Deep Learning

Recurrent Neural Network

Xinfeng Zhang(张新峰)
School of Computer Science and Technology
University of Chinese Academy of Sciences
Email: xfzhang@ucas.ac.cn

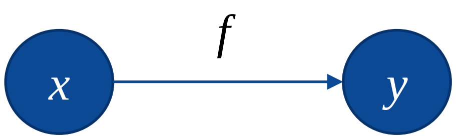
提纲

- 计算图
- 循环神经网络
- 长短时记忆网络
- 其他典型循环神经网络
- 循环神经网络的主要应用
- 中英文术语对照

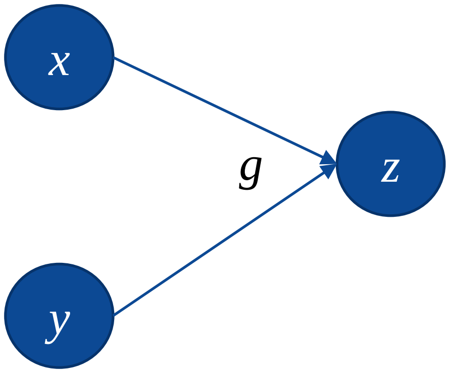
1 计算图

计算图

- **计算图(Computational Graph)**
  - 描述计算结构的一种图
- **计算图的元素**
  - 节点(node)：表示变量，可以是标量、矢量、张量等
  - 边(edge)：表示操作(函数)



$y=f(x)$



$z=g(x, y)$

计算图

❑ 计算图(Computational Graph)

$$y=f\,(g(h(x)))$$

$$u=h(x) \qquad v=g(u) \qquad y=f(v)$$

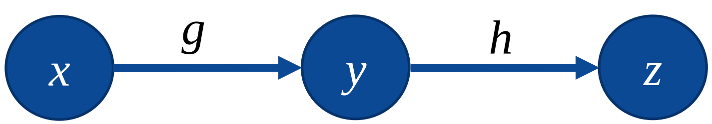
$$x \xrightarrow{h} u \xrightarrow{g} v \xrightarrow{f} y$$

计算图

❑ 链式法则

– Case 1:

$z=f(x)$ ➡ $y=g(x)$, $z=h(y)$

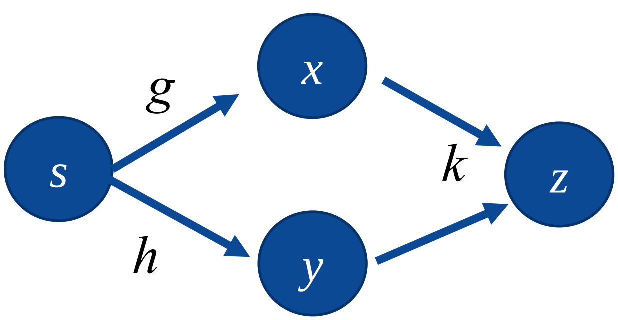


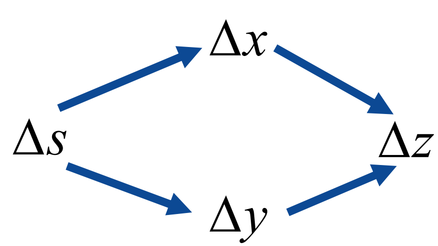
$$\frac{dz}{dx}=\frac{dz}{dy}\frac{dy}{dx}$$

$$\Delta x \rightarrow \Delta y \rightarrow \Delta z$$

– Case 2:

$z=f(s)$ ➡ $x=g(s)$, $y=h(s)$, $z=k(x,y)$





$$\frac{dz}{ds}=\frac{\partial z}{\partial y}\frac{dy}{ds}+\frac{\partial z}{\partial x}\frac{dx}{ds}$$

