



Bài 13: Long Short-Term Memory và Gated Recurrent Unit

Đội ngũ giảng dạy: - Hoàng Quý Phát
- Hoàng Minh Phương
- Huỳnh Chí Kiên
- Vũ Thái Duy
- Nguyễn Vinh Tiệp

Cố vấn khóa học: Huyền Nguyễn, Tiệp Vũ, Thăng Lương



Recurrent Neural Network

Long Short-Term Memory (LSTM)

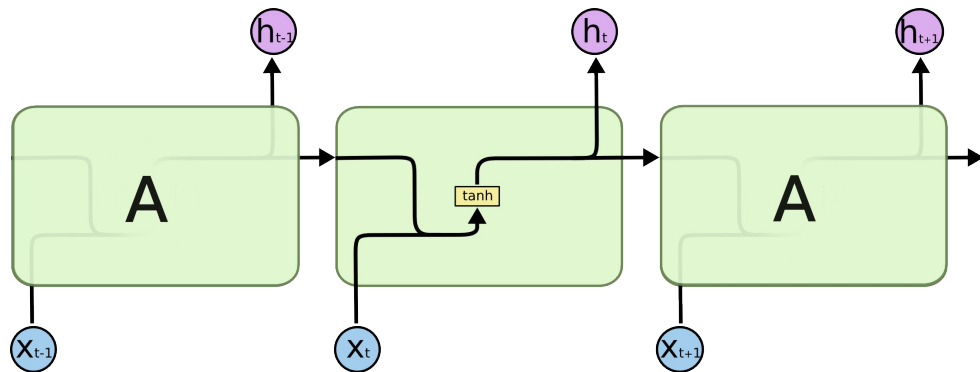
- Mục đích: giúp mạng RNN có thể tận dụng được những thông tin "long term".
- Cách tính hidden units (h) phức tạp hơn
- Ý tưởng chính:
 - Tạo thêm một bộ nhớ (memory) để nhớ được thông tin ở xa trước đó.
 - Cho phép thông tin được đổ xuống bộ nhớ và hidden units mạnh hoặc yếu khác nhau tùy thuộc vào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



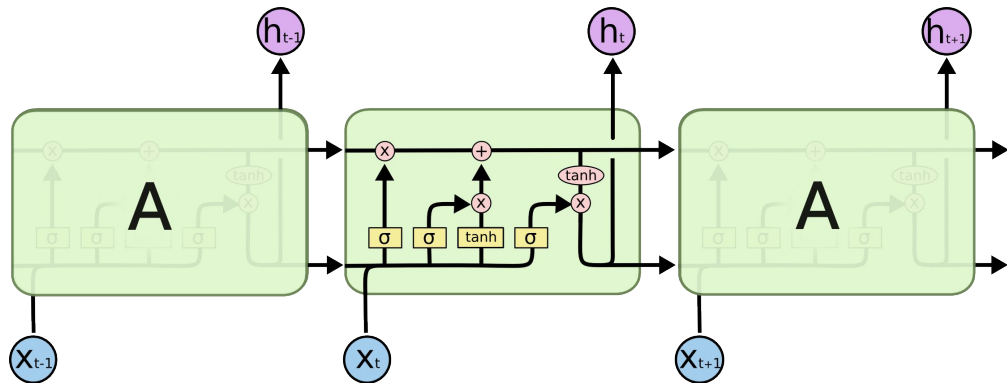
Recurrent Neural Network

Long Short-Term Memory (LSTM)

