



温州大學  
WENZHOU UNIVERSITY

# 深度学习-序列模型

黄海广 副教授

2023年05月

- 01** 序列模型概述
- 02** 循环神经网络(RNN)
- 03** 长短期记忆(LSTM)
- 04** 双向循环神经网络
- 05** 深层循环神经网络

# 1.序列模型概述

3

## 01 序列模型概述

**02** 循环神经网络(RNN)

**03** 长短期记忆(LSTM)

**04** 双向循环神经网络

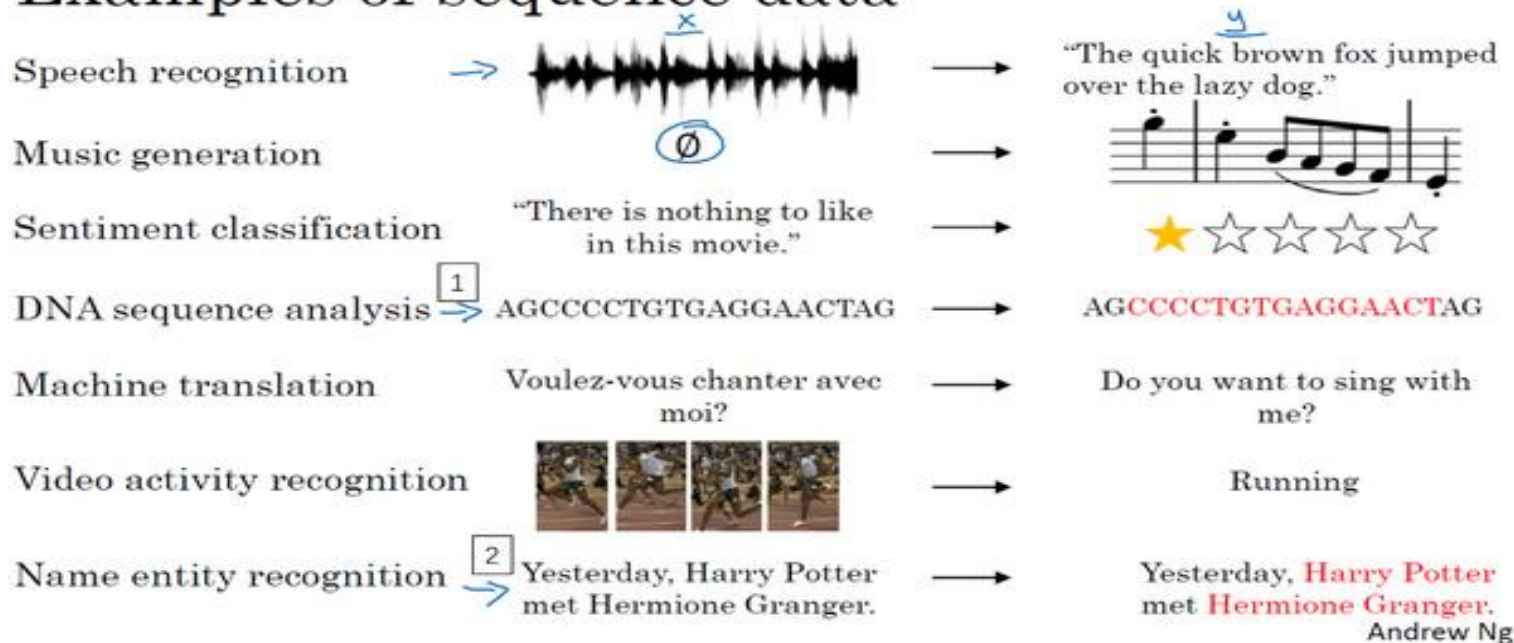
**05** 深层循环神经网络

# 1.序列模型概述

4

循环神经网络（**RNN**）之类的模型在语音识别、自然语言处理和其他领域中引起变革。

## Examples of sequence data



# 数学符号

5

在这里 $x^{<1>}$ 表示**Harry**这个单词，它就是一个第4075行是1，其余值都是0的向量（上图编号1所示），因为那是**Harry**在这个词典里的位置。

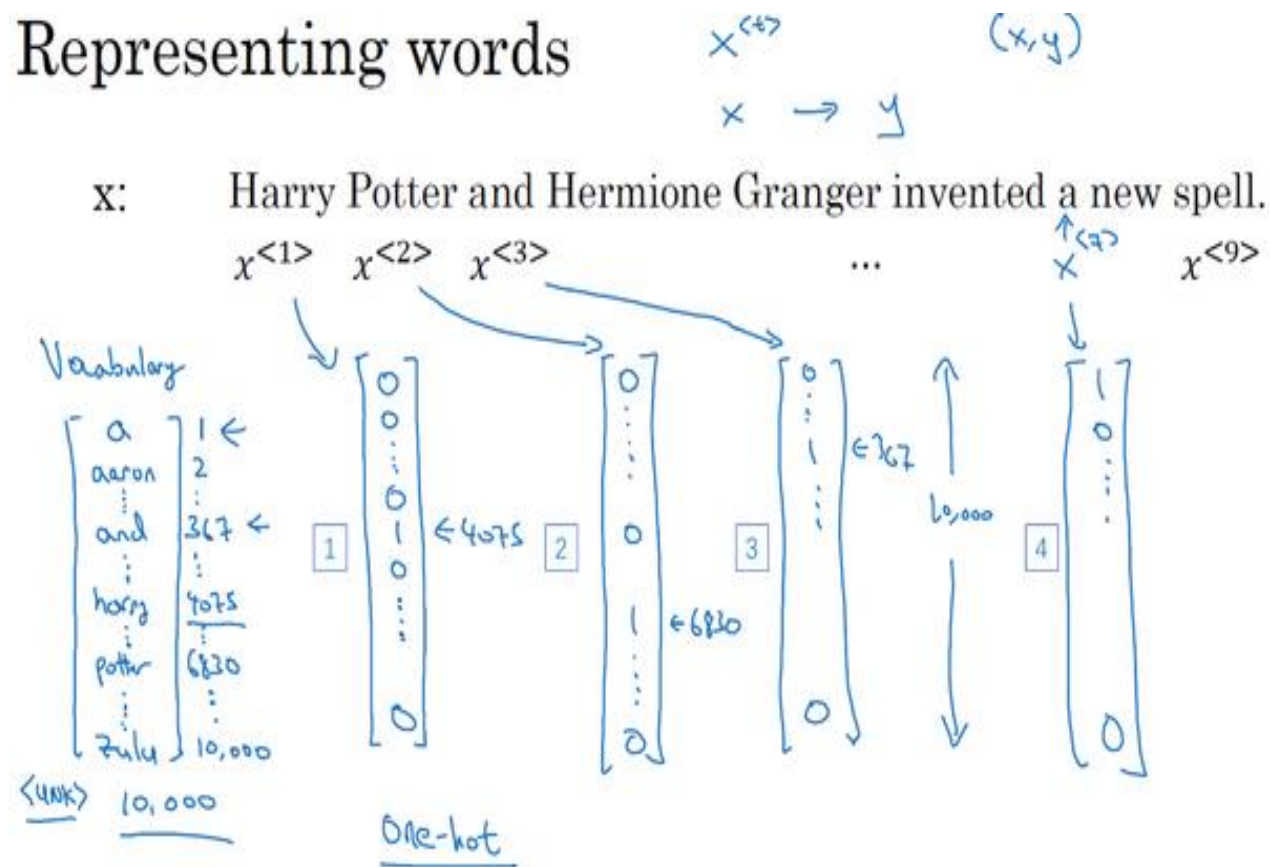
$x^{<2>}$ 是第6830行是1，其余位置都是0的向量（上图编号2所示）。

