19 循环神经网络

概要

- 循环神经网络
  - 实现 RNN 语言模型
  - 截断通过时间反向传播
- 门控循环单元(GRU)
- 长短期记忆网络(LSTM)

循环神经网络

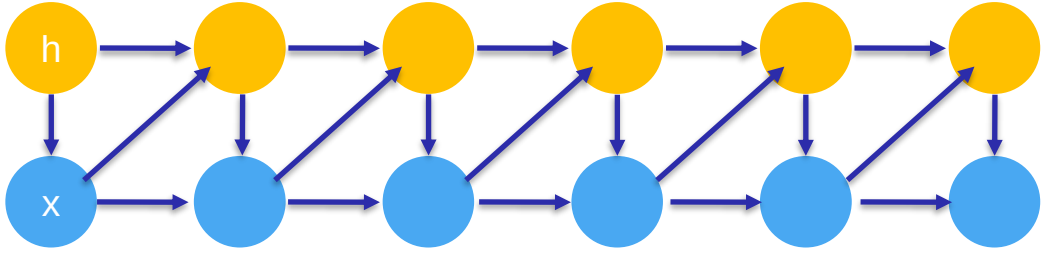
潜变量模型

- 隐含状态h_t总结了有关过去所有的相关信息

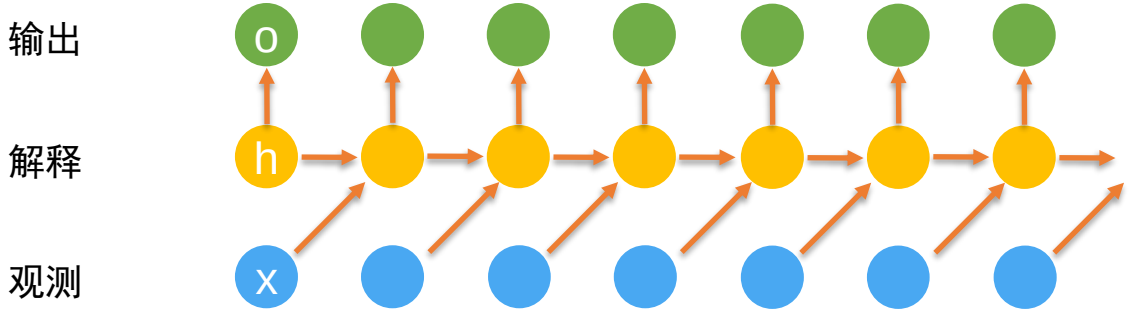
$$h_t = f(x_1, \dots x_{t-1}) = f(h_{t-1}, x_{t-1})$$

