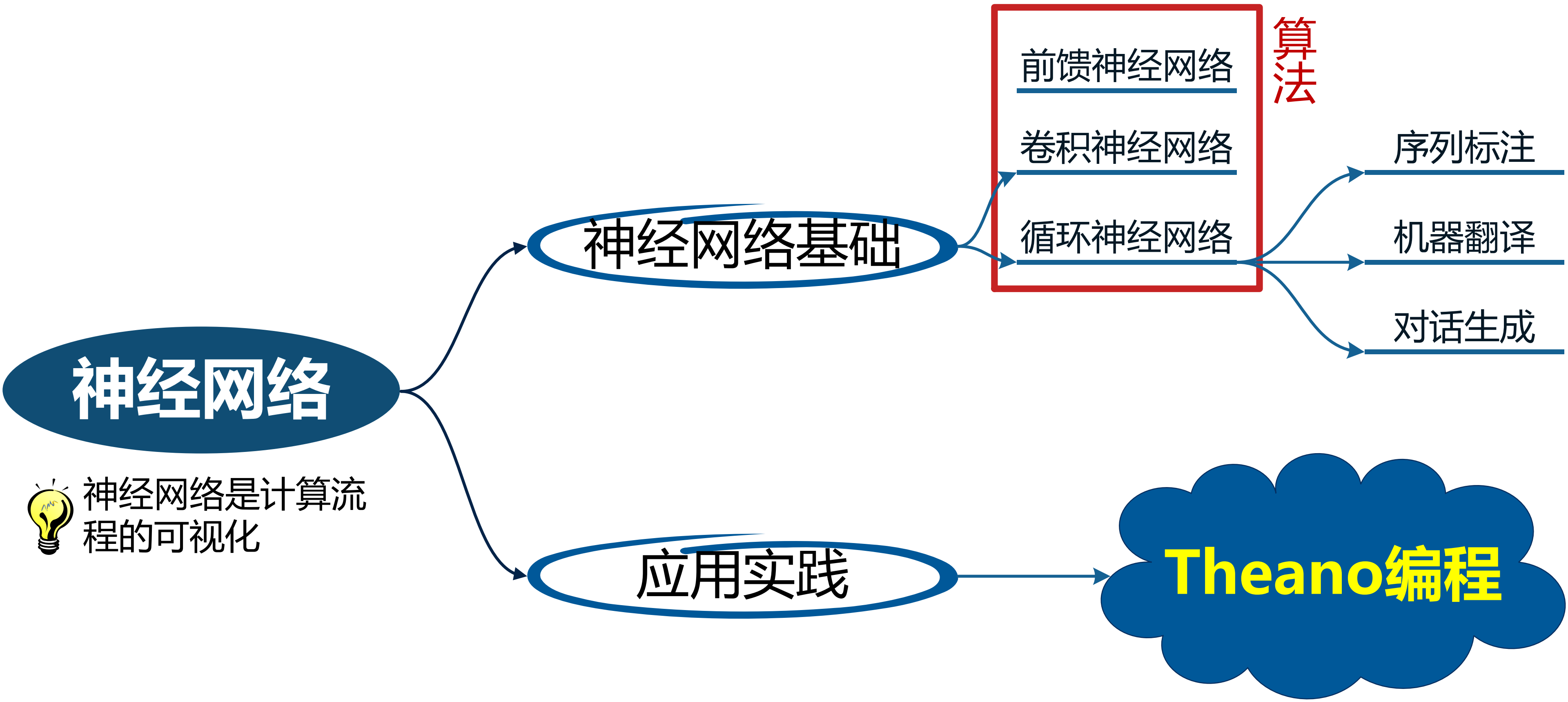
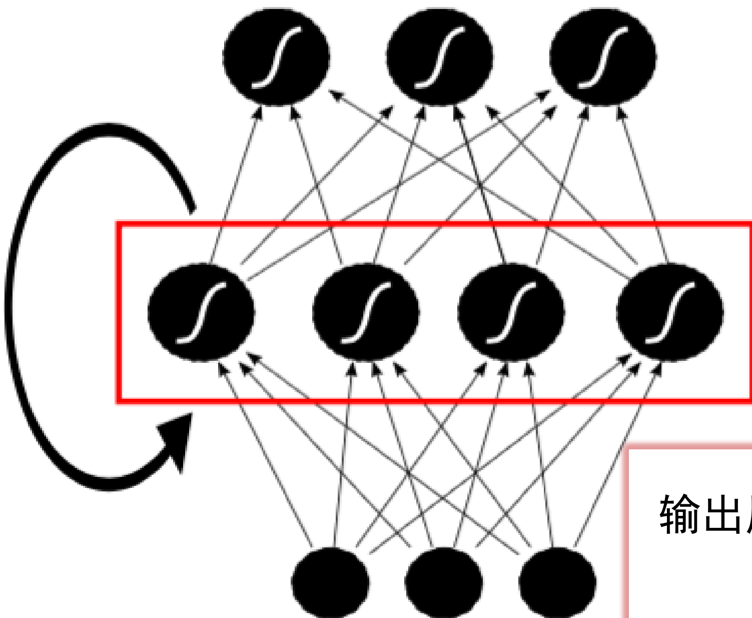
课程专题三：神经网络方法