$$h_t = \sigma(Wx_t + Uh_{t-1})$$



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

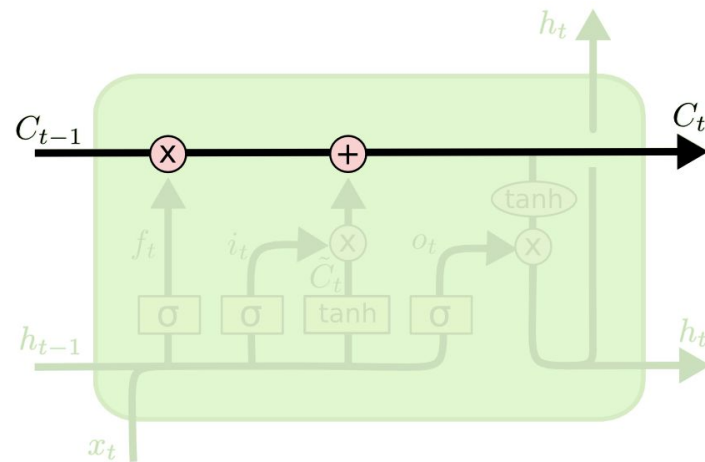




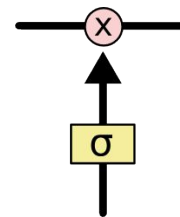
Recurrent Neural Network

Long Short-Term Memory (LSTM)

- 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)



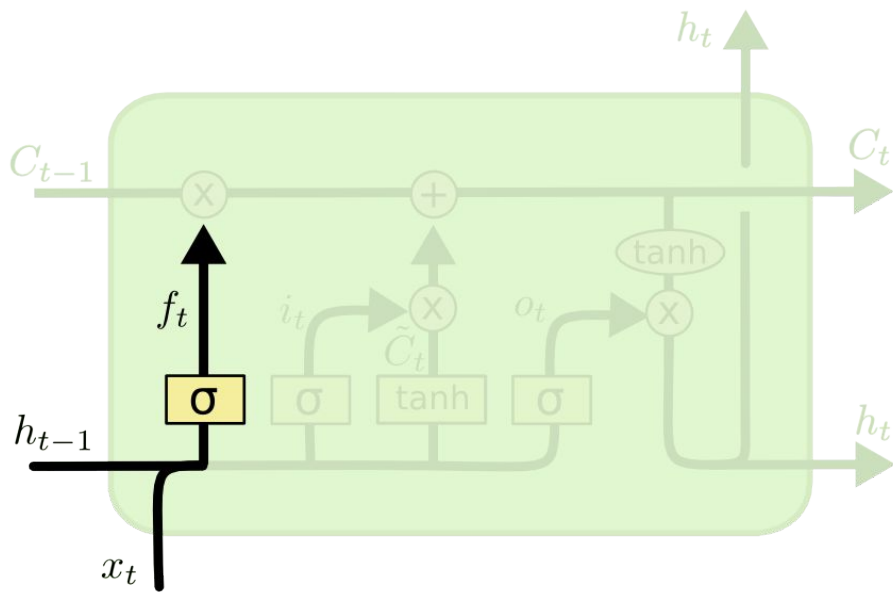
Gates



Recurrent Neural Network

Long Short-Term Memory (LSTM)

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



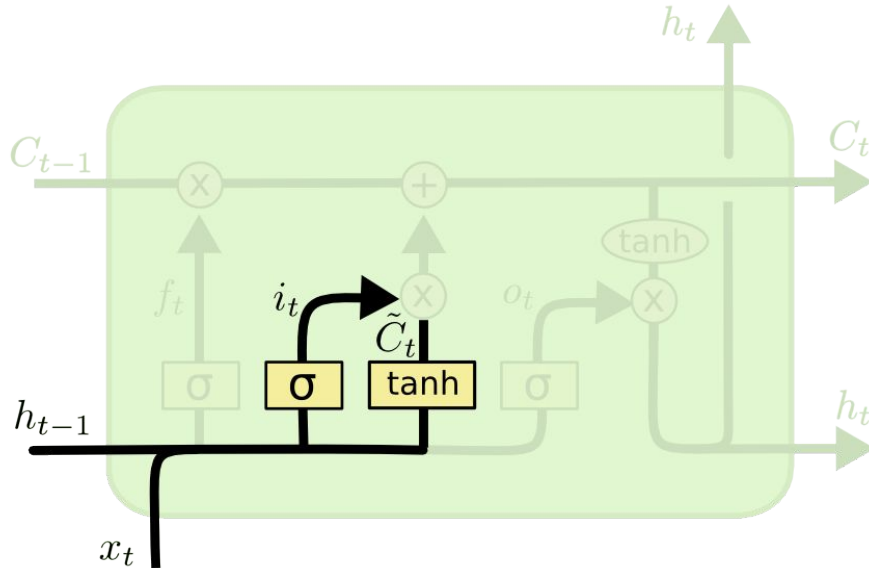
$$f_t = \sigma (W_f x_t + U_f h_{t-1})$$



Recurrent Neural Network

Long Short-Term Memory (LSTM)

- 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 \tilde{C}_t mới sẽ được thêm vào C .



