机器学习
Machine learning

第十章 神经网络与深度学习 (3)
Neural Network and Deep Learning

授课人：周晓飞
zhouxiaofei@iie.ac.cn
2021-12-24

第十章 神经网络与深度学习

Recurrent 网络

四种基本递归结构

1. 输入-输出递归模型（NARX 模型）

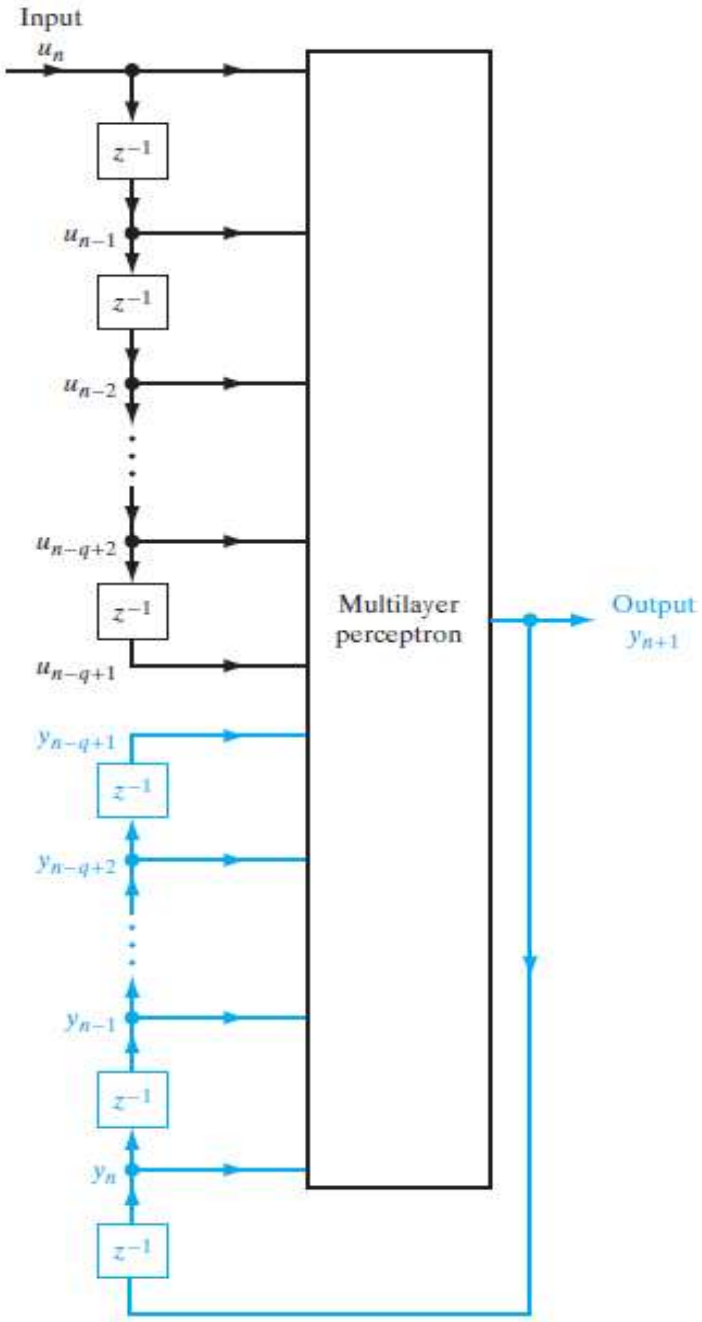
Input–Output Recurrent Model

$$y_{n+1} = F(y_n, \ldots, y_{n-q+1}; u_n, \ldots, u_{n-q+1})$$

输入： 当前输入u_n，过去的输入值$u_{n-1},\ldots,u_{n-q+1}$，

自反馈部分$y_n,\ldots,y_{n-q+1}$

输出： y_{n+1}



Recurrent 网络

四种基本递归结构

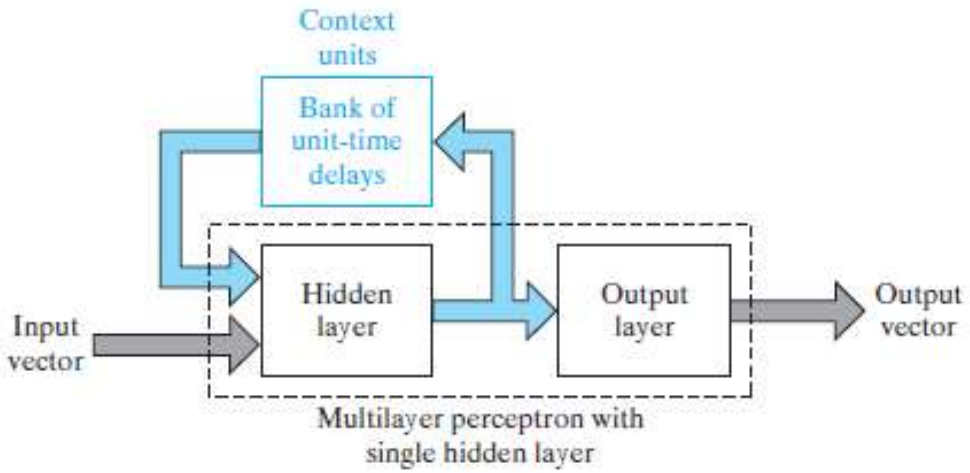
2. 状态空间模型

Simple Recurrent Network(SRN，Elman (1990,1996))

$$\mathbf{x}_{n+1} = \mathbf{a}(\mathbf{x}_n, \mathbf{u}_n)$$

$$\mathbf{y}_n = \mathbf{B}\mathbf{x}_n$$

$a(\ ,\ )$**非线性变换，单隐层神经元；B 线性变换。**

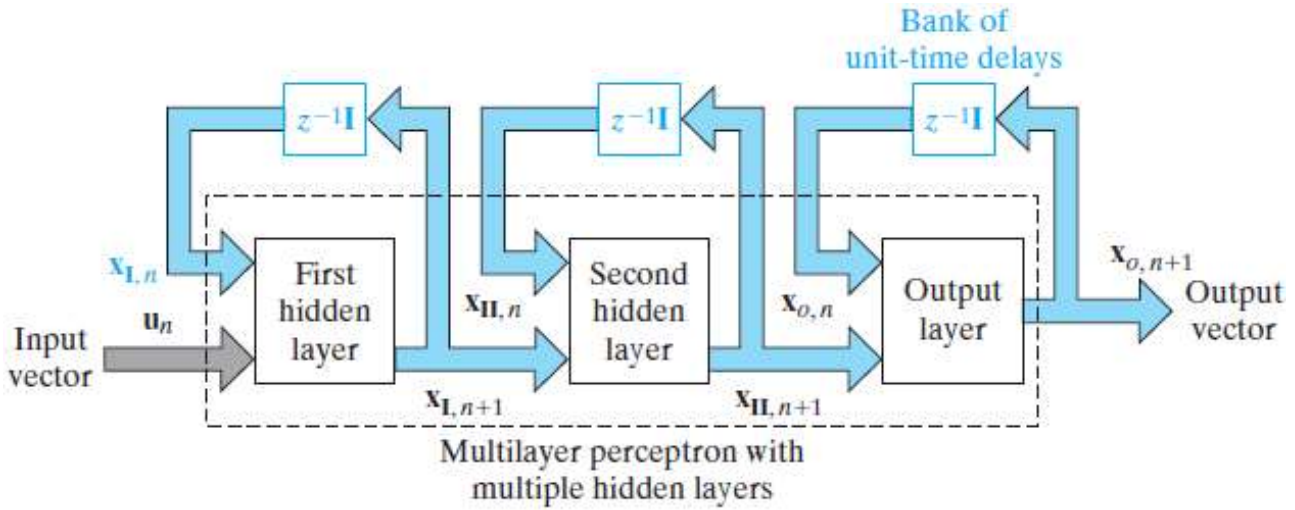


四种基本递归结构

3. 递归多层感知机

Recurrent Multilayer Perceptrons

$$\mathbf{x}_{\mathrm{I},n+1} = \boldsymbol{\phi}_{\mathrm{I}}(\mathbf{x}_{\mathrm{I},n}, \mathbf{u}_n)$$
$$\mathbf{x}_{\mathrm{II},n+1} = \boldsymbol{\phi}_{\mathrm{II}}(\mathbf{x}_{\mathrm{II},n}, \mathbf{x}_{\mathrm{I},n+1})$$
$$\vdots$$
$$\mathbf{x}_{o,n+1} = \boldsymbol{\phi}_{o}(\mathbf{x}_{o,n}, \mathbf{x}_{K,n+1})$$