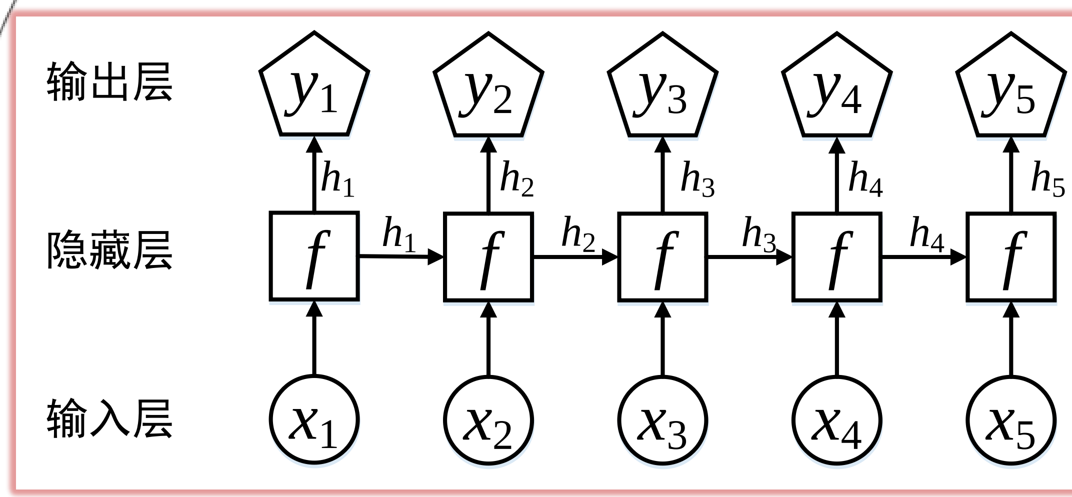


RNN模型介绍

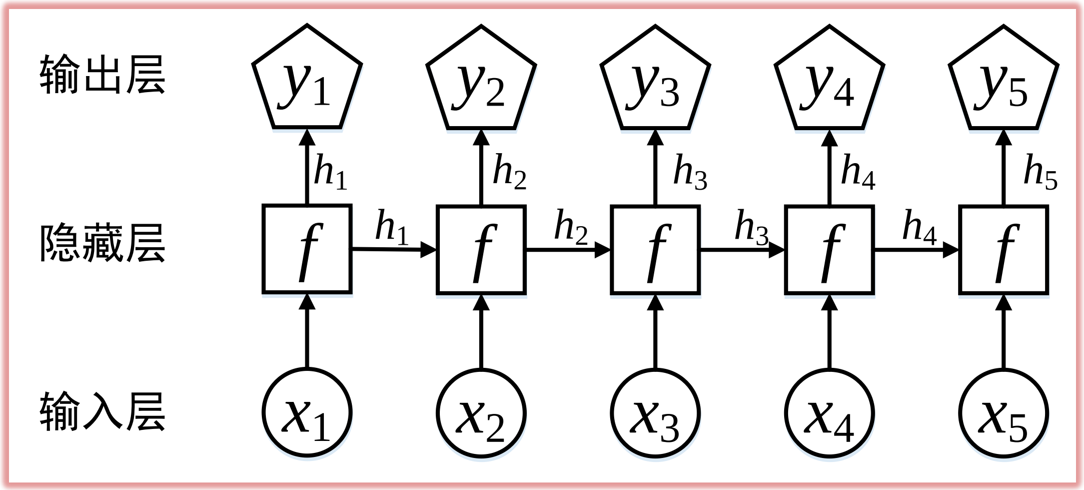
- 神经网络模型中，普遍认为网络的连接参数，蕴含的是“知识”，它像大脑神经元之间的连接，类似的数据重新输入后，经过这些参数的转换，可以复现类似的结论。
- RNN可以学习序列数据，并可以根据上下文复现学习到的序列。

RNN到底长什么样?