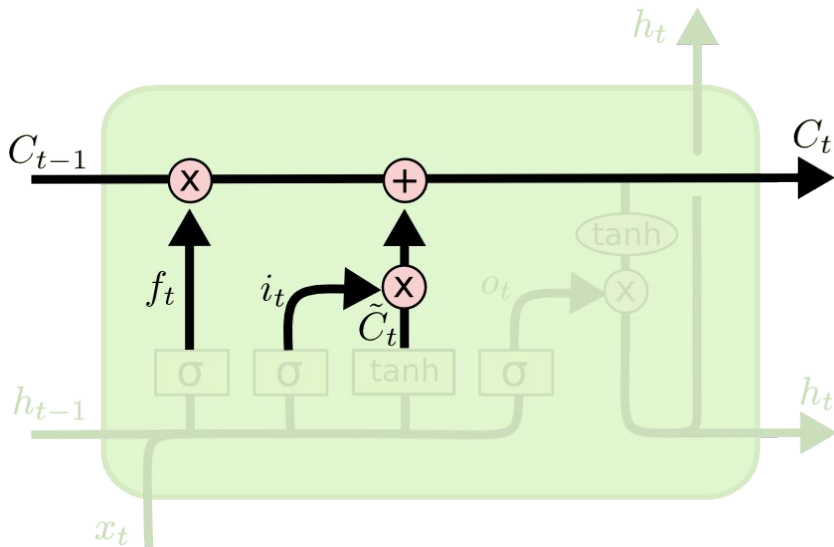
$$i_t = \sigma(W_i x_t + U_i h_{t-1})$$
$$\tilde{C}_t = \tanh(W_C x_t + U_C h_{t-1})$$



Recurrent Neural Network

Long Short-Term Memory (LSTM)

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



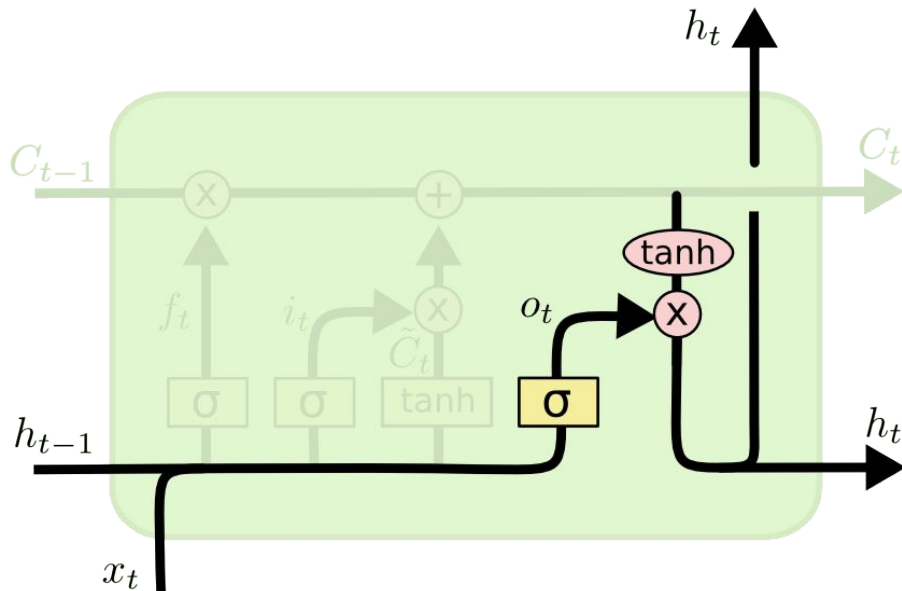
$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t$$



Recurrent Neural Network

Long Short-Term Memory (LSTM)