Recurrent 网络

四种基本递归结构

4. 二阶网络

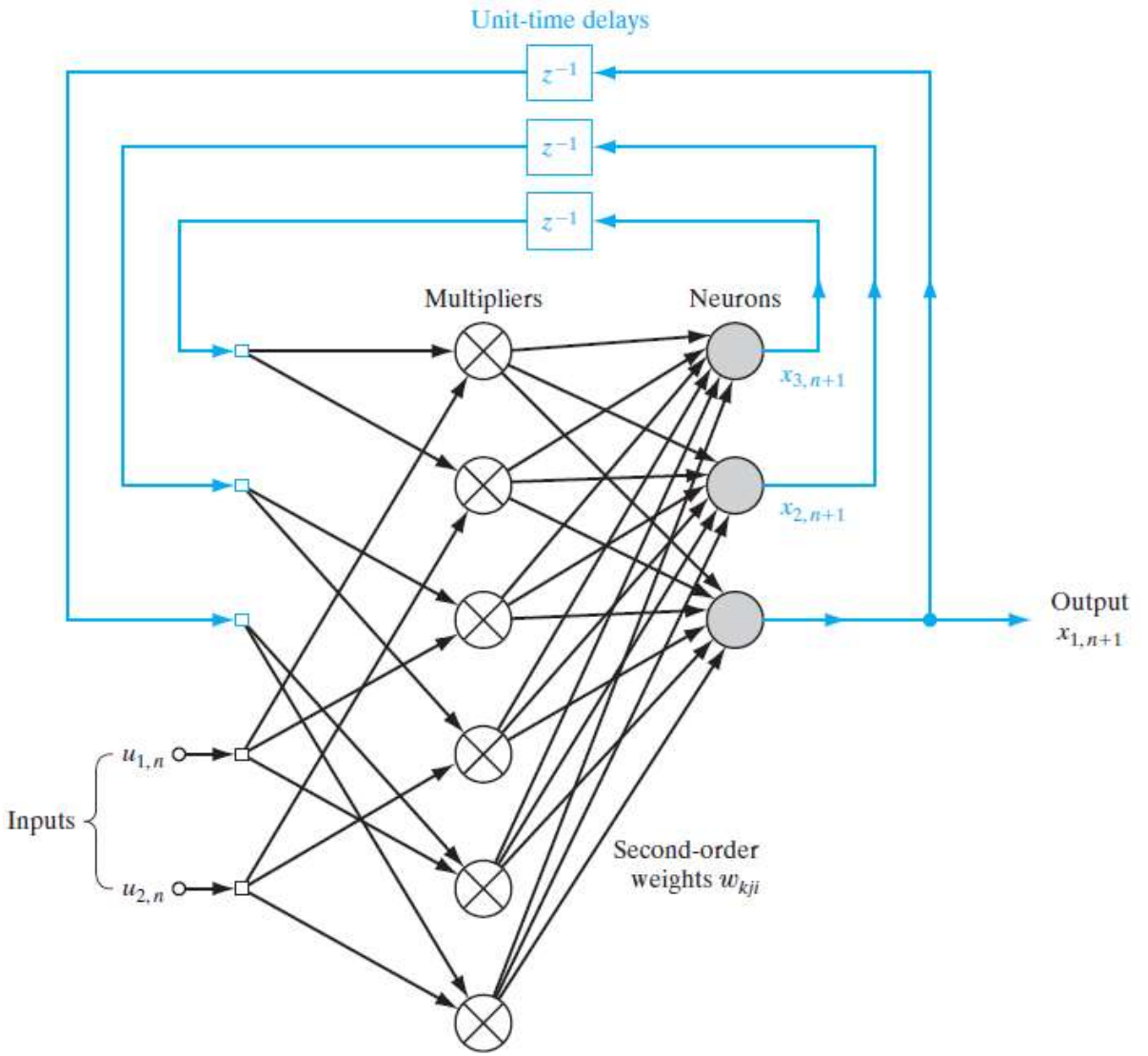
Second-Order Network

$$v_{k,n} = b_k + \sum_i \sum_j w_{kij} x_{i,n} u_{j,n}$$

$$x_{k,n+1} = \varphi(v_{k,n}) = \frac{1}{1 + \exp(-v_{k,n})}$$

状态转换可看作：

$$\delta(x_i, u_j) = x_k$$



Recurrent 网络

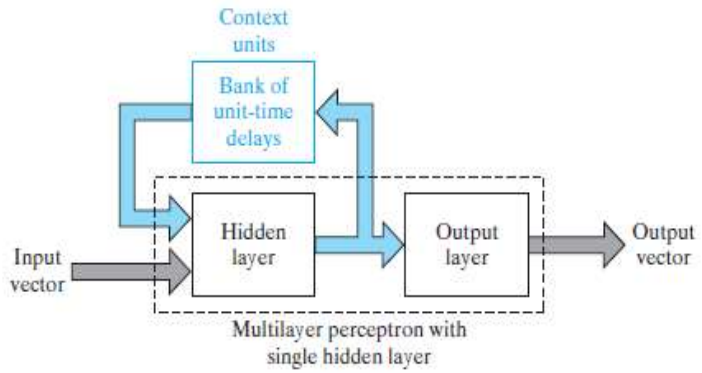
通用逼近定理

如果网络具有充分多的隐藏神经元，任意的非线性动态系统可以由递归神经网络以期望的精度来逼近，对于状态空间的紧致性没有限制。

递归神经网络:

$$\mathbf{x}_{n+1} = \phi(\mathbf{W}_a \mathbf{x}_n + \mathbf{W}_b \mathbf{u}_n)$$
$$\mathbf{y}_n = \mathbf{W}_c \mathbf{x}_n$$

$$\phi: \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_q \end{bmatrix} \rightarrow \begin{bmatrix} \varphi(x_1) \\ \varphi(x_2) \\ \vdots \\ \varphi(x_q) \end{bmatrix}$$



where $\mathbf{W}_a$ is a q-by-q matrix, $\mathbf{W}_b$ is a q-by-m matrix, $\mathbf{W}_c$ is a p-by-q matrix, and $\phi: \mathbb{R}^q \rightarrow \mathbb{R}^q$ is a diagonal map described by

$$\varphi(x) = \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \text{ or a logistic function, } \varphi(x) = \frac{1}{1 + e^{-x}}$$