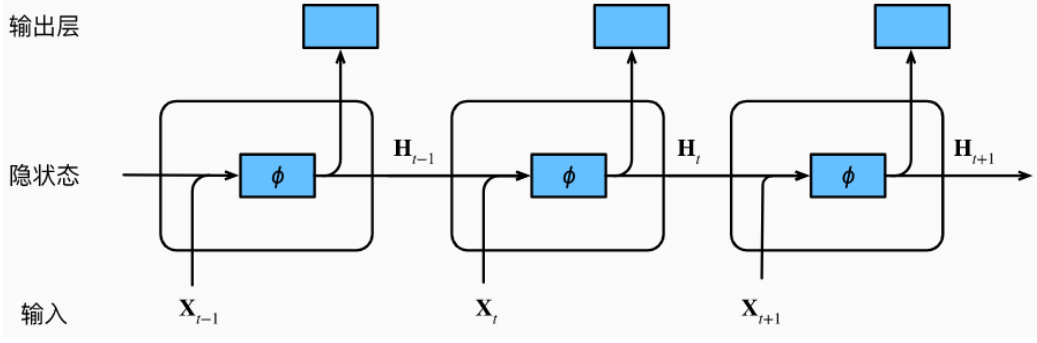
$$p(h_t|h_{t-1}, x_{t-1}),$$

$$p(x_t|h_t, x_{t-1})$$



循环神经网络

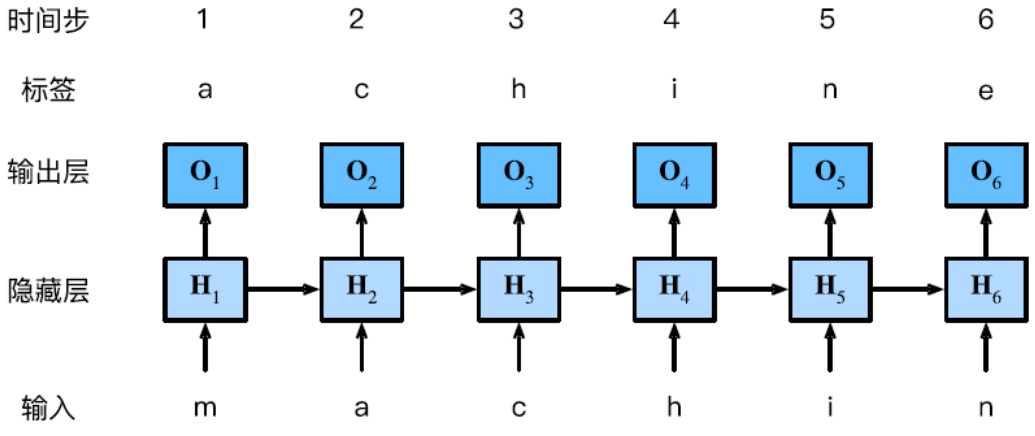




- 隐含状态更新$\mathbf{h}_t = \phi(\mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{W}_{hx}\mathbf{x}_{t-1} + \mathbf{b}_h)$
- 输出更新$\mathbf{o}_t = \phi(\mathbf{W}_{ho}\mathbf{h}_t + \mathbf{b}_o)$
- 计算损失

基于循环神经网络的字符级语言模型

- 目标
  - 根据过去的和当前的词元预测下一个词元，因此将原始序列移位一个词元作为标签
  - Bengio等首先提出使用神经网络进行语言建模
- 字符级语言[illegible]元化为字符/非单词



困惑度(perplexity)

- 一个好的语言模型
  - 能够用高度准确的词元来预测接下来会看到什么
- 衡量一个语言模型的好坏，可以用平均交叉熵

$$\pi = \frac{1}{n}\sum_{i=1}^{n} -\log p(x_t \mid x_{t-1}, \dots)$$

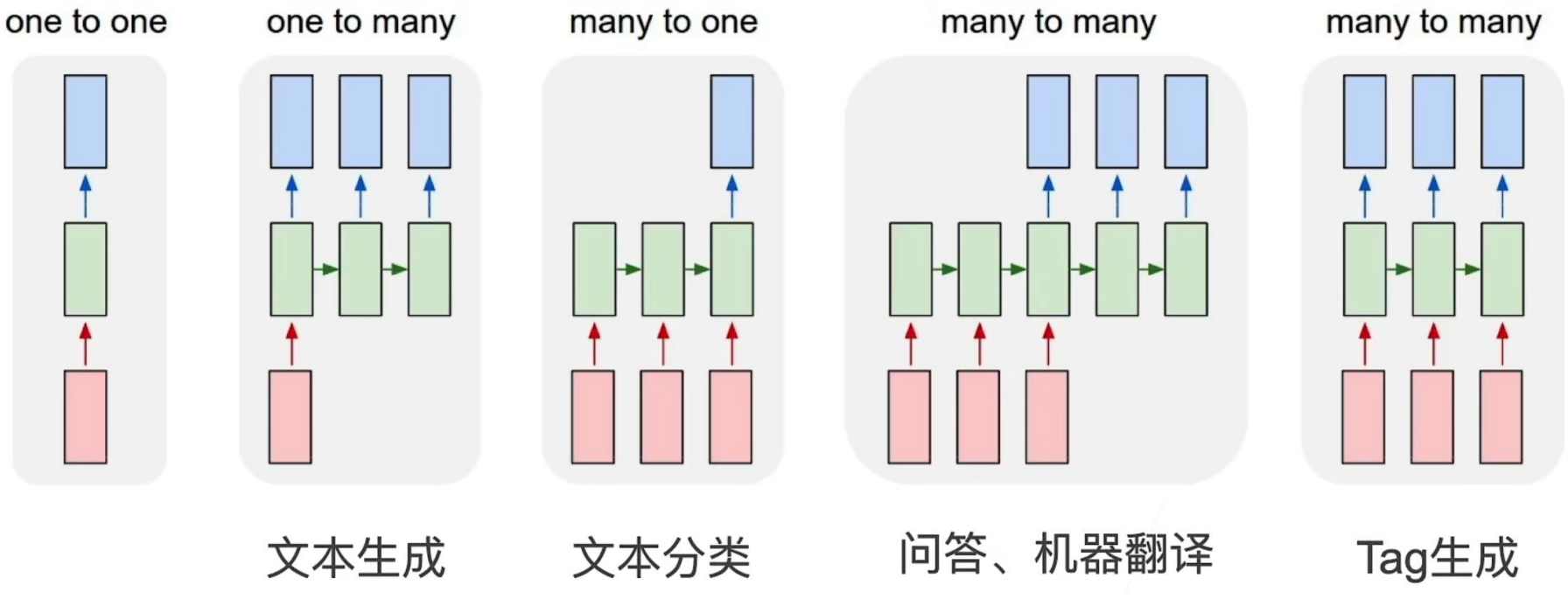
  - p 是语言模型的预测概率, x_t 是真实词
- 历史原因，NLP使用困惑度$\exp(\pi)$来衡量。困惑度
  - 平均每次可能选项
  - 最好的理解是“下一个词元的实际选择数的调和平均数”
  - 1表示完美, 无穷大为最差情况