**and**在词典里排第367，所以 $x^{<3>}$ 就是第367行是1，其余值都是0的向量（上图编号3所示）。

因为**a**是字典第一个单词， $x^{<7>}$ 对应**a**，那么这个向量的第一个位置为1，其余位置都是0的向量（上图编号4所示）。

**Unknow Word**的伪单词，用**<UNK>**作为标记。

## Representing words



如果你的词典大小是10,000的话，那么这里的每个向量都是10,000维的。



# 循环神经网络解决的问题

6

- **卷积神经网络或全连接网络的局限性**
  - 同一层节点之间无关联，从而导致获取时序规则方面功能不足
- **循环神经网络可以解决时序问题**
  - 基于语言模型（LM），故可以捕捉时序规则信息
  - 它是如何实现？

## 2.循环神经网络(RNN)

7

**01** 序列模型概述

**02** 循环神经网络(RNN)

**03** 长短期记忆(LSTM)

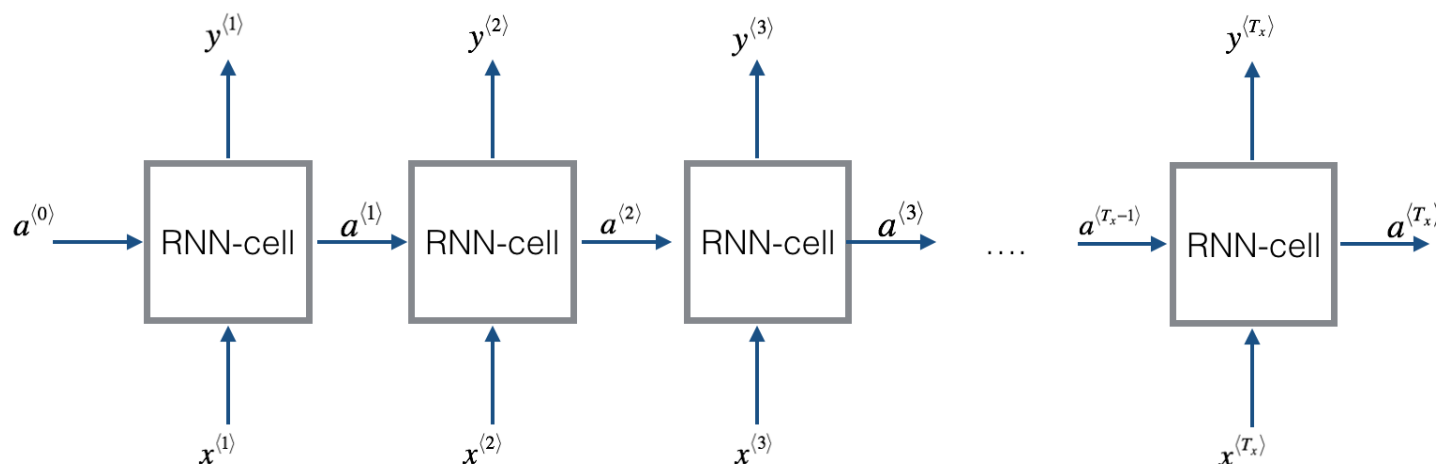
**04** 双向循环神经网络

**05** 深层循环神经网络

## 2.循环神经网络(RNN)

8

### RNN的前向传播



$$a^{(t)} = \tanh(W_{ax}x^{(t)} + W_{aa}a^{(t-1)} + b_a)$$
$$\hat{y}^{(t)} = \text{soft max}(W_{ya}a^{(t)} + b_y)$$

$$a^{<0>} = 0$$

$$a^{<1>} = g_1(W_{aa}a^{<0>} + W_{ax}x^{<1>} + b_a)$$

$$\hat{y}^{<1>} = g_2(W_{ya}a^{<1>} + b_y)$$

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

$$\hat{y}^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

```
rnn=nn.RNN(input size=10,hidden size=20,num layers=2)
```



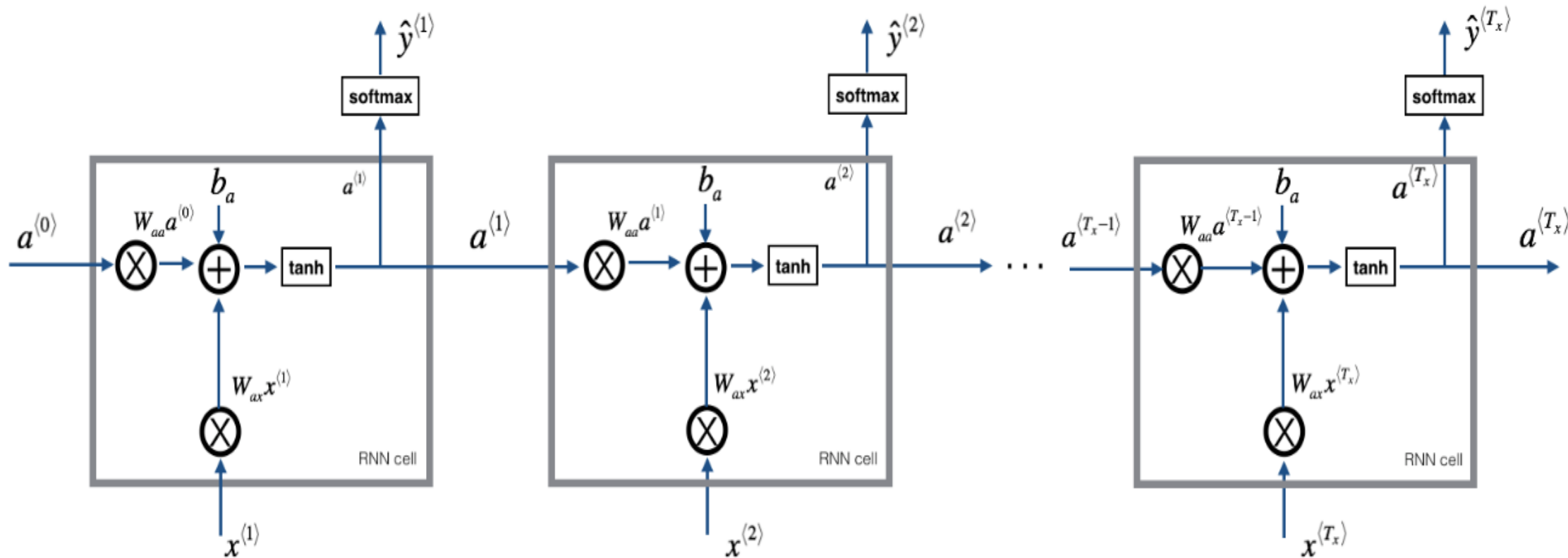
## 2.循环神经网络(RNN)