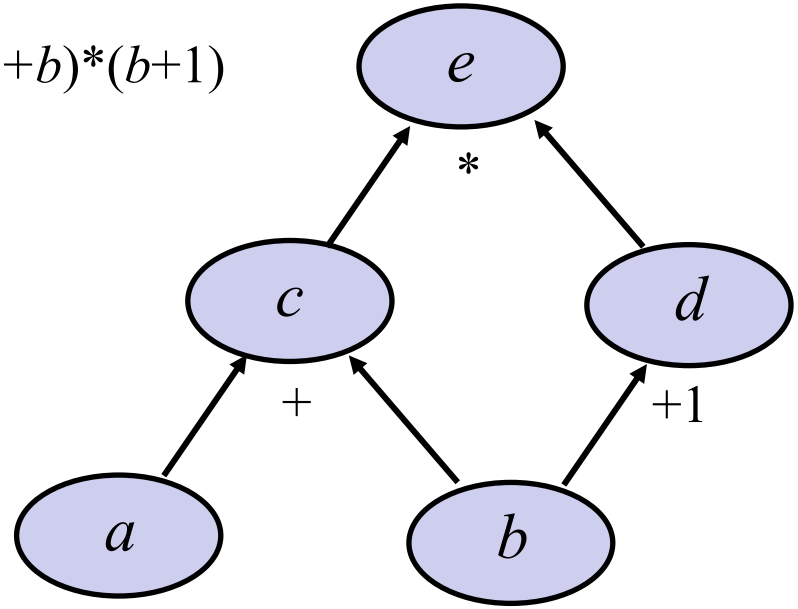
计算图

- 求导示例：
  - $a = 2, b = 1$

    ➡ $c = 3, d = 2, e = 6$

  - $\frac{\partial e}{\partial a} = ?, \frac{\partial e}{\partial b} = ?$

$e=(a+b)*(b+1)$



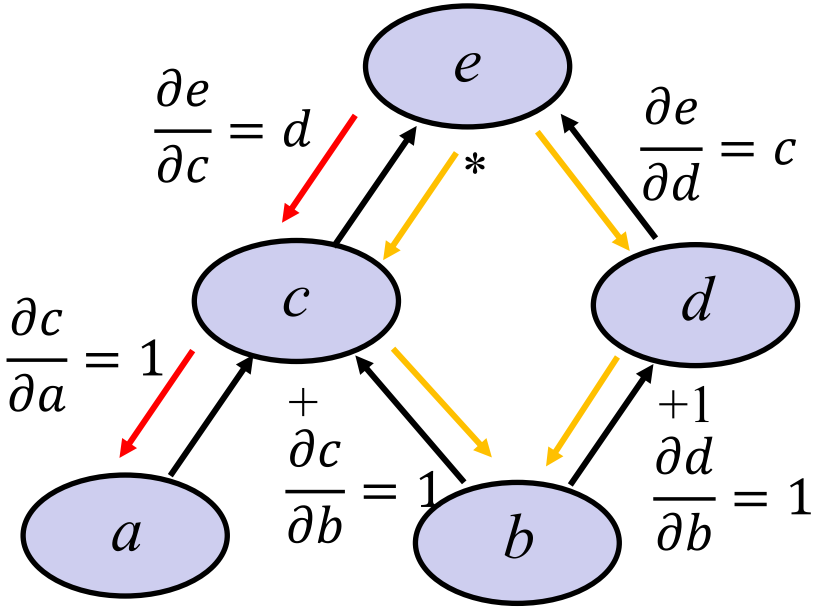
计算图

- 求导示例：
  - $a = 2, b = 1$

    ➡ $c = 3, d = 2, e = 6$

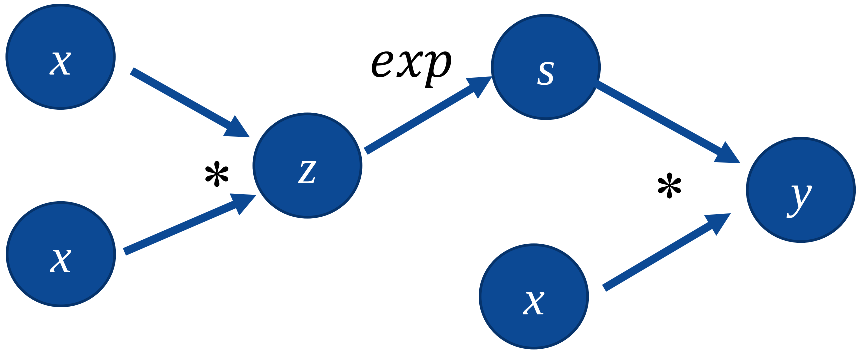
  - $\frac{\partial e}{\partial a} = \frac{\partial e}{\partial c}\frac{\partial c}{\partial a} = d = b + 1 = 2$
  - $\frac{\partial e}{\partial b} = \frac{\partial e}{\partial c}\frac{\partial c}{\partial b} + \frac{\partial e}{\partial d}\frac{\partial d}{\partial b}$