梯度裁剪

- 迭代中计算 T 个时间步上的梯度, 在反向传播过程中产生长度为 $O(T)$ 的矩阵乘法链, 导致数值不稳定
- 梯度裁剪能有效预防梯度爆炸
  - 降低 η 的学习率
  - 一个流行的替代方案：通过将梯度 g 投影回给定半径 (例如 θ) 的球来裁剪梯度 g
    - 如梯度长度超过θ, 那么投影回长度θ

$$\mathbf{g} \leftarrow \min\left(1, \frac{\theta}{\| \mathbf{g} \|}\right)\mathbf{g}$$

更多的应用


one to one
one to many
many to one
many to many
many to many
文本生成
文本分类
问答、机器翻译
Tag生成

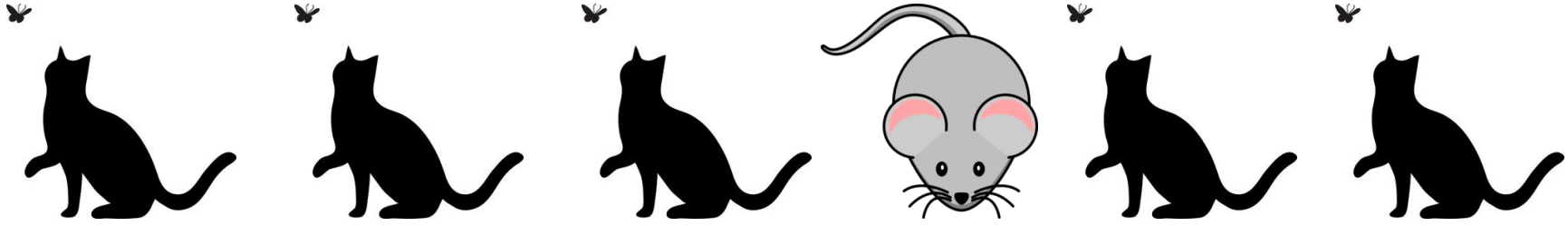
总结

- 循环神经网络的输出取决于当下输入和前一时间的隐变量
- 应用到语言模型中时, 循环神经网络根据当前词预测下一次时刻词
- 通常使用困惑度来衡量语言模型的好坏

门控循环单元(GRU)

在一个序列的注意力

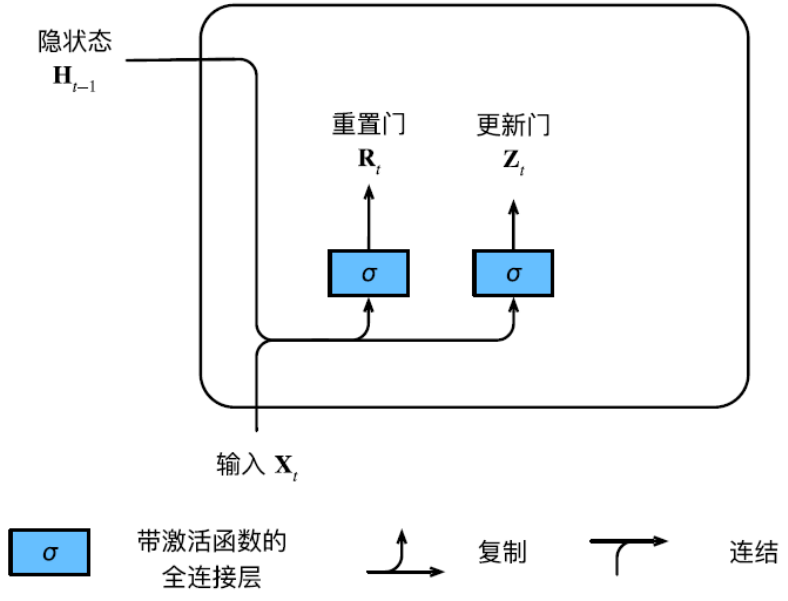
- 并非所有元素都具有同等意义



- 想只记住相关的元素【特别的元素】
  - 能关注的机制(更新门)【 0为与过去无关，1为只和过去有关】【更新的是候选的内容】
  - 能忘记的机制(重置门)【0忘记，1记住】【生成候选内容，不一定会更新】
- 如何实现？
  - 对应的控制变量
  - 隐状态，以及候选隐状态！

门控循环单元

- $\boldsymbol{R}_t = \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xr} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hr} + \boldsymbol{b}_r)$
- $\boldsymbol{Z}_t = \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xz} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hz} + \boldsymbol{b}_z)$
- 重置门：
  - 隐状态权重，控制候选隐状态的生成
- 更新门
  - 输入权重，控制最终隐状态的生成
- 【注】
  - 引入隐状态，则与其相关状态包含
    - 过去隐状态
    - 候选隐状态
      - 过去隐状态
      - 输入