RNN建模



$$p(y_1, y_2, \cdots, y_T \mid x_1, x_2, \cdots, x_T) = \prod_i p(y_i \mid Y_{<i}, X_{\le i})$$

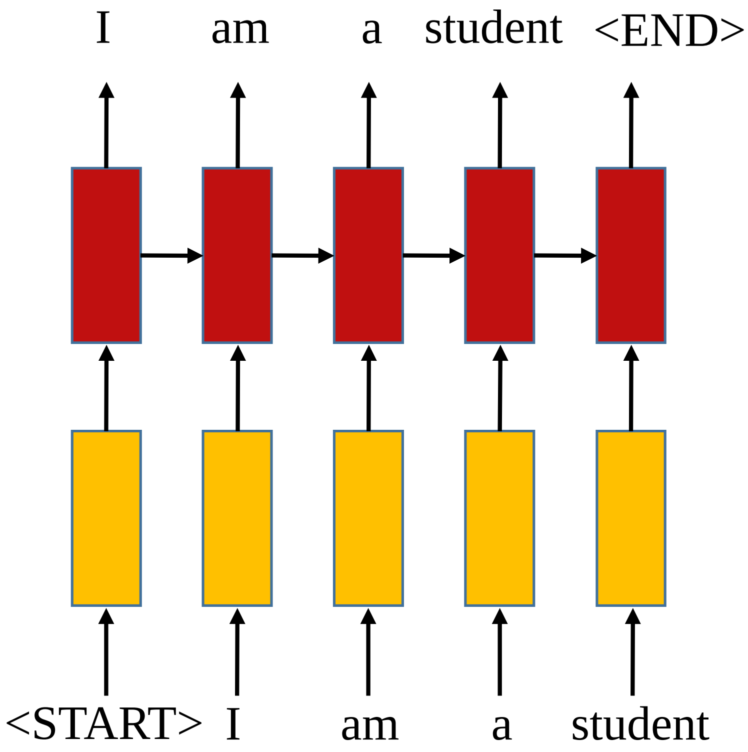
$h_i = f(x_i, h_{i-1})$ $\quad h_0 = 0$ 作为初始输入