9

RNN的前向传播

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

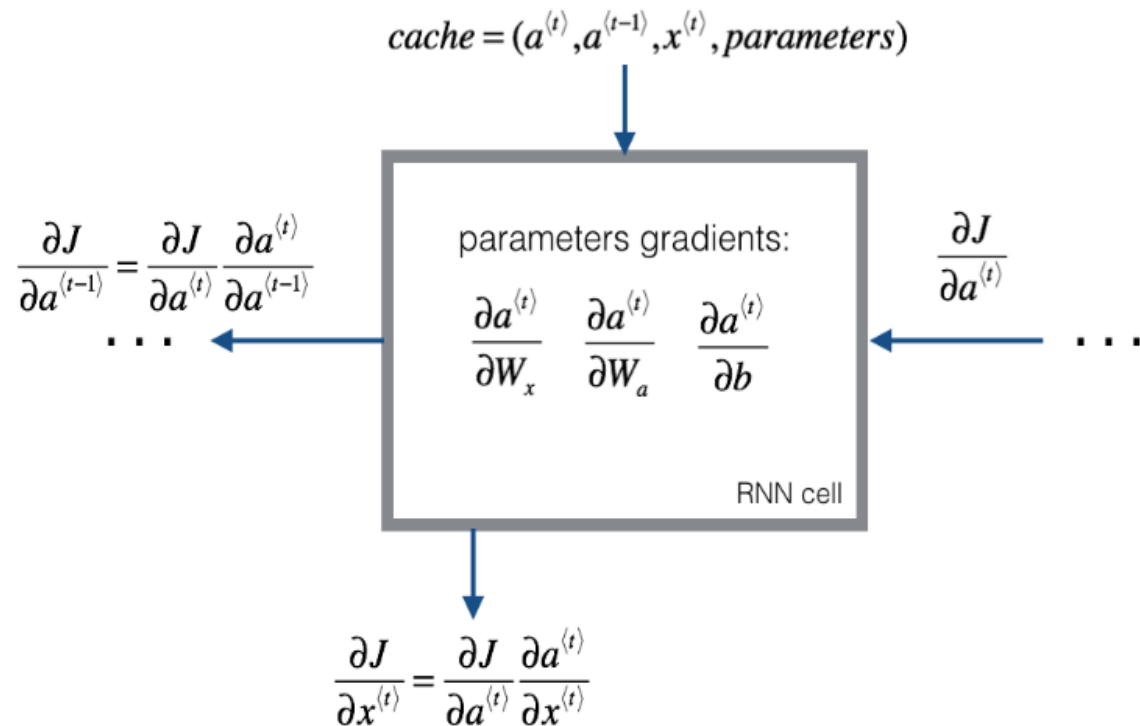
$$\hat{y}^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$



## 2.循环神经网络(RNN)

10

### RNN的反向传播



$$a^{(t)} = \tanh(W_{ax}x^{(t)} + W_{aa}a^{(t-1)} + b)$$

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

$$\frac{\partial a^{(t)}}{\partial W_{ax}} = (1 - \tanh(W_{ax}x^{(t)} + W_{aa}a^{(t-1)} + b)^2) x^{(t)T}$$

$$\frac{\partial a^{(t)}}{\partial W_{aa}} = (1 - \tanh(W_{ax}x^{(t)} + W_{aa}a^{(t-1)} + b)^2) a^{(t-1)T}$$

$$\frac{\partial a^{(t)}}{\partial b} = \sum_{batch} (1 - \tanh(W_{ax}x^{(t)} + W_{aa}a^{(t-1)} + b)^2)$$

$$\frac{\partial a^{(t)}}{\partial x^{(t)}} = W_{ax}^T \cdot (1 - \tanh(W_{ax}x^{(t)} + W_{aa}a^{(t-1)} + b)^2)$$

$$\frac{\partial a^{(t)}}{\partial a^{(t-1)}} = W_{aa}^T \cdot (1 - \tanh(W_{ax}x^{(t-1)} + W_{aa}a^{(t-1)} + b)^2)$$