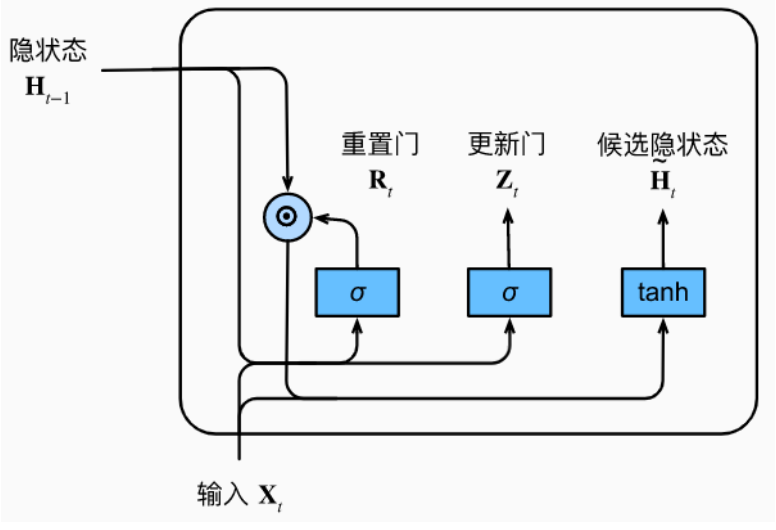


候选隐含状态

- 候选隐状态 (candidate hidden state) $\tilde{\mathbf{H}}_t \in \mathbb{R}^{n \times h}$

$$\tilde{\mathbf{H}}_t = \tanh(\mathbf{X}_t \mathbf{W}_{xh} + (\mathbf{R}_t \odot \mathbf{H}_{t-1}) \mathbf{W}_{hh} + \mathbf{b}_h)$$

  - 符号 $\odot$ 是Hadamard积 (按元素乘积) 运算符
- 合成新的隐状态，作为新的隐含状态的输入分量
  - $\mathbf{R}_t$ 和 $\mathbf{H}_{t-1}$作为输入的多层感知机的结果

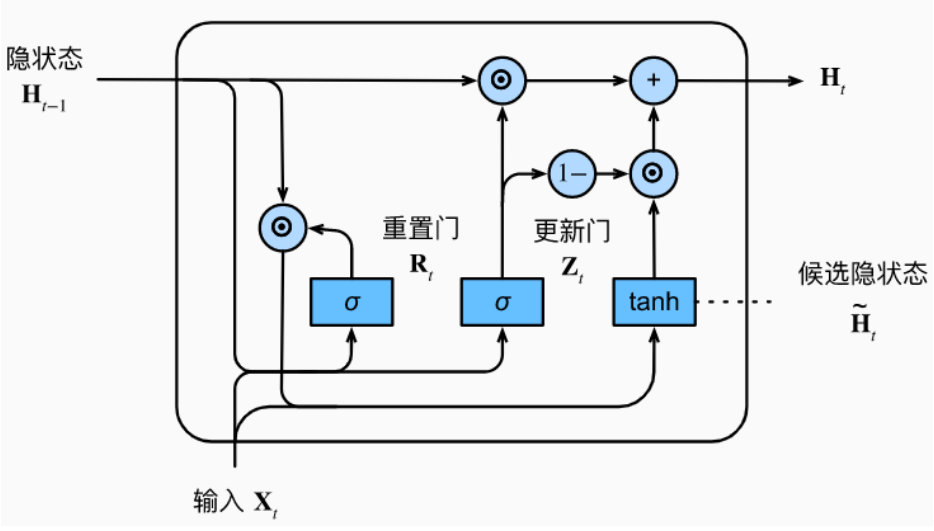


隐含状态

- $\mathbf{H}_{t-1}$ 和 $\tilde{\mathbf{H}}_t$ 之间进行按元素的凸组合更新隐含状态

$$\mathbf{H}_t = \mathbf{Z}_t \odot \mathbf{H}_{t-1} + (1 - \mathbf{Z}_t) \odot \tilde{\mathbf{H}}_t$$

