



# 循环神经网络

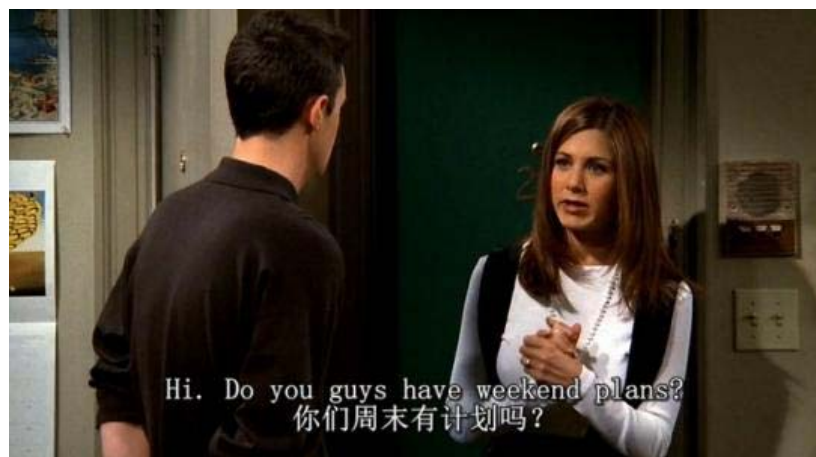
(Recurrent Neural Network)

