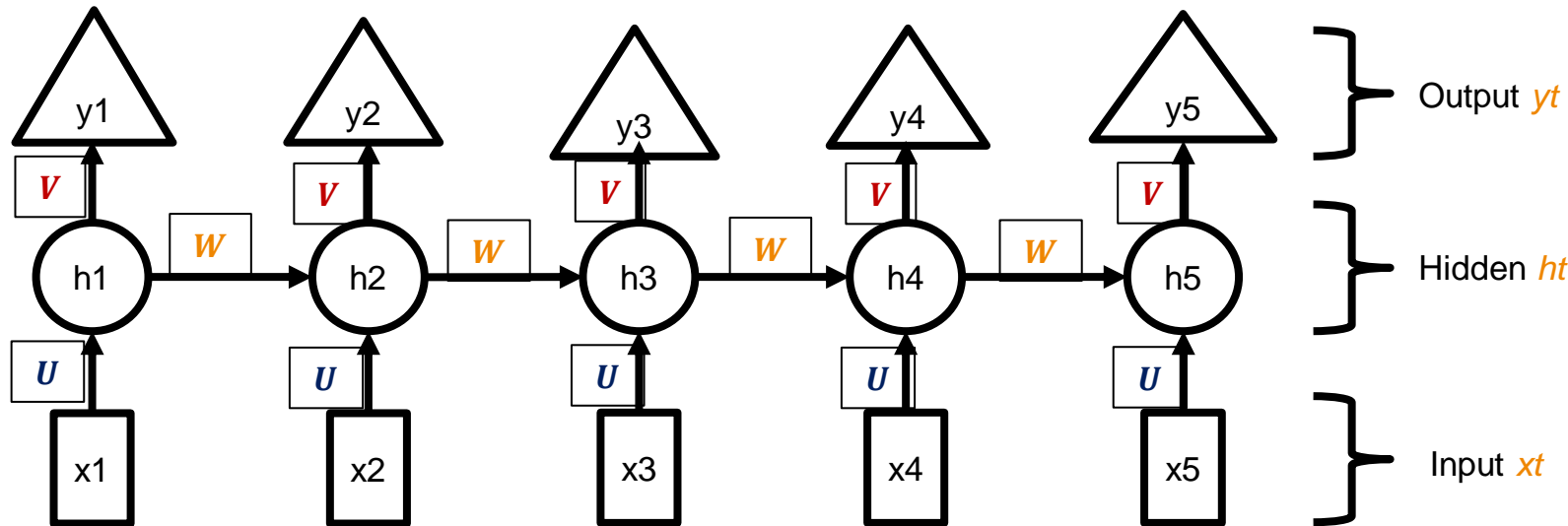


Variants of Recurrent Neural Network



VietAI teaching team



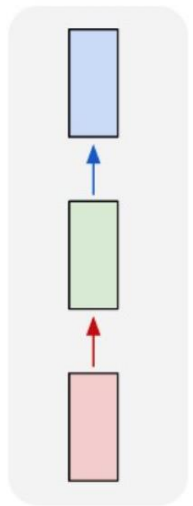
Bước 1: $h_t = \tanh(Ws_{t-1} + Ux_t + b_s)$

Bước 2: $y_t = \text{softmax}(Vh_t + b_y)$

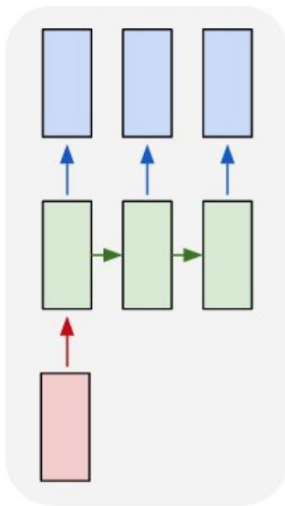
$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

