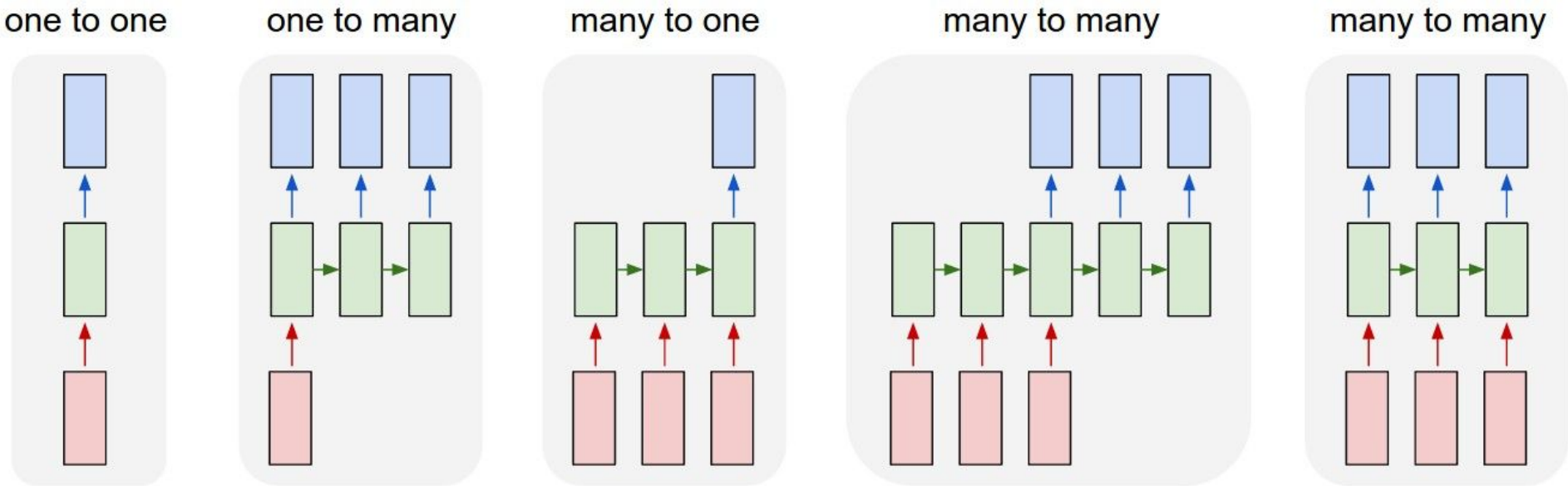
VietAI

Bài 13: LSTM, GRU và các biến thể của Recurrent Neural Network

Tuần 7A

Kiến trúc RNN - Các biến thể



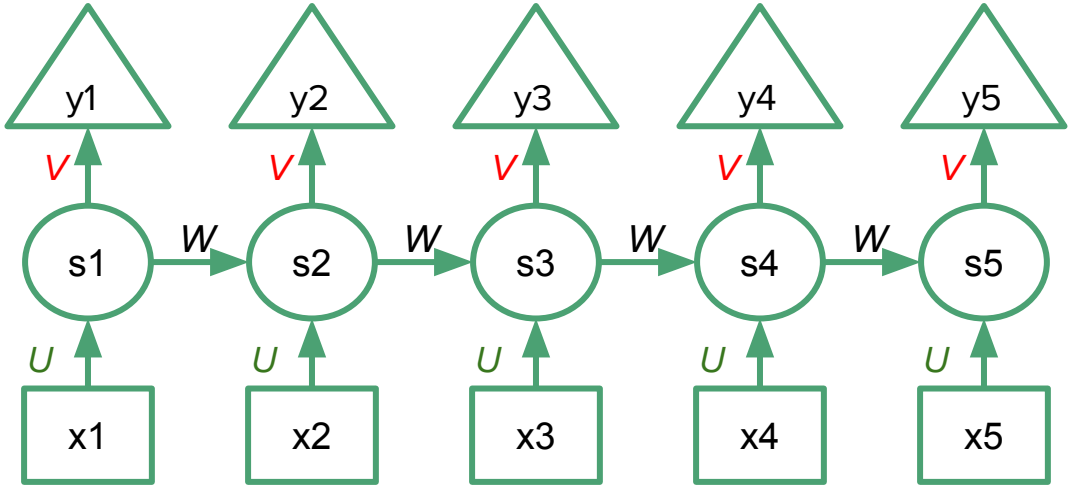
Vd. Text classification, Sentiment Analysis

Vd. Machine Translation, Speech Recognition

Vd. POS Tagging, NER, Language Modelling

(image credits: The Unreasonable Effectiveness of Recurrent Neural Networks, Andrej Karpathy)