$$p(y_i \mid Y_{<i}, X_{\le i}) = \mathrm{softmax}(h_i)$$

h_i试图包含了（$Y_{<i}, X_{\le i}$）中的所有信息

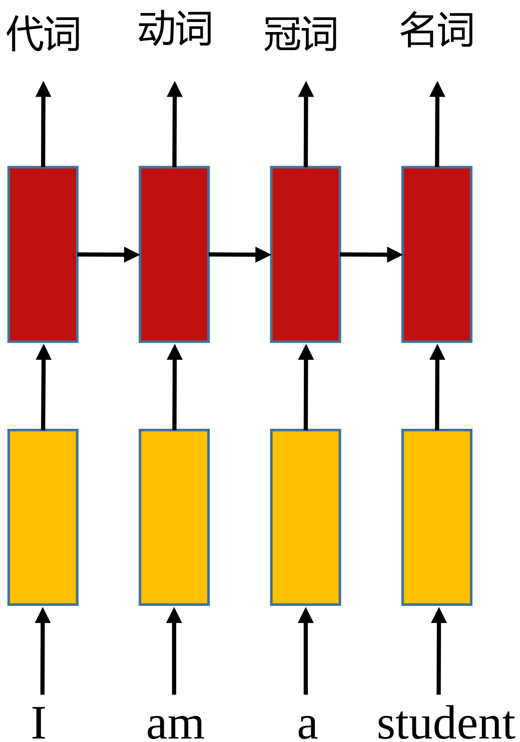
RNN应用介绍

- 序列标注（Sequential Labeling）：语言模型，建模某句话用某种语言生成的概率

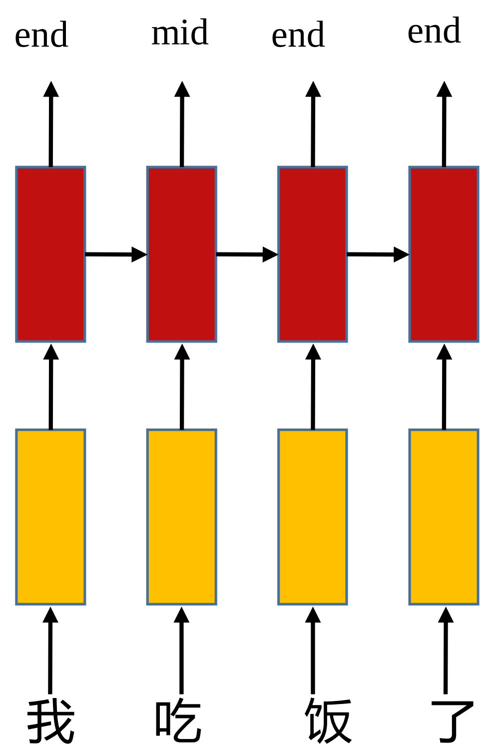


RNN应用介绍

- 序列标注：POS（Part of Speech）标注、分词

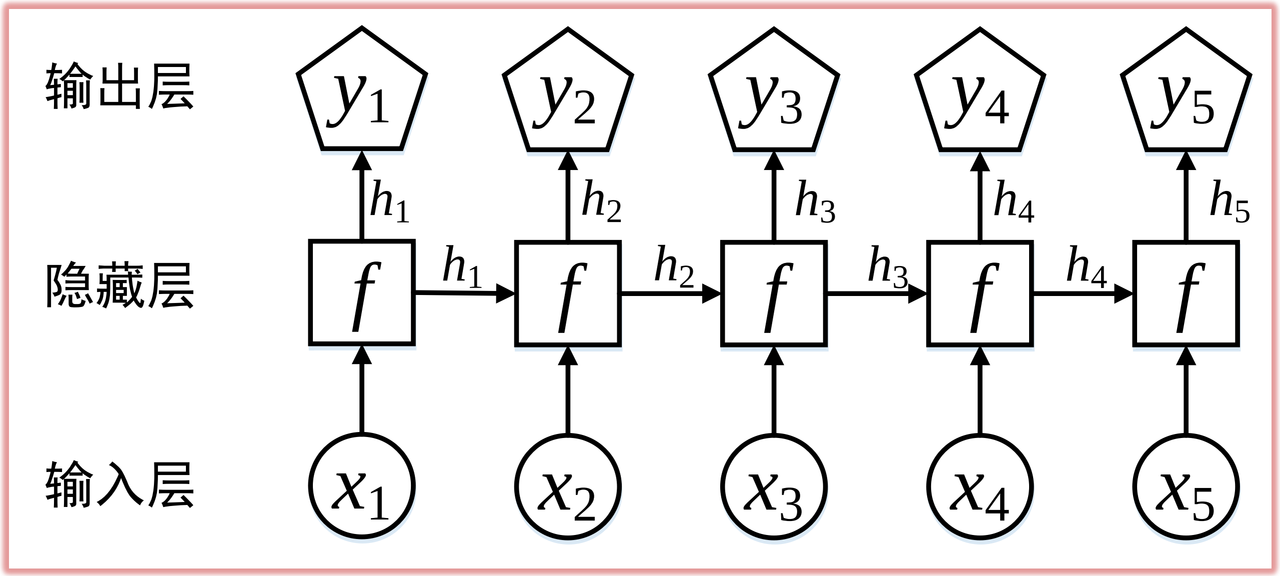


POS



分词

RNN的特点



- 每个时刻的输入数据，与上一时刻输出数据，合并生成当前时刻的输出数据
- 方框部分代表神经元，所有时刻的数据都共享同一组神经元参数

RNN建模

$$p(y_1, y_2, \cdots, y_T \mid x_1, x_2, \cdots, x_T) = \prod_i p(y_i \mid Y_{<i}, X_{\le i})$$