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

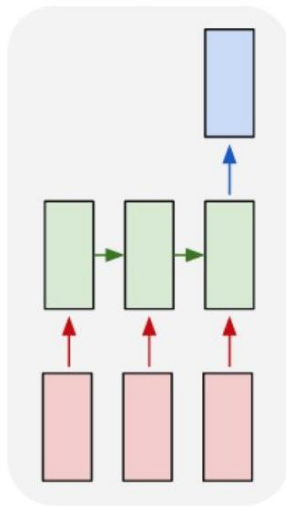
one to one



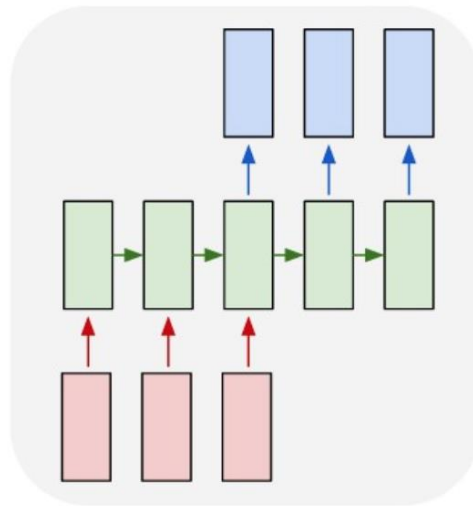
one to many



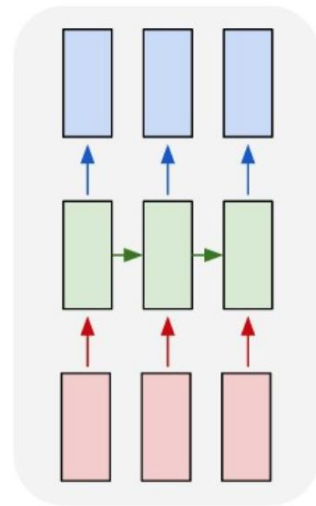
many to one

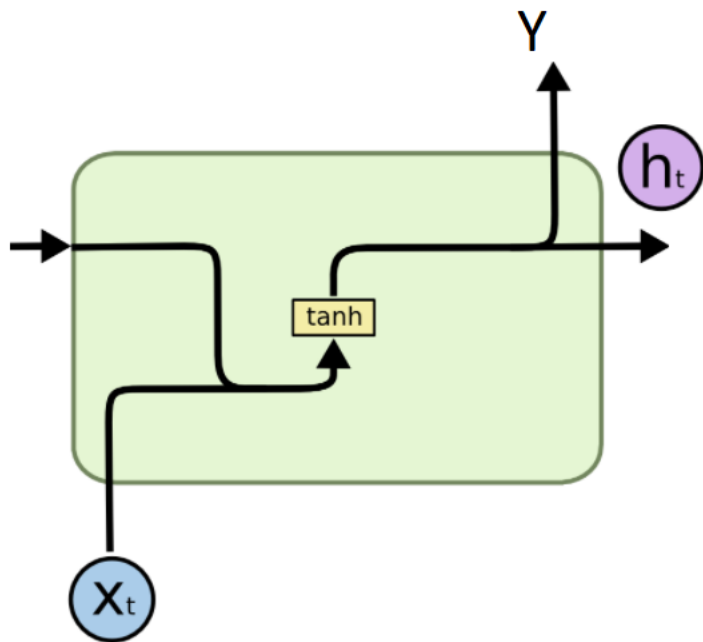


many to many



many to many





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

Bước 2: $y_t = \text{softmax}(Vh_t + b_y)$

Ứng dụng

1. Language model
2. Music generation

Nội dung

1. Long Short Term Memory
2. Gated Recurrent Unit
3. Bidirectional RNN
4. Deep-stacked RNN

Nội dung

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

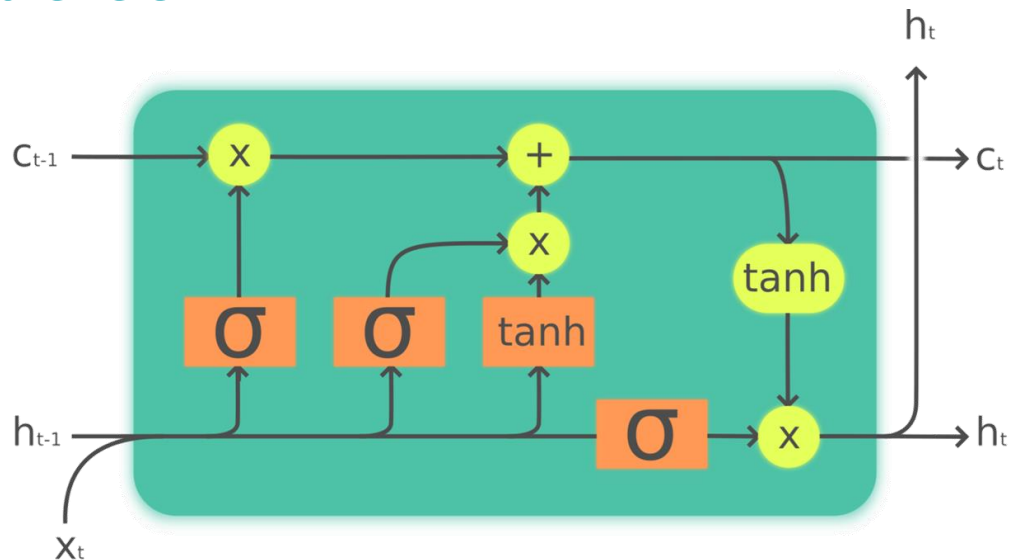
1 Giới thiệu LSTM

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

1 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
- LSTM được phát triển để giải quyết vấn đề vanishing gradient khi huấn luyện với mô hình vanilla RNN

1 Cấu trúc cell



Legend:

Layer



Pointwise op



Copy



1 Công thức LSTM

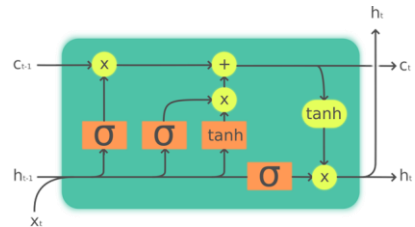
$$\begin{aligned}
 f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
 i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
 o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
 c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
 h_t &= o_t \circ \sigma_h(c_t)
 \end{aligned}$$

- $x_t \in \mathbb{R}^d$: Vector đầu vào
- $f_t \in \mathbb{R}^h$: Vector cổng forget
- $i_t \in \mathbb{R}^h$: Vector cổng input
- $o_t \in \mathbb{R}^h$: Vector cổng output
- $h_t \in \mathbb{R}^h$: Vector đầu ra của mỗi cell
- $c_t \in \mathbb{R}^h$: Vector trạng thái mỗi state

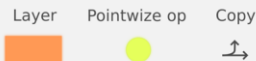
t : Giá trị đang xét tại thời điểm t

Khởi tạo: $c_0 = 0, h_0 = 0$

- σ_g : sigmoid function.
- σ_c : hyperbolic tangent



Legend:

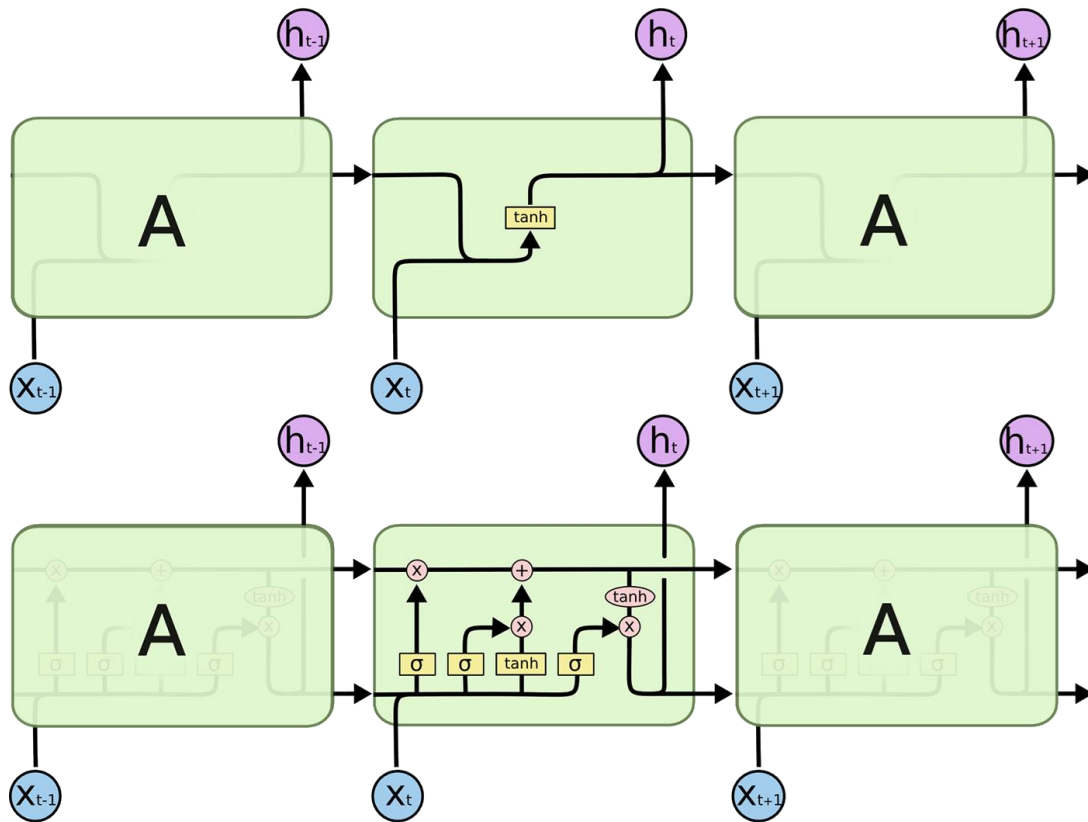


- $W \in \mathbb{R}^{h \times d}$, $U \in \mathbb{R}^{h \times h}$ and $b \in \mathbb{R}^h$:

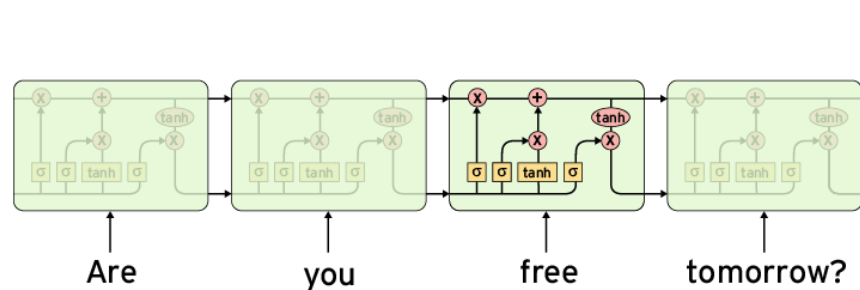
Ma trận weight và bias cần học

d và h là số chiều vector đầu vào và đầu ra

1 So sánh RNN và LSTM

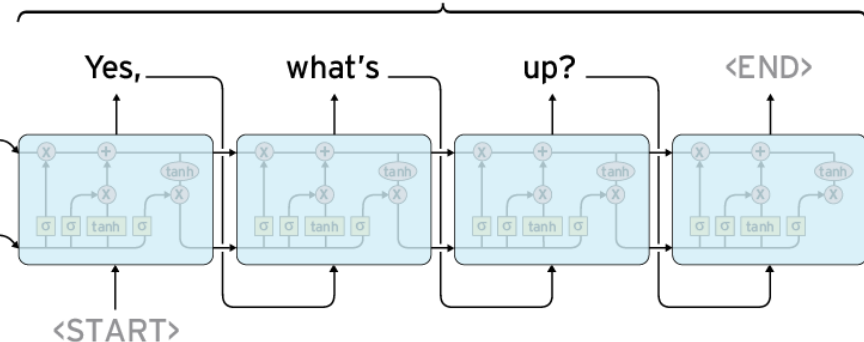


ENCODER



Incoming Email

Reply



DECODER

1 Bài tập ví dụ - LSTM

Bài tập 2:

Hãy tính giá trị h_1 với đầu vào:

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

Các params của mạng:

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

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

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

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

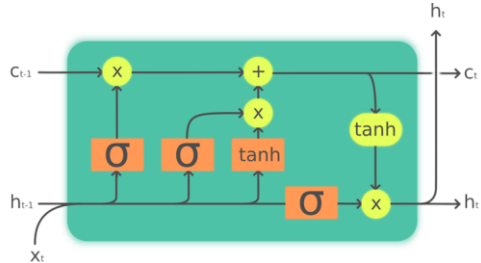
$$U_f = \begin{bmatrix} 1 & 5 & 5 \\ 2 & 1 & 1 \end{bmatrix}$$

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

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

$$U_c = \begin{bmatrix} 1 & -1 & 1 \\ 0 & -2 & 6 \end{bmatrix}$$

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



Legend:

Layer

Pointwise op

Copy



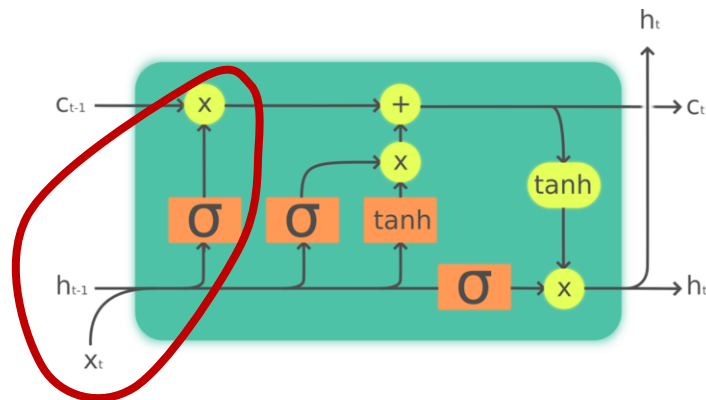
1

Bài tập ví dụ - LSTM

Giải:

1 Ý nghĩa công thức

Bước 1: Tính giá trị vector tại forget gate



Legend:

Layer



Pointwise op



Copy



$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \quad \leftarrow$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

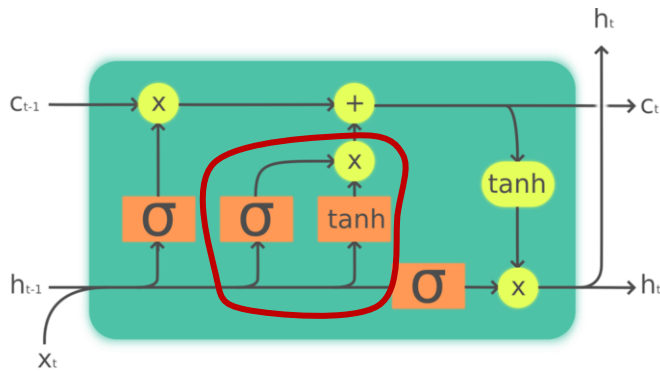
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$

$$h_t = o_t \circ \sigma_h(c_t)$$

1 Ý nghĩa công thức

Bước 2: Tính giá trị vector tại input gate



Legend:

Layer



Pointwise op



Copy



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

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \leftarrow$$

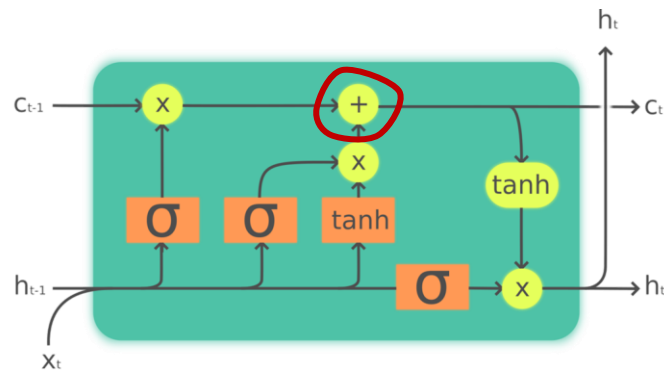
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \leftarrow$$

$$h_t = o_t \circ \sigma_h(c_t)$$

1 Ý nghĩa công thức

Bước 3: Cập nhật c



Legend:

Layer



Pointwise op



Copy



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

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

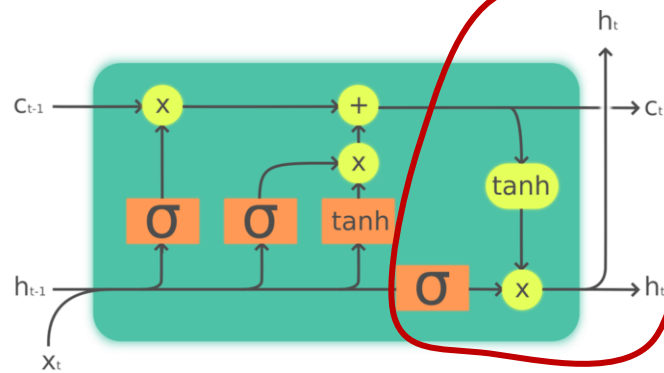
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \leftarrow$$

$$h_t = o_t \circ \sigma_h(c_t)$$

1

Ý nghĩa công thức

Bước 4: Tính giá trị output



Legend:

Layer



Pointwise op



Copy



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

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \leftarrow$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$

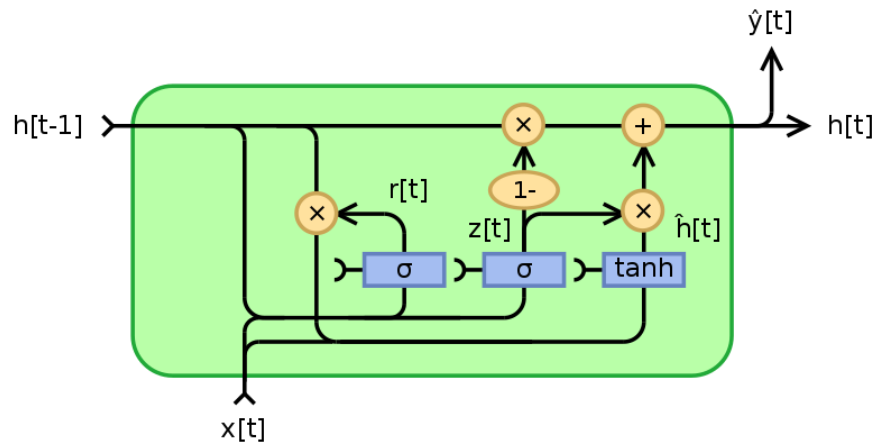
$$h_t = o_t \odot \sigma_h(c_t) \leftarrow$$