刘远超

哈尔滨工业大学

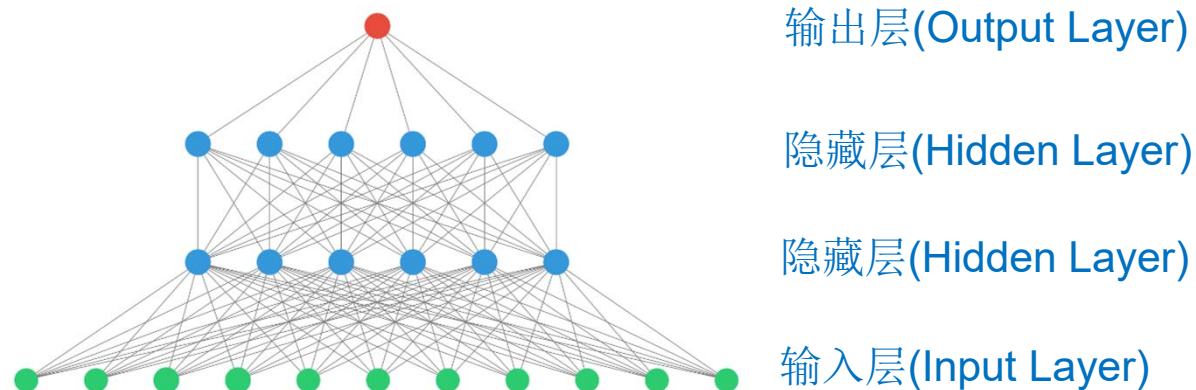
计算机科学与技术学院

# 序列数据

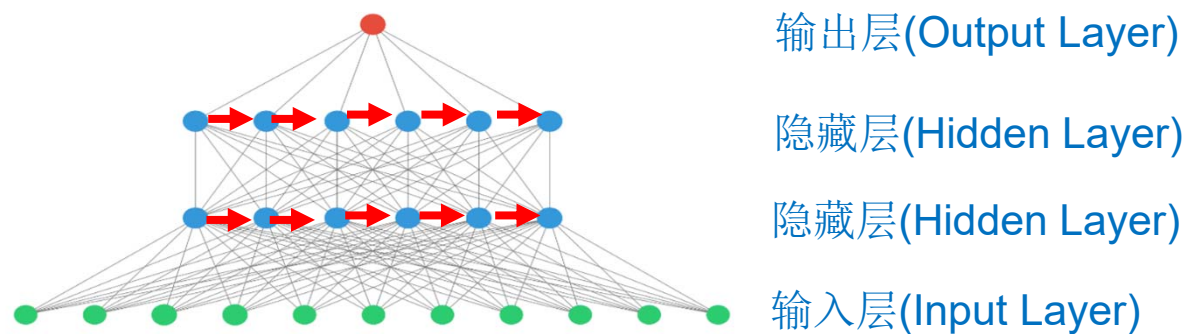


# 什么是循环神经网络？

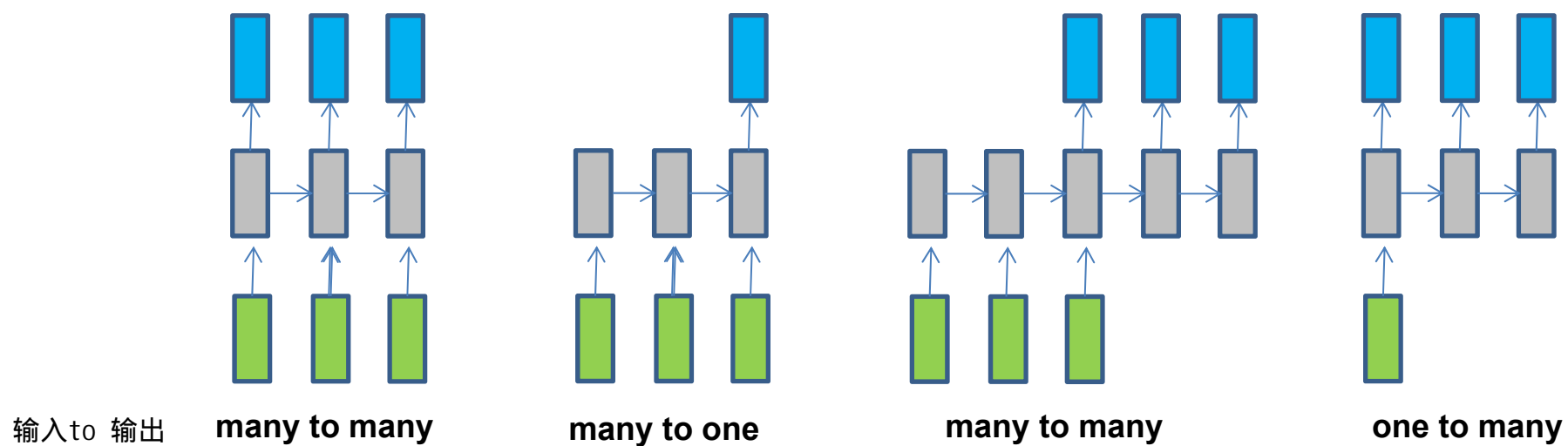
- 传统的神经网络模型，隐藏层的节点之间是无连接的。



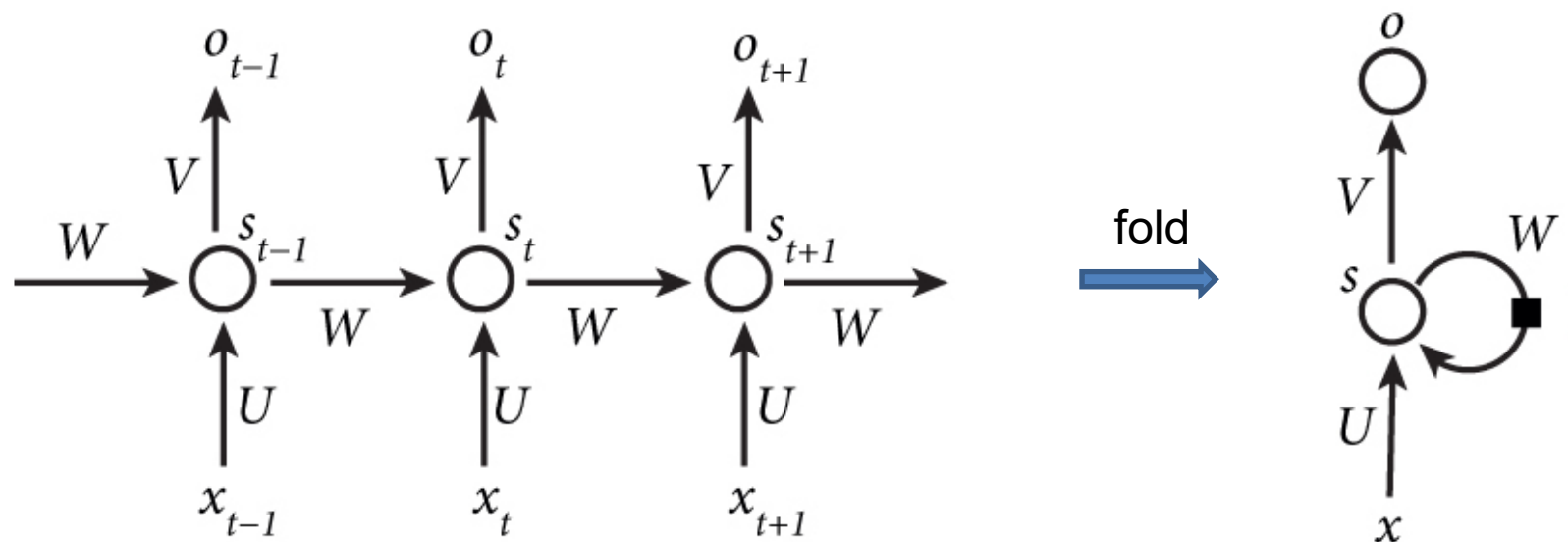
- 循环神经网络 (Recurrent Neural Network, RNN)：隐藏层的节点之间有连接，是主要用于对序列数据进行分类、预测等处理的神经网络。



# RNN序列处理

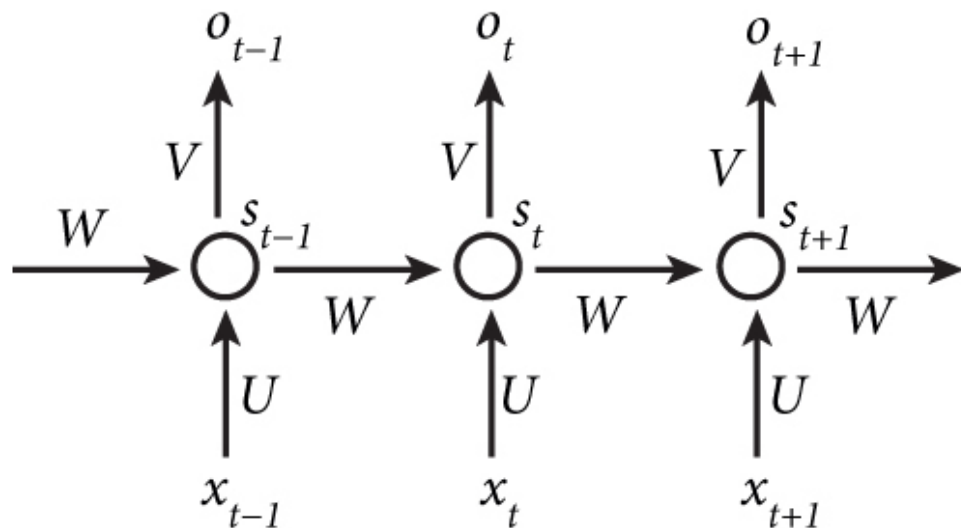


# 最基本的RNN结构



- 输入单元 (input units)为  $\{x_0, x_1, \dots, x_{t-1}, x_t, x_{t+1}, \dots\}$
- 输出单元 (output units)为  $\{o_0, o_1, \dots, o_{t-1}, o_t, o_{t+1}, \dots\}$
- 隐藏单元 (Hidden units)的输出标记为  $\{s_0, s_1, \dots, s_{t-1}, s_t, s_{t+1}, \dots\}$

# 基本RNN的计算过程

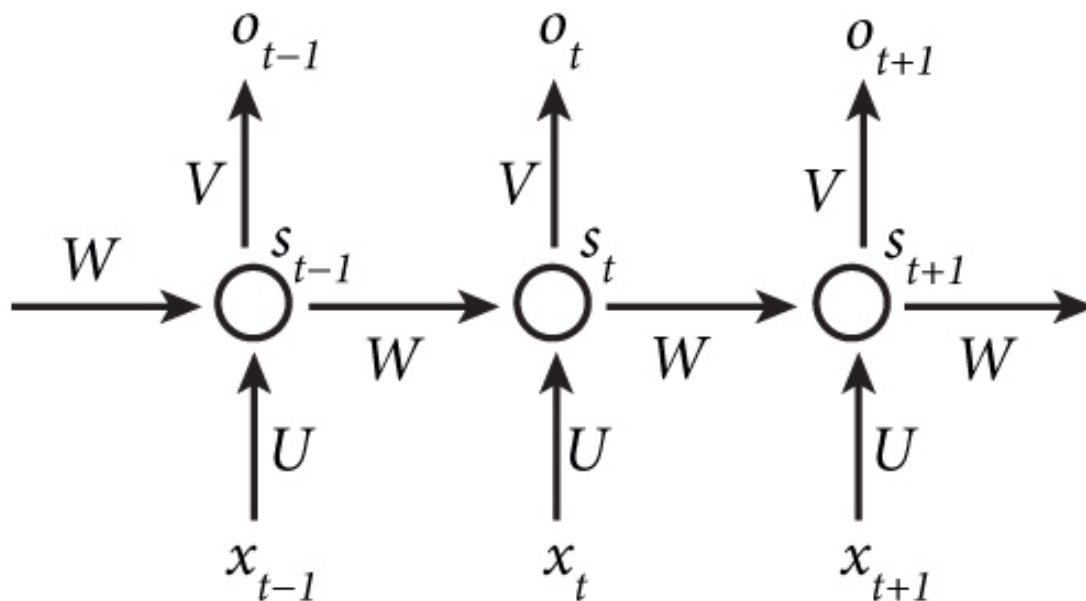


- 输入层:  $\mathbf{x}_t$  表示时刻  $t$  的输入.
- 隐藏层:  $\mathbf{s}_t = f(\mathbf{U}\mathbf{x}_t + \mathbf{W}\mathbf{s}_{t-1})$ . 其中  $f$  是非线性激活函数, 如 [tanh](#).
- 输出层:  $\mathbf{o}_t = \text{softmax}(\mathbf{V}\mathbf{s}_t)$ .