Ôn tập RNN



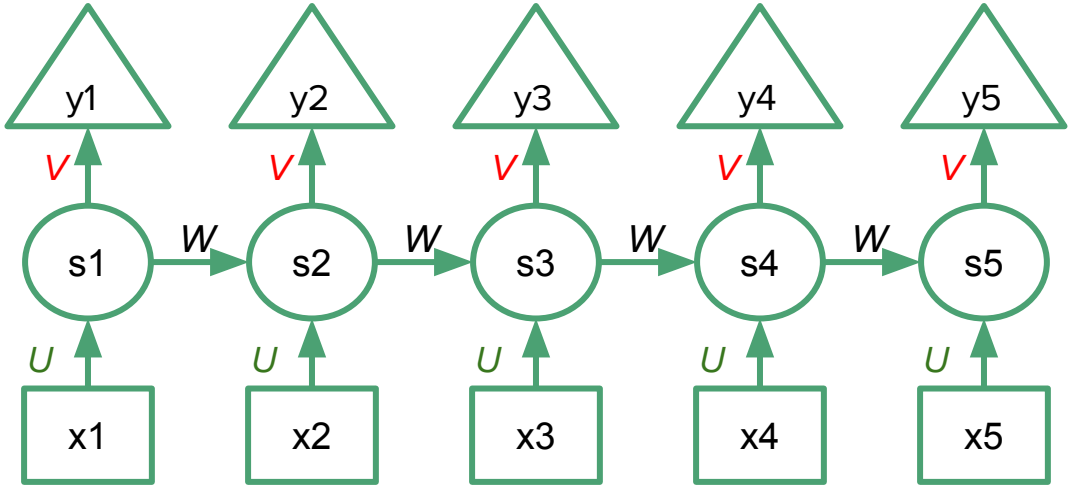
Output *yt*

Hidden *st*

Input *xt*

Ôn tập RNN

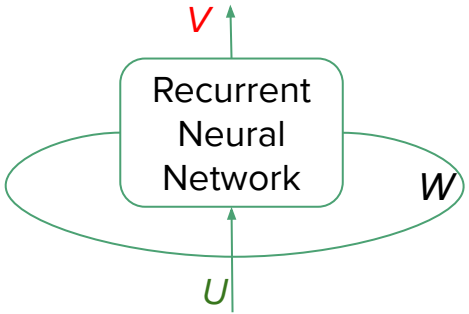
Cấu trúc



Output *yt*

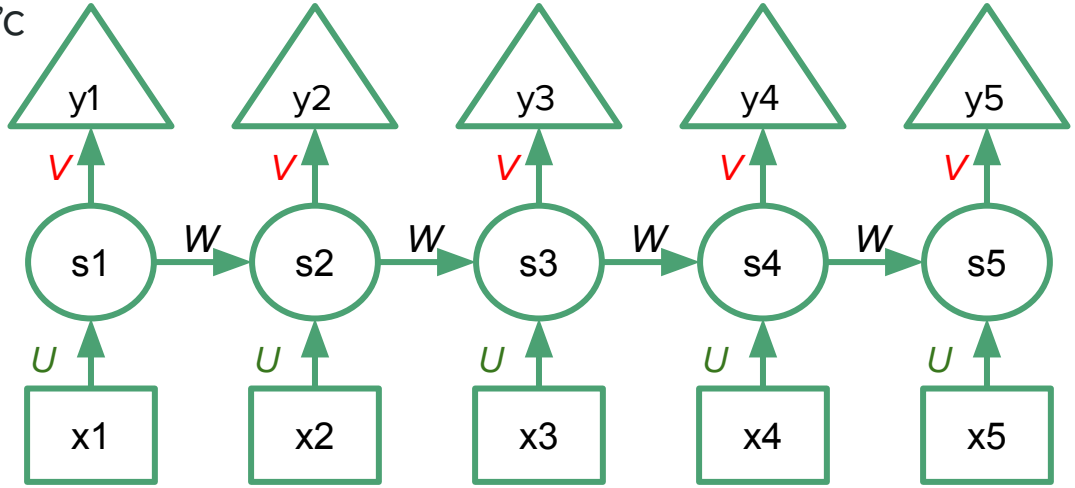
Hidden *st*

Input *xt*



Ôn tập RNN

Công thức



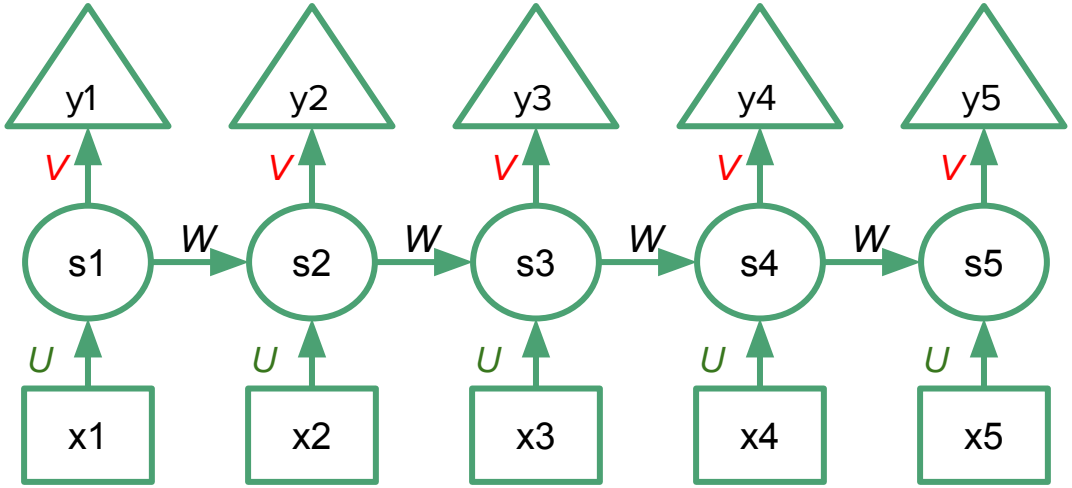
Bước 1: $s_t = \tanh(Ws_{t-1} + Ux_t)$

Bước 2: $y_t = \text{softmax}(Vs_t)$

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

$$\text{softmax}(x_1) = \frac{e^{x_1}}{e^{x_1} + e^{x_2} + \cdots + e^{x_n}}$$

Ôn tập RNN



Hãy tính output *y1*. Biết rằng, params của mạng như sau:

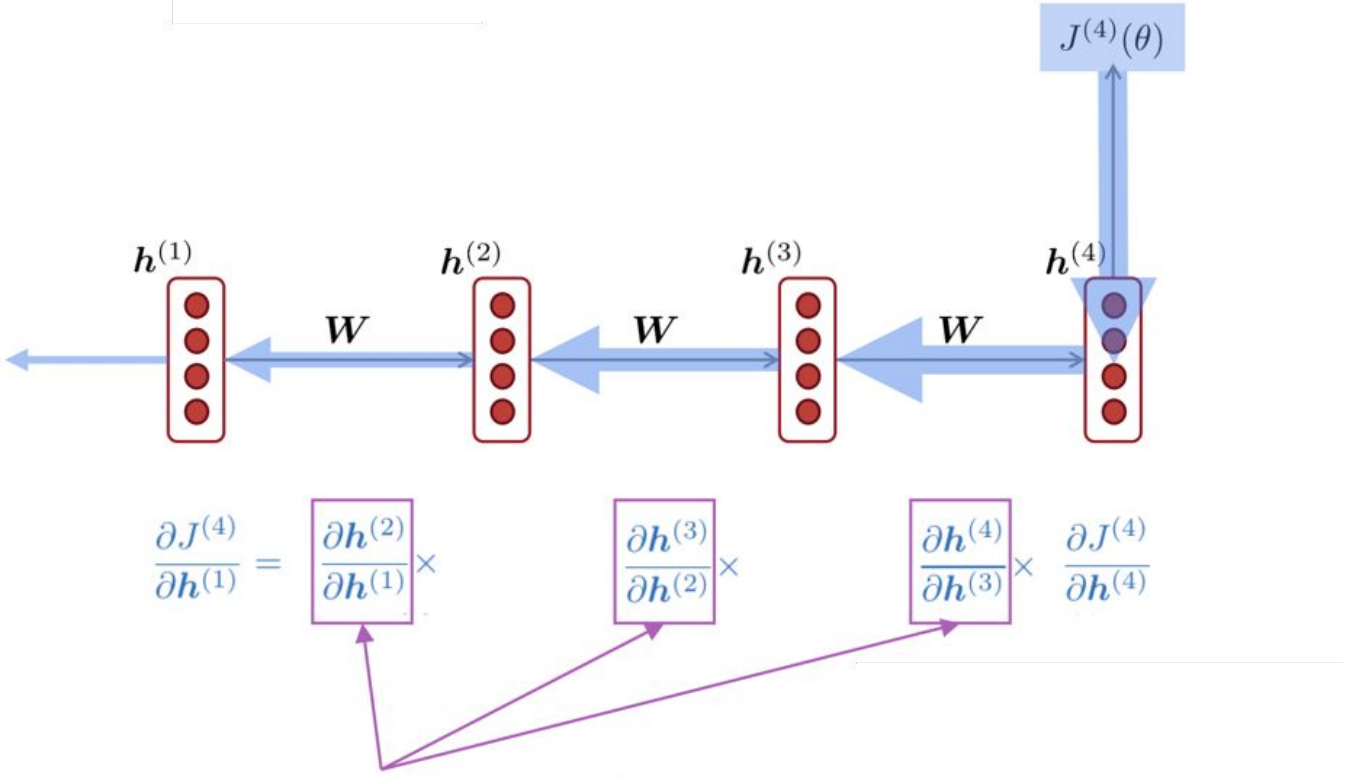
$$\textbf{Input } X = [\ 0.3, 0.1, 0.55, 0.2, 0.7\ ]$$