输出序列

1.56

=

0.1

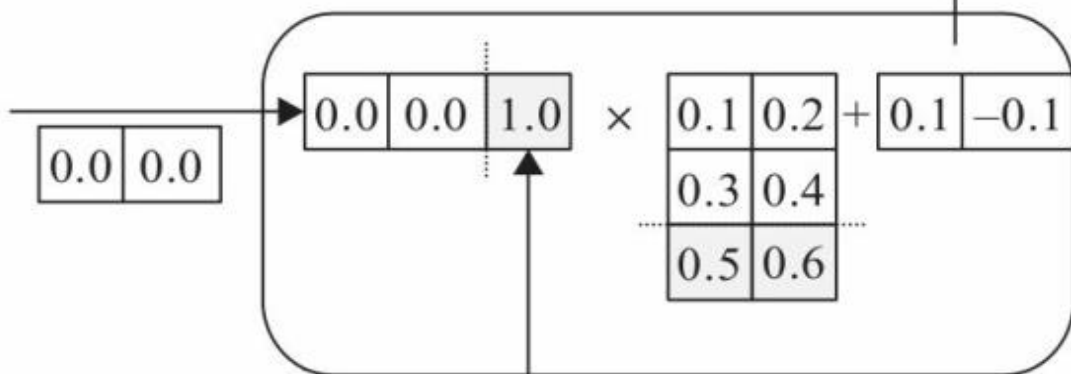
+

1.0

2.0

×

0.537 0.462



输入序列

1

时间序列

$t_0$

2.73

=

0.1

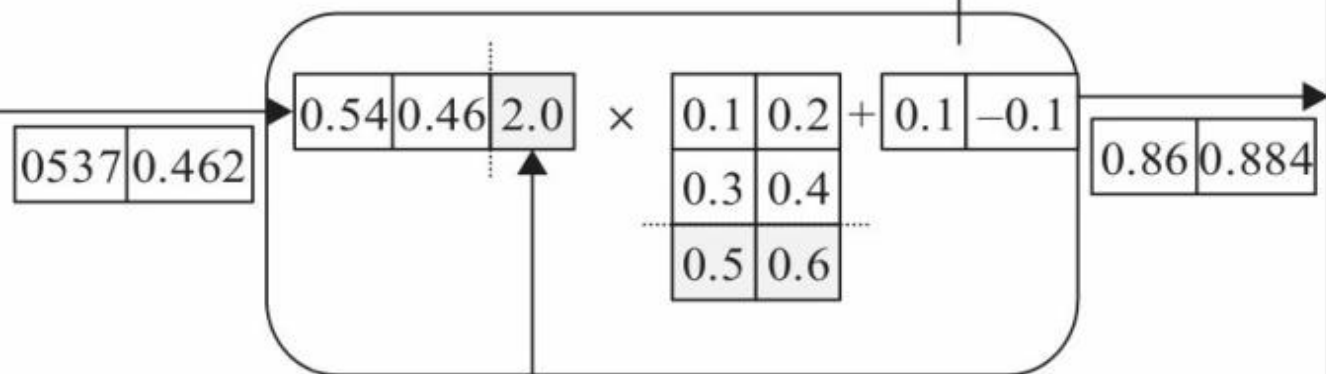
+

1.0

2.0

×

0.86 0.884



2

$t_1$

```
import numpy as np

X = [1, 2]
state = [0.0, 0.0]
w_cell_state = np.asarray([[0.1, 0.2], [0.3, 0.4], [0.5, 0.6]])
b_cell = np.asarray([0.1, -0.1])
w_output = np.asarray([[1.0], [2.0]])
b_output = 0.1

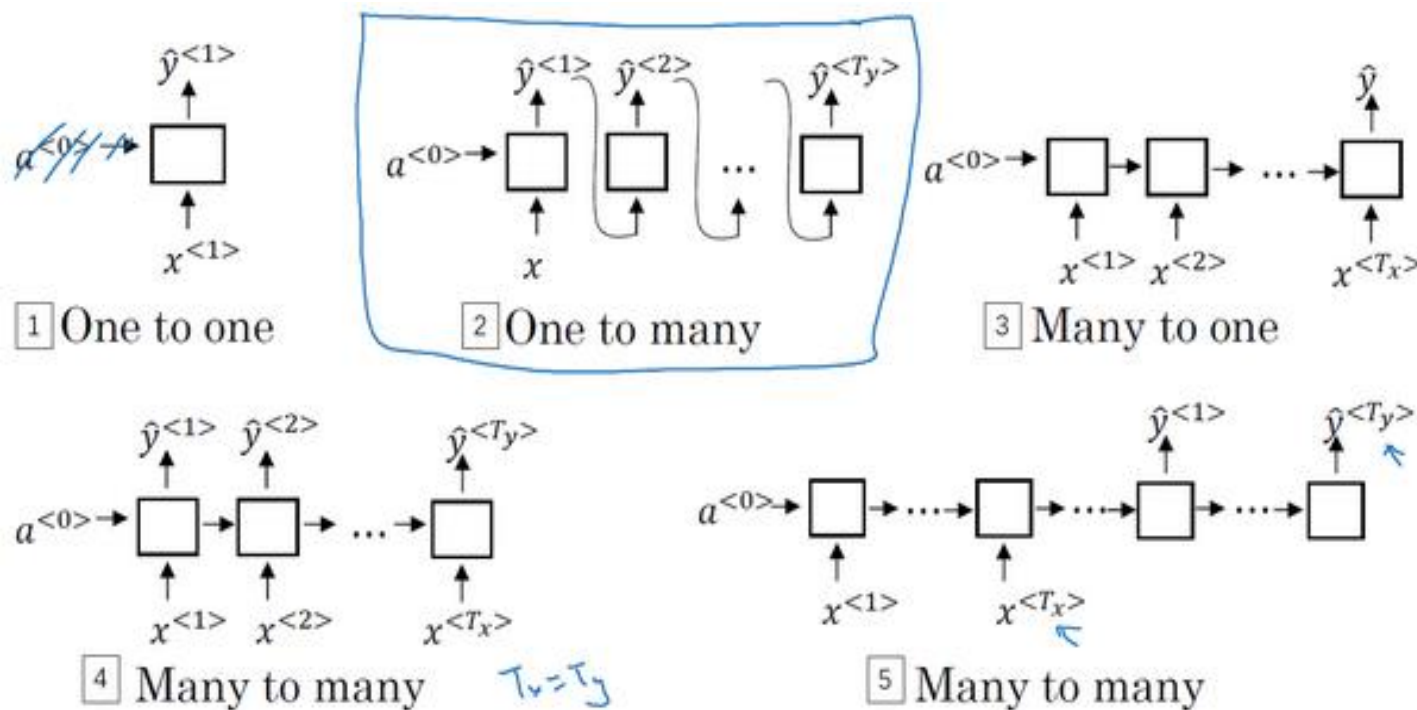
for i in range(len(X)):
    state = np.append(state, X[i])
    before_activation = np.dot(state, w_cell_state) + b_cell
    state = np.tanh(before_activation)
    final_output = np.dot(state, w_output) + b_output
    print("状态值_%i: " % i, state)
    print("输出值_%i: " % i, final_output)
```

```
状态值_0: [0.53704957 0.46211716]
输出值_0: [1.56128388]
状态值_1: [0.85973818 0.88366641]
输出值_1: [2.72707101]
```

## 2.循环神经网络(RNN)

13

### RNN的类型



## 2.循环神经网络(RNN)

14

语言模型和序列生成

**The apple and pear (pair) salad was delicious.**

第一句话的概率是:

$$P(\text{The apple and pair salad}) = 3.2 \times 10^{-13},$$

而第二句话的概率是:

$$P(\text{The apple and pear salad}) = 5.7 \times 10^{-10},$$

# 3.长短期记忆(LSTM)

15

**01** 序列模型概述

**02** 循环神经网络(RNN)

**03** 长短期记忆(LSTM)

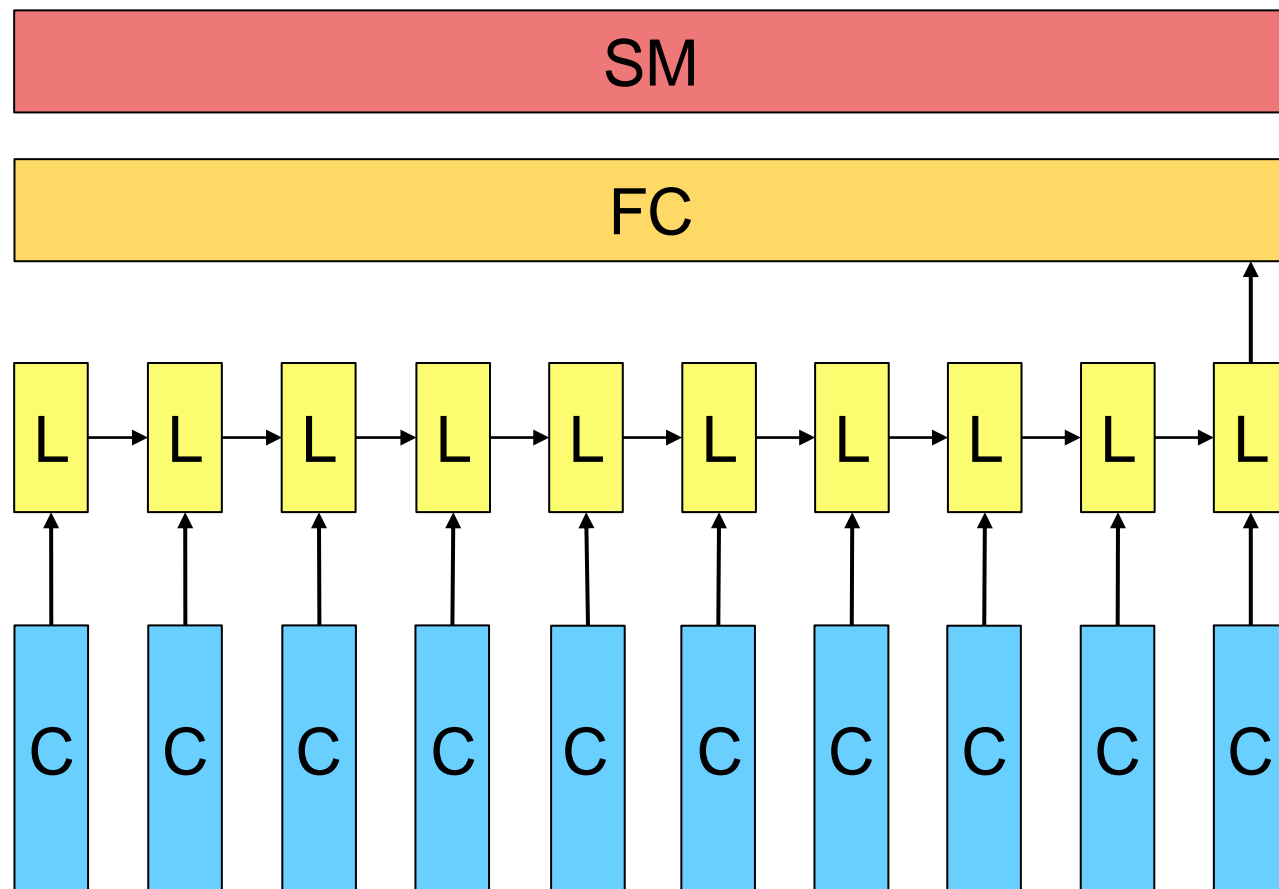
**04** 双向循环神经网络

**05** 深层循环神经网络



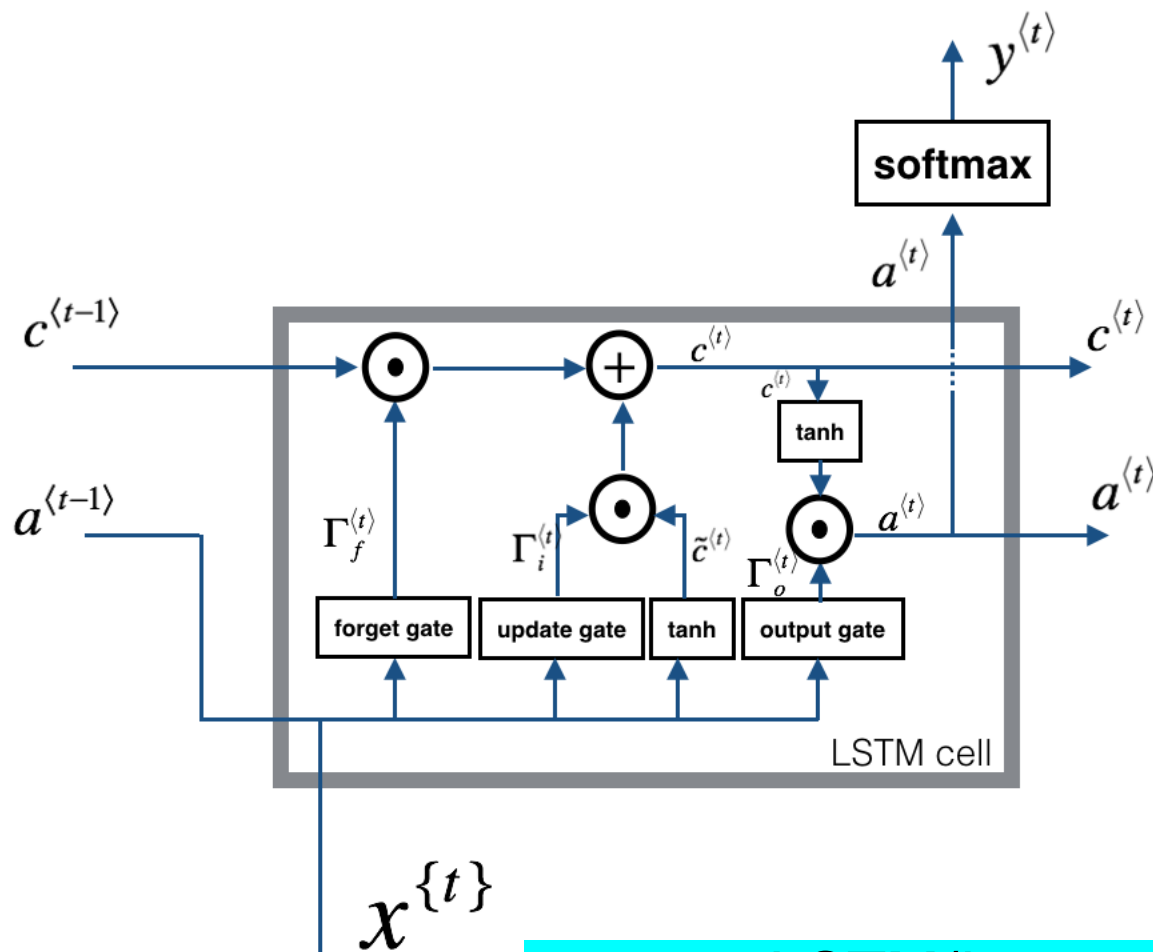
### 3.长短期记忆(LSTM)

16



### 3.长短期记忆(LSTM)

17



$$\Gamma_f^{\langle t \rangle} = \sigma(W_f[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f)$$

$$\Gamma_u^{\langle t \rangle} = \sigma(W_u[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$$

$$\tilde{c}^{\{t\}} = \tanh(W_c[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c)$$

$$c^{\langle t \rangle} = \Gamma_f^{\langle t \rangle} \odot c^{\langle t-1 \rangle} + \Gamma_u^{\langle t \rangle} \odot \tilde{c}^{\langle t \rangle}$$

$$\Gamma_o^{\langle t \rangle} = \sigma(W_o[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o)$$

$$a^{\langle t \rangle} = \Gamma_o^{\langle t \rangle} \odot \tanh(c^{\langle t \rangle})$$

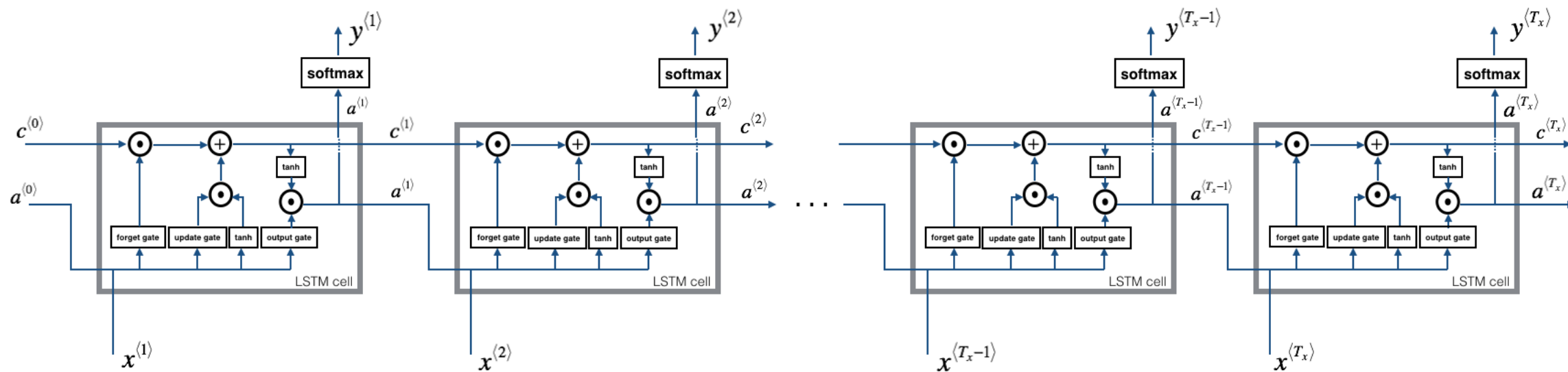
...