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



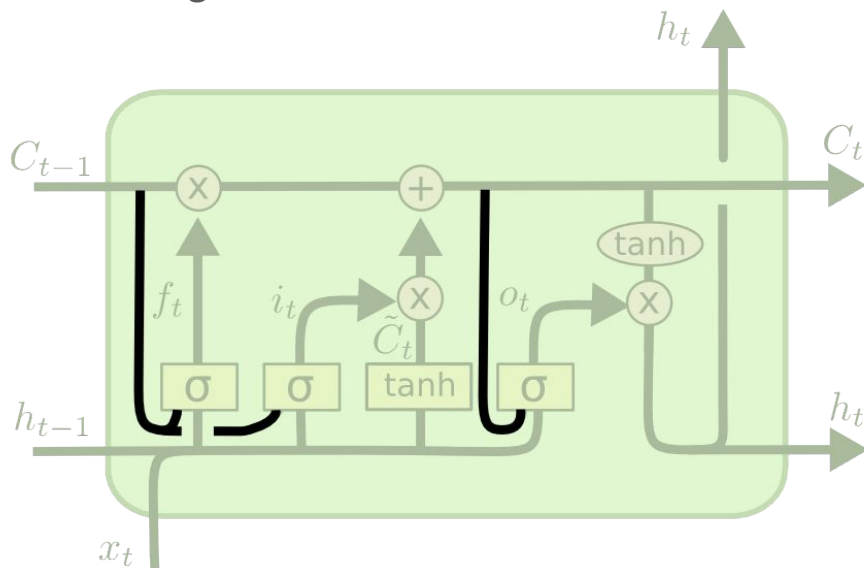
$$o_t = \sigma(W_o x_t + U_o h_{t-1})$$
$$h_t = o_t \circ \tanh(C_t)$$



Recurrent Neural Network

Long Short-Term Memory (LSTM)

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



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

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

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



Recurrent Neural Network

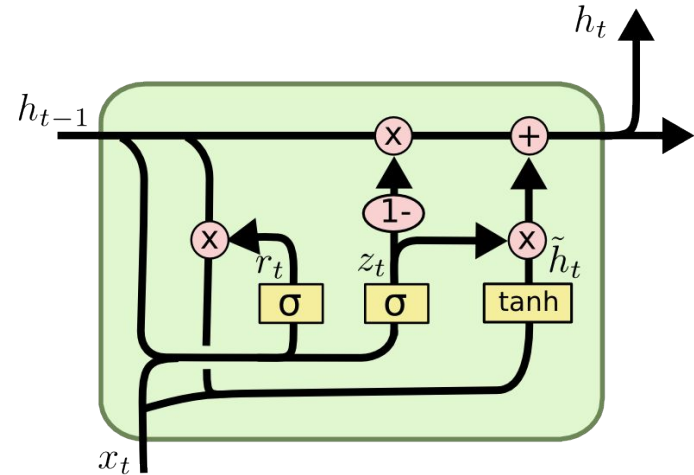
Gated Recurrent Unit (GRU)

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

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

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





Recurrent Neural Network

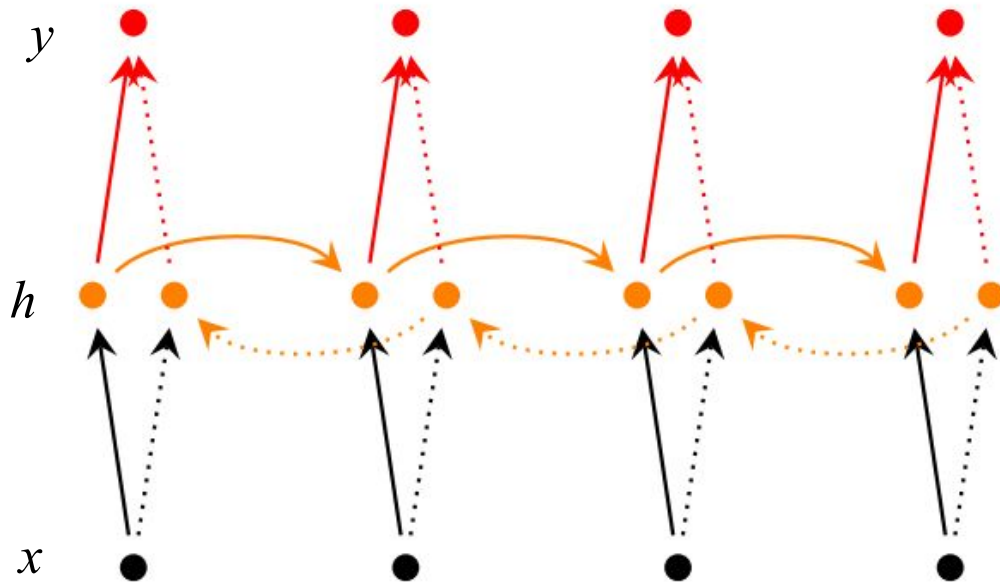
Bidirectional Recurrent Neural Network (BD-RNN)