$$U = 0.75$$

$$W = 0.1$$

$$V = 0.5$$

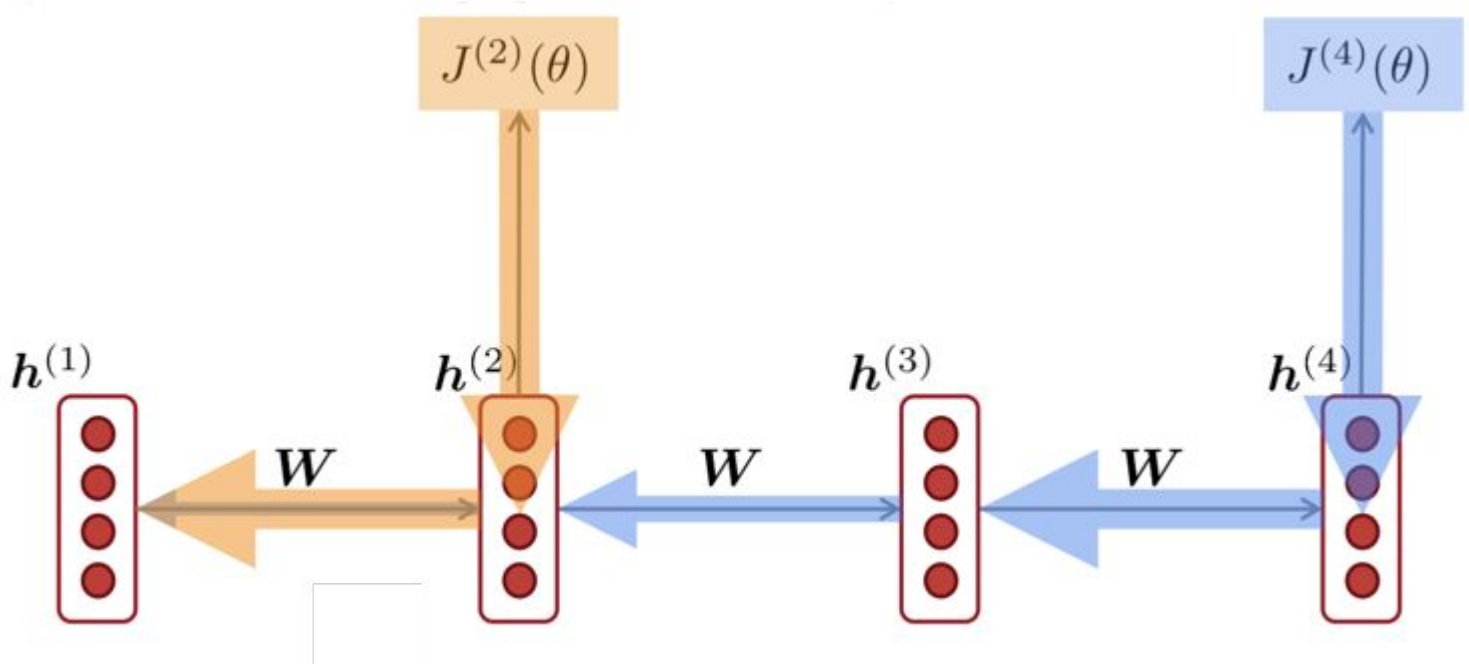
Ôn tập RNN



Vấn đề gì xảy ra nếu các thành phần này nhỏ?

http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture07-fancy-rnn.pdf

Ôn tập RNN



Gradient signal yếu từ các timestep xa, và ngược lại. Điều này gây khó khăn là gì?

http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture07-fancy-rnn.pdf

Outline

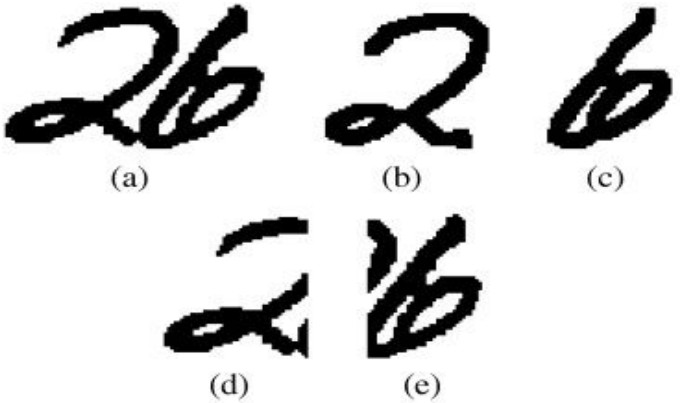
1. Long Short-Term Memory (LSTM)
2. Gated Recurrent Unit (GRU)
3. RNN hai chiều (Bidirectional RNN)
4. RNN nhiều tầng (Deep-stacked RNN)

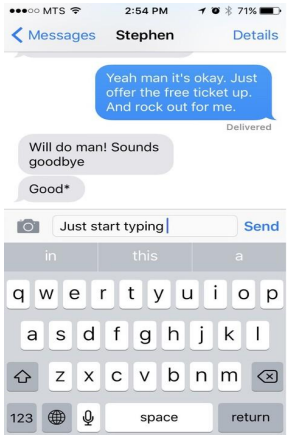
Outline

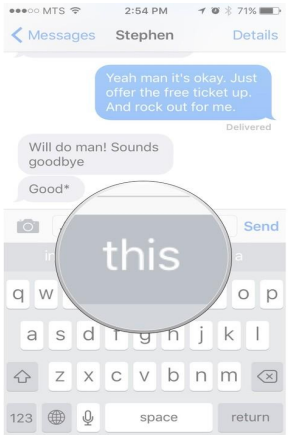
1. Long Short-Term Memory (LSTM)
2. Gated Recurrent Unit (GRU)
3. RNN hai chiều (Bidirectional RNN)
4. RNN nhiều tầng (Deep-stacked RNN)

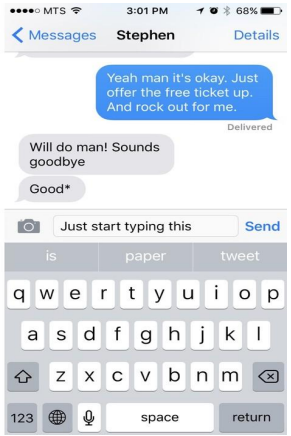
Giới thiệu LSTM

- Long short term memory (LSTM) là một biến thể của RNN
- Bao gồm 4 thành phần chính: **cell state**, **input gate**, **output gate** và **forget gate**
- Các cell có nhiệm vụ nhớ các giá trị trong khoảng thời gian phụ thuộc vào giá trị của cổng **forget gate**
- Ba cổng có nhiệm vụ điều chỉnh luồng thông tin vào và ra khỏi cell









Giới thiệu LSTM

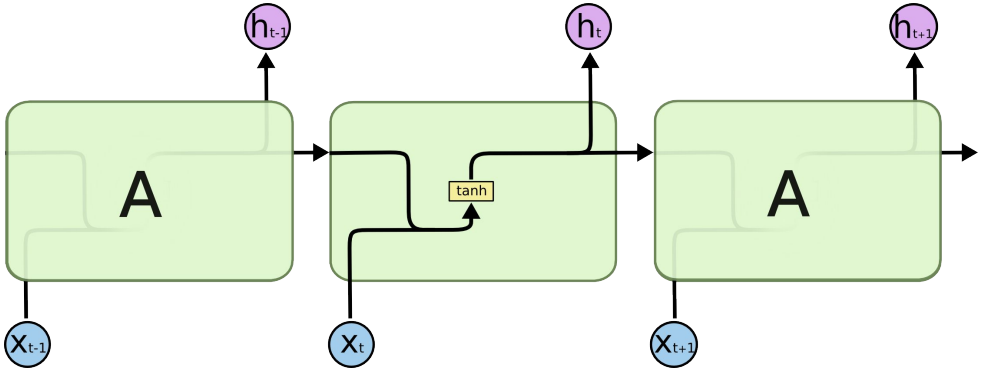
- Mỗi cell trong LSTM network có khả năng xử lý dữ liệu một cách tuần tự
- LSTM rất phù hợp cho bài toán phân loại với dữ liệu theo chuỗi thời gian
- LSTMs được phát triển để giải quyết vấn đề vanishing gradient khi huấn luyện với mô hình vanilla RNN

Long short term memory

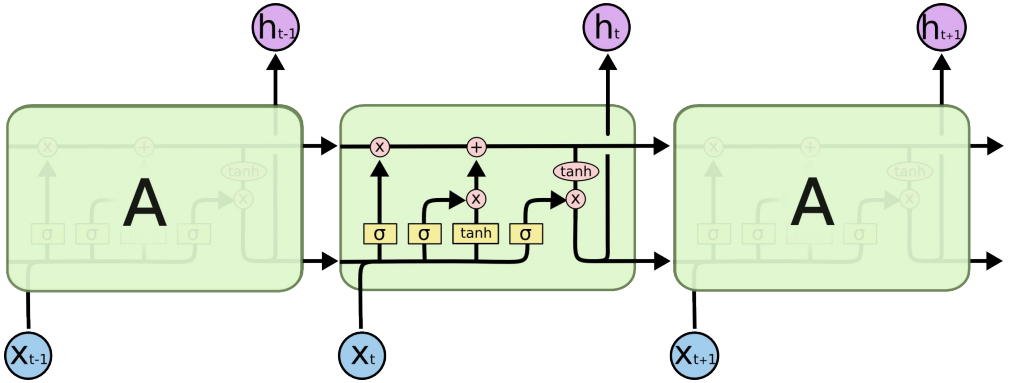
- Mục đích: giúp mạng RNN có thể tận dụng được những thông tin "long term" - giải quyết vấn đề vanishing gradient của vanilla RNN.
- Cách tính hidden units (h) phức tạp hơn
- Ý tưởng chính:
  - Tạo thêm một bộ nhớ (memory) để nhớ được thông tin ở xa trước đó.
  - Cho phép thông tin được đổ xuống bộ nhớ và hidden units mạnh hoặc yếu khác nhau tùy thuộc vào input tại thời điểm đó.