    $= d + c = b + 1 + a + b = 5$



计算图

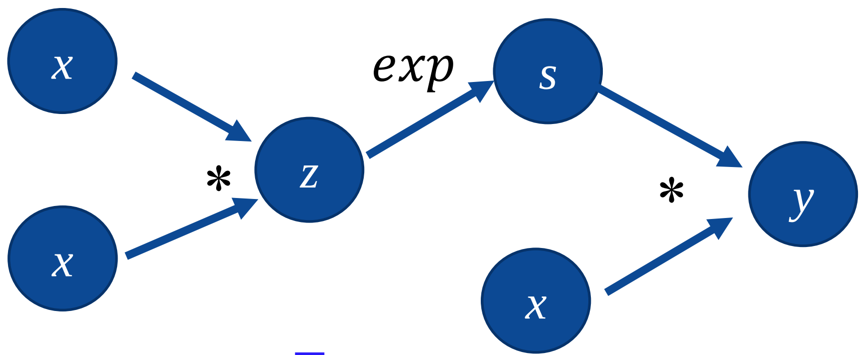
- 参数共享
  - $y = xe^{x^2}$



计算图

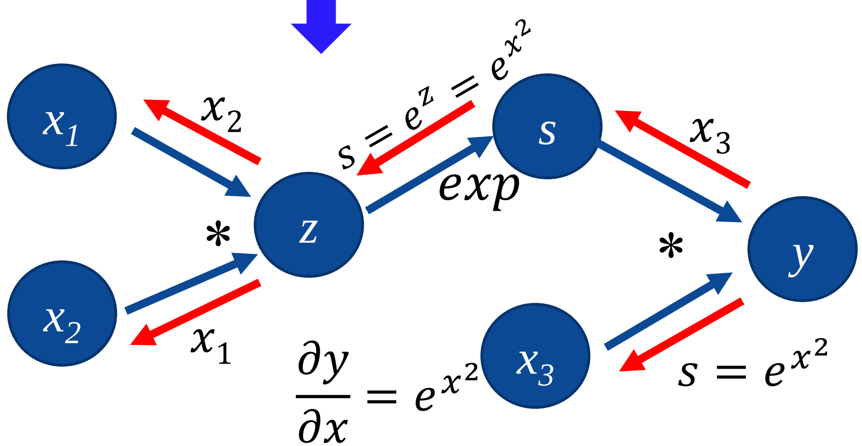
❑ 参数共享

- $y = xe^{x^2}$ ⟹ $\dfrac{\partial y}{\partial x} = 2xe^{x^2}x + e^{x^2}$

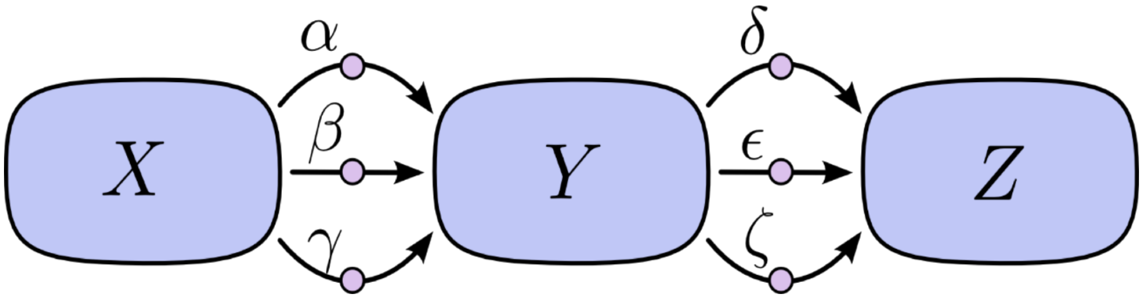


$$\frac{\partial y}{\partial x} = xe^{x^2}x$$

$$\frac{\partial y}{\partial x} = xe^{x^2}x$$



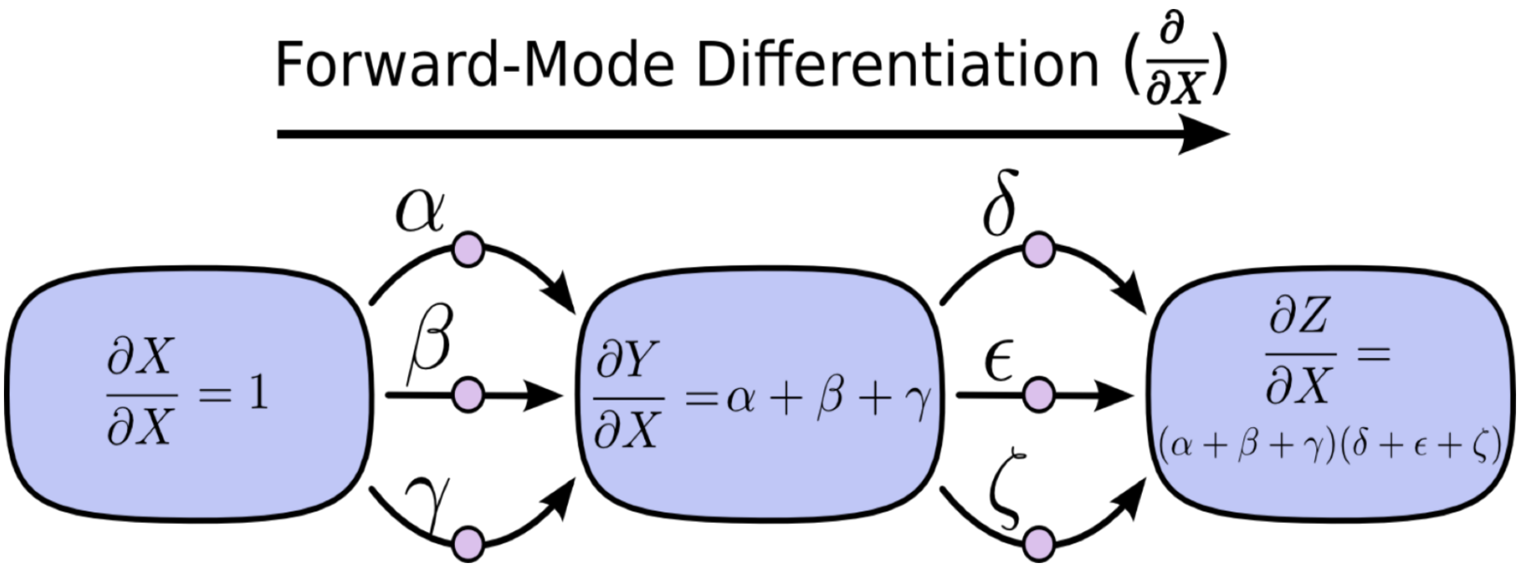
计算图



$$\frac{\partial Z}{\partial X} = \alpha\delta + \alpha\epsilon + \alpha\zeta + \beta\delta + \beta\epsilon + \beta\zeta + \gamma\delta + \gamma\epsilon + \gamma\zeta$$

