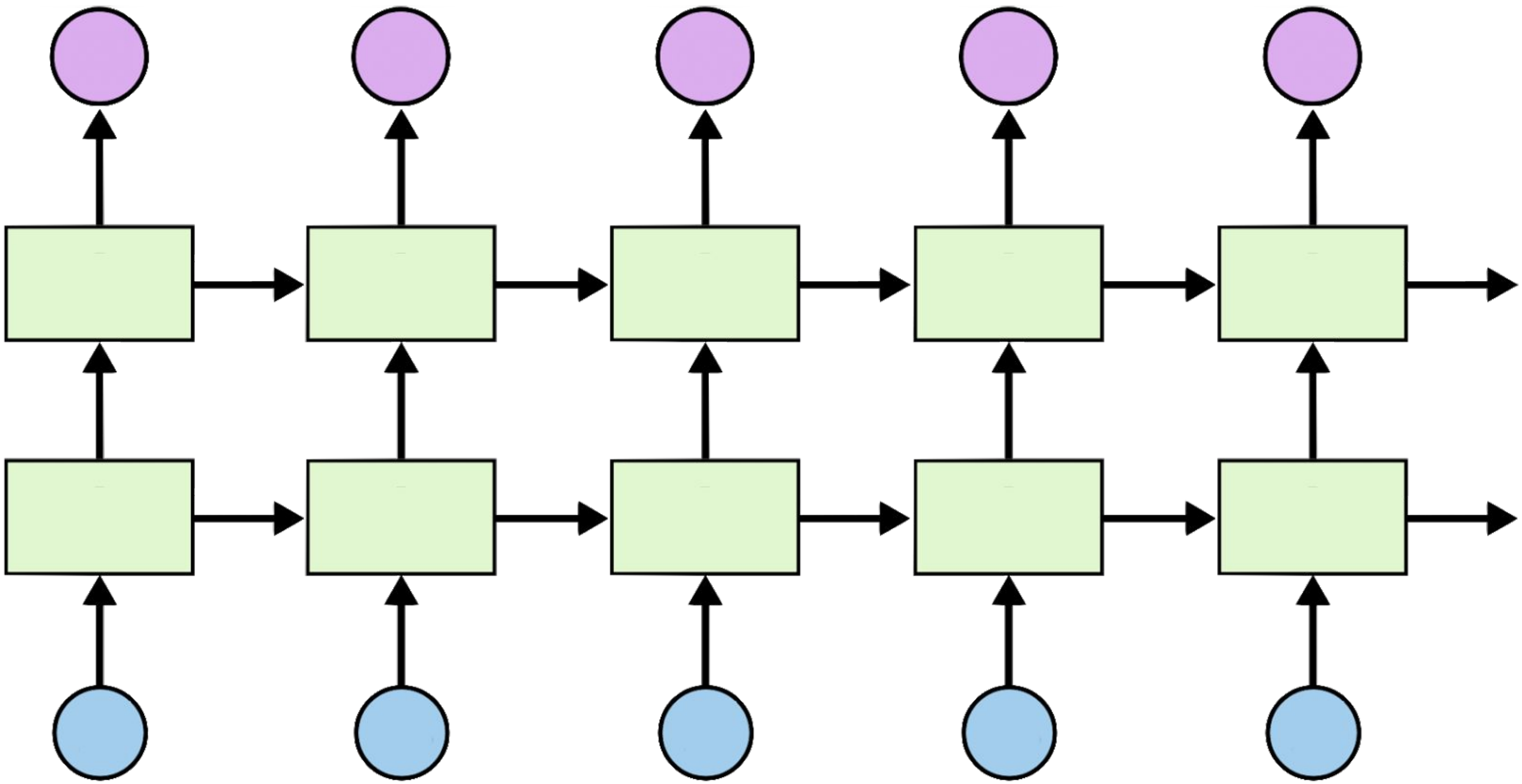
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Bidirectional RNN (LSTM)