Long short term memory

$$h_t = \sigma\left(W x_t + U h_{t-1}\right)$$



$$\begin{aligned}
f_t &= \sigma\left(W_f x_t + U_f h_{t-1}\right) \\
i_t &= \sigma\left(W_i x_t + U_i h_{t-1}\right) \\
\widetilde{C}_t &= \tanh\left(W_C x_t + U_C h_{t-1}\right) \\
C_t &= f_t \circ C_{t-1} + i_t \circ \widetilde{C}_t \\
o_t &= \sigma\left(W_o x_t + U_o h_{t-1}\right) \\
h_t &= o_t \circ \tanh\left(C_t\right)
\end{aligned}$$

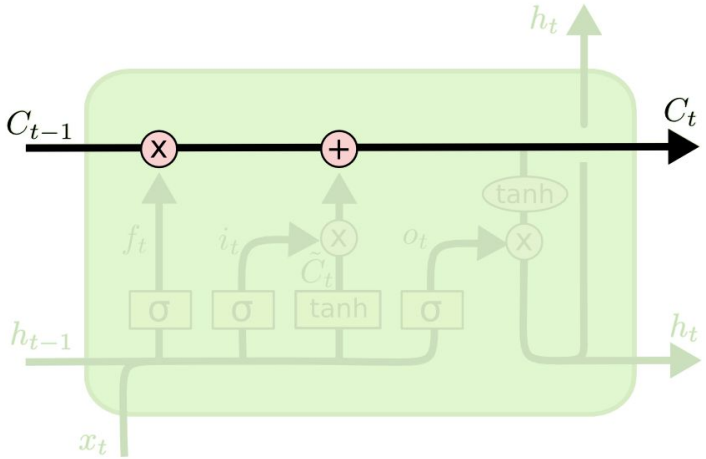


http://colah.github.io/posts/2015-08-Understanding-LSTMs/

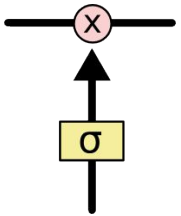
Long short term memory

Cell state

- LSTM có khả năng kiểm soát việc thông tin được thêm vào hoặc bỏ ra thông qua các "cổng" (gates)
- Các gates được tạo thành từ đầu vào là x, h qua hàm sigmoid và kết hợp bởi tích Hadamard.
- Hàm sigmoid có giá trị từ 0 đến 1, quyết định độ "mạnh/yếu" của thông tin đi qua.



Cell state (C)



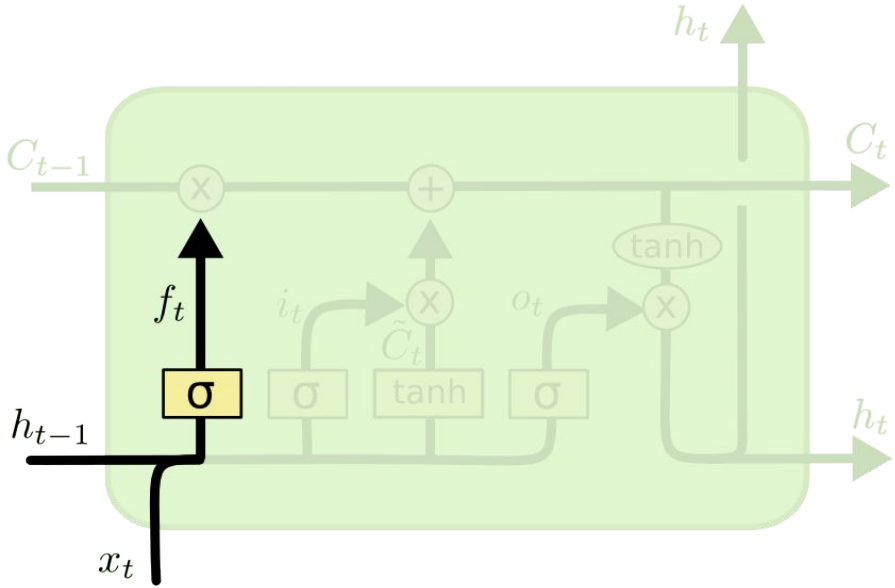
Gates

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

Long short term memory

Forget gate

Forget gate: quyết định bao nhiêu thông tin cũ (C_{t-1}) phải bỏ đi.



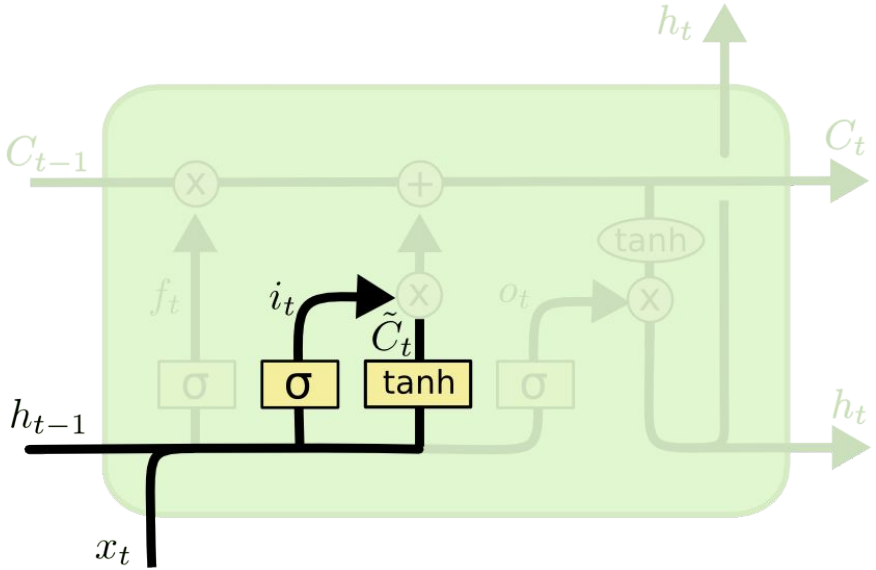
$$f_t = \sigma\left(W_f x_t + U_f h_{t-1}\right)$$

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

Long short term memory

Input gate

Input gate: quyết định thông tin mới nào sẽ được lưu vào cell. Bước này gồm 2 phần: tính input gate và tính giá trị m $\tilde{C}$ sẽ được thêm vào C.



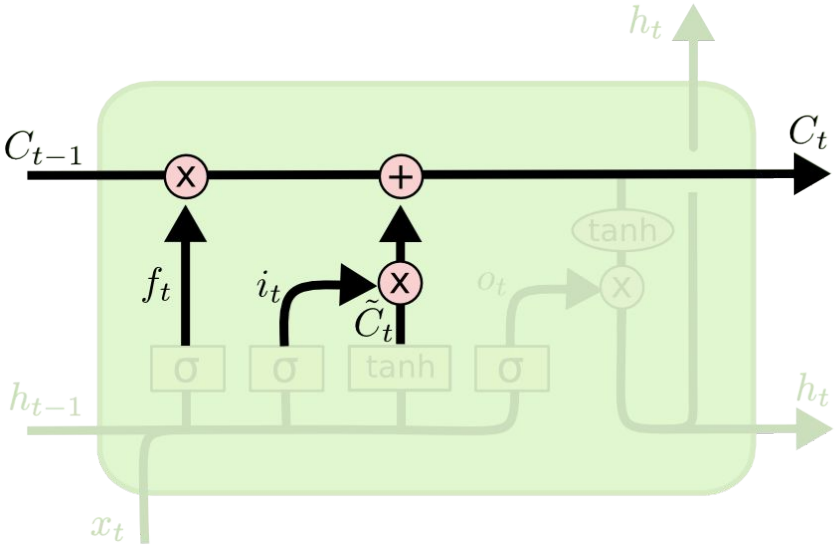
$$i_t = \sigma\left(W_i x_t + U_i h_{t-1}\right)$$
$$\tilde{C}_t = \tanh\left(W_C x_t + U_C h_{t-1}\right)$$

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

Long short term memory

Update Cell state

Cập nhật giá trị của Cell (C) dựa vào giá trị cũ của C (C_{t-1}) và thông tin mới sẽ thêm vào