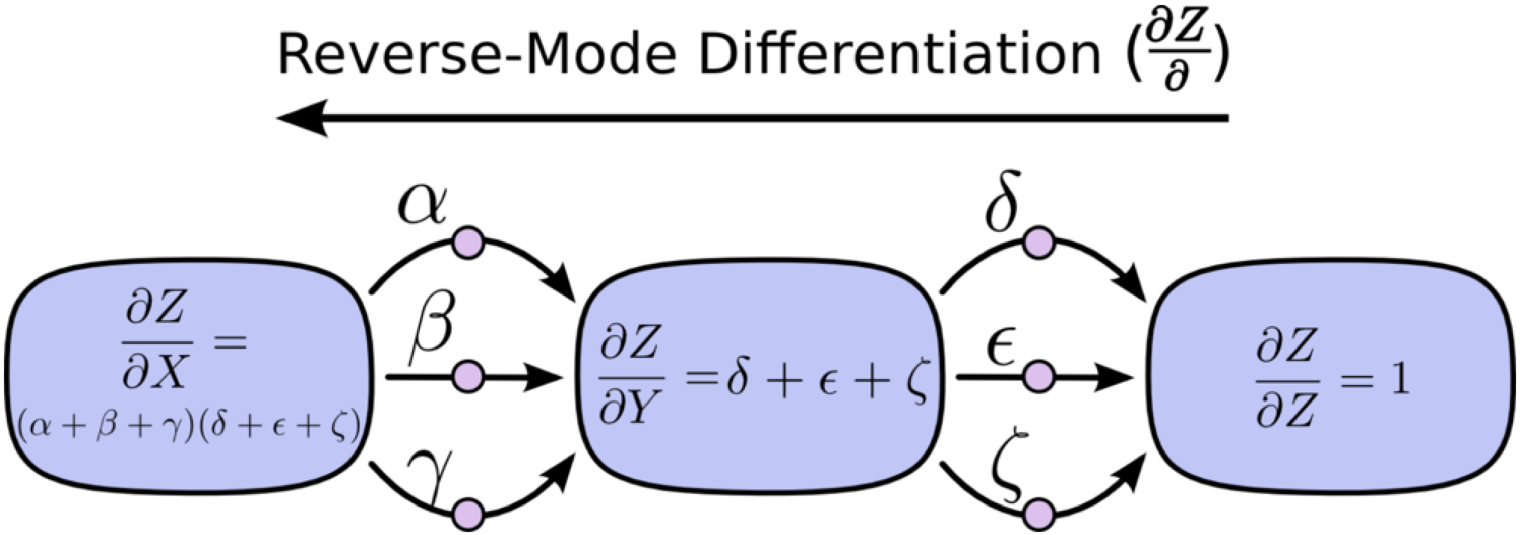
$$\frac{\partial Z}{\partial X} = (\alpha + \beta + \gamma)(\delta + \epsilon + \zeta)$$

计算图



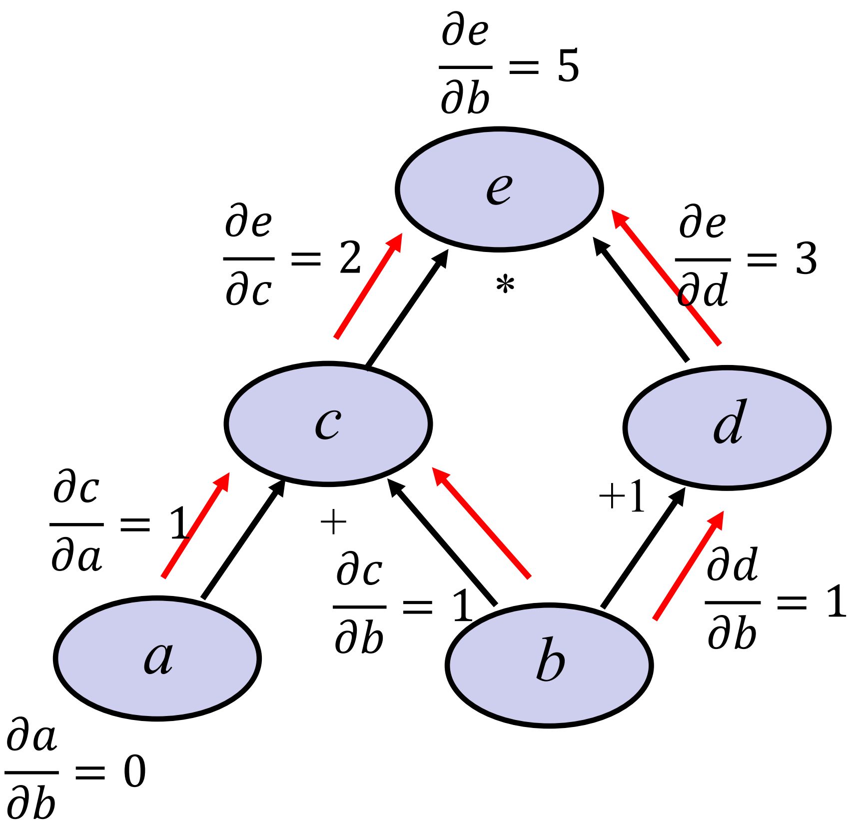
Forward-Mode Differentiation $(\frac{\partial}{\partial X})$
α
β
γ
δ
ϵ
ζ
$\frac{\partial X}{\partial X} = 1$
$\frac{\partial Y}{\partial X} = \alpha + \beta + \gamma$
$\frac{\partial Z}{\partial X} = (\alpha + \beta + \gamma)(\delta + \epsilon + \zeta)$