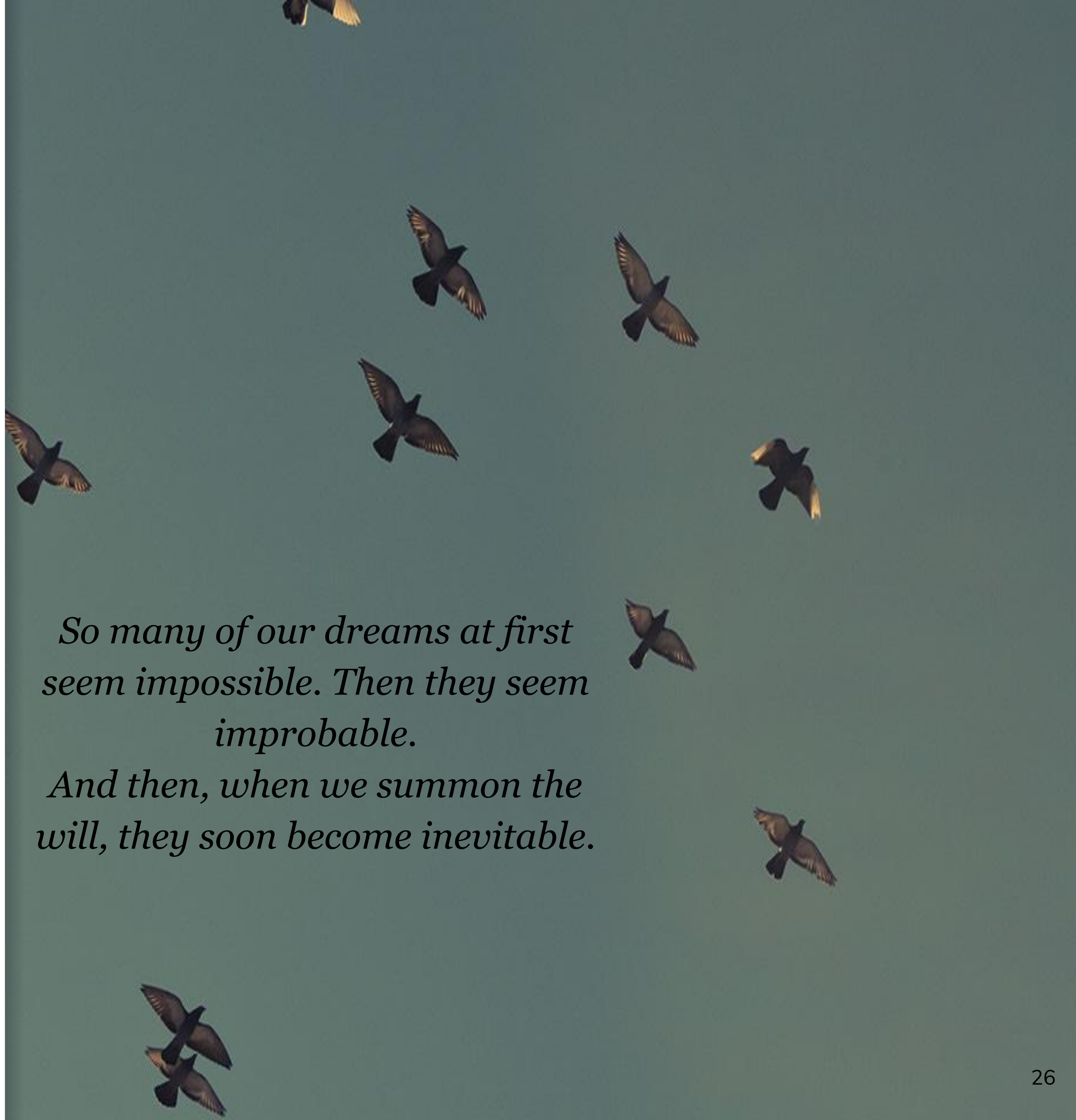


Stacked RNNs (LSTMs)



Introduction to Deep NLP

https://prezi.com/ztuugu7fwtjo/introduction-to-deep-nlp/?utm_campaign=share&utm_medium=copy


So many of our dreams at first seem impossible. Then they seem improbable.
And then, when we summon the will, they soon become inevitable.