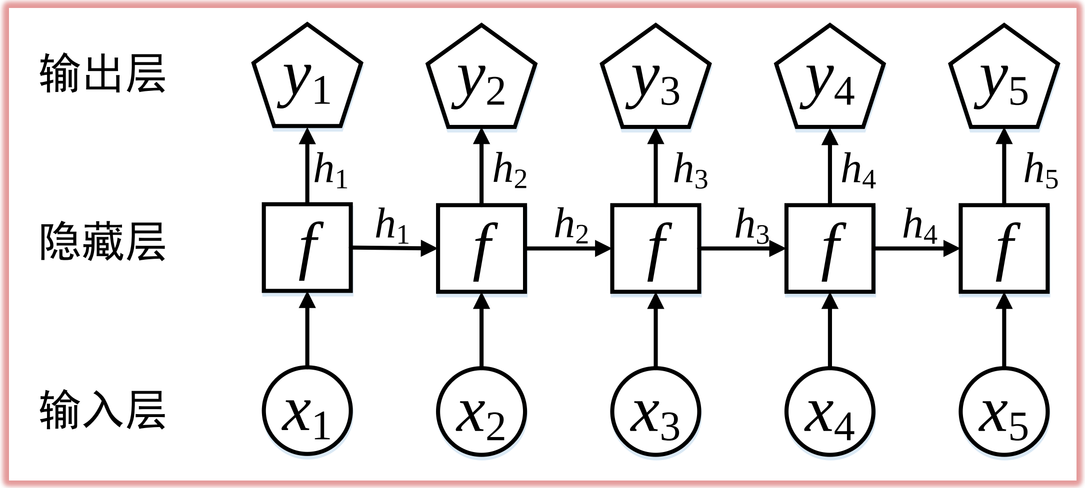
$$h_i = f(x_i, h_{i-1})$$

$$p(y_i \mid Y_{<i}, X_{\le i}) = \mathrm{softmax}(h_i)$$

- 关键点
  - 研究f函数如何建模
  - 研究x_i如何表达
    - x_i对应该词的词向量（Embedding）

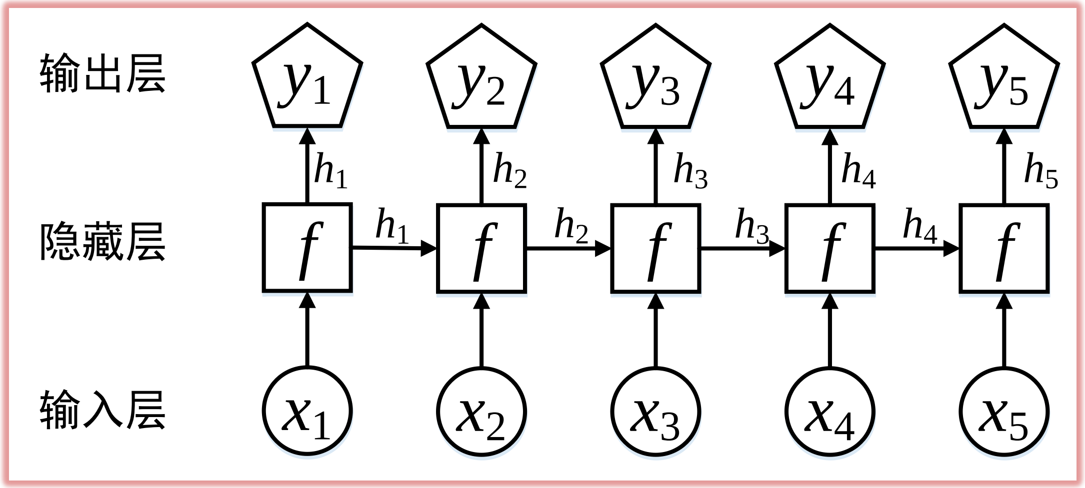
$$W_{k\times n} \cdot \begin{matrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{matrix} = x$$

朴素的f函数：RNN的正向计算



- 令$z_{\mathrm{i}} = Uh_{i-1} + Wx_i + b$ U和W的简单含义
- $h_i = \sigma(z_i)$
- 网络对每个时刻都有输出值$\hat{y}_i$
- 计算h_i可以是简单激活函数，如sigmoid, tanh等；也可以是复杂的多层神经网络。