  - 即根据过去状态，以及融合输入信息的候选隐状态来更新
    - 权重通过更新门控制，0为与过去无关，1为只和过去有关



门控循环单元(GRU)总结

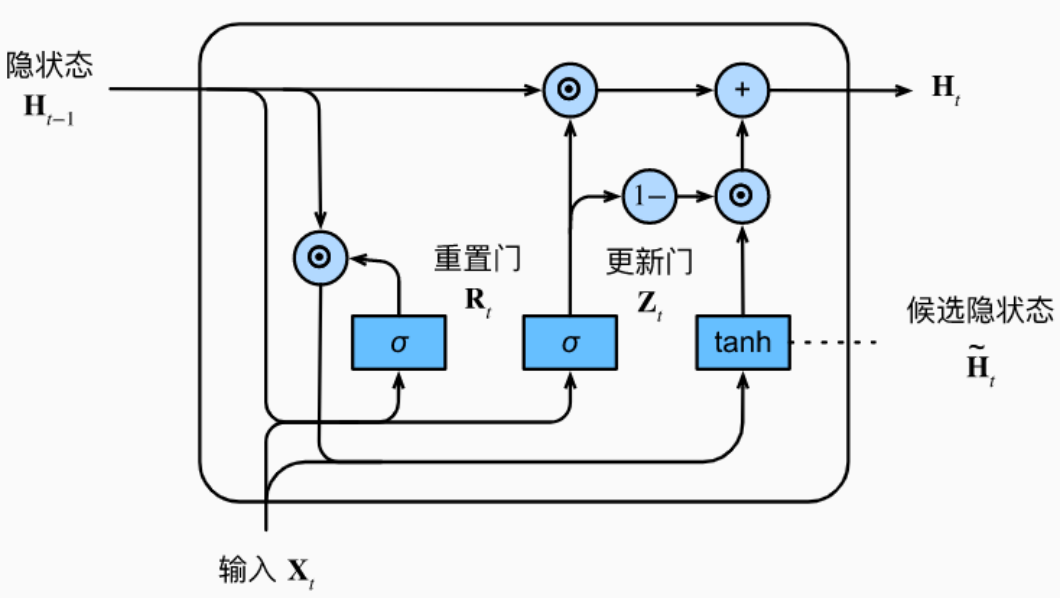
$$\boldsymbol{R}_t = \sigma(\boldsymbol{X}_t \boldsymbol{W}_{xr} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hr} + \boldsymbol{b}_r)$$

$$\boldsymbol{Z}_t = \sigma(\boldsymbol{X}_t \boldsymbol{W}_{xz} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hz} + \boldsymbol{b}_z)$$

- $$\tilde{\boldsymbol{H}}_t = \tanh(\boldsymbol{X}_t \boldsymbol{W}_{xh} + (\boldsymbol{R}_t \odot \boldsymbol{H}_{t-1}) \boldsymbol{W}_{hh} + \boldsymbol{b}_h)$$

$$\boldsymbol{H}_t = \boldsymbol{Z}_t \odot \boldsymbol{H}_{t-1} + (1 - \boldsymbol{Z}_t) \odot \tilde{\boldsymbol{H}}_t$$