Nội dung

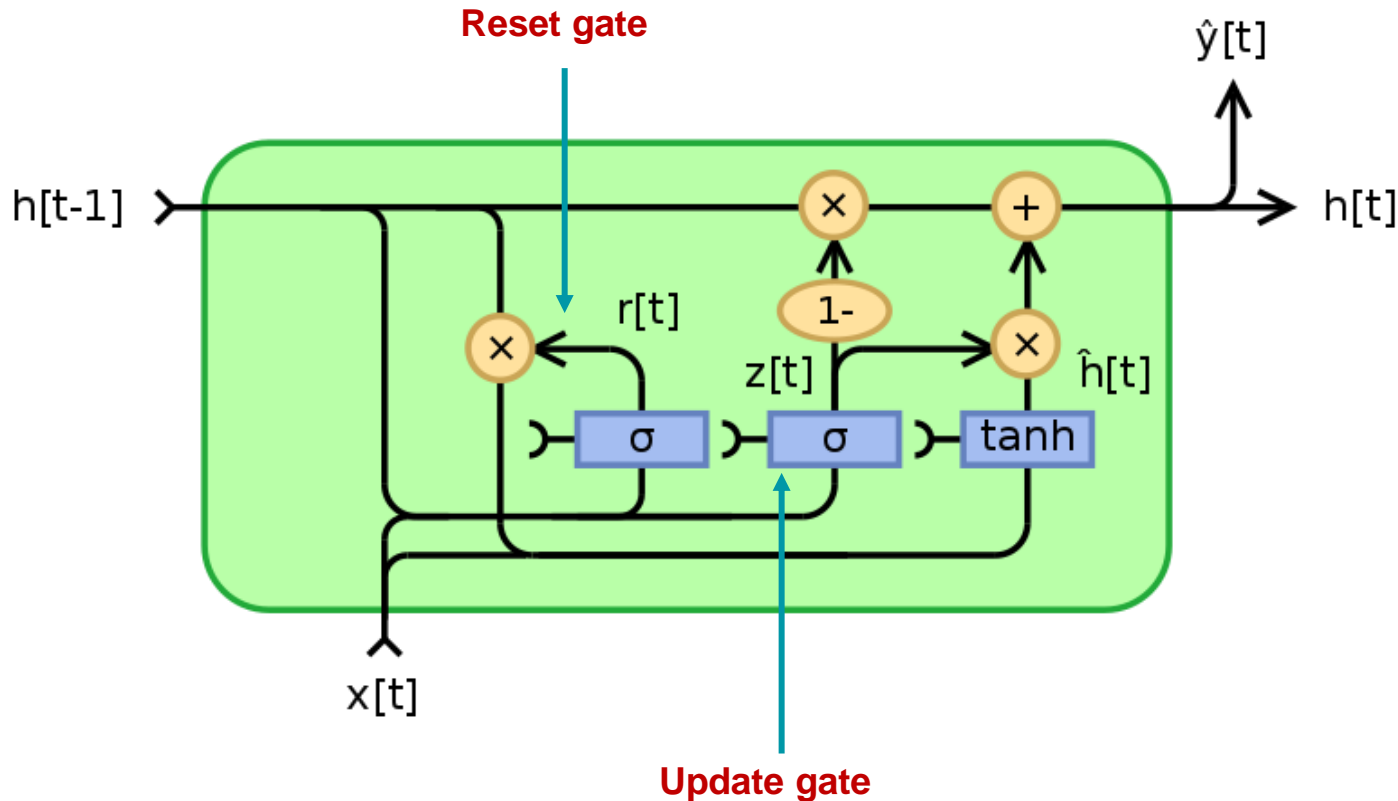
1. Long Short Term Memory
2. **Gated Recurrent Unit (GRU)**
3. Bidirectional RNN
4. Deep-stacked RNN

2 Gated Recurrent Unit

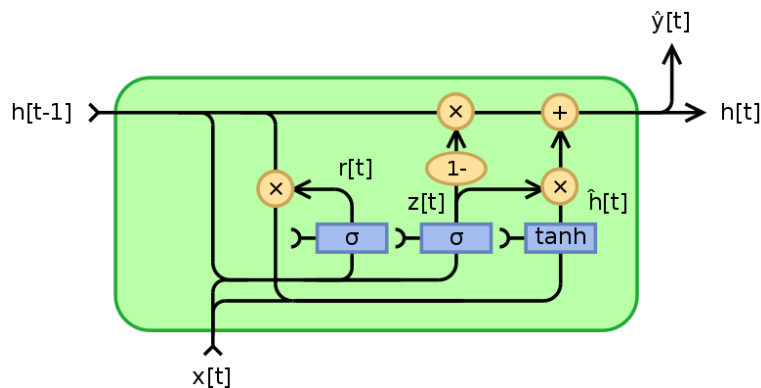
- RNN-based model
- Được giới thiệu lần đầu tiên bởi Kyunghyun Cho et al.
- Hiệu suất ngang bằng với LSTM (trên bài toán music modeling và speech signal modeling)
- Có ít bộ trọng số hơn
- Hiệu suất cao hơn đối với những dataset nhỏ