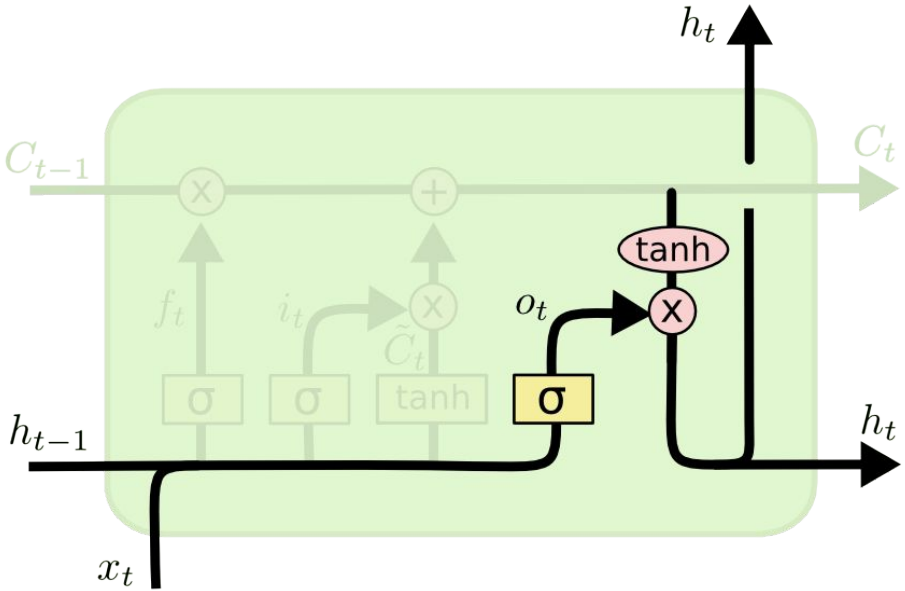
$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t$$

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

Long short term memory

Output gate

Output gate: quyết định giá trị của h dựa vào giá trị của C.

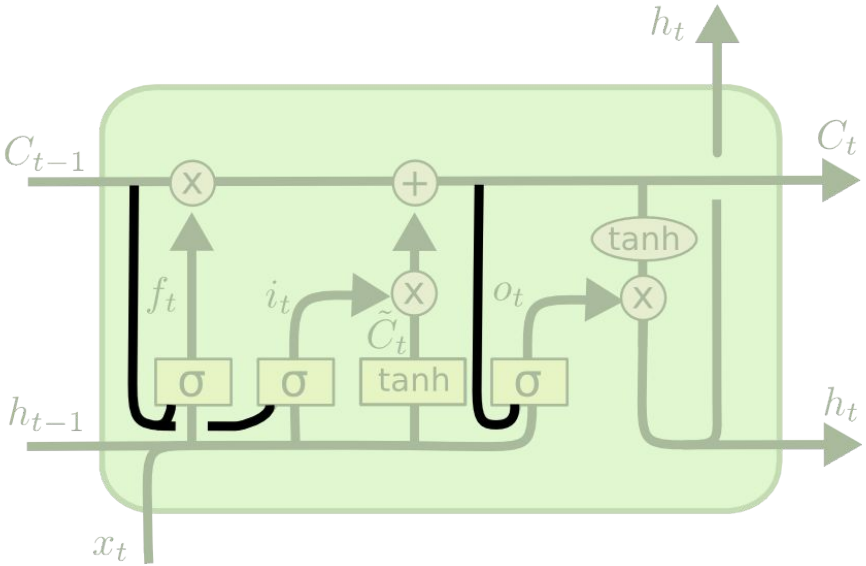


$$o_t = \sigma\left(W_o x_t + U_o h_{t-1}\right)$$
$$h_t = o_t \circ \tanh\left(C_t\right)$$

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

Long short term memory

Một biến thể khác của LSTM là việc sử dụng giá trị của C để tính giá trị của các gates.



$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + Z_f C_{t-1}\right)$$

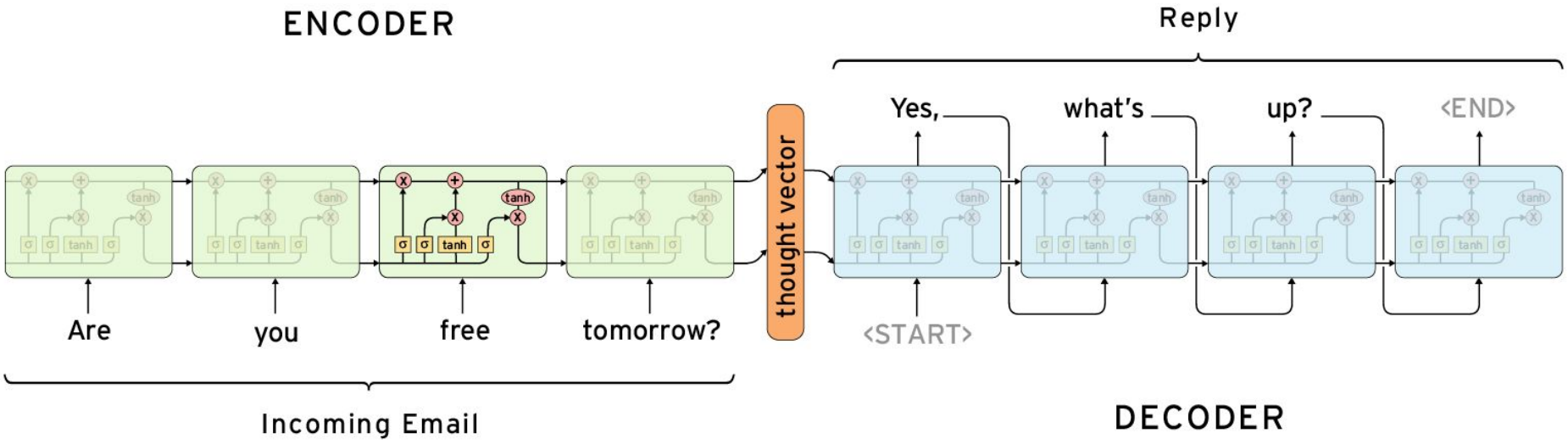
$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + Z_i C_{t-1}\right)$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + Z_o C_t\right)$$

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

Long short term memory

Seq2Seq



Nguồn: WILDML

Long short term memory

Bài tập

Hãy tính giá trị h_1 với đầu vào:

$x_1 = [2 \quad 3 \quad -1]$

Các params của mạng:

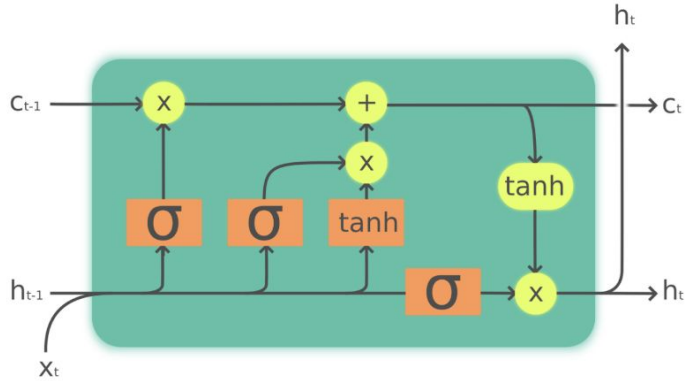
$$W_f = \begin{bmatrix} 1 & -2 & 5 \\ 5 & 5 & 0 \end{bmatrix} \quad U_f = \begin{bmatrix} 1 & 5 \\ 2 & 1 \end{bmatrix}$$

$$W_i = \begin{bmatrix} 0 & 2 & 5 \\ 3 & -4 & 1 \end{bmatrix} \quad U_i = \begin{bmatrix} -1 & -5 \\ 2 & 1 \end{bmatrix}$$

$$W_o = \begin{bmatrix} 1 & 2 & 1 \\ 0 & -1 & 3 \end{bmatrix} \quad U_o = \begin{bmatrix} 2 & 4 \\ 0 & 3 \end{bmatrix}$$

$$W_c = \begin{bmatrix} -2 & 0 & 1 \\ 5 & 1 & 8 \end{bmatrix} \quad U_c = \begin{bmatrix} 1 & -1 \\ 0 & -2 \end{bmatrix}$$