$$O_t = \boldsymbol{H}_t \boldsymbol{W}_{hq} + \boldsymbol{b}_q$$

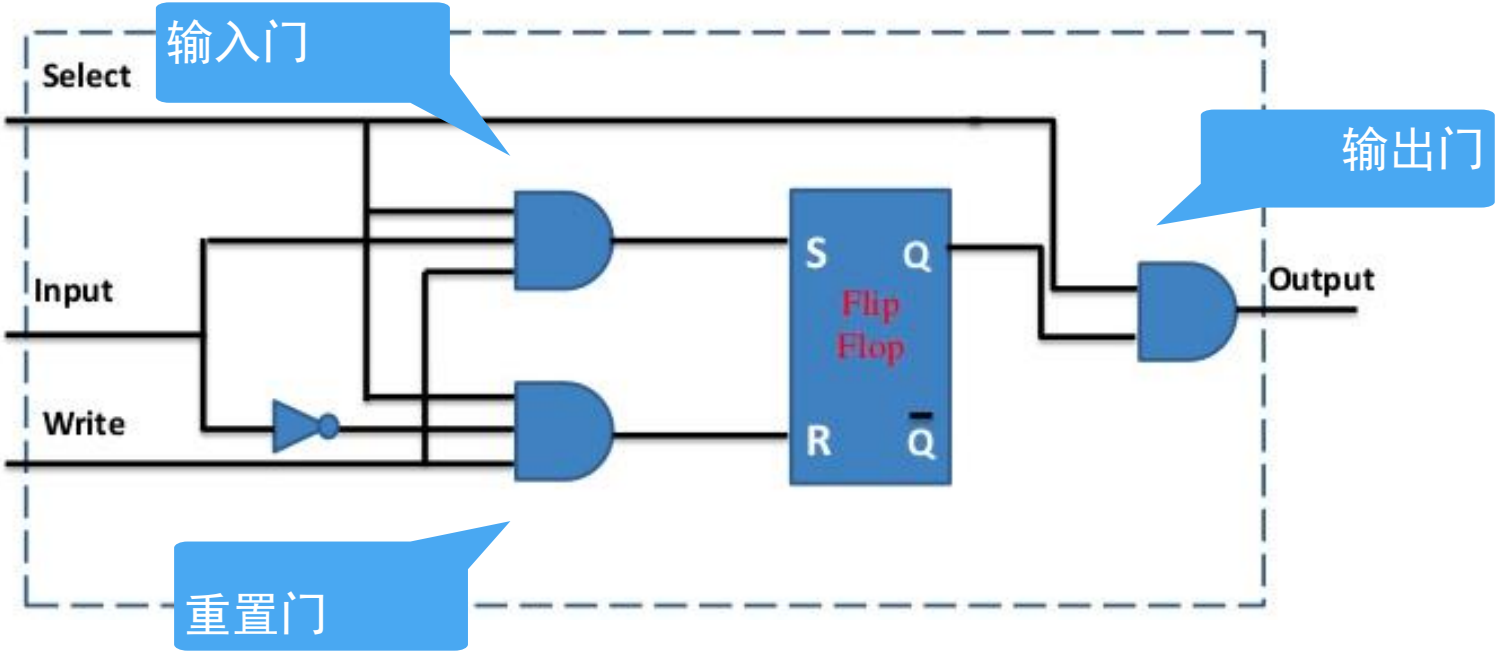


代码 ...

长短期记忆(LSTM)

电门联想

- 长短期记忆网络引入记忆元(memory cell)，或简称为单元(cell)



长短期记忆(LSTM)

- 忘记门
  - 重置单元的内容
- 输入门
  - 决定是否应忽略输入数据
- 输出门
  - 决定是不是使用隐状态

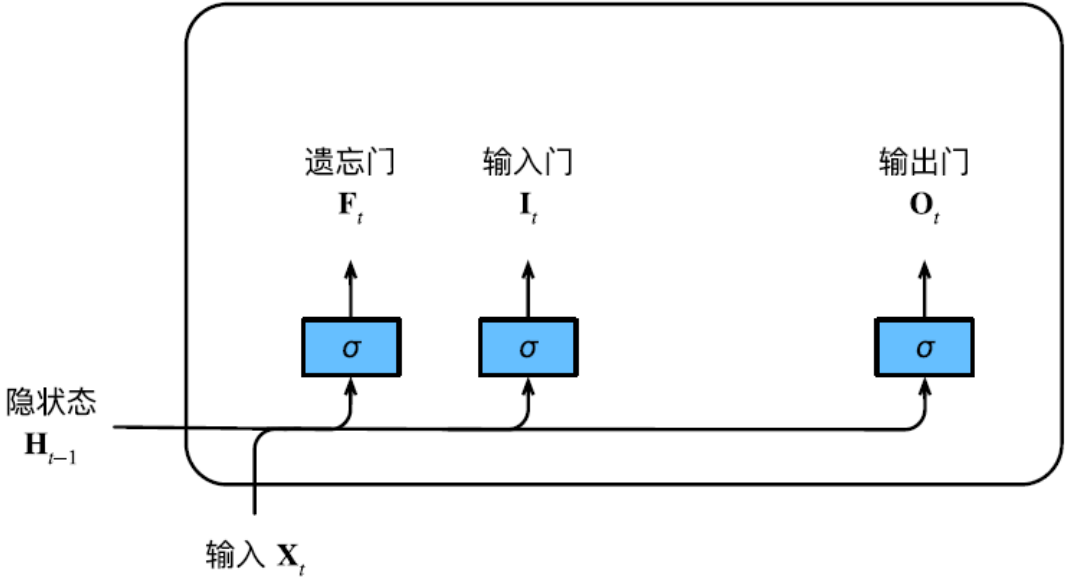
- 机制
  - 隐状态
  - 记忆和候选记忆【可理解为没有归一化的隐状态，因此这三个变量有交叉】

输入门、遗忘门和输出门

$$\boldsymbol{I}_t = \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xi} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hi} + \boldsymbol{b}_i)$$

$$\boldsymbol{F}_t = \sigma\left(\boldsymbol{X}_t\boldsymbol{W}_{xf} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hf} + \boldsymbol{b}_f\right)$$

$$\boldsymbol{O}_t = \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xo} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{ho} + \boldsymbol{b}_o)$$