2 Gated Recurrent Unit



2 Công thức GRU

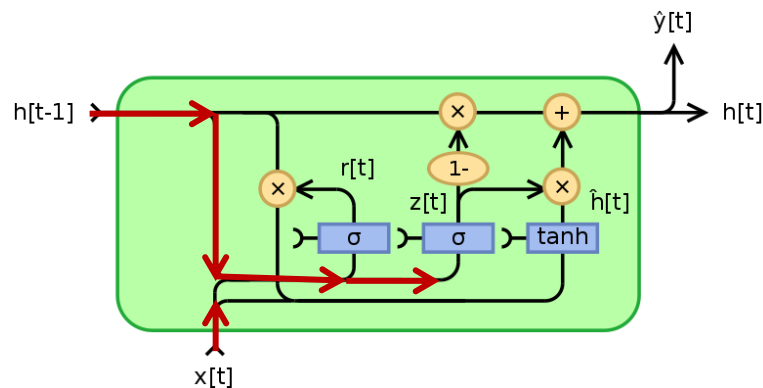


$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z)$$

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

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

2 Công thức GRU

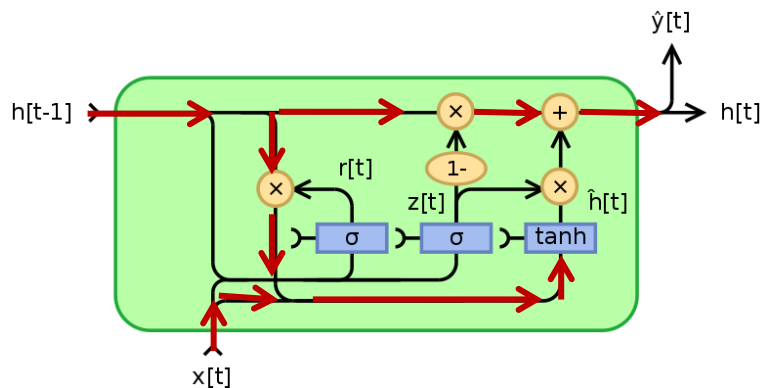


$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z)$$

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

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

2 Công thức GRU



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

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

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

Thông tin update mới: \tilde{h}_t

Bài tập 3:

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

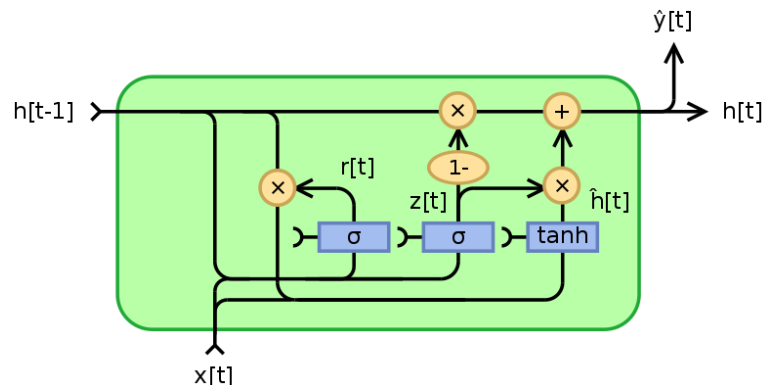
$$x_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Các params của mạng:

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

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

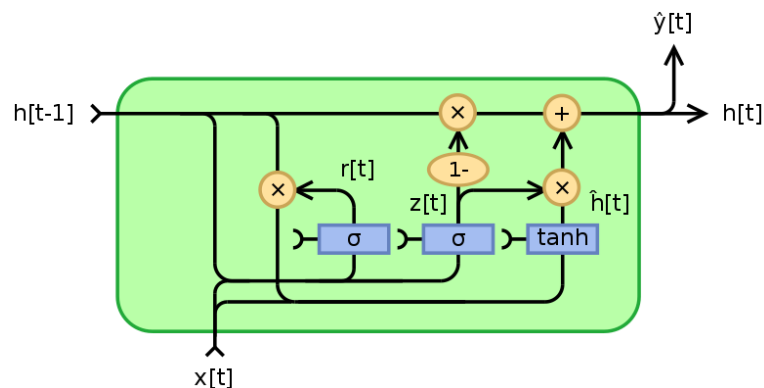
$$W_h = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix} \quad U_h = \begin{bmatrix} 2 & 3 \\ 0 & -2 \end{bmatrix} \quad b_z = b_r = b_h = \begin{bmatrix} 1 & 1 \end{bmatrix}$$



2

Bài tập ví dụ - GRU

Giải:



Hint

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

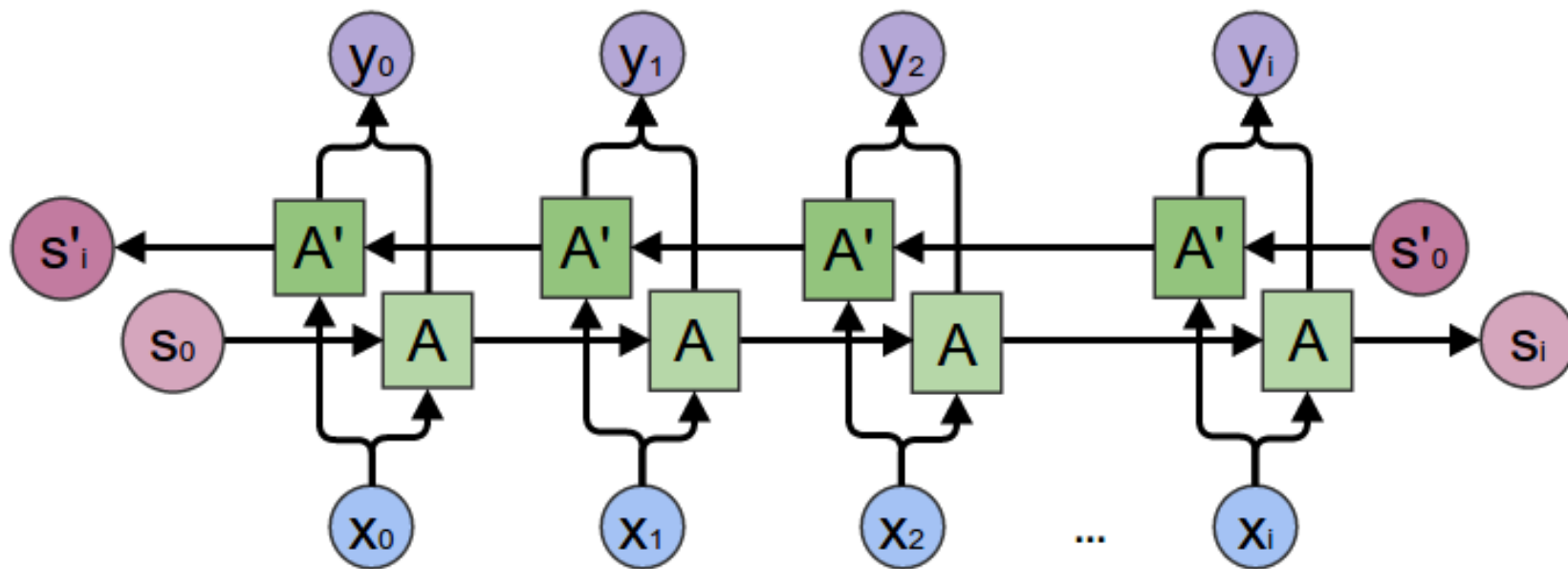
$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r)$$