$$b_i = b_o = b_f = b_c = [5 \quad 2]$$



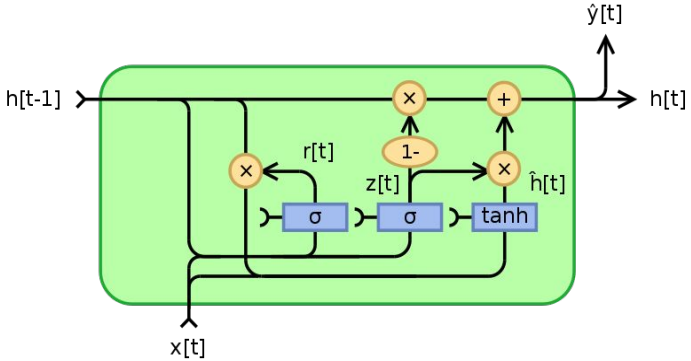
$$f_t = \sigma\left(W_f x_t + U_f h_{t-1}\right)$$
$$i_t = \sigma\left(W_i x_t + U_i h_{t-1}\right)$$
$$\widetilde{C}_t = \tanh\left(W_C x_t + U_C h_{t-1}\right)$$
$$C_t = f_t \circ C_{t-1} + i_t \circ \widetilde{C}_t$$
$$o_t = \sigma\left(W_o x_t + U_o h_{t-1}\right)$$
$$h_t = o_t \circ \tanh\left(C_t\right)$$

Outline

1. Long Short-Term Memory (LSTM)
2. Gated Recurrent Unit (GRU)
3. RNN hai chiều (Bidirectional RNN)
4. RNN nhiều tầng (Deep-stacked RNN)

Gated Recurrent Unit

- RNN-based cell
- Được giới thiệu lần đầu tiên bởi Kyunghyun Cho et al.
- Hiệu suất ngang bằng với LSTM (trên bài toán music modeling và speech signal modeling)
- Có ít bộ trọng số hơn
- Hiệu suất cao hơn đối với những dataset nhỏ



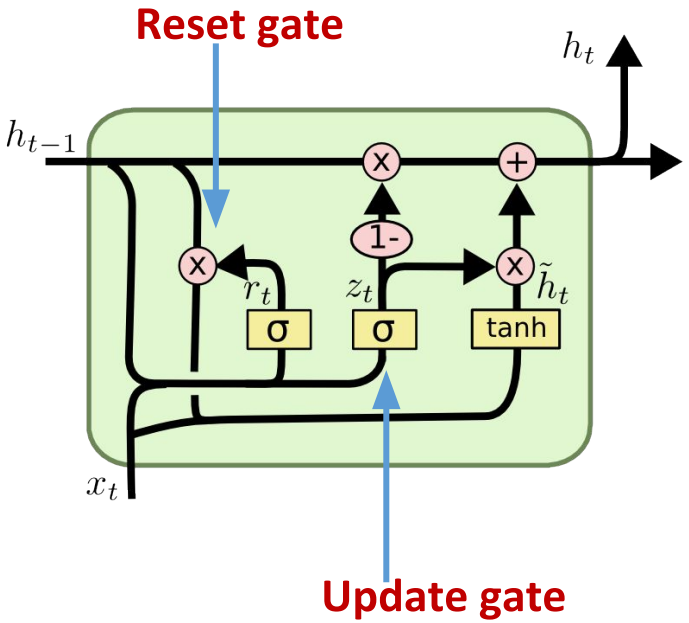
Gated Recurrent Unit

- Update gate: $z_t = \sigma\left(W_z x_t + U_z h_{t-1}\right)$
- Reset gate: $r_t = \sigma\left(W_r x_t + U_r h_{t-1}\right)$
- Nội dung memory mới:

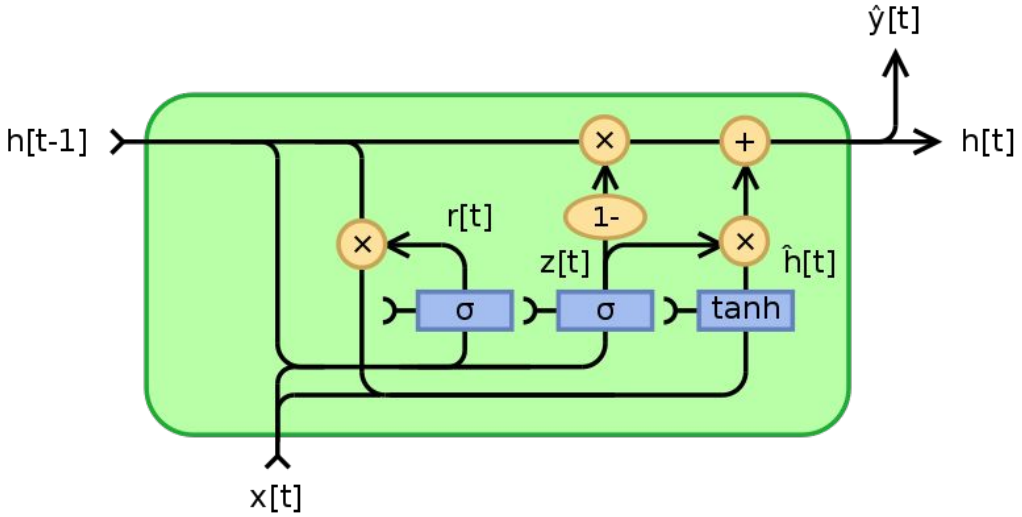
$$\widetilde{h}_t = \tanh\left(W_h x_t + r_t \circ U_h h_{t-1}\right)$$

- Giá trị h được tính kết hợp bởi giá trị h cũ và nội dung mới. Nếu reset gate bằng 0, thì việc tính giá trị $\widetilde{h}_t$ không quan tâm giá trị memory cũ mà chỉ quan tâm thông tin của từ mới.
- Giá trị h cuối cùng được cập nhật:

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

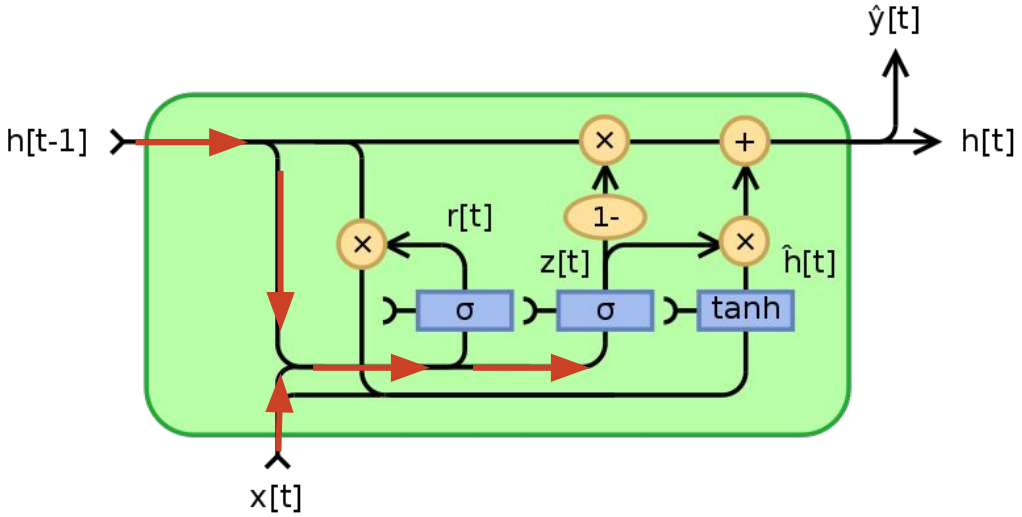


Gated Recurrent Unit



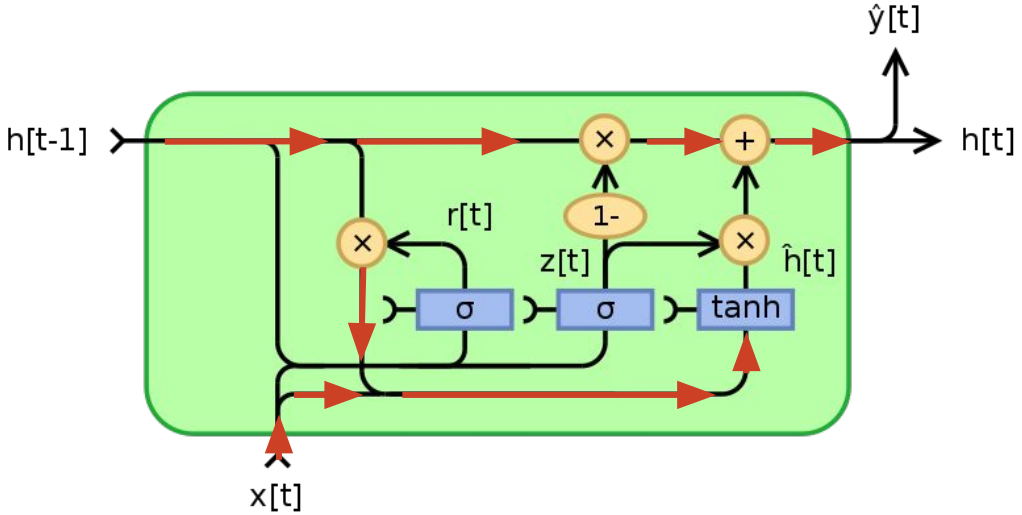
$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z)$$
$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r)$$
$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h)$$