Recurrent 网络

计算能力

定理Ⅰ（Siegelmann and Sontag，1991）

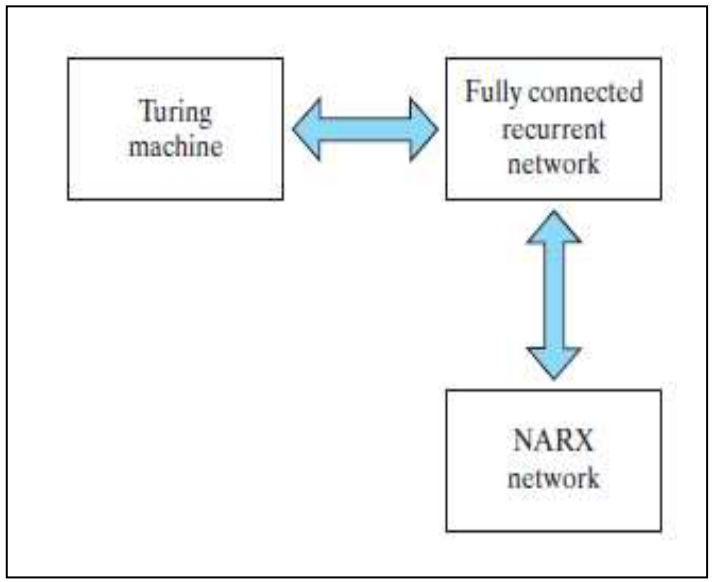
所有图灵机都可由建立在用 sigmoid 激活函数的神经元上的完全连接递归网络模拟。

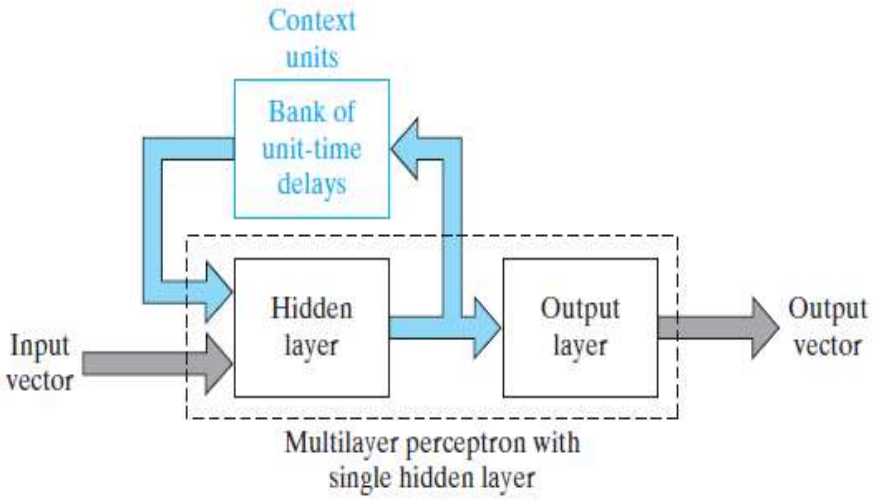
定理Ⅱ（siegelmann 等，1997）

对于 NARX 网络，若具有一隐藏层单元，其激活函数为有界和单侧饱和的并且有一个线性输出神经元，那么不计线性延迟（linear slowdown），它可以模拟用完全连接的具有有界且单侧饱和的激活函数的递归网络。

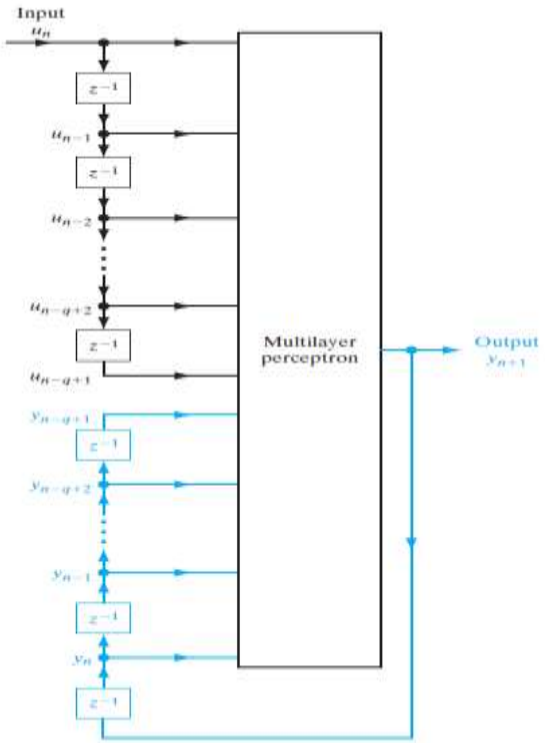
Recurrent 网络

计算能力





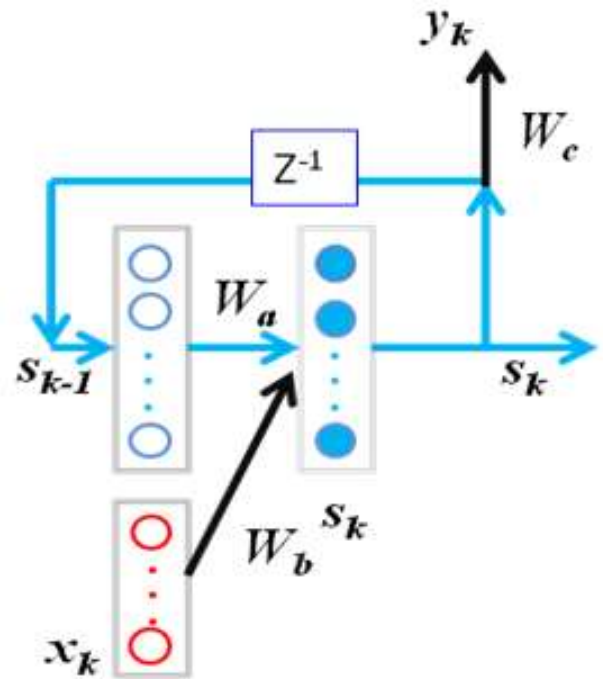
Recurrent Network



NARX Network

Recurrent 网络

RNN(状态空间模型)基本循环单元

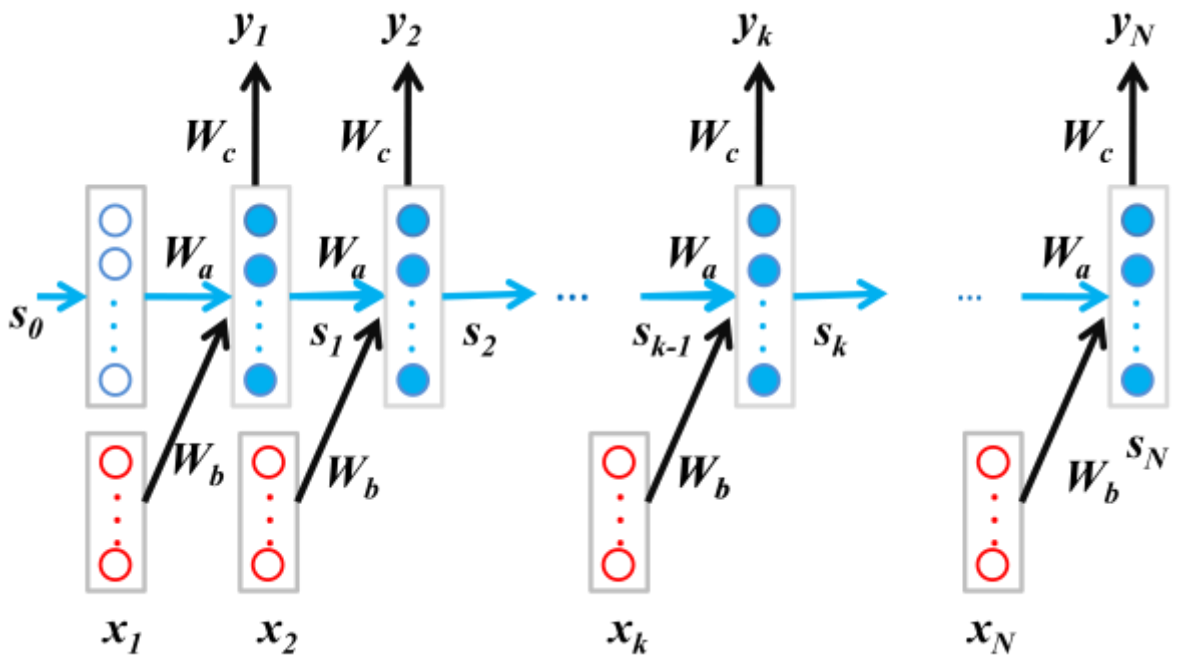


$$\begin{cases} v_k = W_a s_{k-1} + W_b x_k; \\ s_k = \sigma(v_k), \\ y_k = W_c s_k; \end{cases}$$