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

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

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

$$y_t = \text{softmax}([\overleftarrow{h}_t, \overrightarrow{h}_t])$$





Recurrent Neural Network

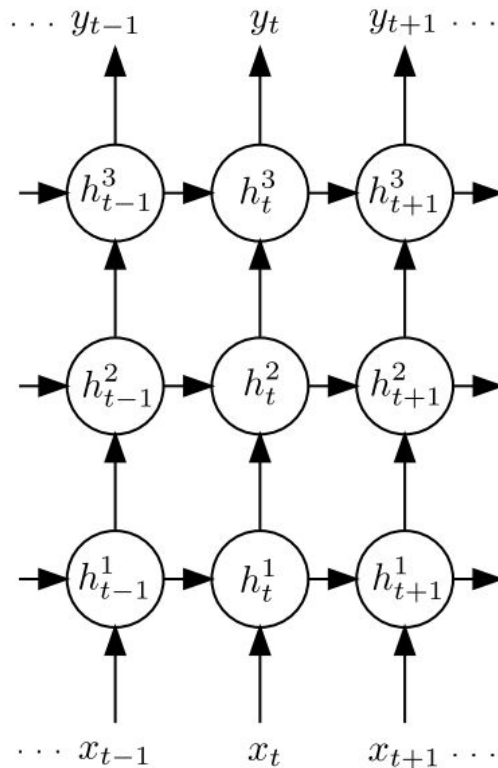
Deep Recurrent Neural Network

- Các RNN được chồng lên nhau.