归一化

其中 softmax 函数的形式  $\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$ 。

# RNN的参数共享



- 传统神经网络中，每一层的参数是不共享的；
- 而在RNNs中，每一步(每一层)都共享参数 $U$ ,  $V$ ,  $W$ 。

# Thanks!







# 长短时记忆网络

(Long Short-Term Memory)

刘远超

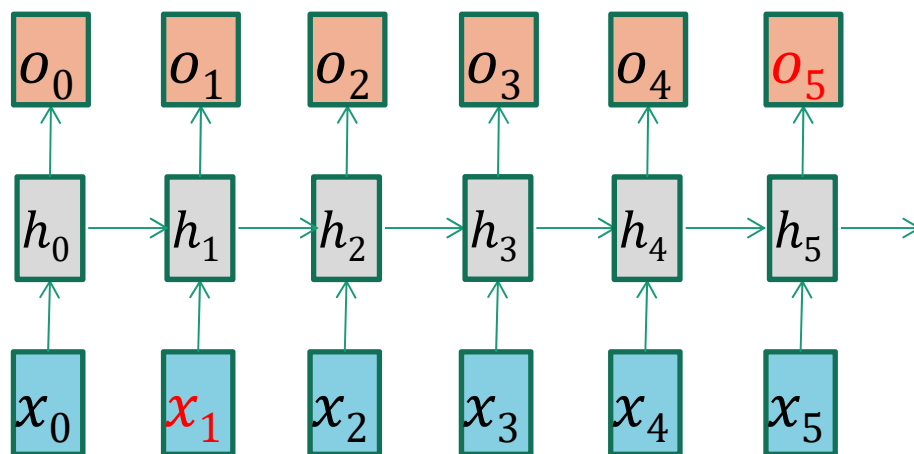
哈尔滨工业大学

计算机科学与技术学院

# 标准RNN可以处理短期依赖

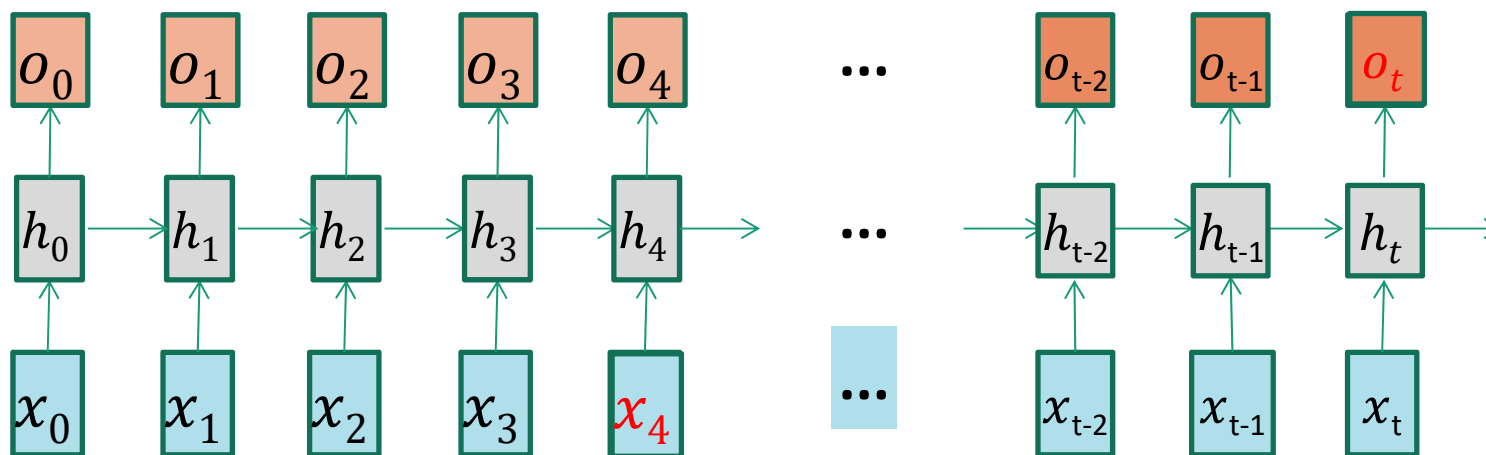
- 标准RNN可以处理不太长的相关信息间隔：

- 例如，预测 “the clouds are in the \_\_\_\_” 空格中的词。



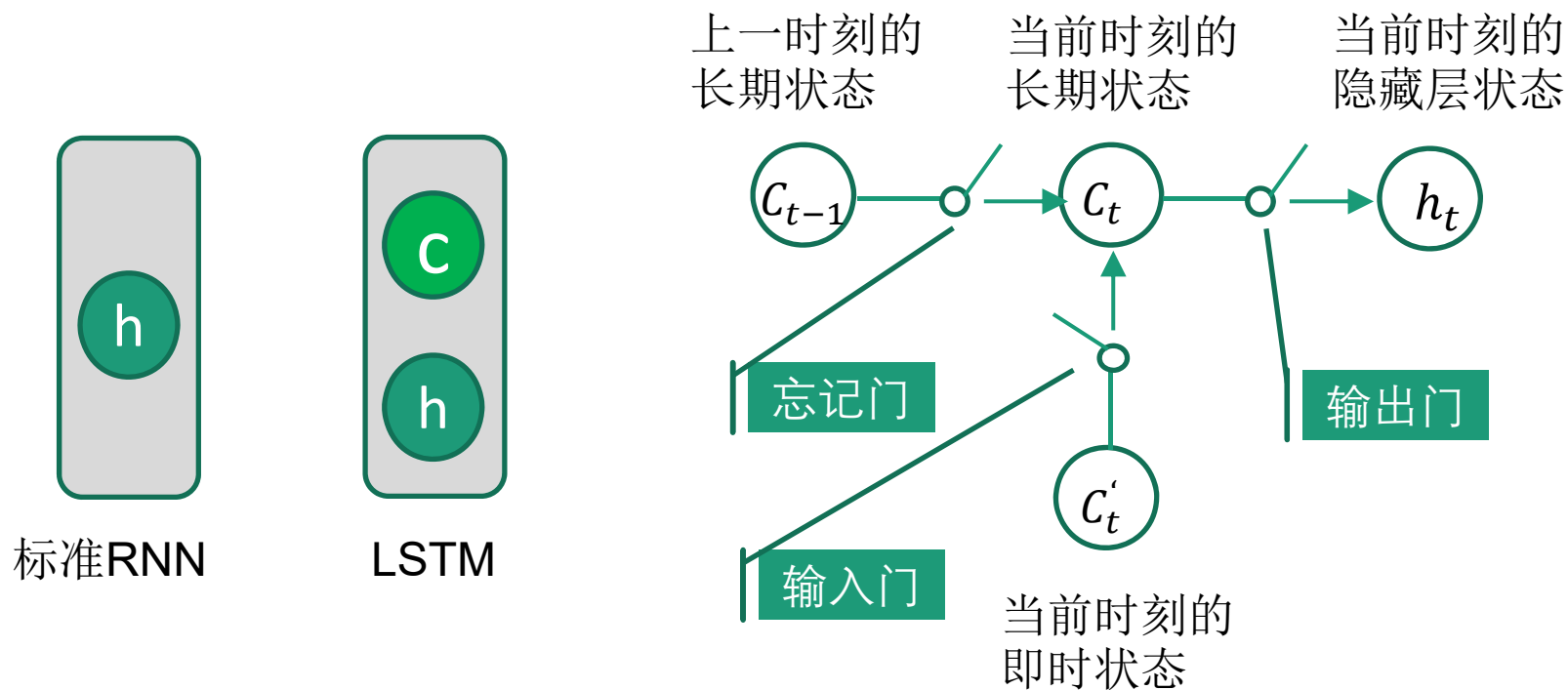
# 标准RNN难以应对长期依赖

- 但标准RNN无法处理更长的上下文间隔，即长期依赖问题。
  - 例如，预测“I grew up in France... I speak fluent \_\_\_\_\_”最后的词。

