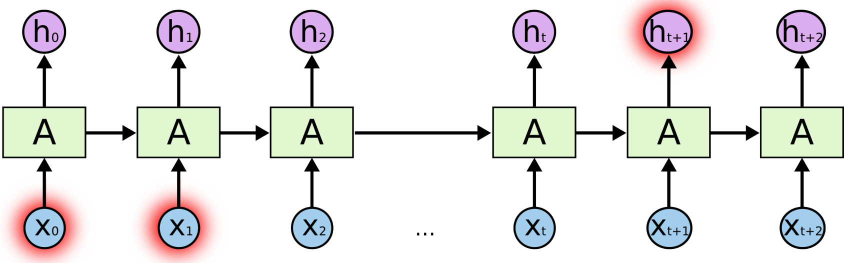
Deep Learning

Vazgen Mikayelyan

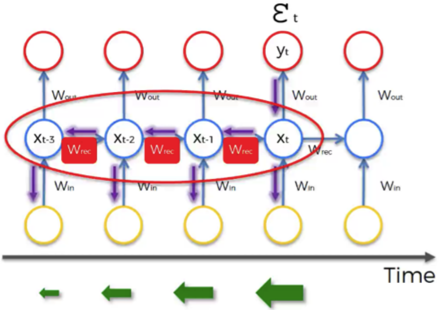
December 1, 2020

FAST DISCOVERING THE FUTURE

Problem of Long Term Dependencies



The Vanishing Gradient Problem



$$\frac{\partial \mathcal{E}}{\partial \theta} = \sum_{1 \le t \le T} \frac{\partial \mathcal{E}_t}{\partial \theta} \quad (3)$$

$$\frac{\partial \mathcal{E}_t}{\partial \theta} = \sum_{1 \le k \le t} \left(\frac{\partial \mathcal{E}_t}{\partial \mathbf{x}_t} \frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} \frac{\partial^+ \mathbf{x}_k}{\partial \theta} \right) \quad (4)$$

$$\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} = \prod_{t \ge i > k} \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_{i-1}} = \prod_{t \ge i > k} \mathbf{W}_{rec}^T diag(\sigma'(\mathbf{x}_{i-1})) \quad (5)$$

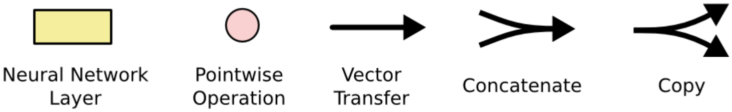
W_{rec} ~ small	⇨	Vanishing
W_{rec} ~ large	⇨	Exploding

Formula Source: Razvan Pascanu et al. (2013)

Deep Learning A-Z

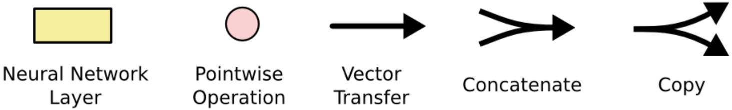
Outline

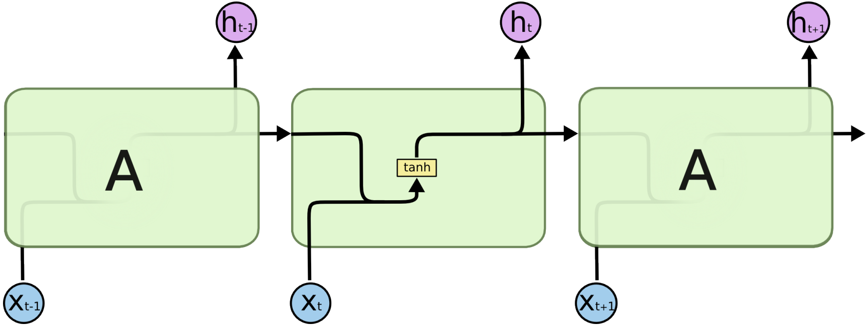
Simple RNN



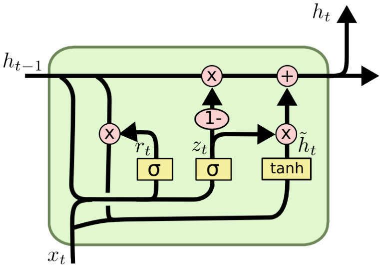
Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy

Simple RNN


Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy


h_{t-1}
h_t
h_{t+1}
A
tanh
A
X_{t-1}
X_t
X_{t+1}

GRU



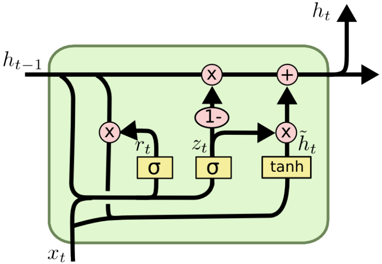
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

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

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

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

GRU

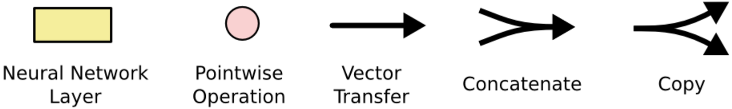


$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

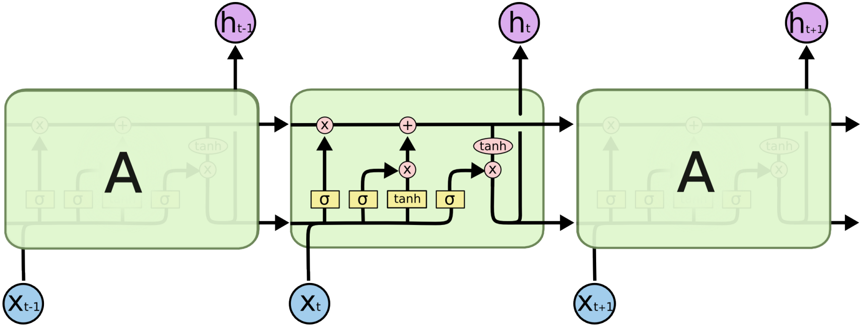
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

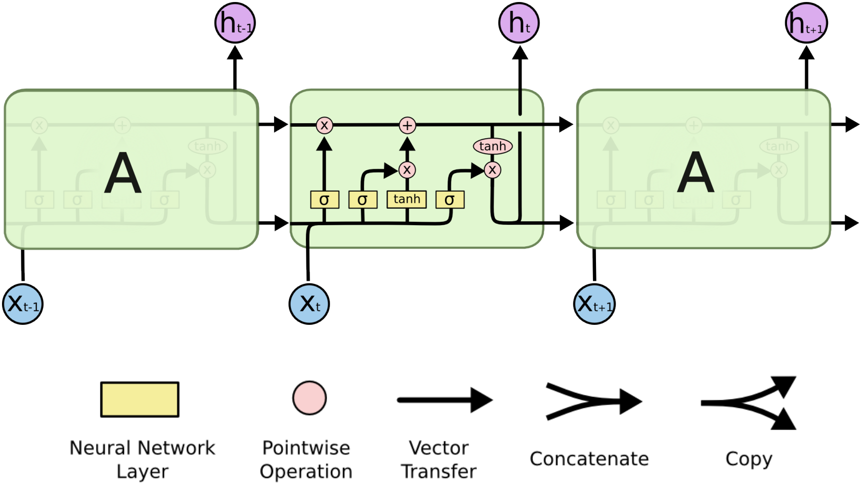
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

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



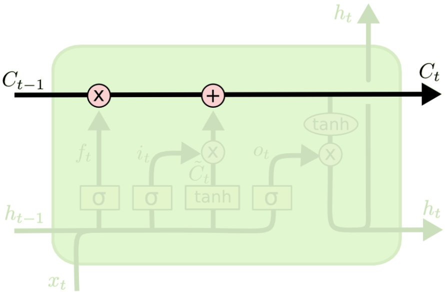
LSTM



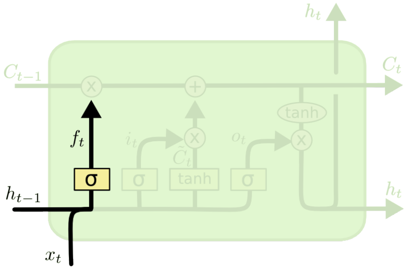


h t-1
h t
h t+1
A
A
σ
σ
tanh
σ
tanh
X t-1
X t
X t+1
Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy

Additional state

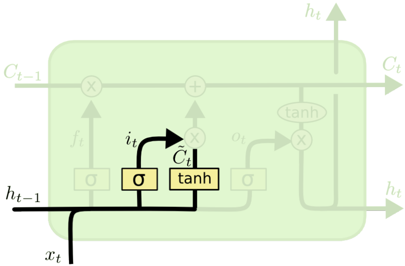


LSTM



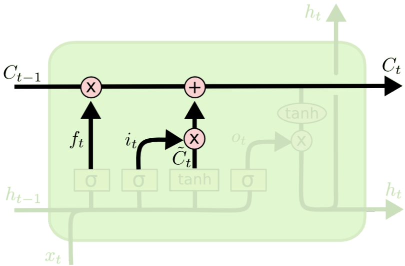
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

LSTM



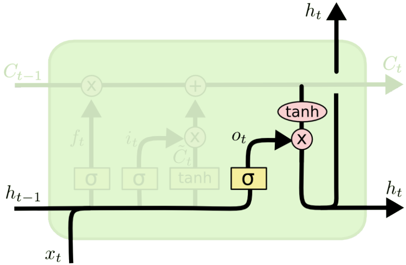
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

LSTM



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

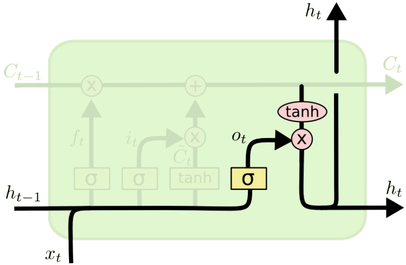
LSTM



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

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

LSTM



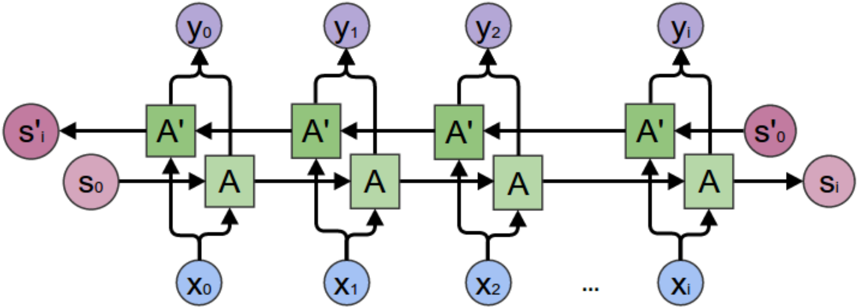
$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$

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

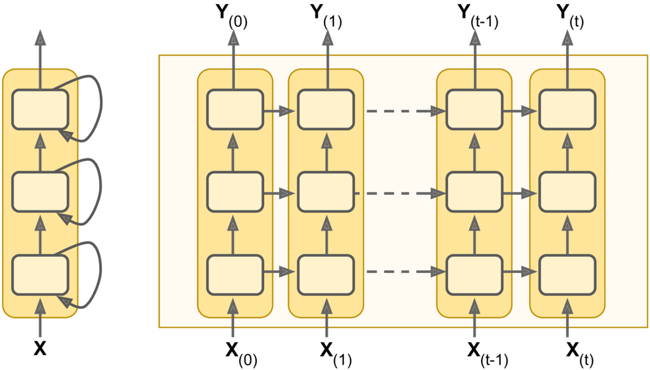
Why to use tanh?

Outline

BiRNN



Deep RNNs



Outline

Basic Idea

- Encode each word in the sentence into a vector using RNNs.

Basic Idea

- Encode each word in the sentence into a vector using RNNs.
- When decoding, perform a convex combination of these vectors, weighted by "attention weights".

Basic Idea

- Encode each word in the sentence into a vector using RNNs.
- When decoding, perform a convex combination of these vectors, weighted by "attention weights".
- Use this combination in picking the next word.

Attention Model