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

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

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





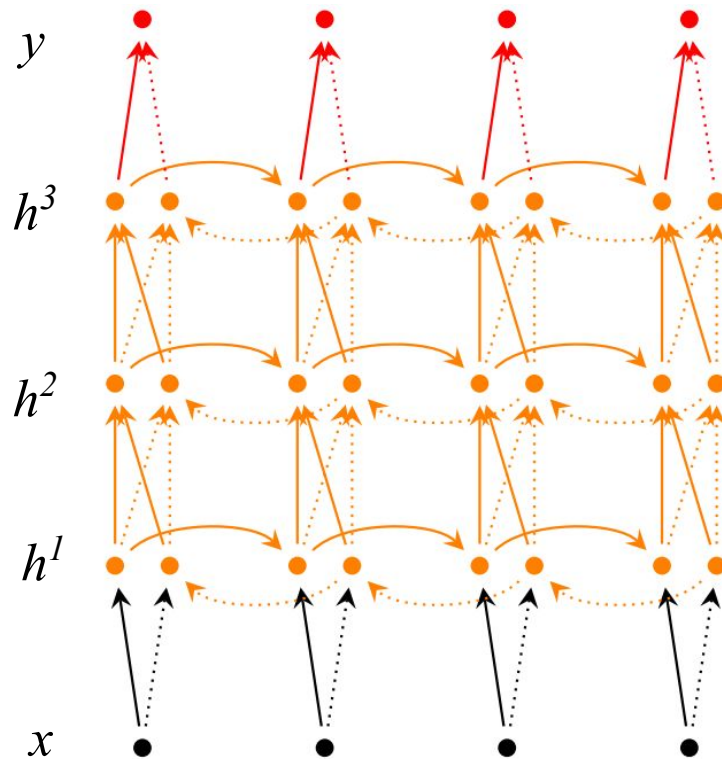
Recurrent Neural Network

Deep Bidirectional Recurrent Neural Network

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

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

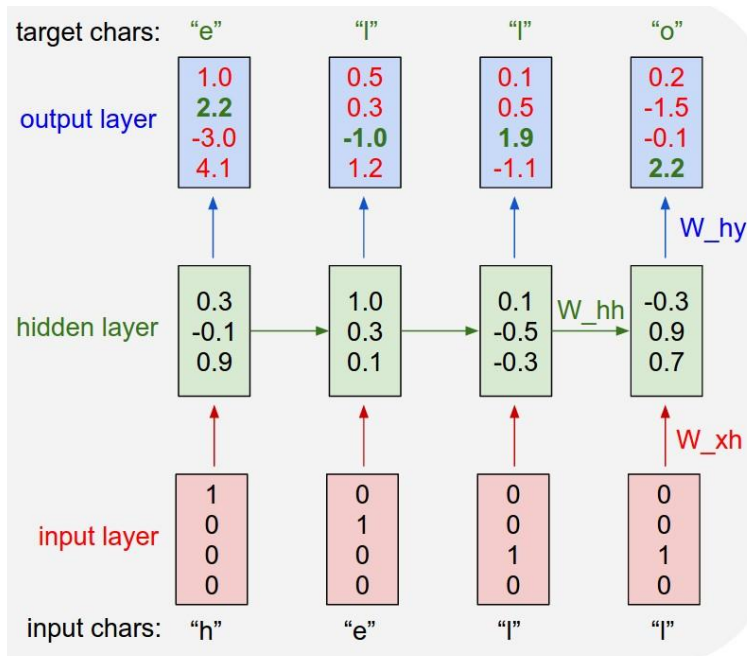
$$y_t = \text{softmax} \left(\left[\overleftarrow{h}_t^i, \vec{h}_t^i \right] \right)$$



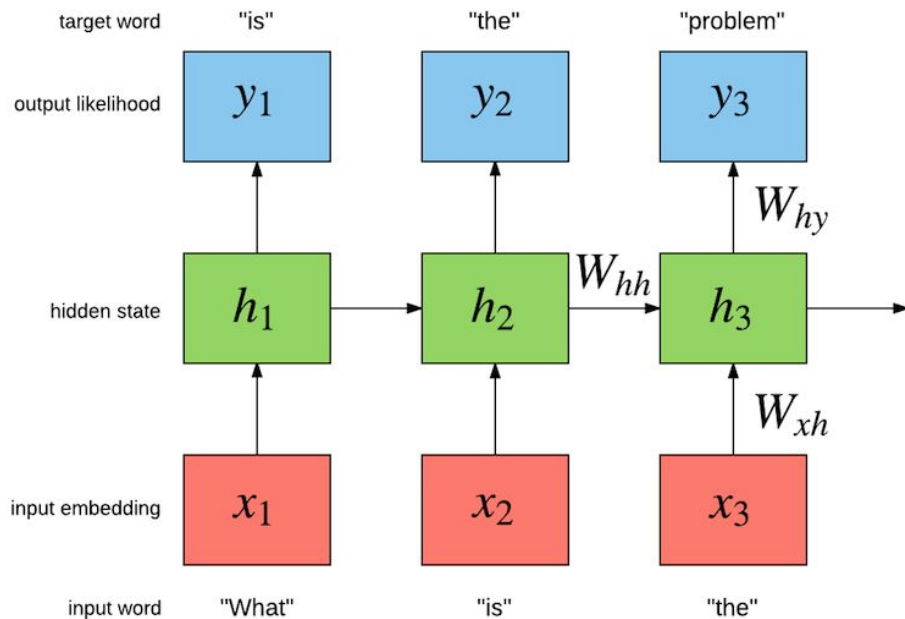


Recurrent Neural Network

Character-level RNN



Character-level RNN



Word-level RNN



Recurrent Neural Network

Ứng dụng

- Language Model
- Sequence Labeling (POS tagging and Named Entity Recognition)
- Machine Translation (Sequence to Sequence)
- Generating Sequences



Tài liệu tham khảo

1. Understanding 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](#)