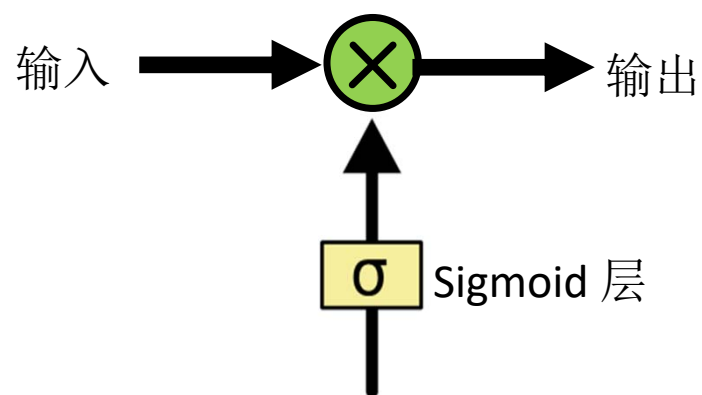


# LSTM 的基本思路

● LSTM(Long Short-Term Memory)，即长短期记忆网络，是RNN的扩展，其通过特殊的结构设计来避免长期依赖问题。

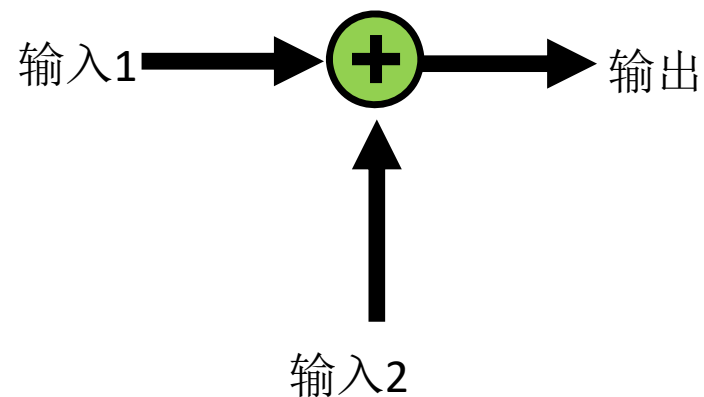


# 神经网络中的门



## 乘法门:

- 为了让信息选择性通过;
- sigmoid 层的输出矩阵中每个元素的范围是[0, 1]