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

Nội dung

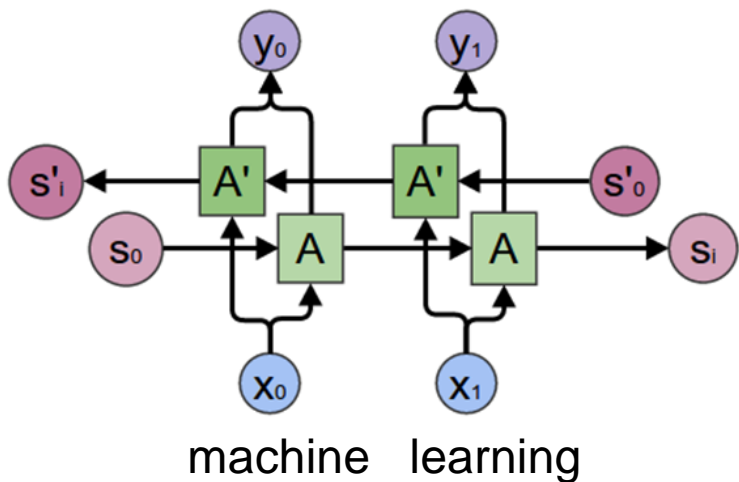
1. Long Short Term Memory
2. Gated Recurrent Unit (GRU)
3. **Bidirectional RNN**
4. Deep-stacked RNN

3 Bidirectional RNN



3 Thách thức

Bài tập 4: Hãy sử dụng one-hot encoding để output của bidirectional RNN sau (bỏ qua tất cả activation function)



machine $\rightarrow [1 \ 0]$

learning $\rightarrow [0 \ 1]$

$$U_{A'} = \begin{bmatrix} 3 & -1 \\ 0 & 1 \end{bmatrix}$$

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

$$V_{A'} = [1 \ 1]$$

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

$$W_A = \begin{bmatrix} 3 & -2 \\ 4 & 1 \end{bmatrix}$$

$$V_A = [-1 \ 0]$$

3

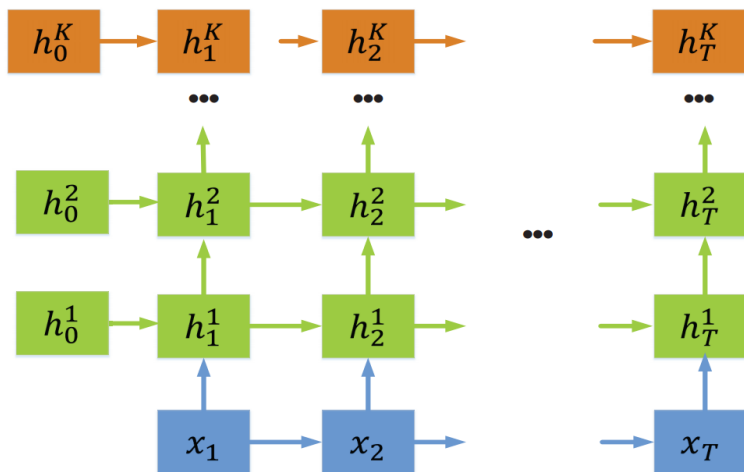
Thách thức

Giải:

Nội dung

1. Long Short Term Memory
2. Gated Recurrent Unit (GRU)
3. Bidirectional RNN
4. **Deep-stacked RNN**

4 Deep-stacked RNN

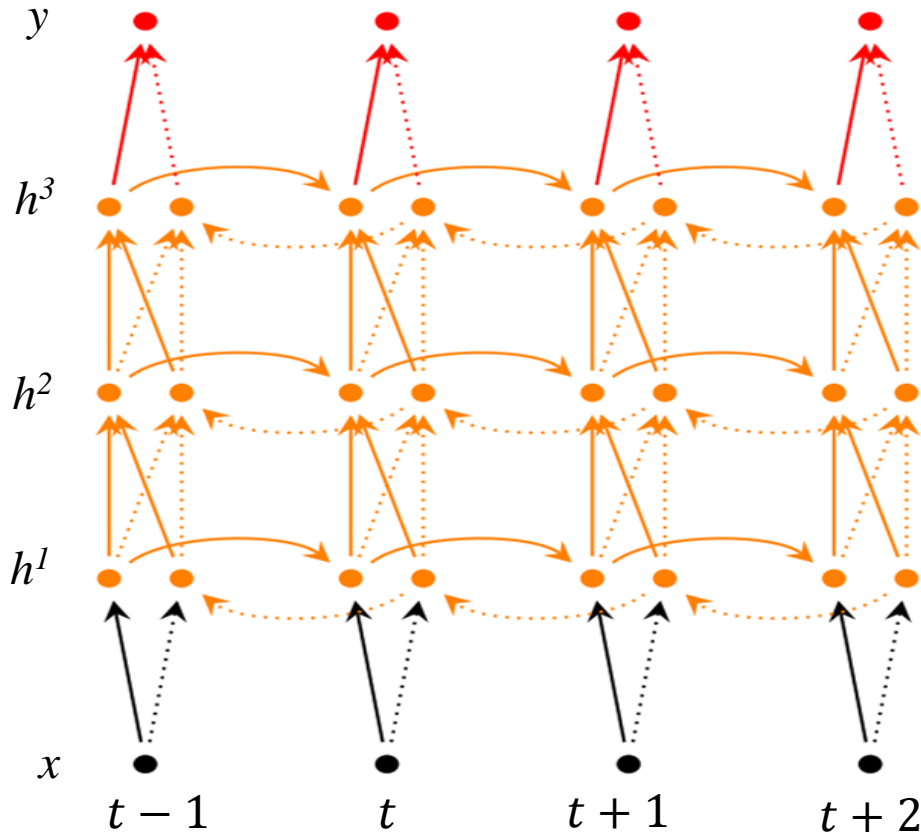


$$h^1_t = \sigma(W^1 x_t + U^1 h^1_{t-1})$$

$$h^2_t = \sigma(W^2 h^1_t + U^2 h^2_{t-1})$$

$$h^3_t = \sigma(W^3 h^2_t + U^3 h^3_{t-1})$$

4 Deep-stacked RNN



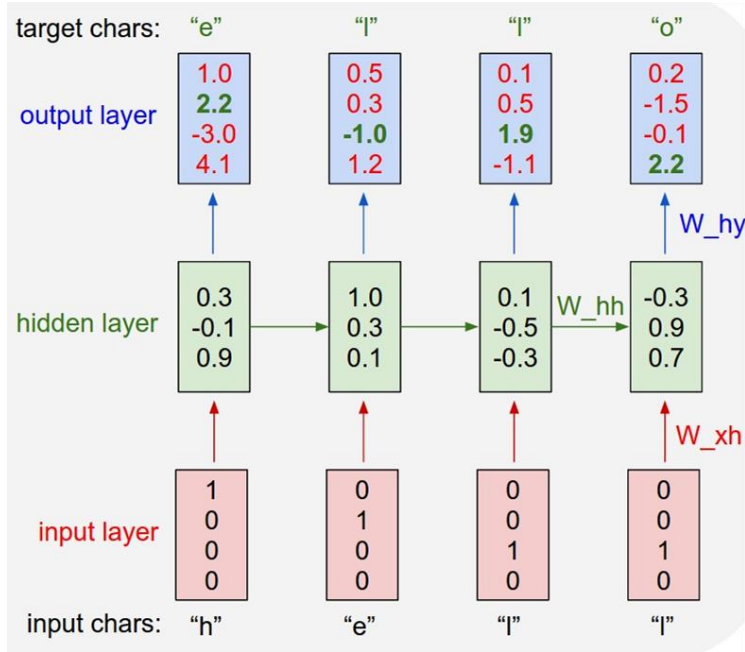
$$\vec{h}_t^i = f \left(\vec{W}^i x_t + \vec{U}^i \vec{h}_{t-1}^i \right)$$

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

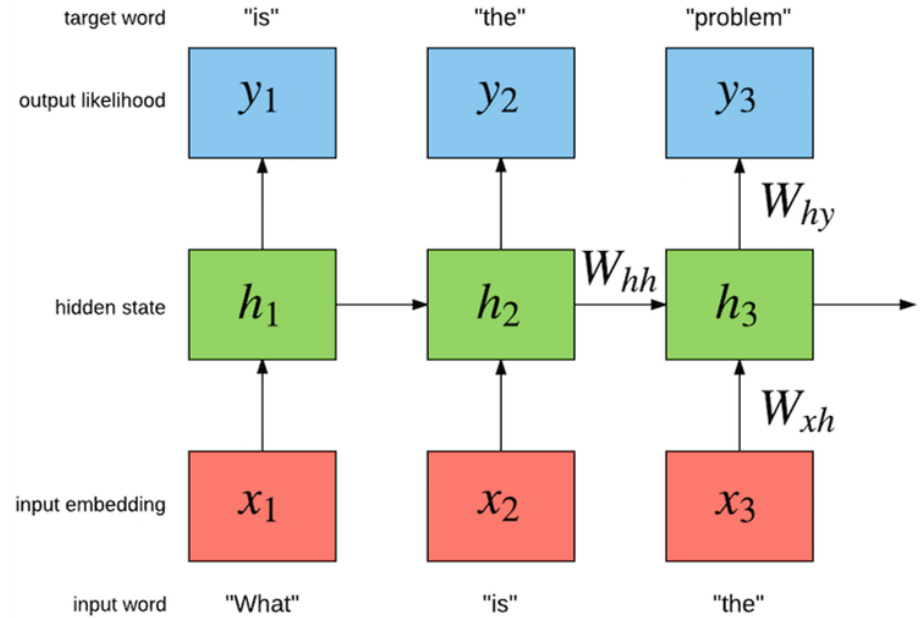
$$y_t = \text{softmax} \left(\left[\overleftarrow{h}_t^i, \vec{h}_t^i \right] \right)$$

4

Deep-stacked RNN



Character-level RNN



Word-level RNN