`lstm=nn.LSTM(input size=10,hidden size=20,num layers=2)`

# 3.长短期记忆(LSTM)

18

## LSTM的前向传播



### 3.长短期记忆(LSTM)

19

#### LSTM的反向传播

$$d\Gamma_o^{\langle t \rangle} = da_{next} * \tanh(c_{next}) * \Gamma_o^{\langle t \rangle} * (1 - \Gamma_o^{\langle t \rangle})$$

$$d\tilde{c}^{\langle t \rangle} = dc_{next} * \Gamma_i^{\langle t \rangle} + \Gamma_o^{\langle t \rangle} (1 - \tanh(c_{next})^2) * i_t * da_{next} * \tilde{c}^{\langle t \rangle} * (1 - \tanh(\tilde{c})^2)$$

$$d\Gamma_u^{\langle t \rangle} = dc_{next} * \tilde{c}^{\langle t \rangle} + \Gamma_o^{\langle t \rangle} (1 - \tanh(c_{next})^2) * \tilde{c}^{\langle t \rangle} * da_{next} * \Gamma_u^{\langle t \rangle} * (1 - \Gamma_u^{\langle t \rangle})$$

$$d\Gamma_f^{\langle t \rangle} = dc_{next} * \tilde{c}_{prev} + \Gamma_o^{\langle t \rangle} (1 - \tanh(c_{next})^2) * c_{prev} * da_{next} * \Gamma_f^{\langle t \rangle} * (1 - \Gamma_f^{\langle t \rangle})$$

# 3.长短期记忆(LSTM)

20

## LSTM的反向传播

### 参数求导

$$dW_f = d\Gamma_f^{(t)} * \begin{pmatrix} a_{prev} \\ x_t \end{pmatrix}^T$$

$$dW_c = d\tilde{c}^{(t)} * \begin{pmatrix} a_{prev} \\ x_t \end{pmatrix}^T$$

$$dW_o = d\Gamma_o^{(t)} * \begin{pmatrix} a_{prev} \\ x_t \end{pmatrix}^T$$

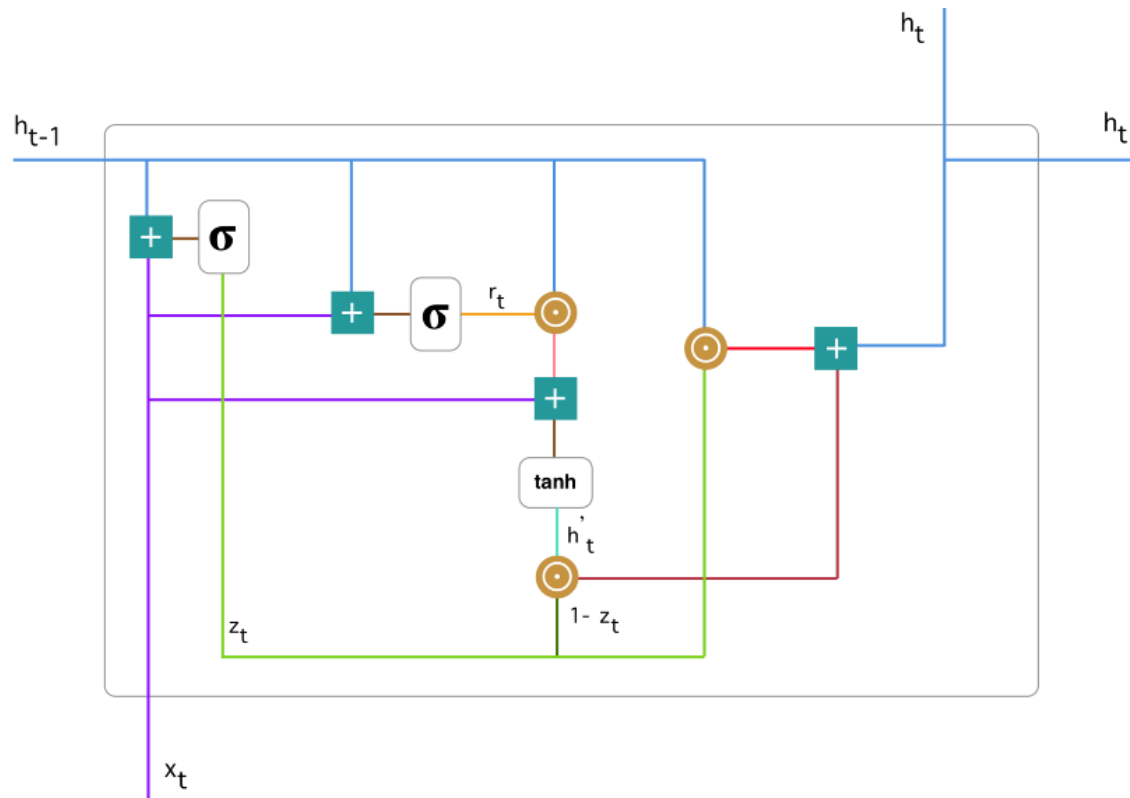
为了计算 $db_f, db_u, db_c, db_o$  你需要对  $d\Gamma_f^{(t)}, d\Gamma_u^{(t)}, d\tilde{c}^{(t)}, d\Gamma_o^{(t)}$  水平方向求和. 得到:

$$da_{prev} = W_f^T * d\Gamma_f^{(t)} + W_u^T * d\Gamma_u^{(t)} + W_c^T * d\tilde{c}^{(t)} + W_o^T * d\Gamma_o^{(t)}$$

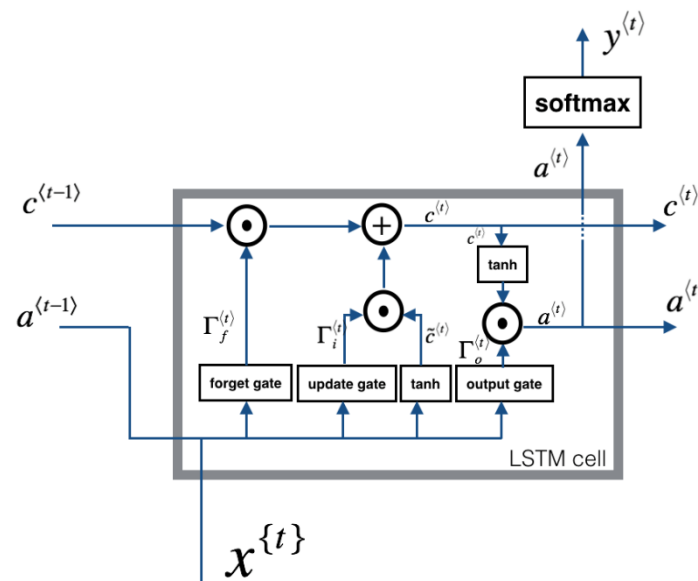
# 3.长短期记忆(LSTM)

21

## GRU



## LSTM



$$\begin{aligned} \Gamma_f^{(t)} &= \sigma(W_f[a^{(t-1)}, x^{(t)}] + b_f) \\ \Gamma_u^{(t)} &= \sigma(W_u[a^{(t-1)}, x^{(t)}] + b_u) \\ \tilde{c}^{(t)} &= \tanh(W_c[a^{(t-1)}, x^{(t)}] + b_c) \\ c^{(t)} &= \Gamma_f^{(t)} \circ c^{(t-1)} + \Gamma_u^{(t)} \circ \tilde{c}^{(t)} \\ \Gamma_o^{(t)} &= \sigma(W_o[a^{(t-1)}, x^{(t)}] + b_o) \\ a^{(t)} &= \Gamma_o^{(t)} \circ \tanh(c^{(t)}) \end{aligned}$$