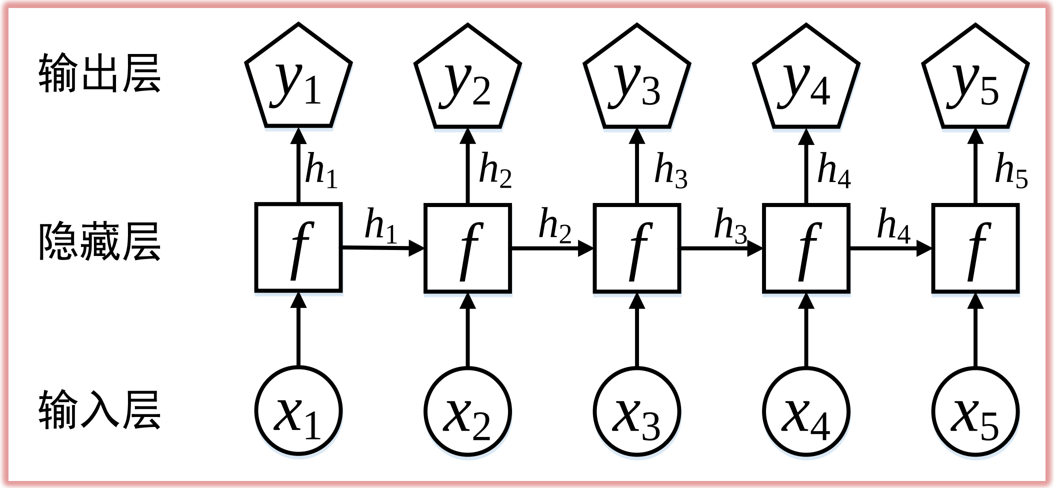
网络代价函数

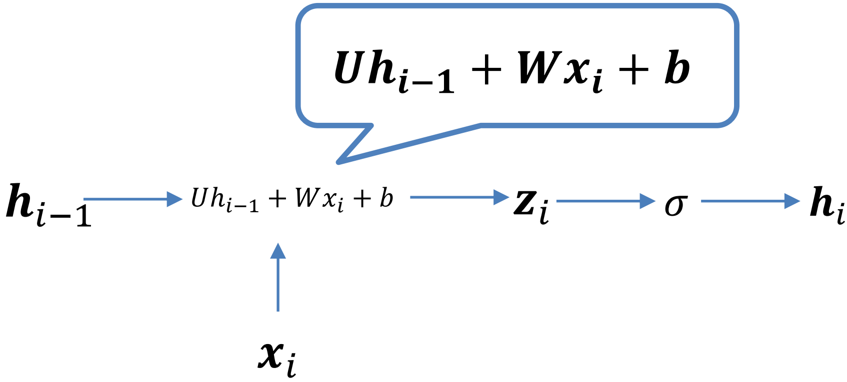


其代价函数可以是平方误差，在**一个序列**上的代价函数表示为

$$C = \sum_i \frac{1}{2} \|\hat{y}_i - y_i\|^2$$

误差反向传播





不妨假设总共有T 个时间步。我们令$\boldsymbol{\delta}_i = \frac{\partial C}{\partial \boldsymbol{z}_i} = \frac{\partial C_{\geq i}}{\partial \boldsymbol{z}_i}$，且$C_i$代表$y_i$对应的误差，则：

$$\boldsymbol{\delta}_i = \frac{\partial C_i}{\partial \boldsymbol{h}_i}\frac{\partial \boldsymbol{h}_i}{\partial \boldsymbol{z}_i} + \frac{\partial C_{>i}}{\partial \boldsymbol{z}_{i+1}}\frac{\partial \boldsymbol{z}_{i+1}}{\partial \boldsymbol{z}_i} = \frac{\partial C_i}{\partial \boldsymbol{h}_i} \odot \sigma'(\boldsymbol{z}_i) + \boldsymbol{\delta}_{i+1} \odot (U\sigma'(\boldsymbol{z}_i))$$

对f 中的某个参数θ，其导数为$\frac{\partial C}{\partial \theta} = \sum_i \frac{\partial C}{\partial z_i}\frac{\partial z_i}{\partial \theta} = \sum_i \boldsymbol{\delta}_i \frac{\partial z_i}{\partial \theta}$

$\frac{\partial C_i}{\partial \boldsymbol{h}_i}, \frac{\partial z_i}{\partial \theta}$ 以及求取其他参数的技巧与全连接神经网络相同，这里不再列举。