其中 $,\sigma(v_k)=[\sigma(v_{k1}),\sigma(v_{k2}),...,\sigma(v_{km})]^T$， $\sigma(x)=\dfrac{1}{1+\exp(-ax)}$;

Recurrent 网络

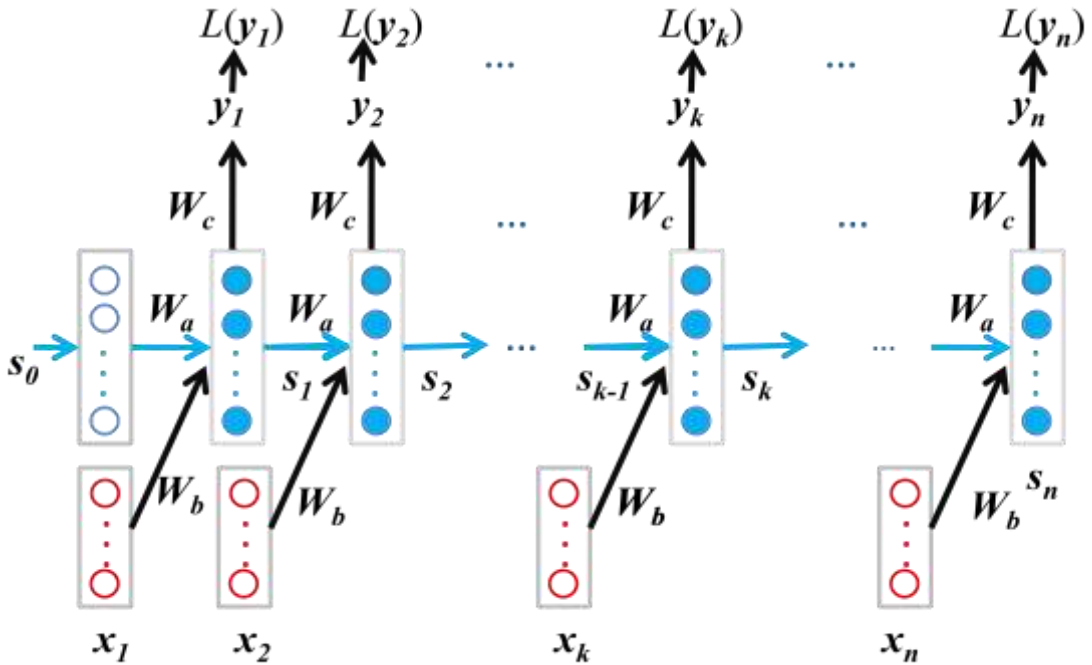
RNN 延展开后



Recurrent 网络

RNN 分回合训练

整体结构（一个回合输入 n 个样本）



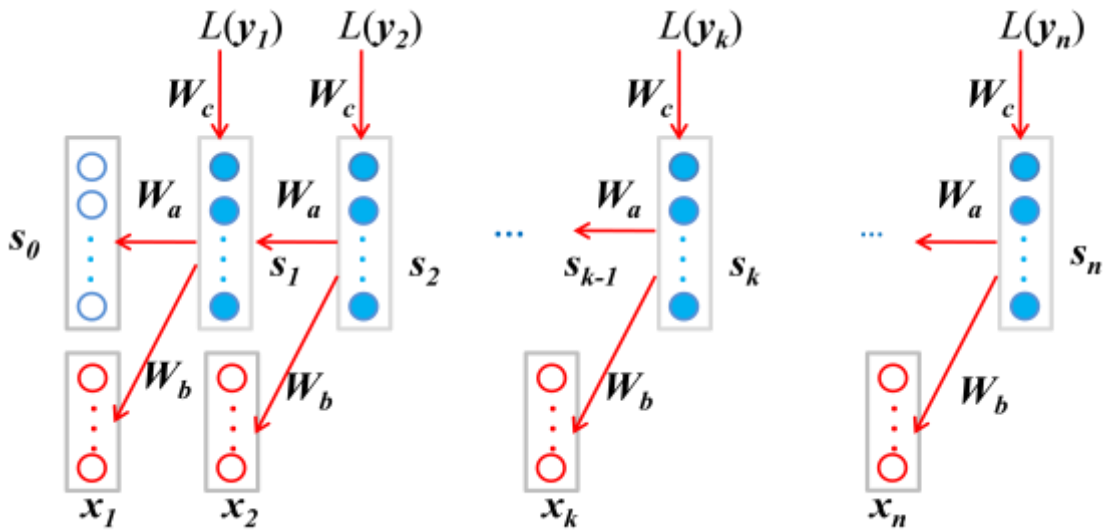
求 W_a, W_b, W_c?