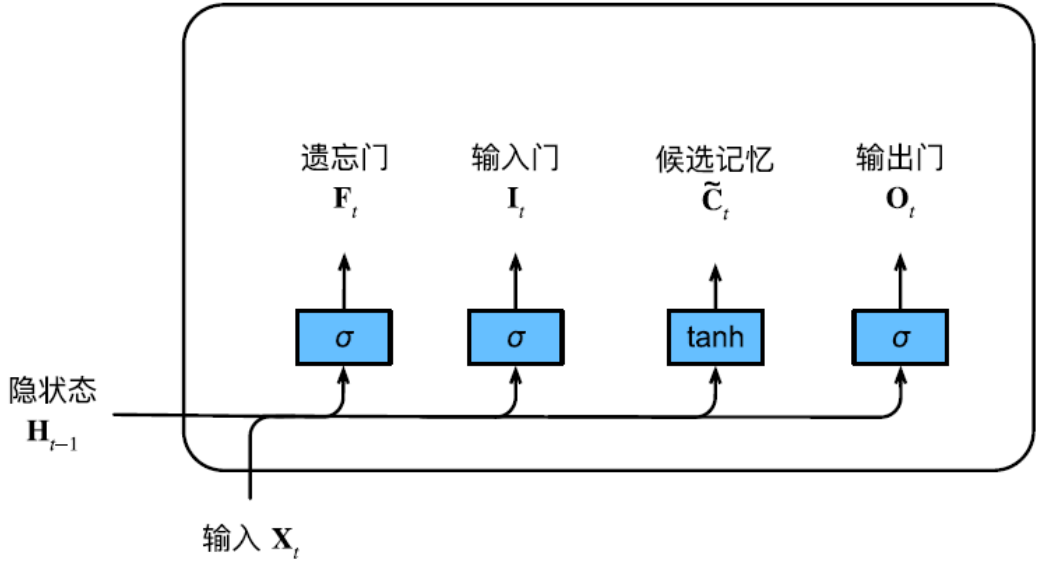


候选记忆元

- 候选记忆元 (candidate memory cell) $\tilde{\mathbf{C}}_t \in \mathbb{R}^{n\times h}$ 。函数的值范围为 $(-1,1)$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{X}_t\mathbf{W}_{xc} + \mathbf{H}_{t-1}\mathbf{W}_{hc} + \mathbf{b}_c),$$

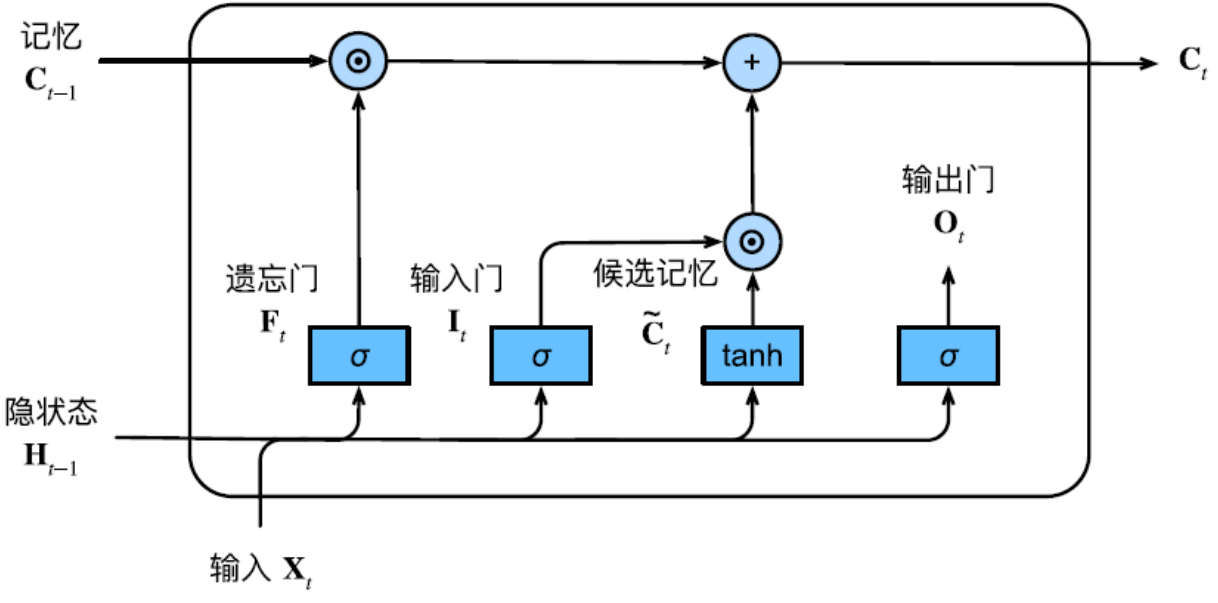
  - 其中 $\mathbf{W}_{xc} \in \mathbb{R}^{d\times h}$ 和 $\mathbf{W}_{hc} \in \mathbb{R}^{h\times h}$ 是权重参数, $\mathbf{b}_c \in \mathbb{R}^{1\times h}$ 是偏置参数。



记忆元

- 门控循环单元中有机制控制输入和遗忘 (或跳过)。在LSTM中, 输入门 $\mathbf{I}_t$ 控制采用多少来自 $\tilde{\mathbf{C}}_t$ 的新数据，而遗忘门 $\mathbf{F}_t$ 控制保留多少过去的记忆元 $\mathbf{C}_{t-1} \in \mathbb{R}^{n \times h}$ 的内容：