Gated Recurrent Unit



$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z)$$
$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r)$$
$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h)$$

Gated Recurrent Unit



$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z)$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r)$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h)$$

Nội dung update mới $\tilde{h_t}$

Gated Recurrent Unit

Bài tập

Hãy tính giá trị $\hat{y}_1$ với đầu vào:

$x_1 = [1 \quad 0]$

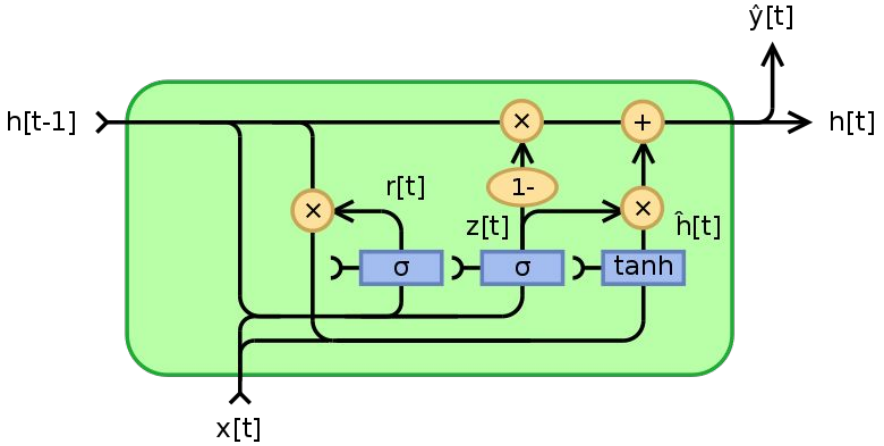
Các params của mạng:

$$W_z = \begin{bmatrix} 1 & 3 \\ 2 & 0 \end{bmatrix} \qquad U_z = \begin{bmatrix} 3 & 0 \\ -1 & 1 \end{bmatrix}$$

$$W_r = \begin{bmatrix} 1 & -1 \\ -6 & -1 \end{bmatrix} \qquad U_r = \begin{bmatrix} 1 & 1 \\ 4 & -3 \end{bmatrix}$$

$$W_h = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix} \qquad U_h = \begin{bmatrix} 2 & 3 \\ 0 & -2 \end{bmatrix}$$

$$b_z = b_r = b_h = [1 \quad 1]$$



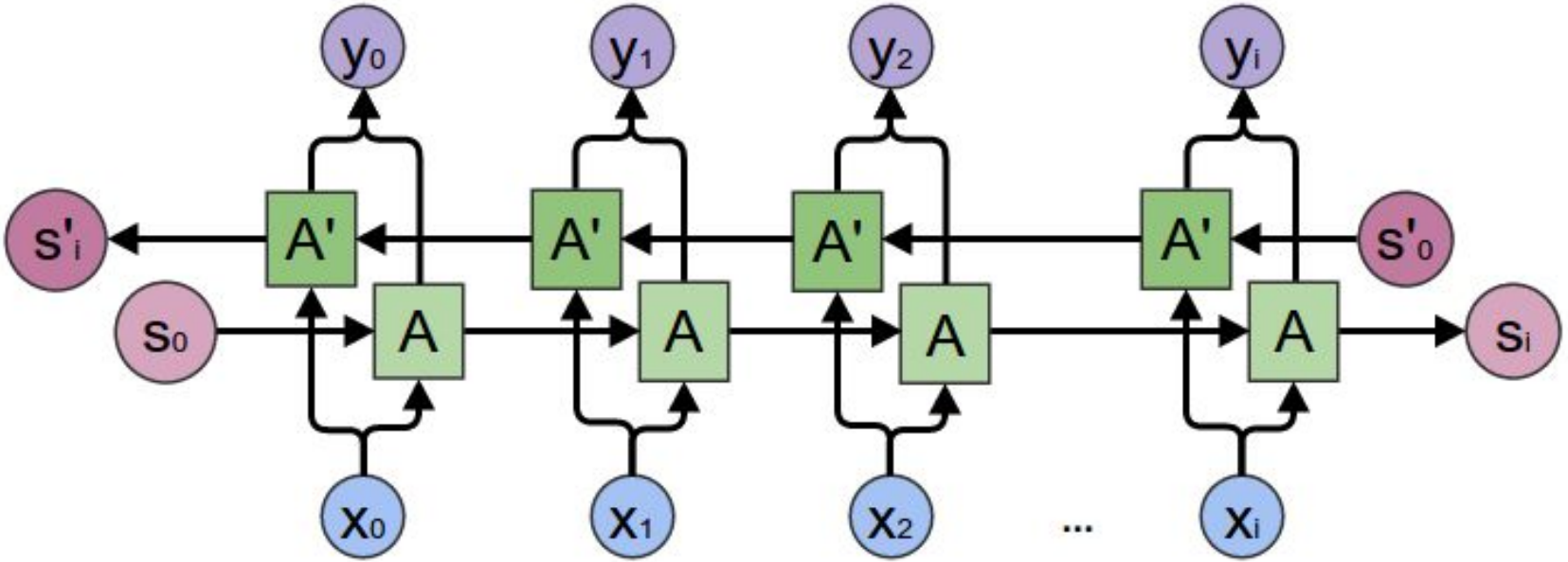
$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z)$$
$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r)$$
$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h)$$

Nguồn: WILDML

Outline

1. Long Short-Term Memory (LSTM)
2. Gated Recurrent Unit (GRU)
3. RNN hai chiều (Bidirectional RNN)
4. RNN nhiều tầng (Deep-stacked RNN)

Bidirectional Neural Network

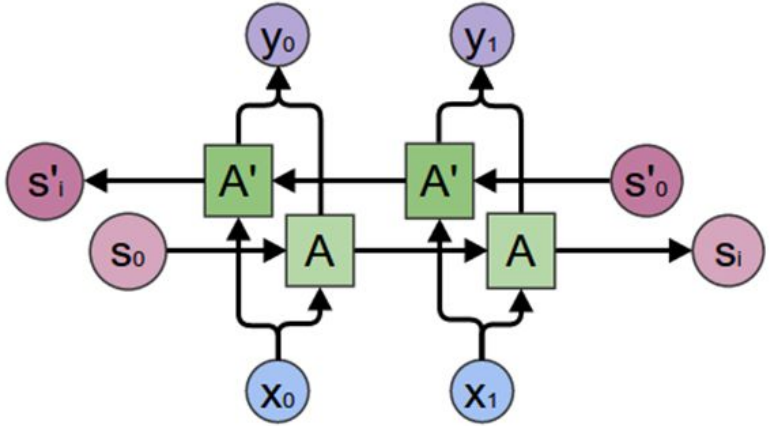


Có mấy bộ trọng số cần điều chỉnh?