W_a, W_b 的梯度依赖于局部梯度，W_a 相当于多层感知机隐层；W_c 相当于输出层。

Recurrent 网络

RNN 分回合训练

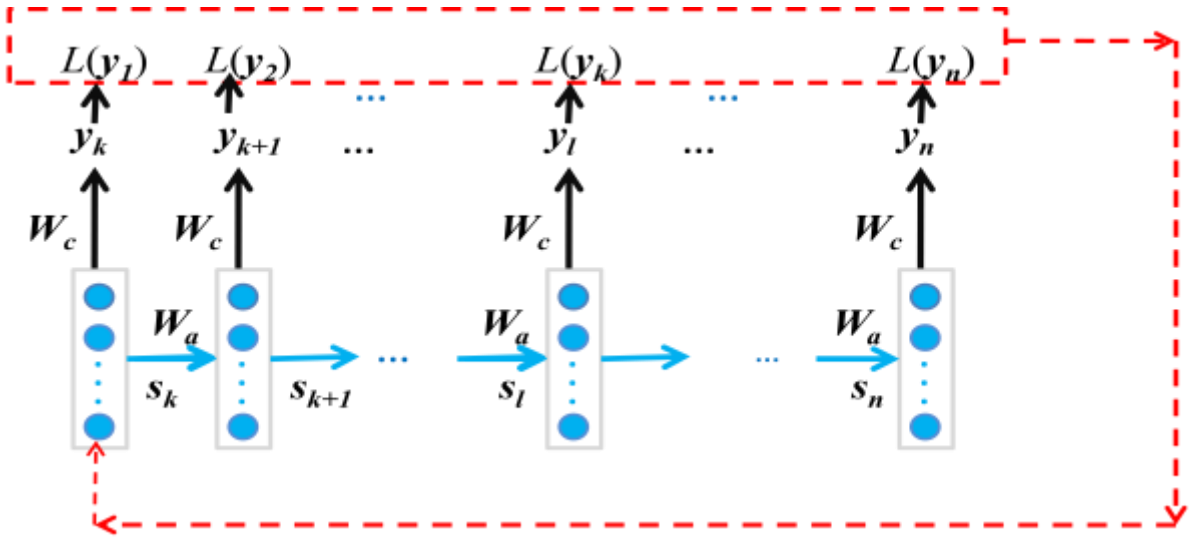
梯度是反向传播



Recurrent 网络

RNN 分回合训练

求局部梯度 $\delta_k = \dfrac{\partial L}{\partial \boldsymbol{v}_k}$

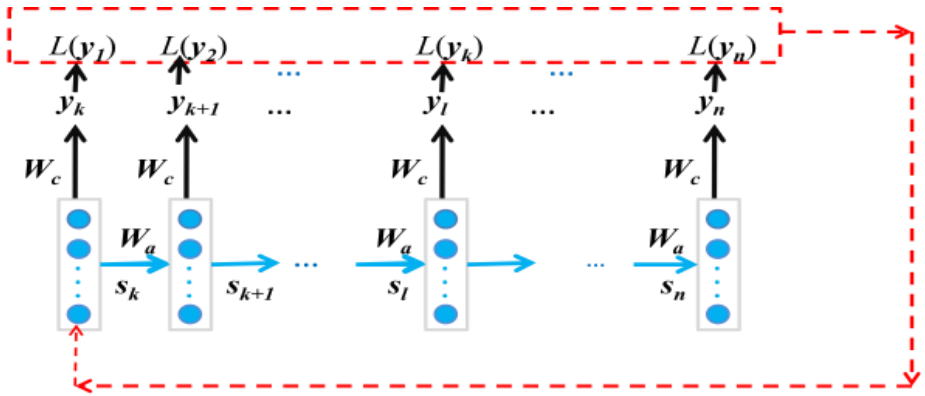


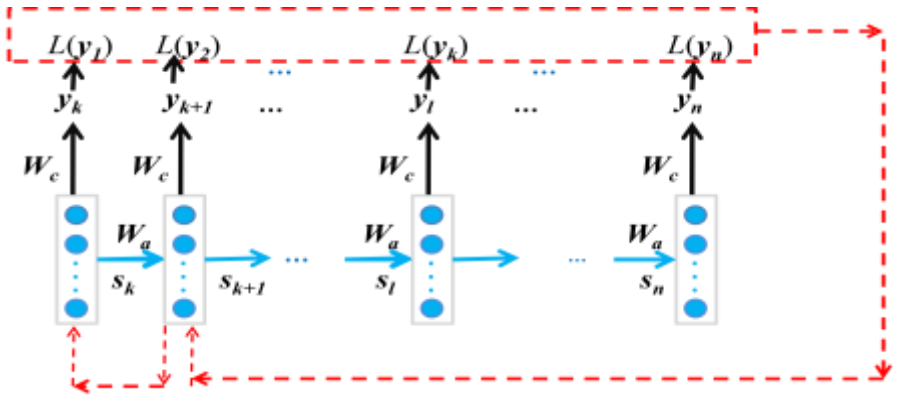
- **总体损失函数：** $L = \sum_{k=1}^{n} L(y_k)$

Recurrent 网络

RNN 分回合训练

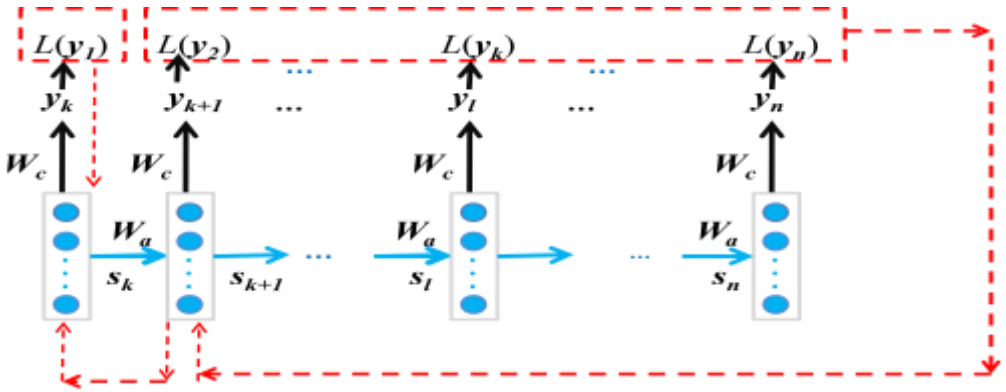
局部梯度迭代





当 $k=n$ 时(最后一个输入)，W_a 只依赖 $L(y_k)$;

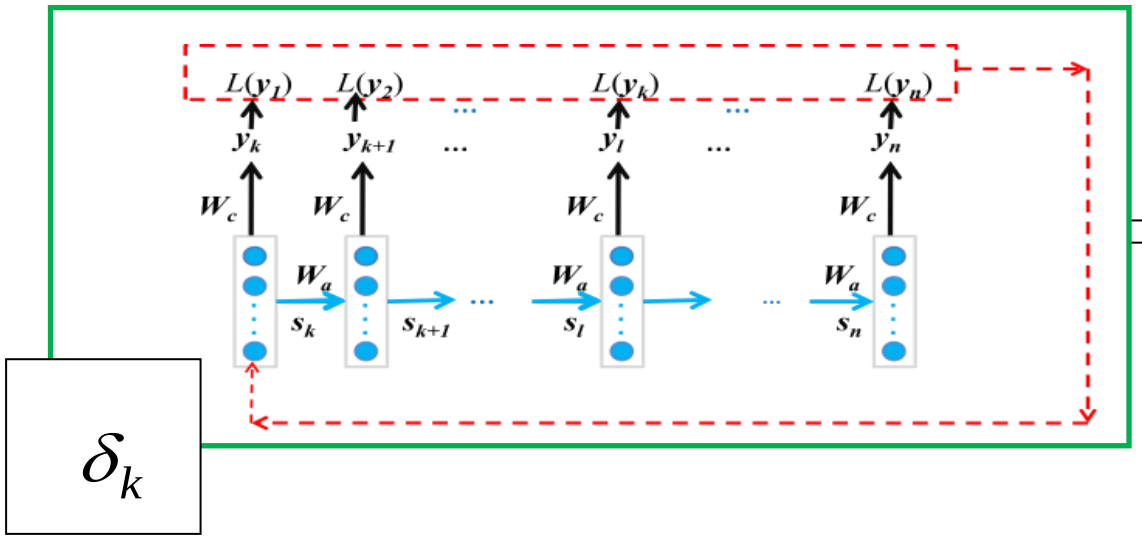
否则，W_a 依赖于 $L(y_k),\cdots L(y_n)$;