计算图

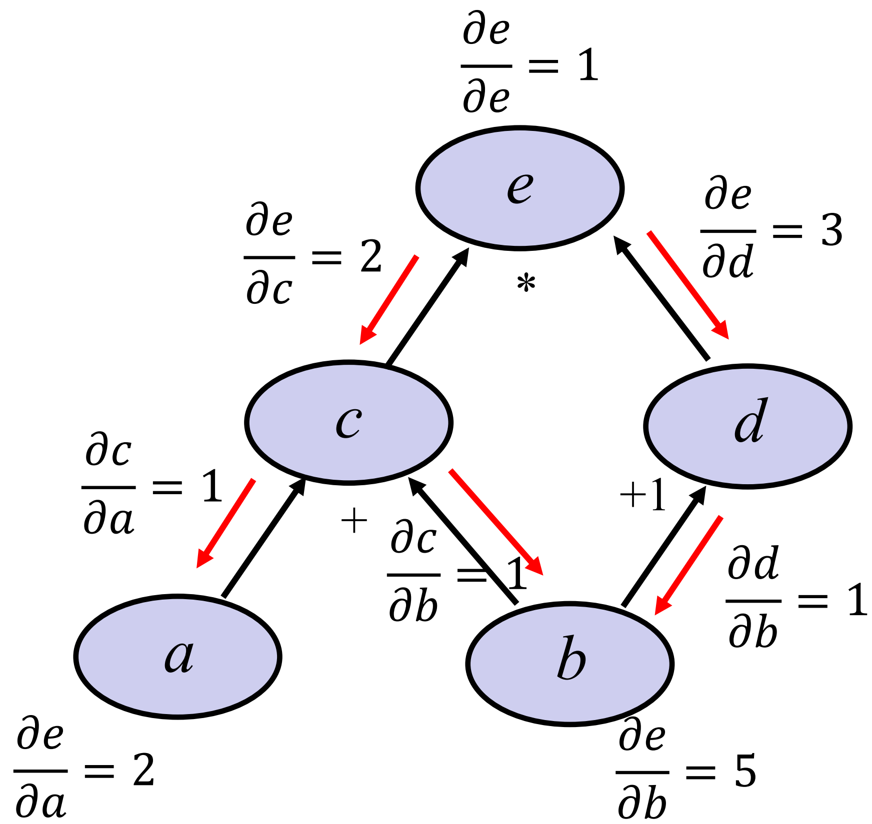

Reverse-Mode Differentiation ($\frac{\partial Z}{\partial}$)
α
β
γ
δ
ϵ
ζ
$\frac{\partial Z}{\partial X} = (\alpha + \beta + \gamma)(\delta + \epsilon + \zeta)$
$\frac{\partial Z}{\partial Y} = \delta + \epsilon + \zeta$
$\frac{\partial Z}{\partial Z} = 1$