Bidirectional Neural Network

Bài tập

Hãy sử dụng one-hot encoding để output của bidirectional RNN sau (bỏ qua tất cả activation function)



machine → [1 0]

learning → [0 1]

$$U = \begin{bmatrix} 3 & -1 \\ 0 & 1 \end{bmatrix}$$

$$W = \begin{bmatrix} 0 & 2 \\ 2 & 1 \end{bmatrix} \quad A'$$

$$V = [1 \quad 1]$$

$$U = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

$$W = \begin{bmatrix} 3 & -2 \\ 4 & 1 \end{bmatrix} \quad A$$

$$V = [-1 \quad 0]$$

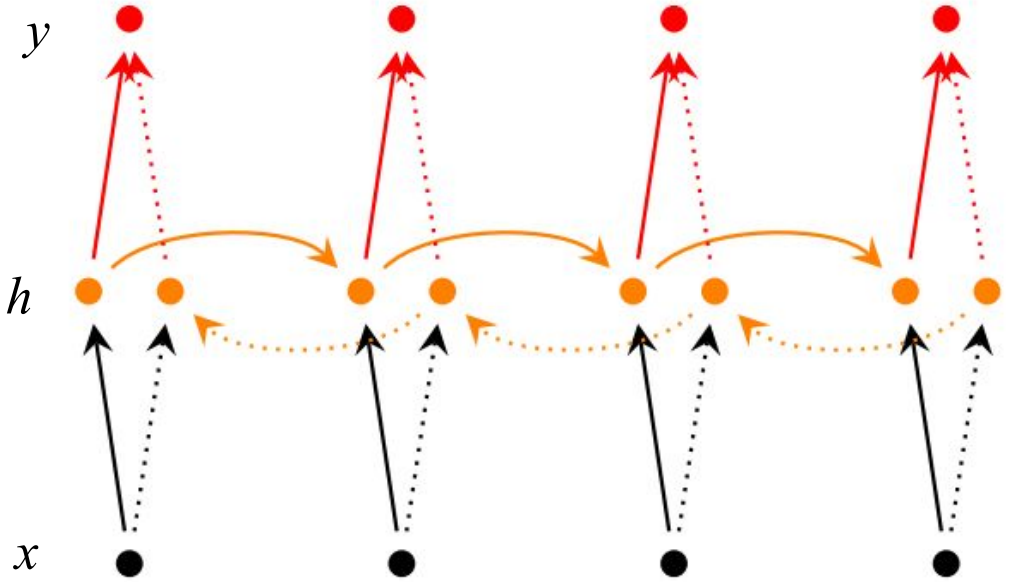
Bidirectional Neural Network

$h_t = \left[\overleftarrow{h}_t, \overrightarrow{h}_t\right]$ biểu diễn thông tin trước và sau của thời điểm đang xét.

$$\overrightarrow{h}_t = f\left(\overrightarrow{W}x_t + \overrightarrow{U}\,\overrightarrow{h}_{t-1}\right)$$

$$\overleftarrow{h}_t = f\left(\overleftarrow{W}x_t + \overleftarrow{U}\,\overleftarrow{h}_{t+1}\right)$$

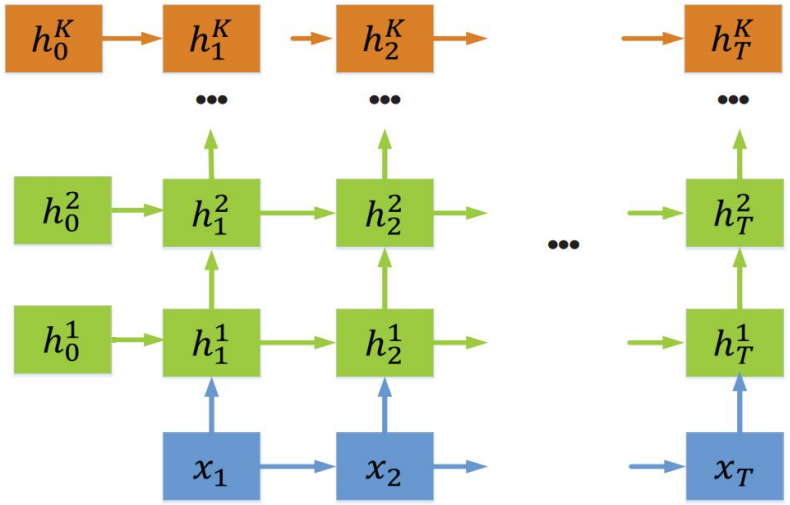
$$y_t = \mathrm{g}\left(\left[\overleftarrow{h}_t, \overrightarrow{h}_t\right]\right)$$



Outline

1. Long Short-Term Memory (LSTM)
2. Gated Recurrent Unit (GRU)
3. RNN hai chiều (Bidirectional RNN)
4. RNN nhiều tầng (Deep-stacked RNN)

Deep Recurrent Neural Network



$$h_t^1 = \sigma\left(W^1 x_t + U^1 h_{t-1}^1\right)$$

$$h_t^2 = \sigma\left(W^2 h_t^1 + U^2 h_{t-1}^2\right)$$

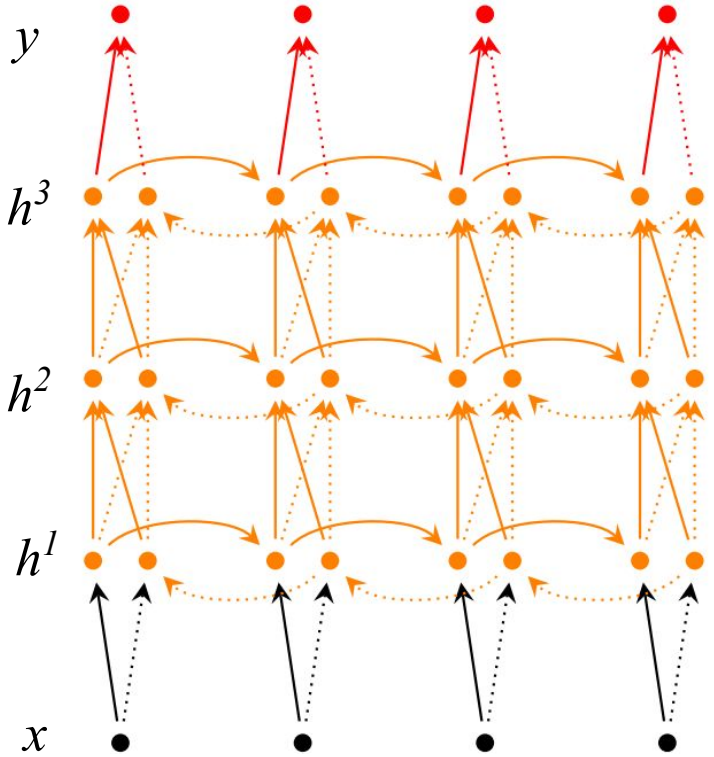
$$h_t^3 = \sigma\left(W^3 h_t^2 + U^3 h_{t-1}^3\right)$$

Deep Recurrent Neural Network

$$\overrightarrow{h}_t^i = f\left(\overrightarrow{W}^i x_t + \overrightarrow{U}^i \overrightarrow{h}_{t-1}^i\right)$$

$$\overleftarrow{h}_t^i = f\left(\overleftarrow{W}^i x_t + \overleftarrow{U}^i \overleftarrow{h}_{t+1}^i\right)$$

$$y_t = \mathrm{g}\left(\left[\overleftarrow{h}_t^i, \overrightarrow{h}_t^i\right]\right)$$



Tài liệu tham khảo

1. Understading LSTM Networks - Chris Olah http://colah.github.io/posts/2015-08-Understanding-LSTMs/
2. S. Hochreiter and J. Schmidhuber, Long Short-Term Memory, Neural Computation 1997
3. Kyunghuyn Cho et al., Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, 2014
4. http://www.cs.toronto.edu/~graves/handwriting.html
5. http://www.cs.toronto.edu/~ilya/rnn.html
6. Andrej Karpathy, The Unreasonable Effectiveness of Recurrent Neural Networks

Recurrent Neural Network

Challenge

Với công thức đã học

Hãy sử dụng tensorflow để xây dựng graph cho quá trình lan truyền thuận, sử dụng 1 LSTM cell (không dùng API dựng sẵn cho LSTM, e.g.

tf.nn.rnn_cell.LSTMCell)

(Có thể sử dụng lại input và kết quả tính được ở bài tập trên để kiểm tra tính chính xác của mô hình)