Recurrent 网络

RNN 分回合训练

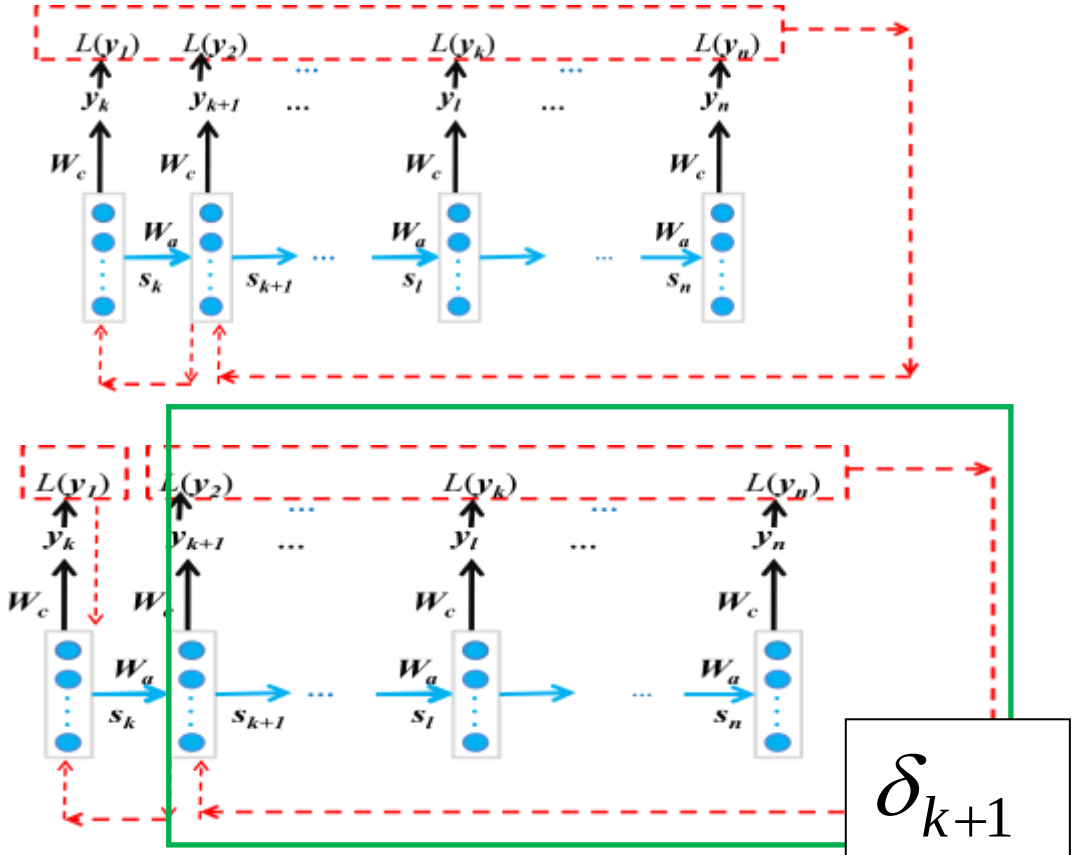
局部梯度迭代



δ_k

当 $k=n$ 时(最后一个输入)，W_a 只依赖 $L(y_k)$;

否则，W_a 依赖于 $L(y_k),\cdots L(y_n)$;



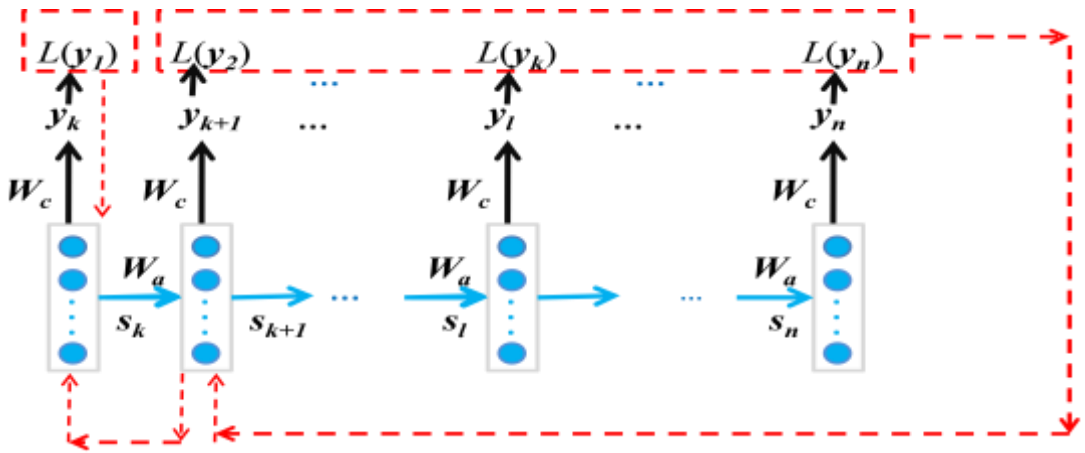
δ_{k+1}

RNN 分回合训练

局部梯度迭代

采用 BP 算法，迭代求取局部梯度

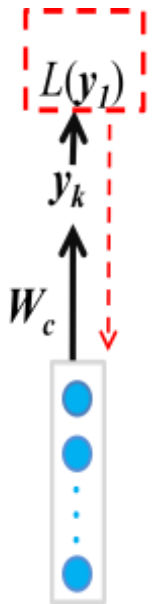
$$\boldsymbol{\delta}_k = \frac{\partial L}{\partial \boldsymbol{v}_k} = \frac{\partial\left(\sum_{l=k}^{n} L(\boldsymbol{y}_l)\right)}{\partial \boldsymbol{v}_k} = \frac{\partial L(\boldsymbol{y}_k)}{\partial \boldsymbol{v}_k} + \frac{\partial\left(\sum_{l=k+1}^{n} L(\boldsymbol{y}_l)\right)}{\partial \boldsymbol{v}_k} == \frac{\partial L(\boldsymbol{y}_k)}{\partial \boldsymbol{v}_k} + \frac{\partial \boldsymbol{v}_{k+1}}{\partial \boldsymbol{v}_k}\boldsymbol{\delta}_{k+1}$$



RNN 分回合训练

其中：

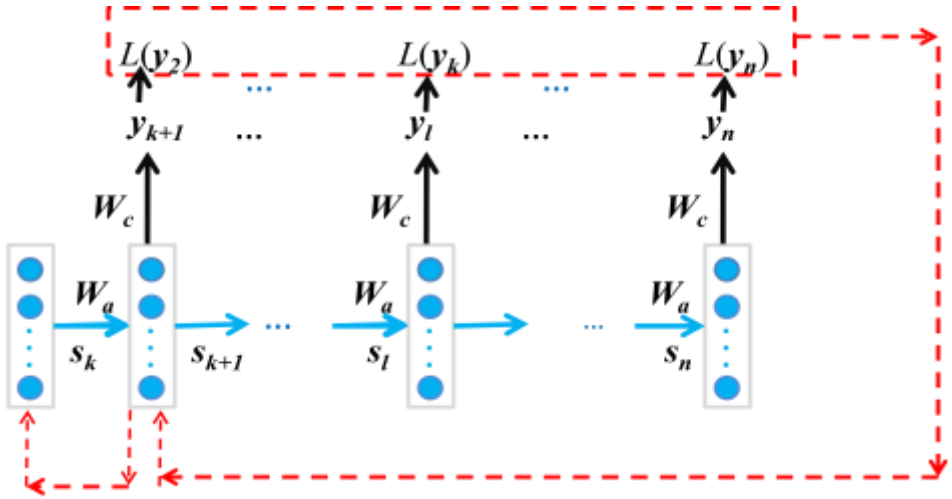
$$\frac{\partial L(\boldsymbol{y}_k)}{\partial \boldsymbol{v}_k}=\frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{v}_k}\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{s}_k}\frac{\partial L(\boldsymbol{y}_k)}{\partial \boldsymbol{y}_k}=diag(\boldsymbol{\sigma}'(\boldsymbol{v}_k))\boldsymbol{W}_c^T L'(\boldsymbol{y}_k)$$



Recurrent 网络

RNN 分回合训练

$$\frac{\partial \boldsymbol{v}_{k+1}}{\partial \boldsymbol{v}_k}\boldsymbol{\delta}_{k+1}=\frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{v}_k}\frac{\partial \boldsymbol{v}_{k+1}}{\partial \boldsymbol{s}_k}\boldsymbol{\delta}_{k+1}=diag(\boldsymbol{\sigma}'(\boldsymbol{v}_k))\boldsymbol{W}_a^T\boldsymbol{\delta}_{k+1}$$



$$\text{当}k=n,\quad \boldsymbol{\delta}_k=\frac{\partial L}{\partial \boldsymbol{v}_k}=\frac{\partial L(\boldsymbol{y}_k)}{\partial \boldsymbol{v}_k}=diag(\boldsymbol{\sigma}'(\boldsymbol{v}_k))\boldsymbol{W}_c^T L'(\boldsymbol{y}_k)$$

$$\text{当}k\neq n,\ \boldsymbol{\delta}_k=\frac{\partial L}{\partial \boldsymbol{v}_k}=\frac{\partial L(\boldsymbol{y}_k)}{\partial \boldsymbol{v}_k}+\frac{\partial \boldsymbol{s}_{k+1}}{\partial \boldsymbol{v}_k}\boldsymbol{\delta}_{k+1}=diag(\boldsymbol{\sigma}'(\boldsymbol{v}_k))(\boldsymbol{W}_c^T L'(\boldsymbol{y}_k)+\boldsymbol{W}_a^T\boldsymbol{\delta}_{k+1});$$

