

# 5-2 Recurrent Neural Network II

Zhonglei Wang

WISE and SOE, XMU, 2025

# Contents

1. GRU

2. LSTM

# GRU

1. 其是 Gated Recurrent Unit (Cho et al., 2014) 的缩写
2. 可通过重置门 (reset gate) 和更新门 (update gate)  
两个门控机制捕捉长依赖关系

[Cho, K., van Merriënboer, B., Bahdanau, D. and Bengio, Y.(2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.]

# 重置门 (Reset gate)

1. 重置门 (reset gate) 控制我们在多大程度上忘记过去

2. 重置门 (reset gate) 的相关计算如下:

- $\Gamma_r^{<i>} = \sigma \left\{ \mathbf{W}_r \left( \mathbf{x}^{<i>T}, \mathbf{a}^{<i-1>T} \right)^T + \mathbf{b}_r \right\}$ : 与  $\mathbf{a}^{<i-1>}$  具有相同维度
- $\tilde{\mathbf{a}}^{<i>} = \tanh \left\{ \mathbf{W} \left( \mathbf{x}^{<i>T}, (\Gamma_r^{<i>} \circ \mathbf{a}^{<i-1>})^T \right)^T + \mathbf{b} \right\}$
- $\Gamma_r^{<i>}$  控制我们得到  $\tilde{\mathbf{a}}^{<i>}$  时, 多大程度地 “忘记过去”

3. 模型参数:

- $\mathbf{W}_r, \mathbf{b}_r$  用于计算更新门  $\Gamma_r^{<i>}$
- $\mathbf{W}, \mathbf{b}$  用于计算  $\tilde{\mathbf{a}}^{<i>}$

# 更新门 (Update gate)

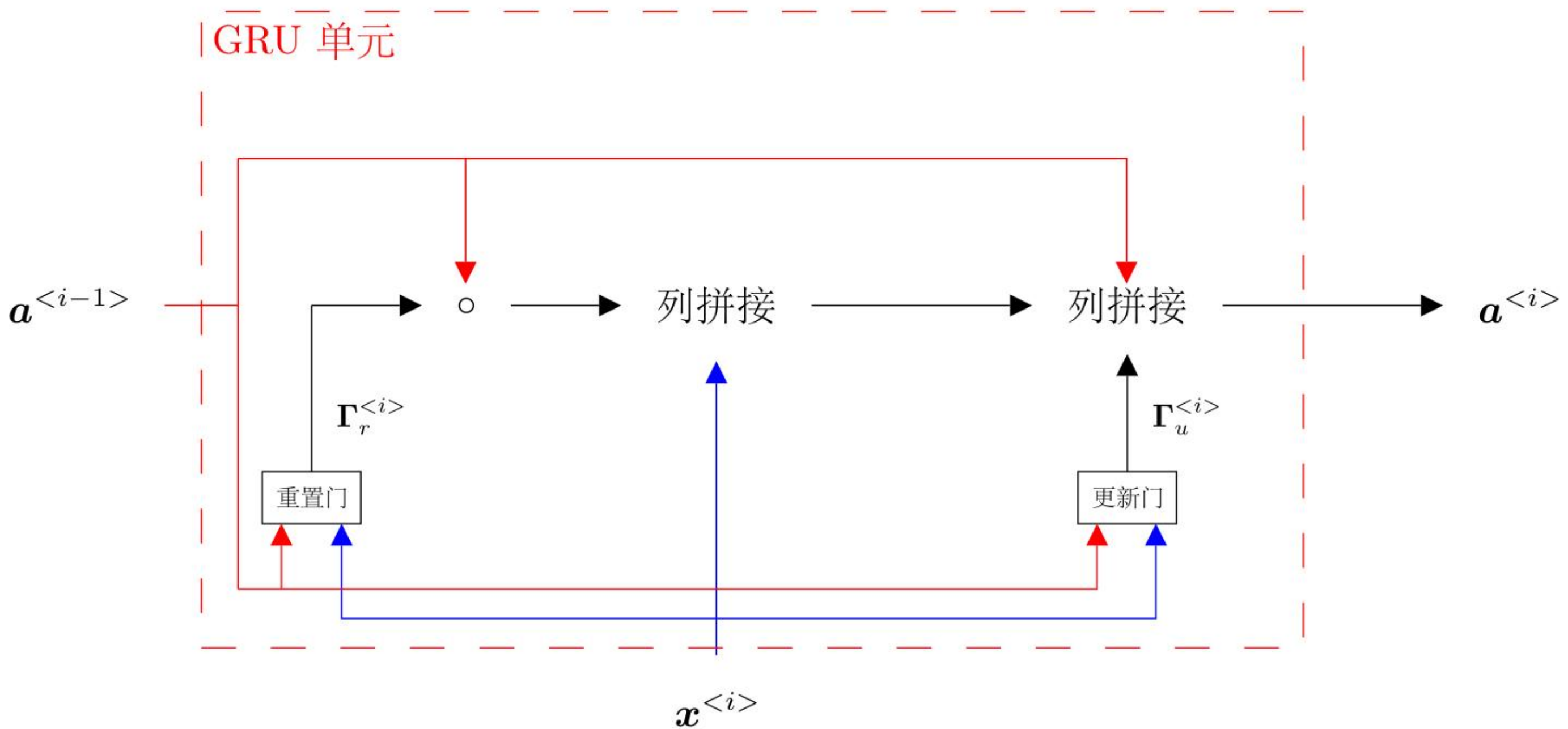
1. 更新门 (update gate) 用于得到  $\mathbf{a}^{<i>}$
2. 更新门 (update gate) 的相关计算如下:

- $\mathbf{\Gamma}_u^{<i>} = \sigma \left\{ \mathbf{W}_u \left( \mathbf{x}^{<i>\top}, \mathbf{a}^{<i-1>\top} \right)^\top + \mathbf{b}_u \right\}$  : 与  $\mathbf{a}^{<i-1>}$  具有相同维度
- $\mathbf{a}^{<i>} = (1 - \mathbf{\Gamma}_u^{<i>}) \circ \mathbf{a}^{<i-1>} + \mathbf{\Gamma}_u^{<i>} \circ \tilde{\mathbf{a}}^{<i>}$
- $\mathbf{A} \circ \mathbf{B}$ : 两个矩阵  $\mathbf{A}$  和  $\mathbf{B}$  的 Hadamard 乘积 (对应位置元素相乘)

3. 模型参数:

- $\mathbf{W}_u, \mathbf{b}_u$  用于计算  $\mathbf{\Gamma}_u^{<i>}$

# 流程图





# LSTM

1. 是 Long Short-Term Memory unit 的缩写
2. 通过更新门 (update gate)、遗忘门 (forget gate) 以及输出门 (output gate) 等门控机制捕捉不同位置的相关关系
3. 首先, 产生一个“备用”激活值

$$\tilde{\mathbf{c}}^{<i>} = \tanh \left\{ \mathbf{W}_c \left( \mathbf{x}^{<i>^T}, \mathbf{a}^{<i-1>^T} \right)^T + \mathbf{b}_c \right\}$$

[部分内容来自于吴恩达教授在 Coursera 上的课程 Deep Learning Specialization]

# LSTM

1. 输入门 (input gate) 的相关计算如下:

$$\mathbf{\Gamma}_i^{<i>} = \sigma \left\{ \mathbf{W}_i \left( \mathbf{x}^{<i>^T}, \mathbf{a}^{<i-1>^T} \right)^T + \mathbf{b}_i \right\} : \text{与 } \mathbf{c}^{<i-1>} \text{ 具有相同维度}$$

2. 遗忘门 (forget gate) 的相关计算如下:

$$\mathbf{\Gamma}_f^{<i>} = \sigma \left\{ \mathbf{W}_f \left( \mathbf{x}^{<i>^T}, \mathbf{a}^{<i-1>^T} \right)^T + \mathbf{b}_f \right\} : \text{与 } \mathbf{c}^{<i-1>} \text{ 具有相同维度}$$

3. 以上两个门控机制用于得到一个“激活值”

$$\mathbf{c}^{<i>} = \mathbf{\Gamma}_i^{<i>} \circ \tilde{\mathbf{c}}^{<i>} + \mathbf{\Gamma}_f^{<i>} \circ \tilde{\mathbf{c}}^{<i-1>}$$

4. 关于前面序列的“记忆”可被遗忘门 (forget gate)  $\mathbf{\Gamma}_f^{<i>}$  “抹去”



# LSTM

1. 输出门 (output gate) 的相关计算如下:

$$\mathbf{\Gamma}_o^{<i>} = \sigma \left\{ \mathbf{W}_o \left( \mathbf{x}^{<i>^T}, \mathbf{a}^{<i-1>^T} \right)^T + \mathbf{b}_o \right\} : \text{与 } \mathbf{c}^{<i-1>} \text{ 具有相同维度}$$

2. 输出门 (output gate) 用于得到

$$\mathbf{a}^{<i>} = \mathbf{\Gamma}_o^{<i>} \circ \mathbf{c}^{<i>}$$

3. 因此, 记忆可以被输出门 (output gate)  $\mathbf{\Gamma}_o^{<i>}$  “进一步控制”