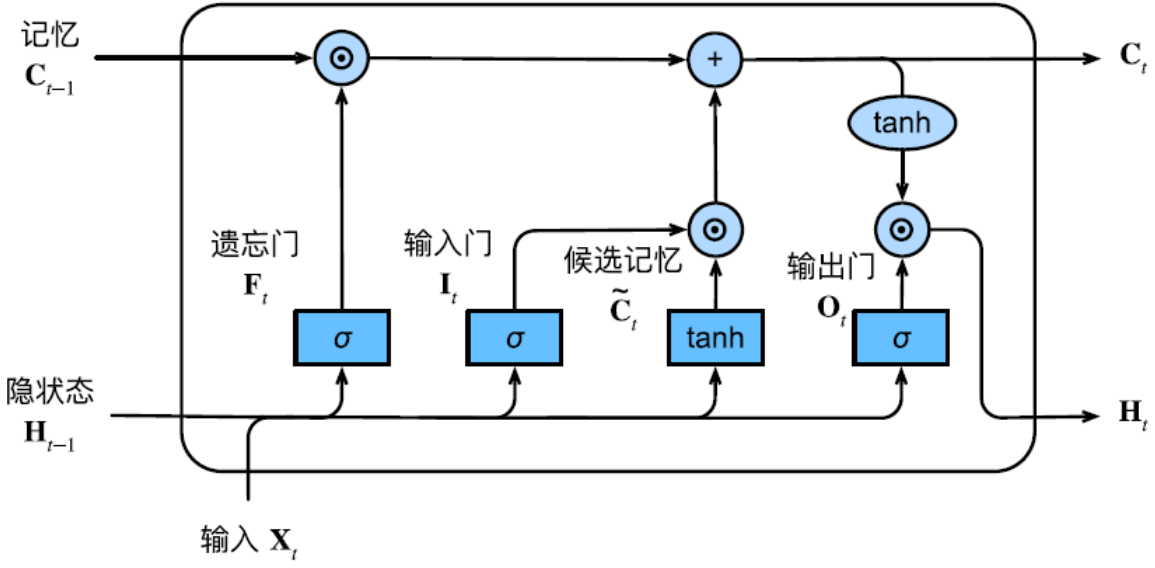
$$\mathbf{C}_t = \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \tilde{\mathbf{C}}_t$$



隐状态

- 隐状态 $\mathbf{H}_t \in \mathbb{R}^{n \times h}$

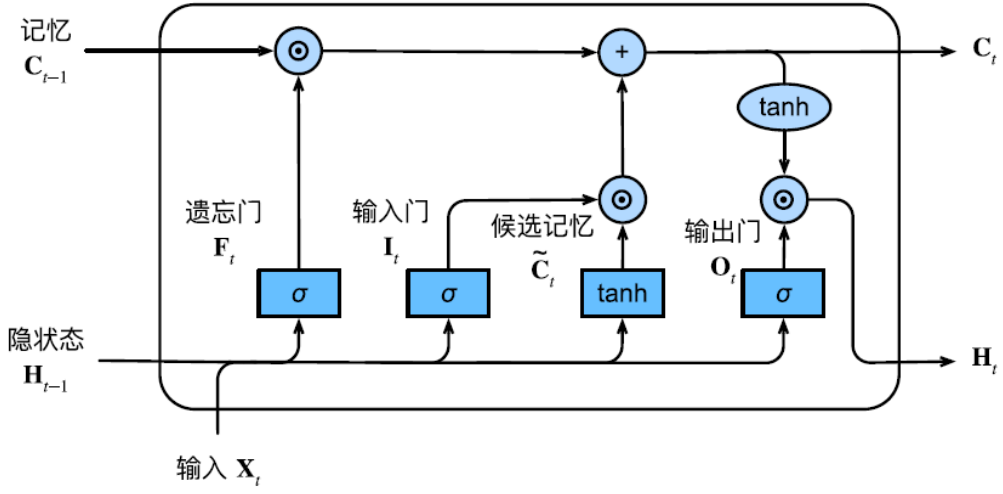
$$\mathbf{H}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t)$$



长短期记忆(LSTM)总结

$$
\begin{aligned}
\boldsymbol{I}_t &= \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xi} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hi} + \boldsymbol{b}_i)\\
\boldsymbol{F}_t &= \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xf} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hf} + \boldsymbol{b}_f)\\
\boldsymbol{O}_t &= \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xo} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{ho} + \boldsymbol{b}_o)\\
\tilde{\boldsymbol{C}}_t &= \tanh(\boldsymbol{X}_t\boldsymbol{W}_{xc} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hc} + \boldsymbol{b}_c)\\
\boldsymbol{C}_t &= \boldsymbol{F}_t \odot \boldsymbol{C}_{t-1} + \boldsymbol{I}_t \odot \tilde{\boldsymbol{C}}_t\\
\boldsymbol{H}_t &= \boldsymbol{O}_t \odot \tanh(\boldsymbol{C}_t)
\end{aligned}
$$

- 输出
  - 输入
  - 隐状态
    - 输出
    - 记忆
      - 过去记忆(遗忘门控制)
      - 候选记忆(输入门控制)
        - 过去隐状态，输入



- 遗忘门和输入门，输出门
  - 生成信息用于参数权重控制
- 长短期记忆网络的隐藏层输出包括“隐状态”和“记忆元”。只有隐状态会传递到输出层，而记忆元完全属于内部信息。