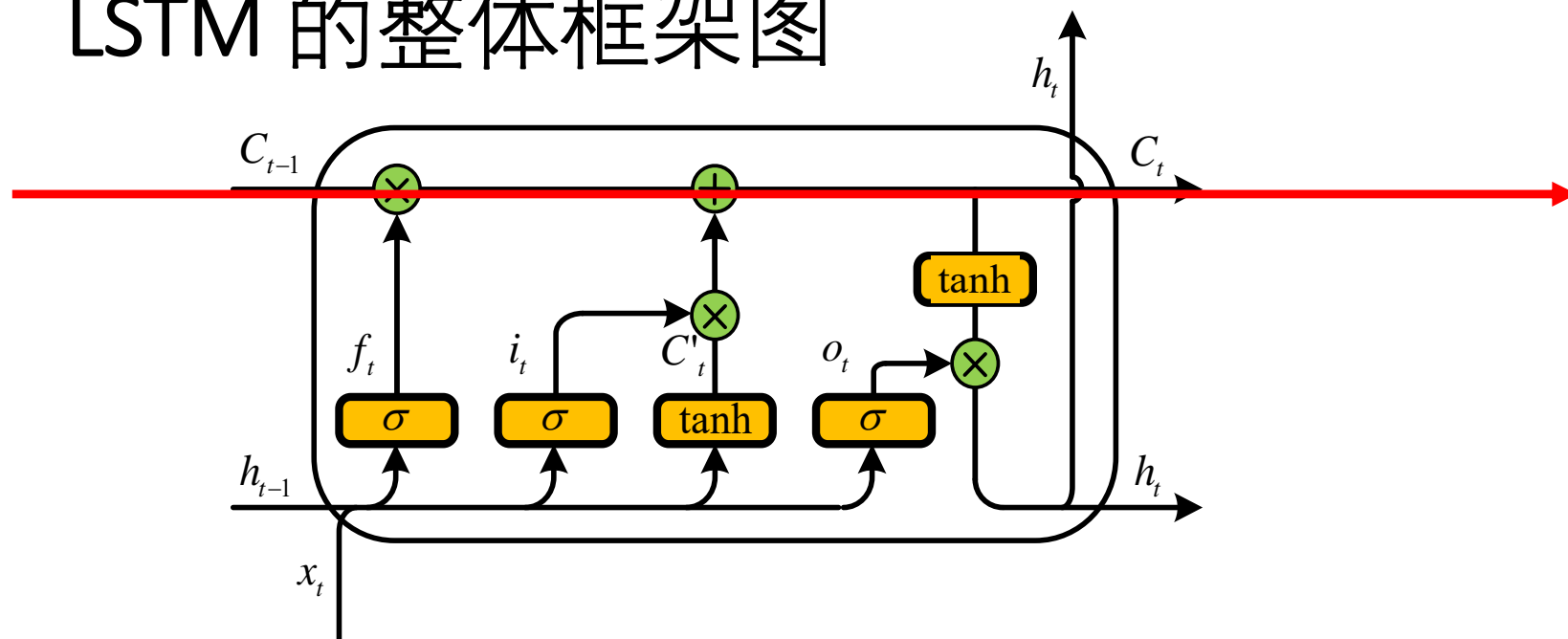



## 加法门:


- 在输入1基础上更新输入2的信息


因此，LSTM中忘记门和输出门要用到乘法门。输入门要用到加法门。


# LSTM 的整体框架图




  $\Rightarrow$  神经网络层

  $\Rightarrow$  逐点操作

  $\Rightarrow$  传输向量

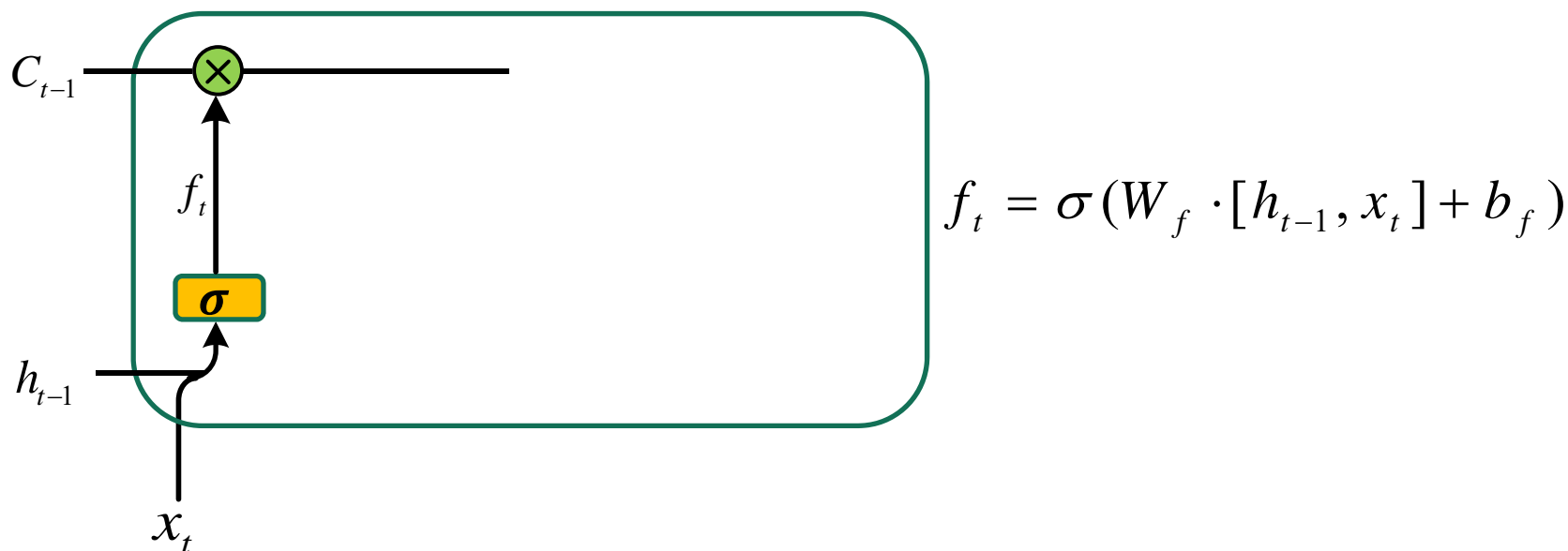
  $\Rightarrow$  向量被复制

  $\Rightarrow$  向量的连接

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

# LSTM的计算过程(1)

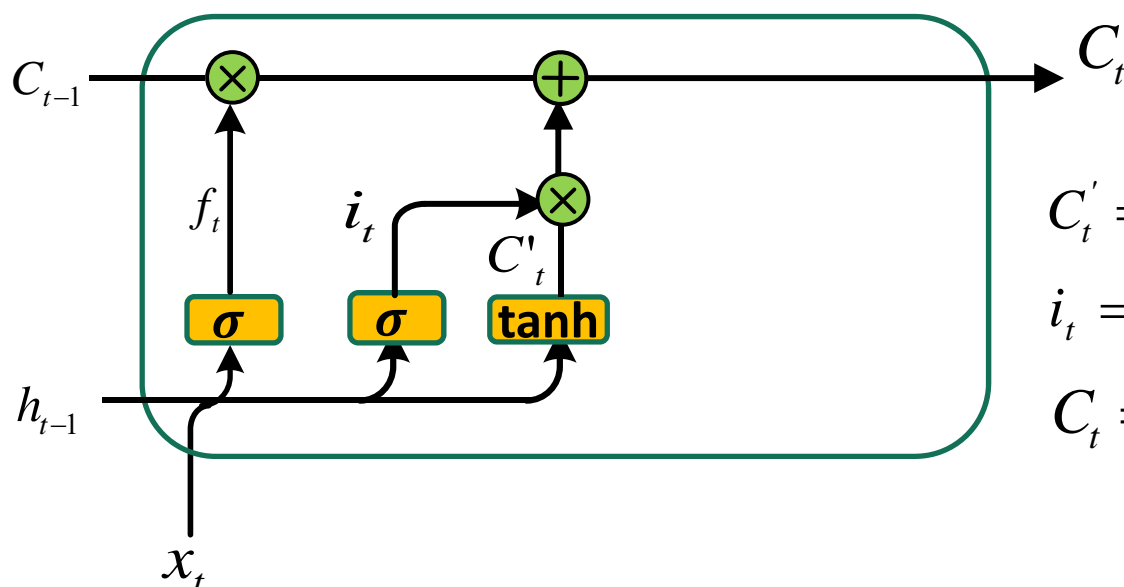
忘记信息：从长期状态中**丢弃某些信息**。



- 忘记门层  $f_t$  的输入为  $h_{t-1}$  和  $x_t$ ，输出的矩阵中每个元素为 0 到 1 之间的数值，并与细胞状态矩阵  $C_{t-1}$  中的每个对应位置元素相乘。
- 语言模型例子：... *Germany* .... I grew up in *France*... I speak fluent \_\_\_\_。

## LSTM的计算过程(2)

新记忆信息：将新信息存放在长期状态中。



$$C'_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

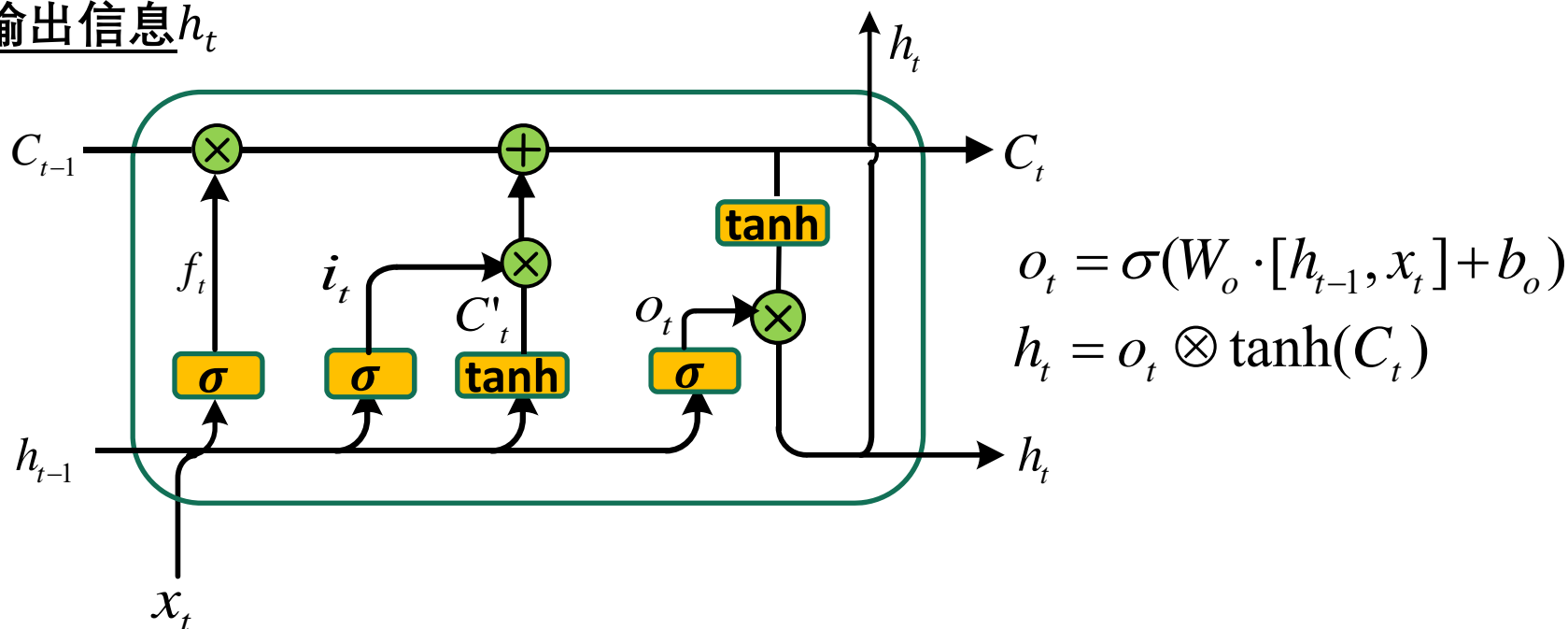
$$C_t = f_t \otimes C_{t-1} + i_t \otimes C'_t$$