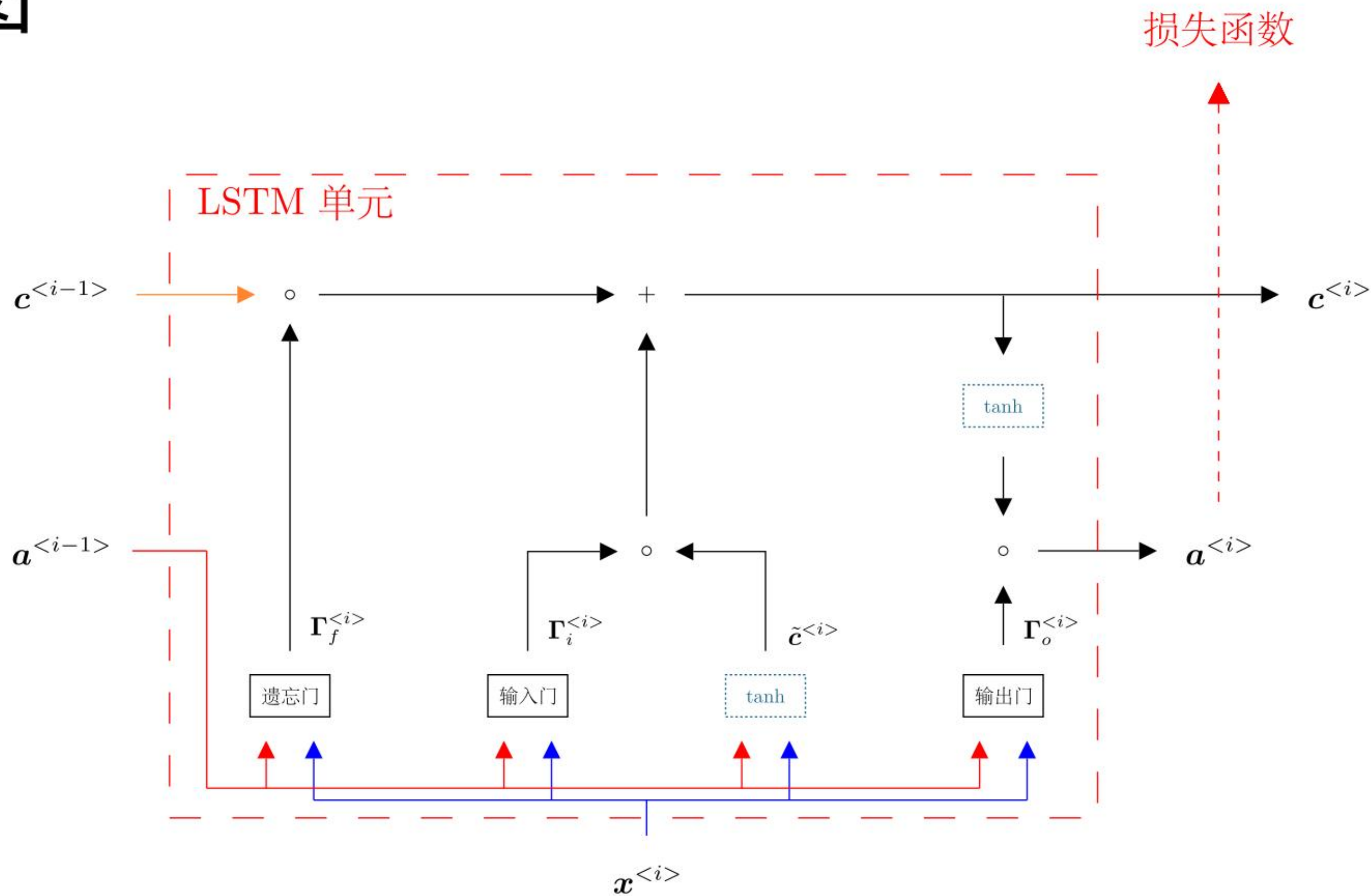
4. 初始化  $\mathbf{a}^{<0>} = \mathbf{c}^{<0>} = 0$

# LSTM

## 1. 模型参数包括:

- $W_c, b_c$  用于计算“备用”激活值
- $W_i, b_i$  用于输入门 (input gate) 相关计算
- $W_f, b_f$  用于遗忘门 (forget gate) 相关计算
- $W_o, b_o$  用于输出门 (output gate) 相关计算

# 流程图



# 深层循环神经网络

1. 我们可通过纵向叠加已经讨论过的“循环结构”增加模型深度