注：RNN训练方法有很多种，这里只是列出其中一种。

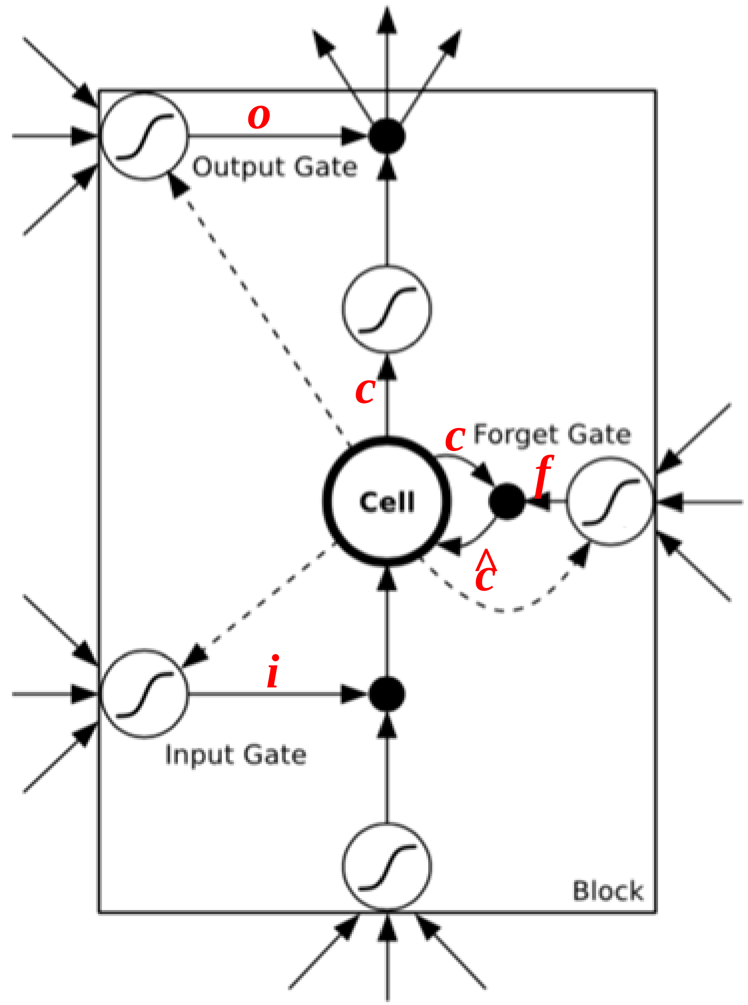
多元复合函数求导法则

- $C = f\big(z_1(\theta), z_2(\theta),\ \ldots,\ z_T(\theta)\big)$

- $\frac{\partial C}{\partial \theta} = \sum_i \frac{\partial C}{\partial z_i}\frac{\partial z_i}{\partial \theta} = \sum_i \boldsymbol{\delta}_i \frac{\partial z_i}{\partial \theta}$

f函数：Long Short Term Memory (LSTM)

$\boxed{f}$



LSTM是最常用的一种RNN激活函数，包含输出门o、遗忘门f 和输入门i。它的定义很复杂，如下式：

$$\mathbf{i}_t = \sigma(\mathbf{W}_{wi}\mathbf{w}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1}) \tag{1}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{wf}\mathbf{w}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1}) \tag{2}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{wo}\mathbf{w}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1}) \tag{3}$$

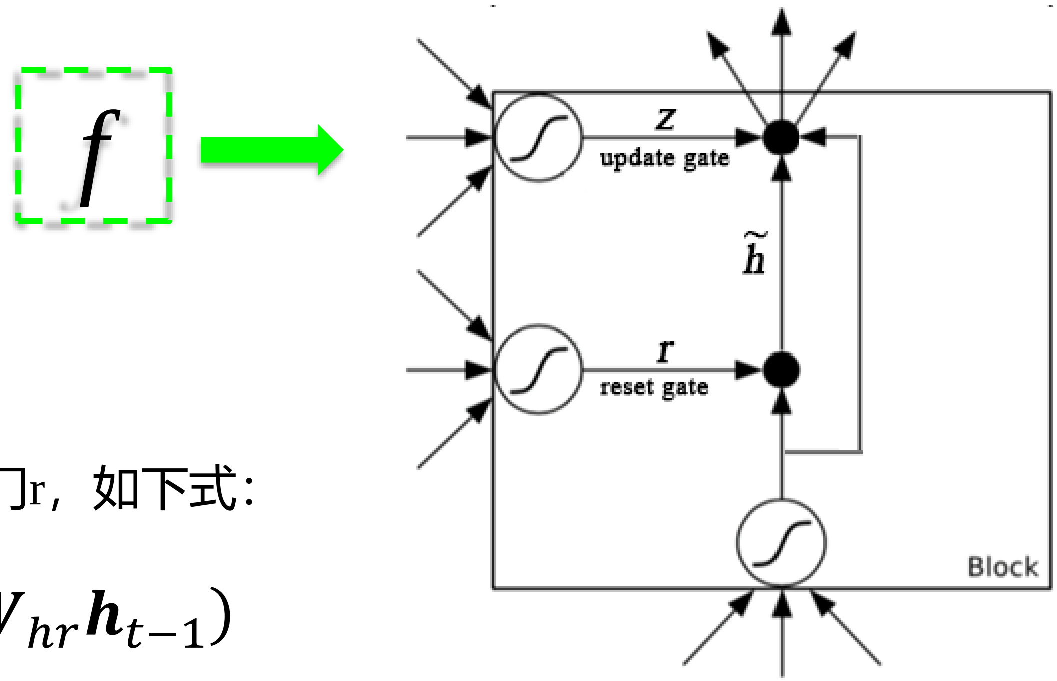
$$\hat{\mathbf{c}}_t = tanh(\mathbf{W}_{wc}\mathbf{w}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1}) \tag{4}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t \tag{5}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{c}_t) \tag{6}$$