## 2.循环神经网络(RNN)

22

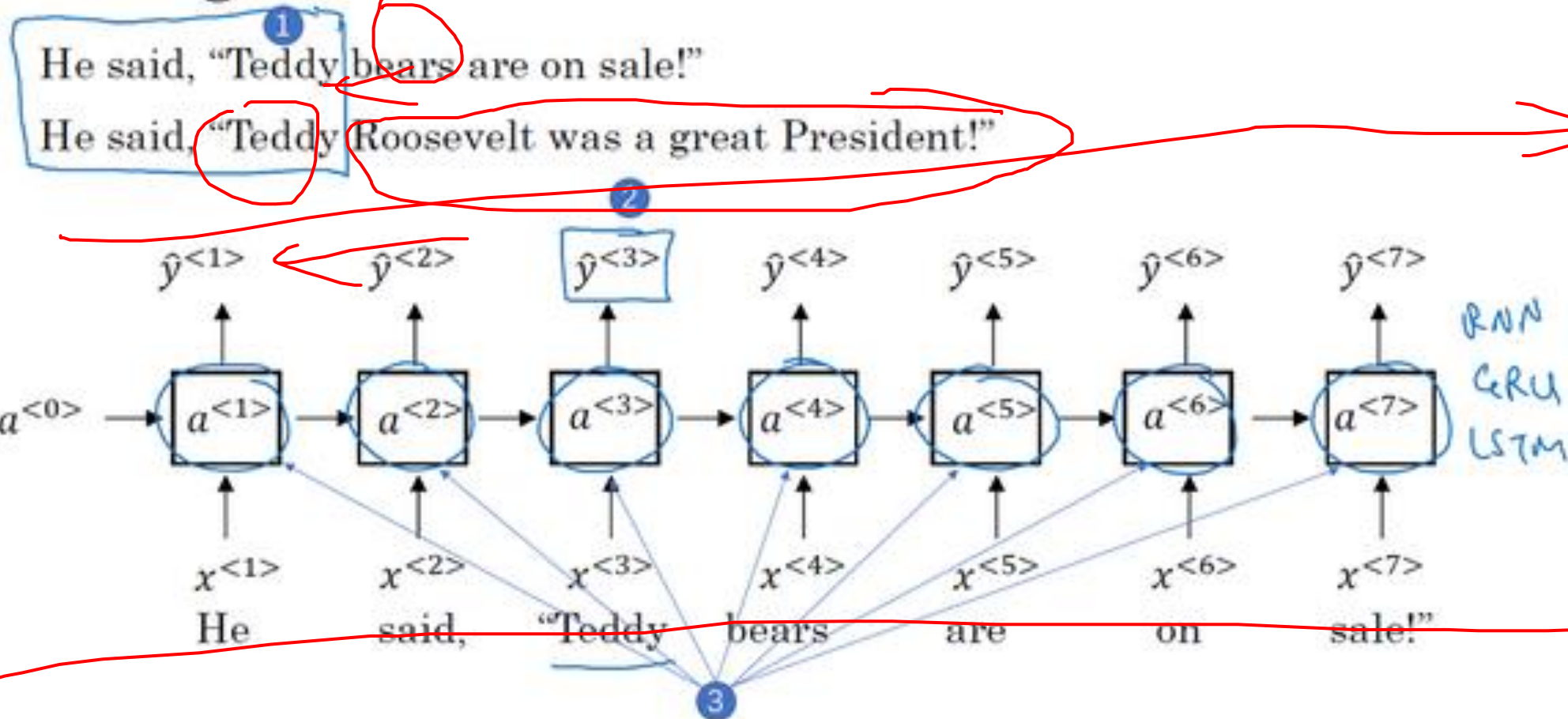
- 01** 序列模型概述
- 02** 循环神经网络(RNN)
- 03** 长短期记忆(LSTM)
- 04** 双向循环神经网络
- 05** 深层循环神经网络