RNN 分回合训练

参数更新

$$\boldsymbol{W}_a^{(k)} = \boldsymbol{W}_a^{(k)} - \eta_1 \boldsymbol{\delta}_k \frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{W}_a^{(k)}}, \qquad \boldsymbol{W}_a = \frac{1}{n}\sum_{k=1}^{n}\left(\boldsymbol{W}_a^{(k)} - \eta_1 \boldsymbol{\delta}_k \frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{W}_a^{(k)}}\right) = \boldsymbol{W}_a - \frac{\eta_1}{n}\boldsymbol{\delta}_k \sum_{k=1}^{n}\left(\frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{W}_a}\right) = \boldsymbol{W}_a - \eta \boldsymbol{\delta}_k \sum_{k=1}^{n} \boldsymbol{s}_{k-1}^T$$

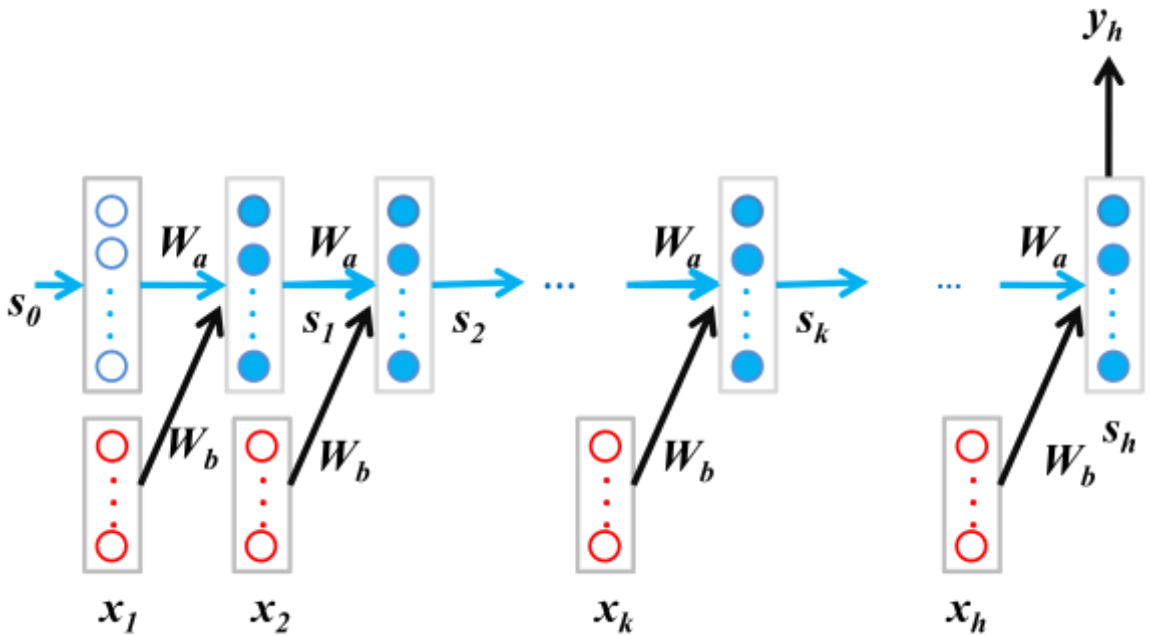
$$\boldsymbol{W}_c^{(k)} = \boldsymbol{W}_c^{(k)} - \eta_2 \frac{\partial L(\boldsymbol{y}_k)}{\partial \boldsymbol{W}_c}, \qquad \boldsymbol{W}_c = \frac{1}{n}\sum_{k=1}^{n}\left(\boldsymbol{W}_c^{(k)} - \eta_2 \frac{\partial L(\boldsymbol{y}_k)}{\partial \boldsymbol{W}_c}\right) = \boldsymbol{W}_c - \frac{\eta_2}{n}\sum_{k=1}^{n}\left(\frac{\partial L(\boldsymbol{y}_k)}{\partial \boldsymbol{W}_c}\right) = \boldsymbol{W}_c - \eta \sum_{k=1}^{n} \boldsymbol{s}_k^T$$

$$\boldsymbol{W}_b^{(k)} = \boldsymbol{W}_b^{(k)} - \eta_3 \boldsymbol{\delta}_k \frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{W}_b}, \qquad \boldsymbol{W}_b = \frac{1}{n}\sum_{k=1}^{n}\left(\boldsymbol{W}_b^{(k)} - \eta_3 \boldsymbol{\delta}_k \frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{W}_b}\right) = \boldsymbol{W}_b - \frac{\eta_3}{n}\boldsymbol{\delta}_k \sum_{k=1}^{n}\frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{W}_b} = \boldsymbol{W}_b - \eta \boldsymbol{\delta}_k \sum_{k=1}^{n} \boldsymbol{x}_k^T$$

Recurrent 网络

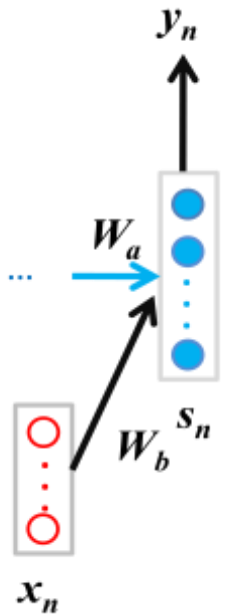
RNN 分回合训练

截断深度为 h 的学习（课堂作业）



RNN 连续训练

实时递归学习(连续训练)



$$\boldsymbol{\delta}_k = \frac{\partial L(\boldsymbol{y}_k)}{\partial \boldsymbol{v}_k} = \frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{v}_k}\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{s}_k}\frac{\partial L(\boldsymbol{y}_k)}{\partial \boldsymbol{y}_k} = diag(\boldsymbol{\sigma}'(\boldsymbol{v}_k))\boldsymbol{W}_c^T L'(\boldsymbol{y}_k)$$

$$\boldsymbol{W}_a = \boldsymbol{W}_a - \eta\boldsymbol{\delta}_k \frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{W}_a} = \boldsymbol{W}_a - \eta\boldsymbol{\delta}_k \boldsymbol{s}_{k-1}^T,$$

$$\boldsymbol{W}_b = \boldsymbol{W}_b - \eta\boldsymbol{\delta}_k \frac{\partial \boldsymbol{v}_k}{\partial \boldsymbol{W}_b} = \boldsymbol{W}_b - \eta\boldsymbol{\delta}_k \boldsymbol{x}_k^T,$$