- 包含三个部分：1) 首先，一个  $\tanh$  层创建一个新的候选值向量；2) 然后，sigmoid 层即输入门层  $i_t$  控制候选向量的哪些元素被更新；3) 新的信息被加入到状态中。
- 语言模型例子： ... **Germa**y .... I grew up in **France**... I speak fluent \_\_\_\_。



## LSTM的计算过程(3)

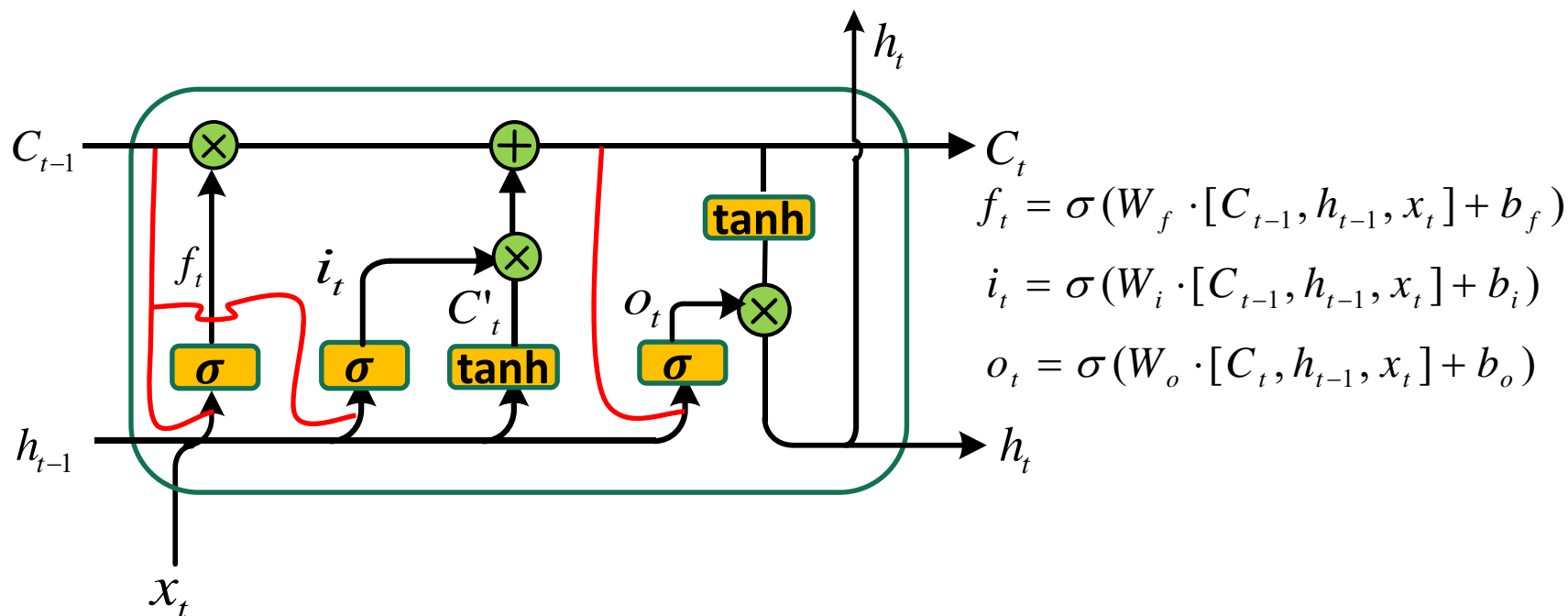
输出信息  $h_t$



- 通过 sigmoid 层，来确定将输出哪些信息，即得到输出门  $o_t$ 。
- 然后把长期状态通过 tanh 层进行处理，然后将其与经输出门过滤后的信息相乘，得到要输出的  $h_t$ 。

# LSTM 的变体(1)

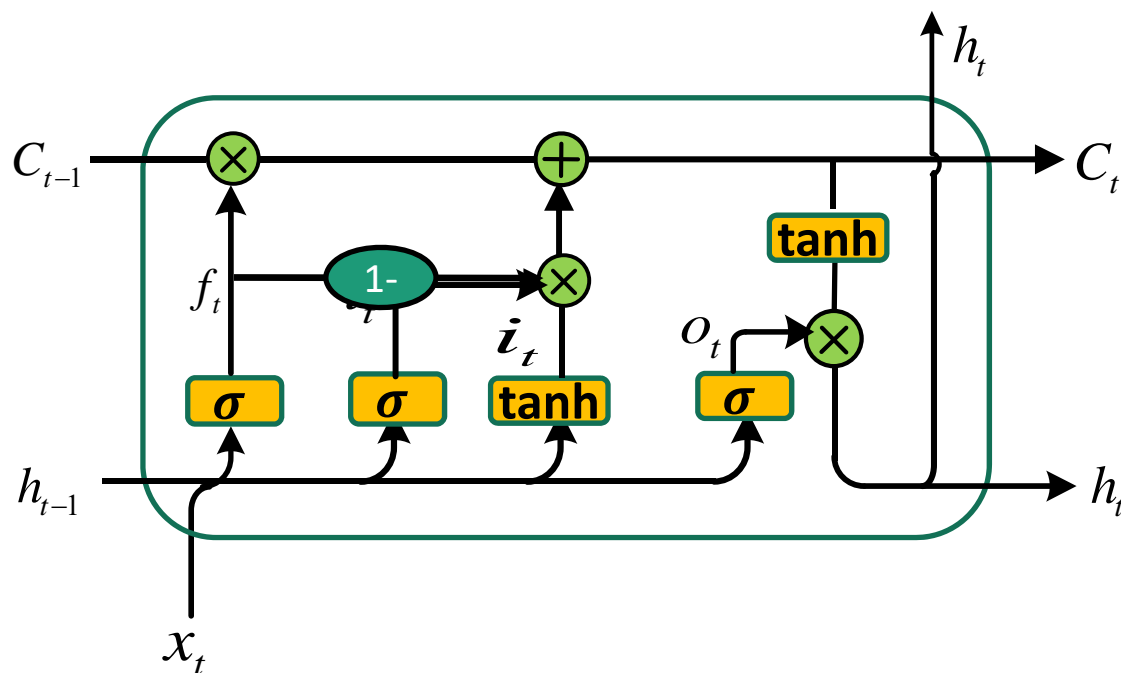
- 由 [Gers & Schmidhuber \(2000\)](#) 提出，增加了“peephole connection”。门层也接受长期状态的输入。



Gers, F. A., & Schmidhuber, J. (2000). Recurrent Nets that Time and Count. *IEEE International Joint Conference on Neural Networks (Vol.3, pp.189-194 vol.3)*. IEEE.