https://mp.weixin.qq.com/s/y2kV4ye2zr1HYvZd3APeWA

f函数： Gated Recurrent Unit (GRU)



GRU包含更新门z 和重置门r，如下式：

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_{wr}\boldsymbol{w}_t + \boldsymbol{W}_{hr}\boldsymbol{h}_{t-1})$$

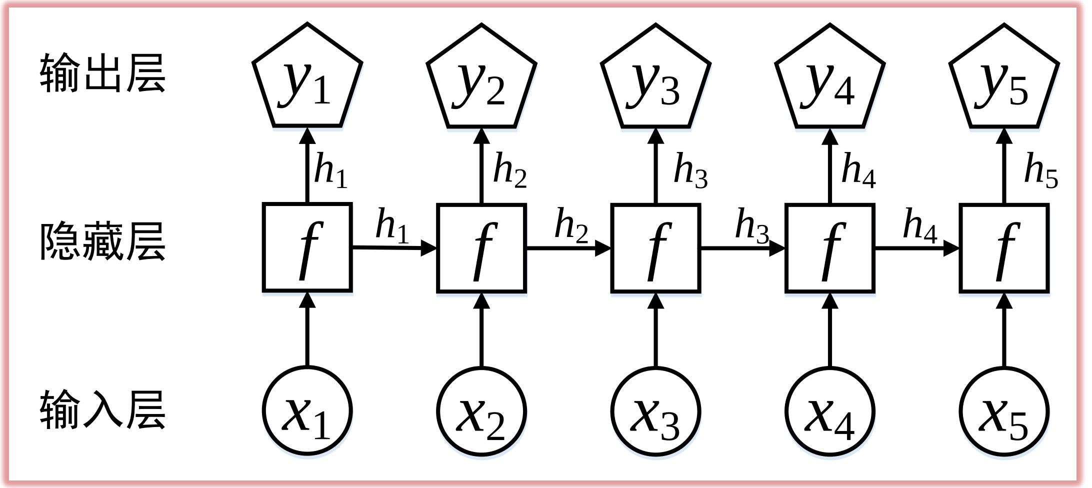
$$\boldsymbol{z}_t = \sigma(\boldsymbol{W}_{wz}\boldsymbol{w}_t + \boldsymbol{W}_{hz}\boldsymbol{h}_{t-1})$$

$$\widetilde{\boldsymbol{h}}_t = \sigma(\boldsymbol{W}_{wr}\boldsymbol{w}_t + \boldsymbol{U}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}))$$

$$\boldsymbol{h}_t = \boldsymbol{z}_t \odot \boldsymbol{h}_{t-1} + (1 - \boldsymbol{z}_t) \odot \widetilde{\boldsymbol{h}}_t$$

RNN总结

- 对“序列”进行建模，解决序列标注等实际问题



$$p(y_1, y_2, \cdots, y_T \mid x_1, x_2, \cdots, x_T) = \prod_i p(y_i \mid Y_{<i}, X_{\le i})$$

$$h_i = f(x_i, h_{i-1}) \qquad h_0 = 0 \text{ 作为初始输入}$$

$$p(y_i \mid Y_{<i}, X_{\le i}) = \mathrm{softmax}(h_i)$$

f函数使用***LSTM***，h_i**试图包含了（$Y_{<i}, X_{\le i}$）中的所有信息**

作业

- 阅读RNN解决实际问题的示例代码
  - http://deeplearning.net/tutorial/lstm.html
- 更多有用的材料
  - http://karpathy.github.io/2015/05/21/rnn-effectiveness/
  - http://colah.github.io/posts/2015-08-Understanding-LSTMs/

谢谢！