$$\boldsymbol{W}_c = \boldsymbol{W}_c - \eta\boldsymbol{s}_k^T,$$

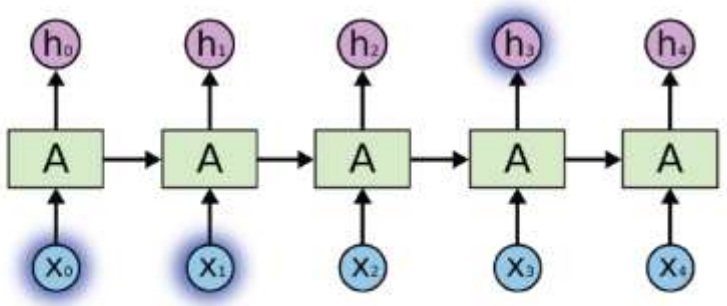
Recurrent 网络

RNN 长期依赖

RNN可以较好地处理Short-Term依赖关系

举例

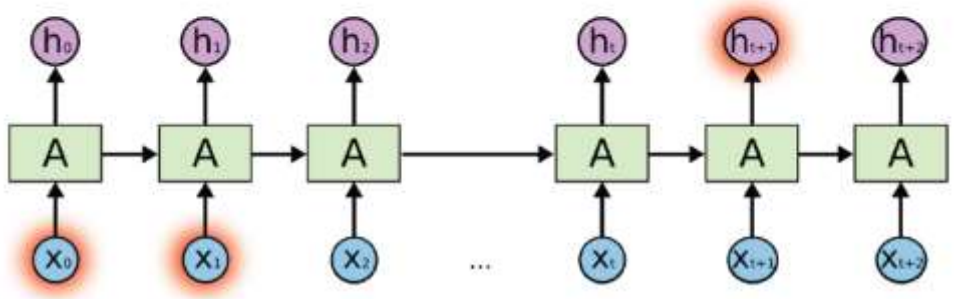
– The clouds are in the *sky*



RNN无法较好地处理Long-Term依赖关系

举例：

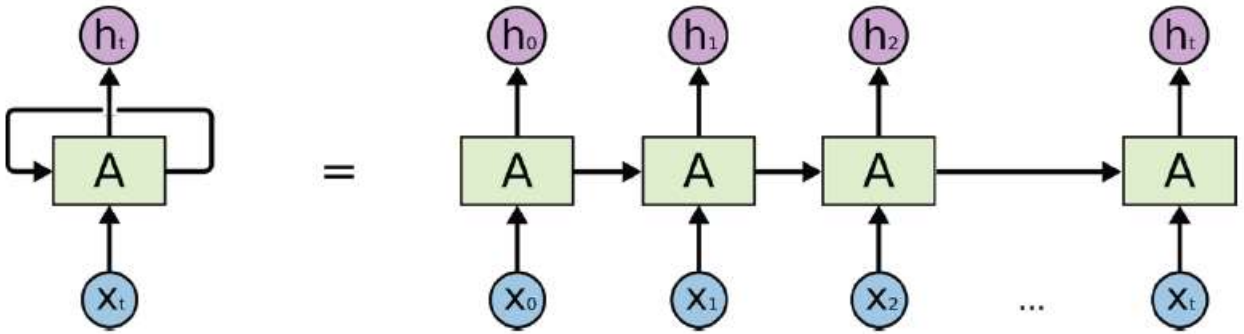
– I grew up in France··· I speak fluent *French*

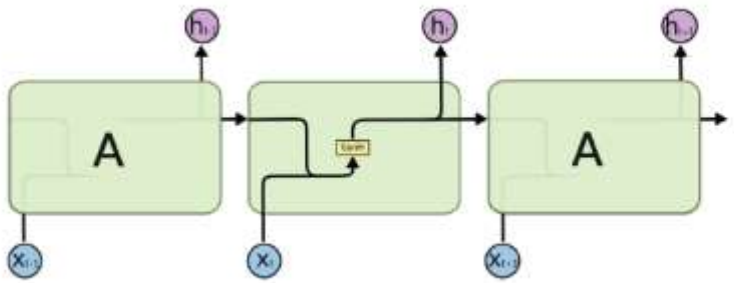


Recurrent 网络

RNN 扩展的递归结构

标准 RNN 结构





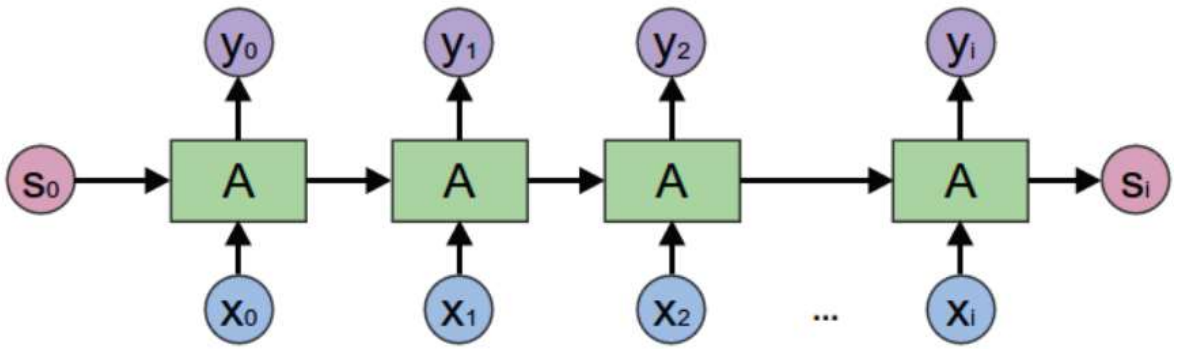
$$h_t = \sigma\left(W^{(hh)}h_{t-1} + W^{(hx)}x_{[t]}\right)$$

$$\hat{y}_t = \text{softmax}\left(W^{(S)}h_t\right)$$

RNN 扩展的递归结构

RNN 任务

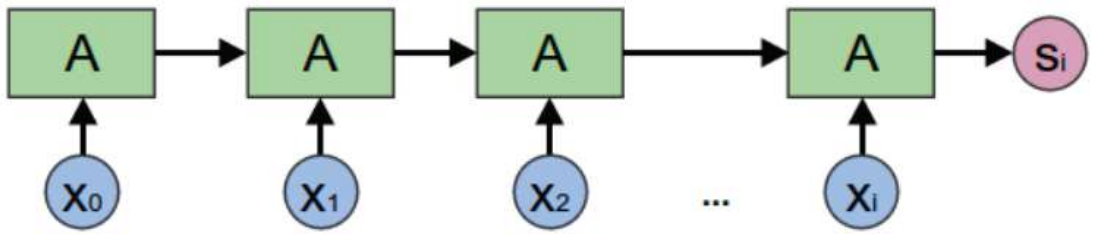
- Sequence to Sequence



RNN 扩展的递归结构

RNN 任务

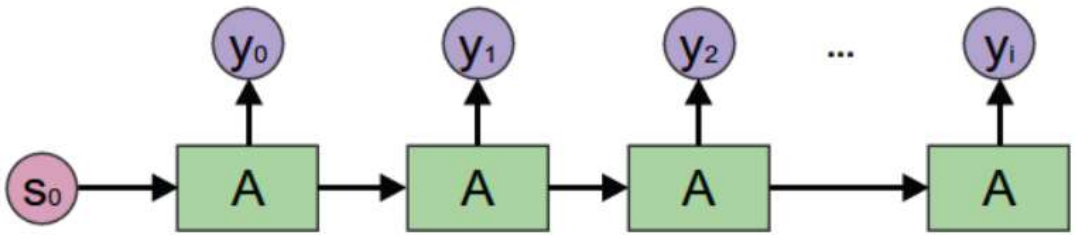
- Encoding



RNN 扩展的递归结构

RNN 任务

- Decoding



本讲参考文献

1. 邱锡鹏，《深度学习与自然语言处理》Slides@CCF ADL 20160529。

2. Stanford Class-CS231n: Convolutional Neural Networks for Visual Recognition。

3. 第十一届中国中文信息学会暑期学校，暨中国中文信息学会《前沿技术讲习班》，201607。

4. Simon Haykin，Neural Network and Learning Machine. 3rd