## LSTM 的变体(2)

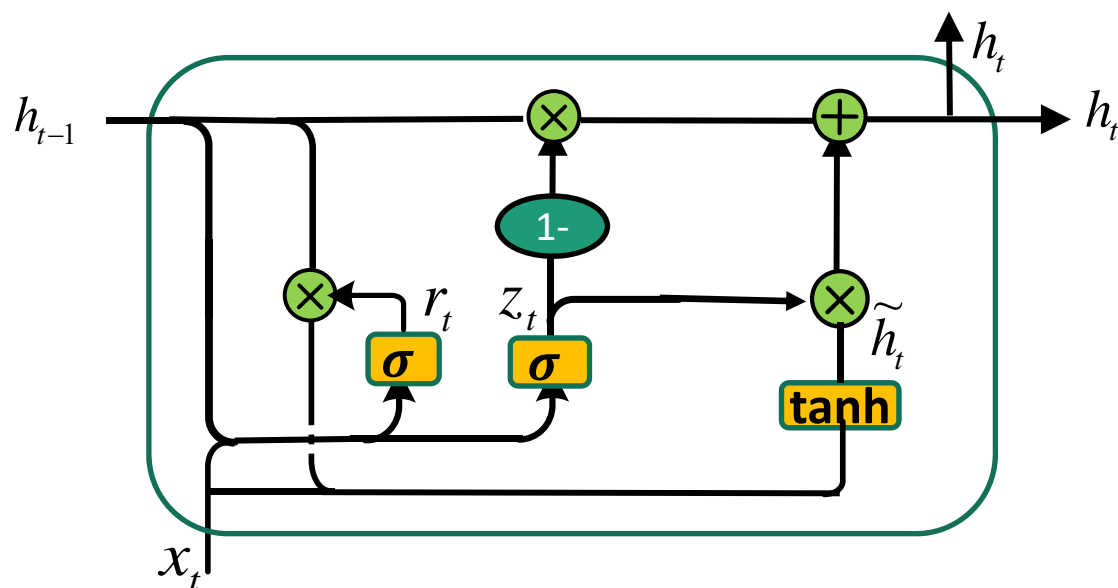
- 耦合(coupled)遗忘和输入单元：将遗忘和新记忆两个过程耦合，即只遗忘那些有新元素来填充的元素。



$$C_t = f_t \otimes C_{t-1} + i_t \otimes C'_t$$
$$C_t = f_t \otimes C_{t-1} + (1 - f_t) \otimes C'_t$$

# LSTM 的变体(3)--GRU

- 即Gated Recurrent Unit [Cho, et al. \(2014\)](#)，混合了长期状态和隐藏状态。



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

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

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

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

- GRU只有两个门: 重置 (reset) 门 $r$ 和更新 (update) 门 $z$ ，取消了LSTM中的output门。 $r$ 和 $z$ 共同控制了如何从之前的隐藏状态 ( $h_{t-1}$ ) 计算获得新的隐藏状态 ( $h_t$ )。

Cho K, Van Merriënboer B, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. Eprint Arxiv, 2014.

# Thanks!





# 双向循环神经网络和注意力机制

(Bidirectional RNN and Attention Mechanism)