# 4.双向循环神经网络

23

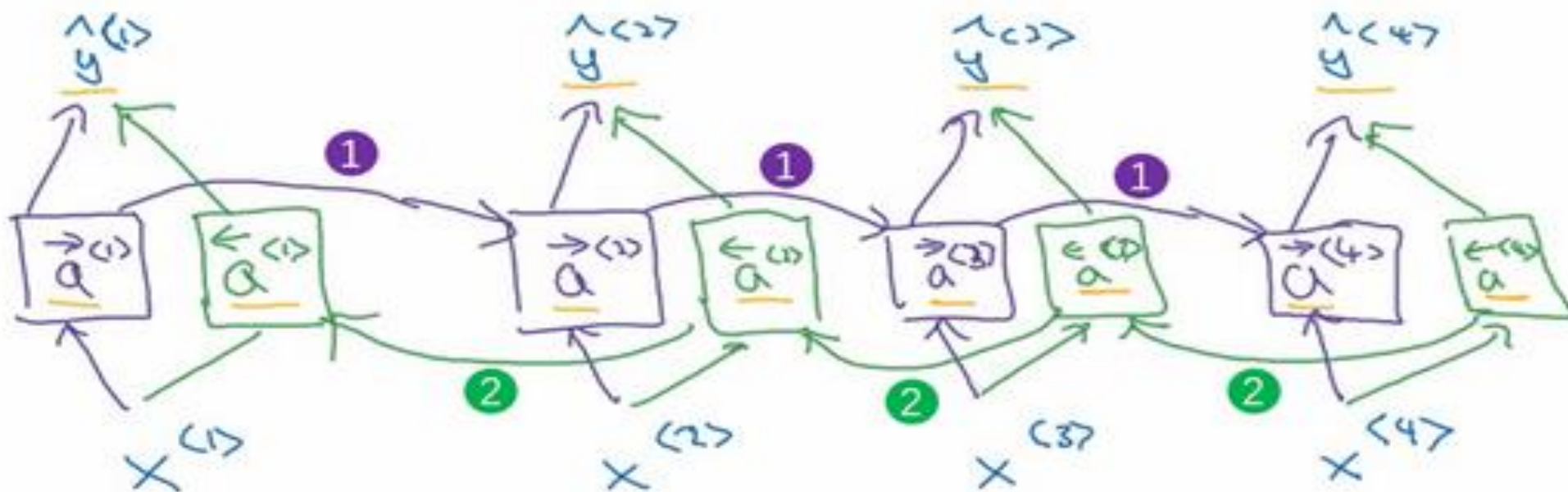
Getting information from the future



## 4.双向循环神经网络

24

### Bidirectional RNN (BRNN)



Acyclic graph

# 5. 深层循环神经网络

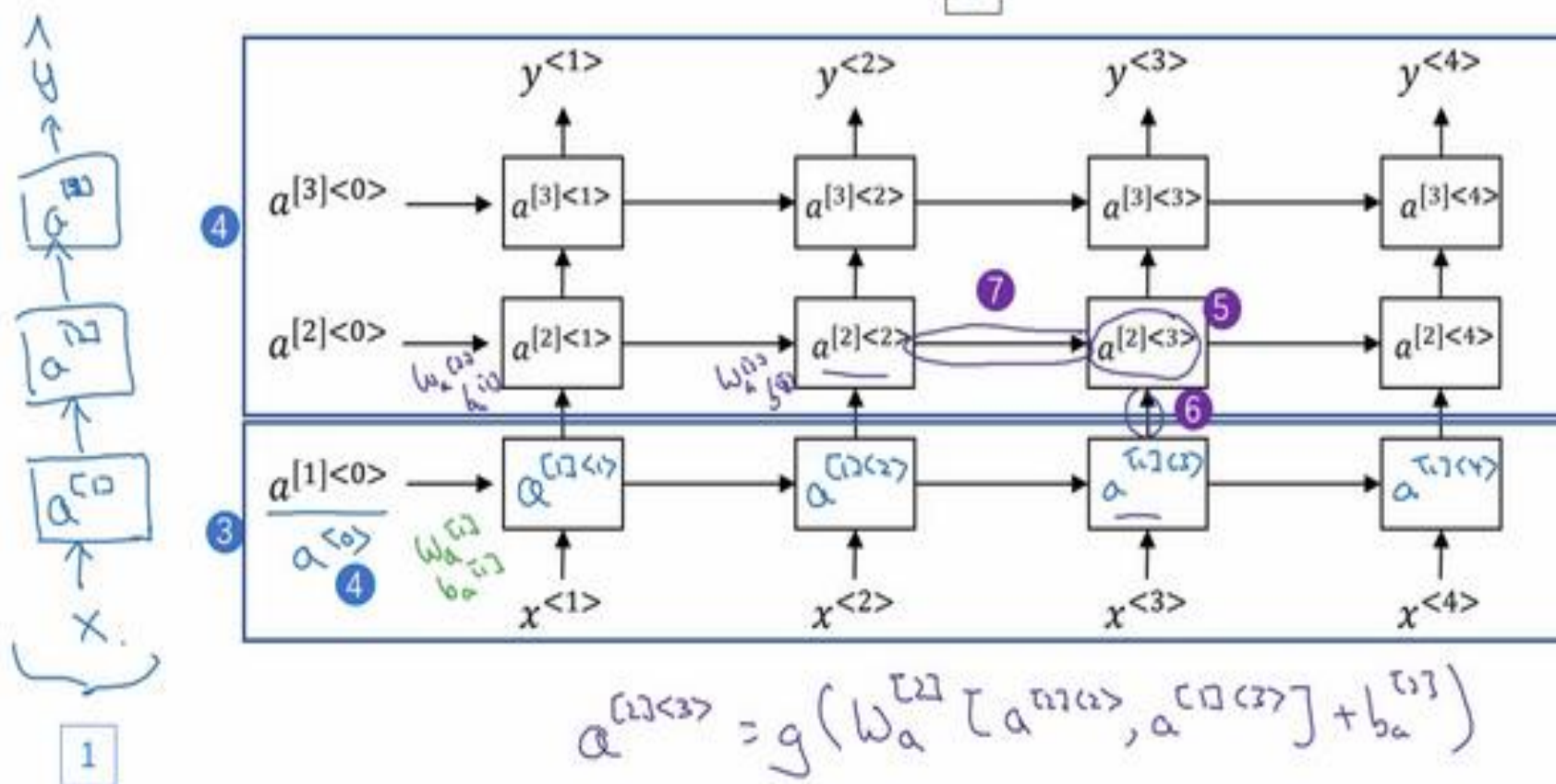
25

- 01** 序列模型概述
- 02** 循环神经网络(RNN)
- 03** 长短期记忆(LSTM)
- 04** 双向循环神经网络
- 05** 深层循环神经网络

# 5. 深层循环神经网络

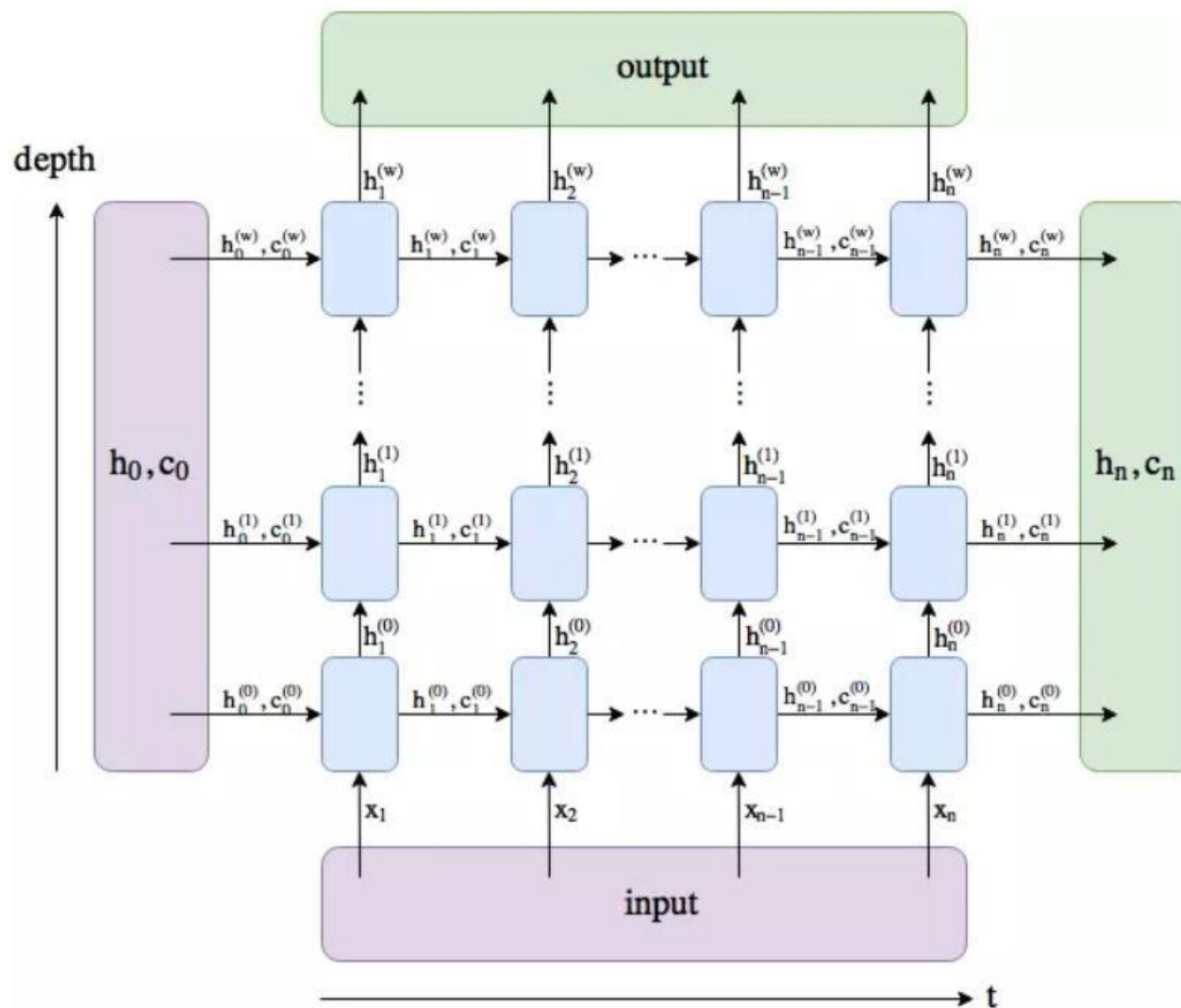
26

Deep RNN example



# 5. 深层循环神经网络

27



1. IAN GOODFELLOW等, 《深度学习》, 人民邮电出版社, 2017
2. Andrew Ng, <http://www.deeplearning.ai>



谢谢!