刘远超

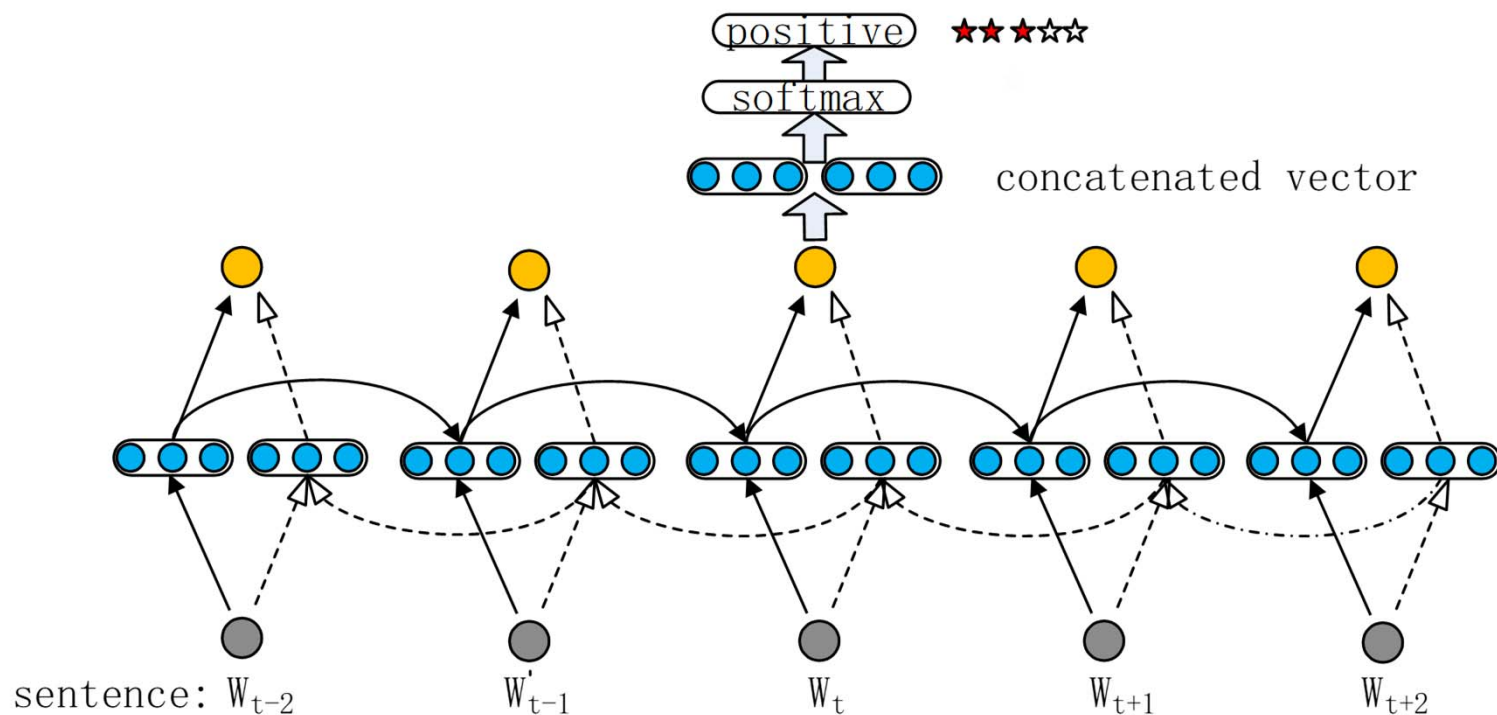
哈尔滨工业大学

计算机科学与技术学院

# 双向RNN(Bidirectional RNNs)

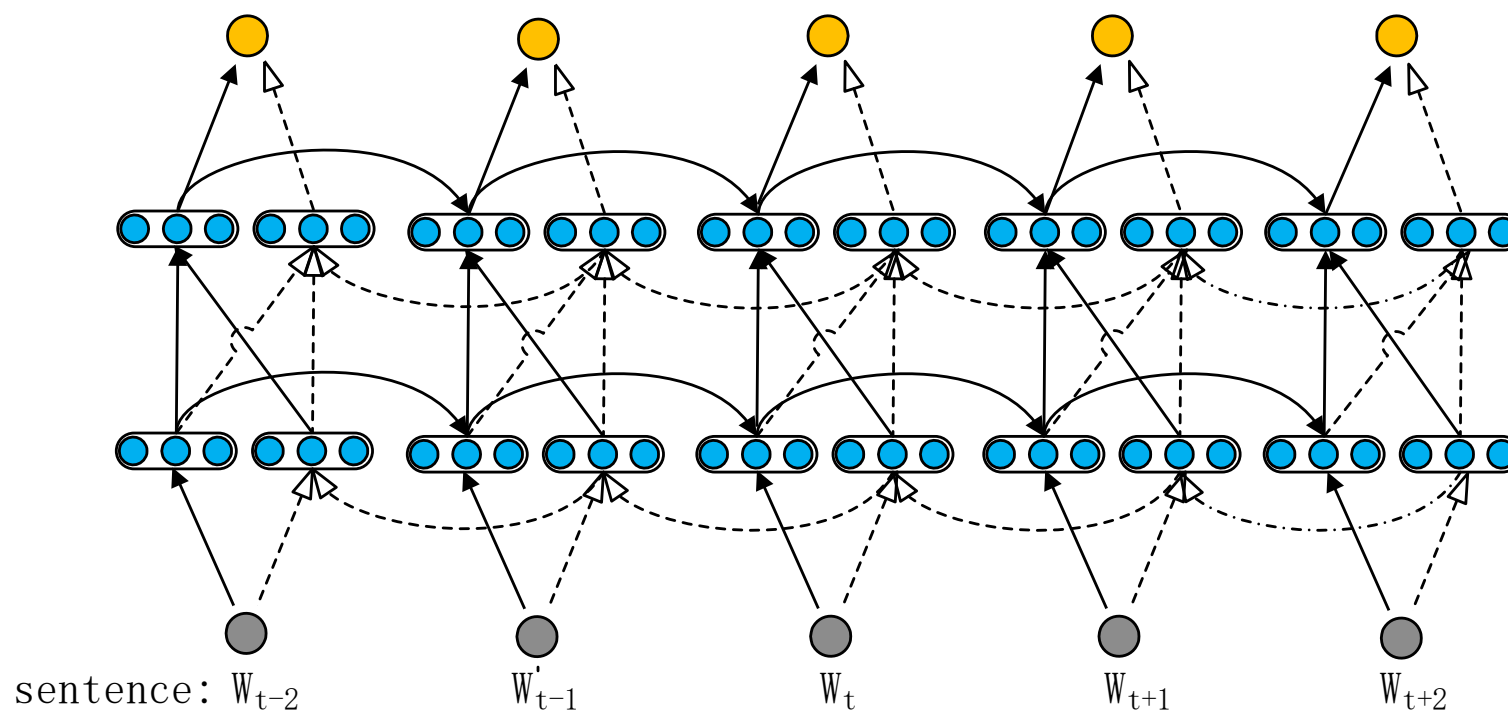
- 在很多应用中，当前步，即第 $t$ 步的输出与前面的序列和后面的序列都有关。

例如：“我喜欢宠物，家里养了一（zhi）可爱的小花猫。”，则括号内填“只”还是“支”？



Schuster M, Paliwal K K. **Bidirectional recurrent neural networks**[J]. *Signal Processing, IEEE Transactions on*, 1997, 45(11): 2673-2681.

# 深层双向RNN(Deep Bidirectional RNNs)



Graves A, Mohamed A R, Hinton G. Speech Recognition with Deep Recurrent Neural Networks[J]. Acoustics Speech & Signal Processing . icassp. international Conference on, 2013:6645 - 6649.



# 注意力模型 (Attention model)

- 注意力模型（机制）是受到了人类注意力机制的启发。

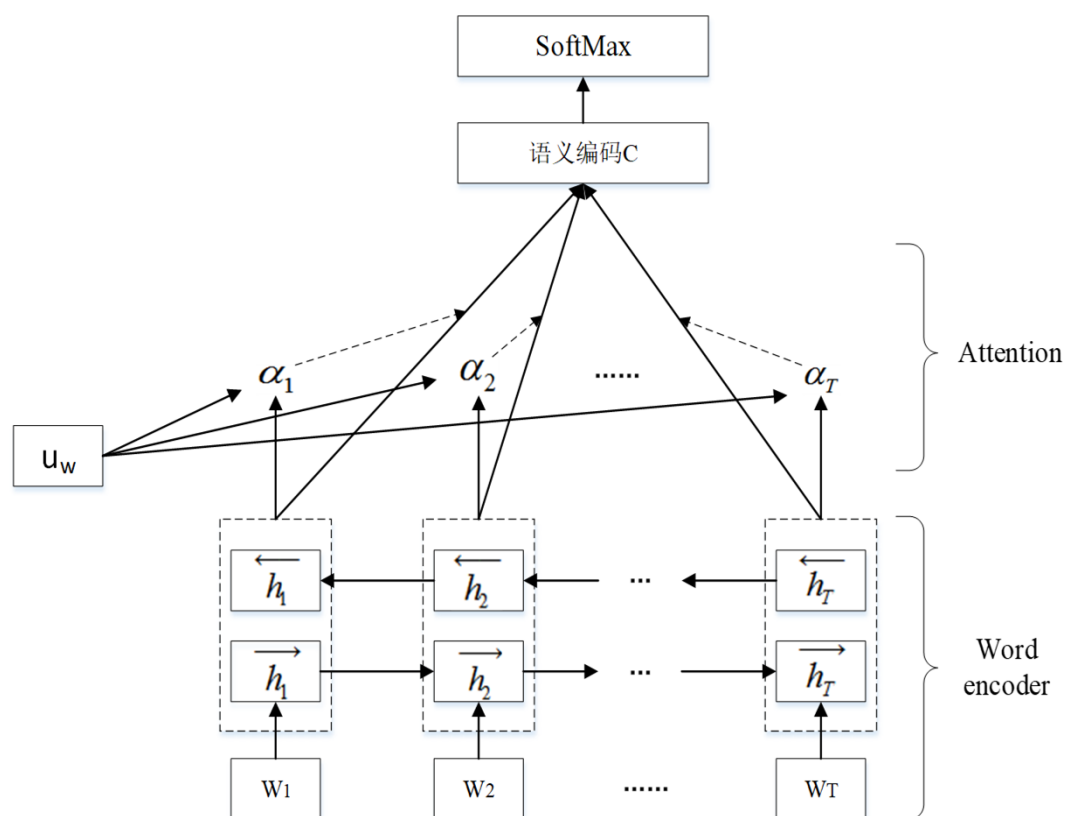


- Google mind团队<sup>1</sup>在RNN模型上使用了attention机制进行图像分类。
- 后来Bahdanau等人<sup>2</sup>将attention机制应用到NLP领域中。如问答系统、自动文摘、文本分类等。

1. Mnih, V., Heess, N., Graves, A., et al. Recurrent models of visual attention. NIPS 2014

2. Bahdanau, D., Cho, K., & Bengio, Y. Neural machine translation by jointly learning to align and translate. NIPS 2014

# 注意力模型基本原理



$$1) \theta_t = \tanh(W_w h_t + b_w)$$

其中  $h_t = [\vec{h}_t, \overleftarrow{h}_t]$ ,  
 $W_w, b_w$  为权重参数和偏置参数

$$2) \alpha_t = \frac{\exp(\theta_t^T u_w)}{\sum_{t=1}^T \exp(\theta_t^T u_w)}$$

其中  $u_w$  为上下文向量  
 $\alpha_t$  为不同词的注意力概率分布

$$3) C = \sum_{t=1}^T \alpha_t h_t$$

其中  $C$  为带有注意分布的语义编码

<https://github.com/richliao/textClassifier>

Yang Z, Yang D, Dyer C, et al. Hierarchical attention networks for document classification[C]//Proceedings of NAACL-HLT. 2016: 1480-1489.

# Thanks!