计算图

- ❑ ***a* = 2, *b* = 1**



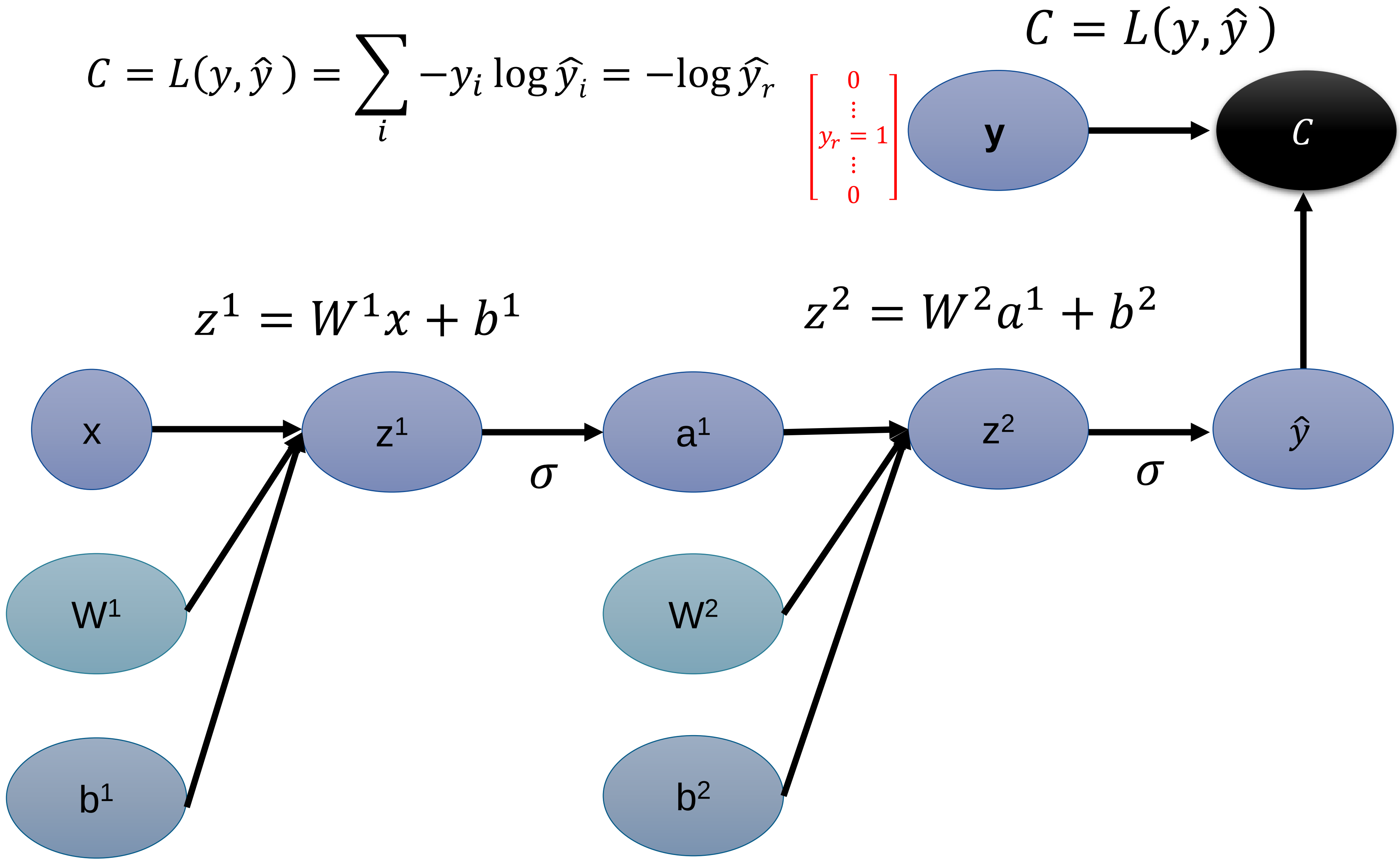
Forward mode



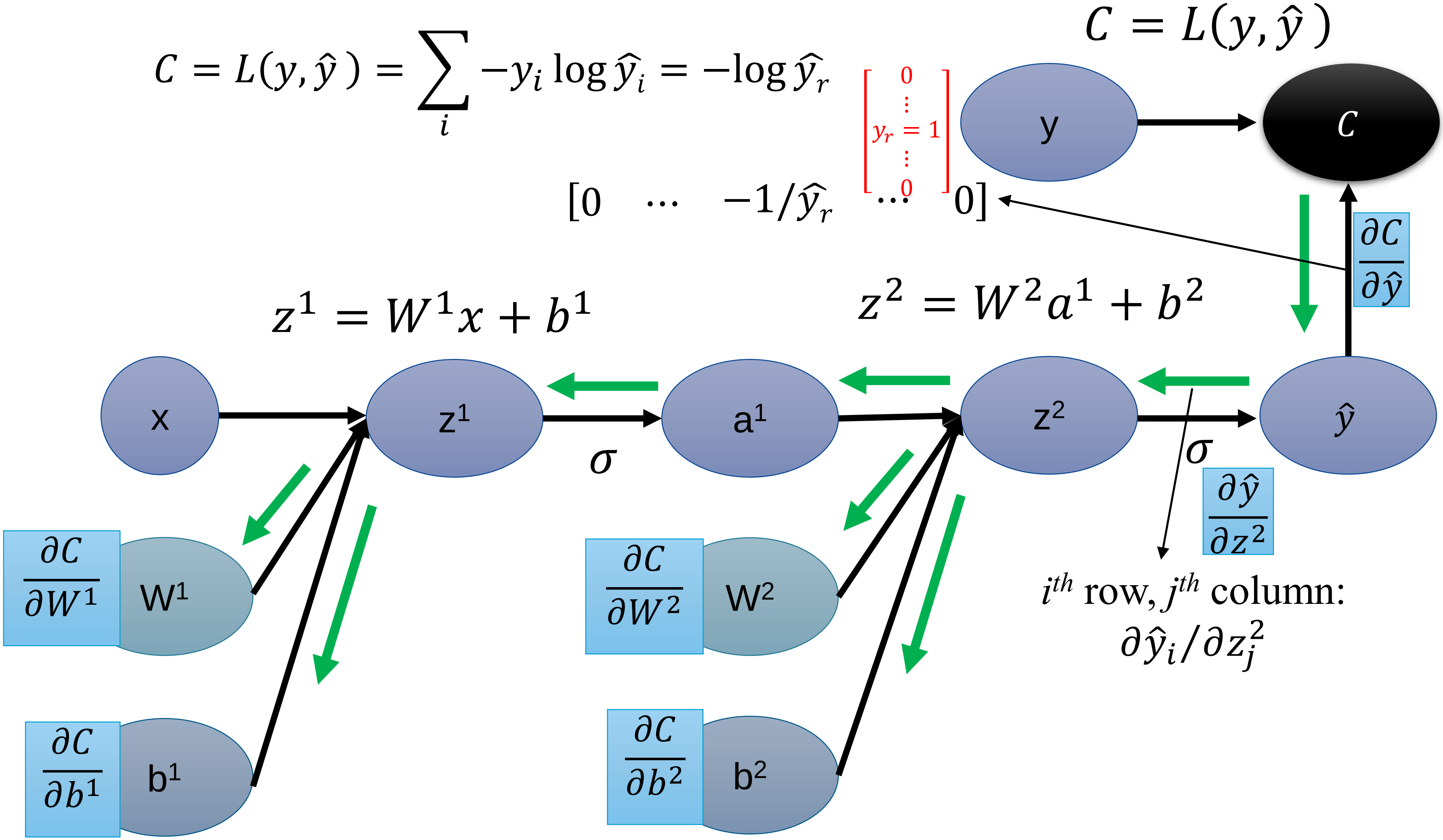
Reverse mode

计算图

$$C = L(y,\hat{y}) = \sum_i -y_i \log \hat{y}_i = -\log \hat{y}_r$$



计算图

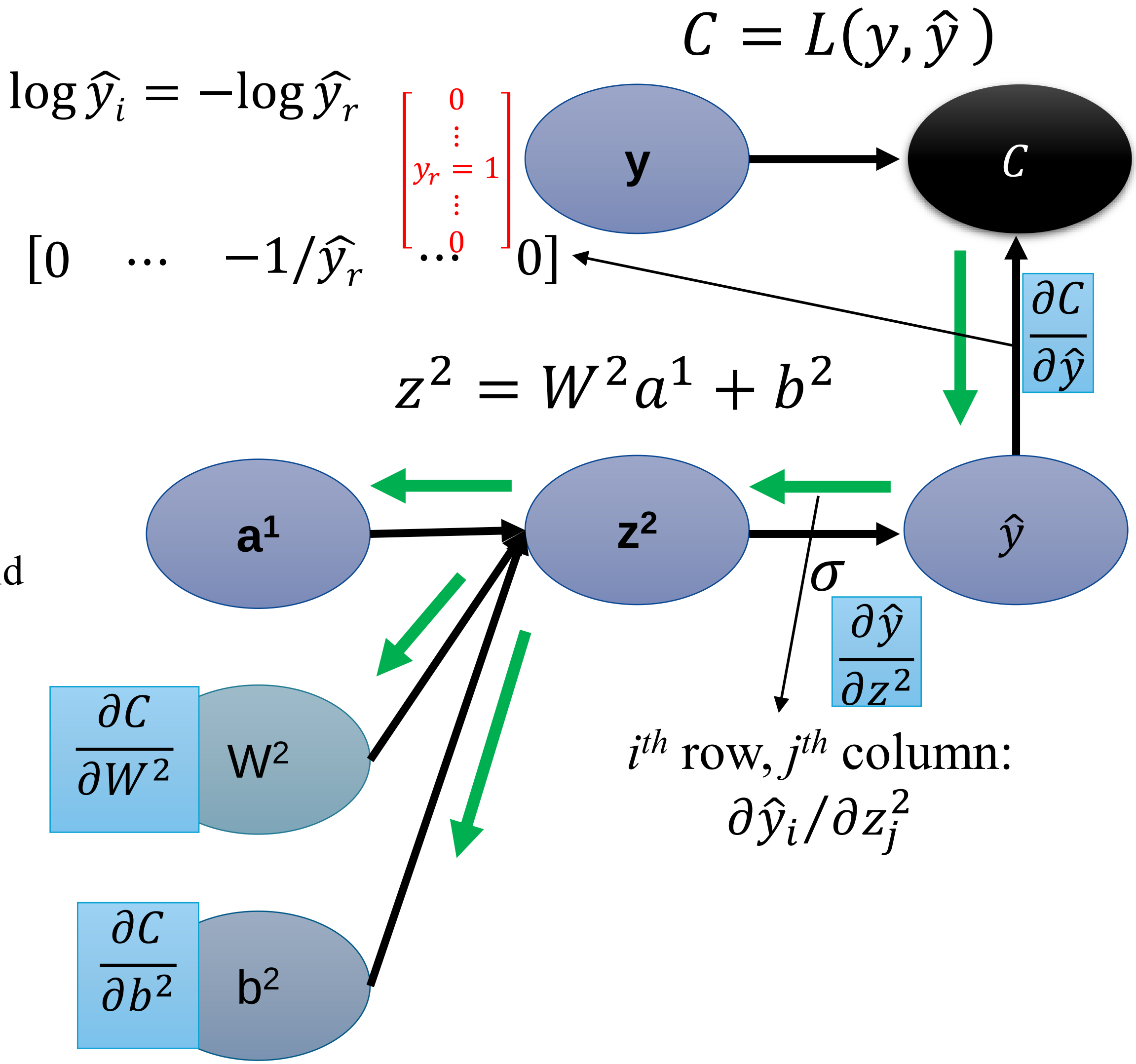


$C = L(y, \hat{y}) = \sum_i -y_i \log \hat{y}_i = -\log \hat{y}_r$
$C = L(y, \hat{y})$
$\begin{bmatrix} 0 \\ \vdots \\ y_r = 1 \\ \vdots \\ 0 \end{bmatrix}$
$[0 \quad \cdots \quad -1/\hat{y}_r \quad \cdots \quad 0]$
$z^1 = W^1 x + b^1$
$z^2 = W^2 a^1 + b^2$
x
z^1
a^1
z^2
$\hat{y}$
y
C
σ
σ
$\frac{\partial C}{\partial \hat{y}}$
$\frac{\partial \hat{y}}{\partial z^2}$
i^{th} row, j^{th} column:
$\partial \hat{y}_i / \partial z_j^2$
$\frac{\partial C}{\partial W^1}$
W^1
$\frac{\partial C}{\partial b^1}$
b^1
$\frac{\partial C}{\partial W^2}$
W^2
$\frac{\partial C}{\partial b^2}$
b^2

计算图

$$C = L(y, \hat{y}) = \sum_i -y_i \log \hat{y}_i = -\log \hat{y}_r$$

$$\frac{\partial \hat{\boldsymbol{y}}}{\partial \boldsymbol{z}^2} = \left.\underbrace{\boxed{}}_{|\boldsymbol{z}^2|}\right\} |\hat{\boldsymbol{y}}|$$



例如：当激活函数是ReLU或者Sigmoid

- $i \neq j$: $\partial \hat{y}_i / \partial z_j^2 = 0$
- $i = j$: $\partial \hat{y}_i / \partial z_i^2 = \sigma'(z_i^2)$

2

循环神经网络

(Recurrent Neural Network)

为什么需要RNN

- **CNN已经取得巨大成功，为何还需要RNN?**
- **序列数据建模**
  - 文本：是字母和词汇的序列
  - 语音：是音节的序列
  - 视频：图像帧的序列
  - 时态数据：气象观测数据，股票交易数据、房价数据等
- **例子：词性标注**
  - 我/n,爱/v 购物/n,
  - 我/n在/pre华联/n购物/v

RNN发展史

J. Hopfield

Hopfield networks

1982

J. Elman

Elman Network

1990

1986

Jordan Network

M. Jordan

1990

BPTT

P. Werbos

早期（1980-1990）

Hochreiter & Schmidhube

LSTM

1997

BRNN

Schuster & Paliwal

中期（1990-2010）

Cho, et al

GRU

2014

Joulin & Mikolov

Stack RNN

2015

Neural turing machine

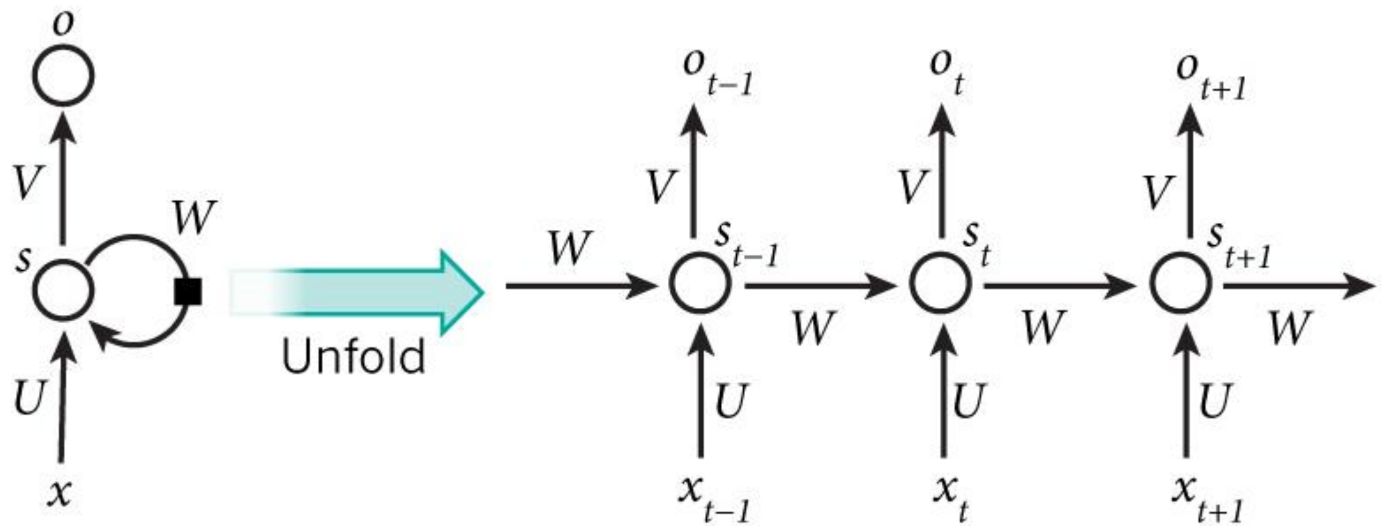
A. Graves

当前（2010 - ）

循环神经网络的定义

- 循环神经网络是一种人工神经网络，它的节点间的连接形成一个遵循时间序列的有向图
  - A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a temporal sequence
- 核心思想
  - 样本间存在顺序关系，每个样本和它之前的样本存在关联。通过神经网络在时序上的展开，我们能够找到样本之间的序列相关性

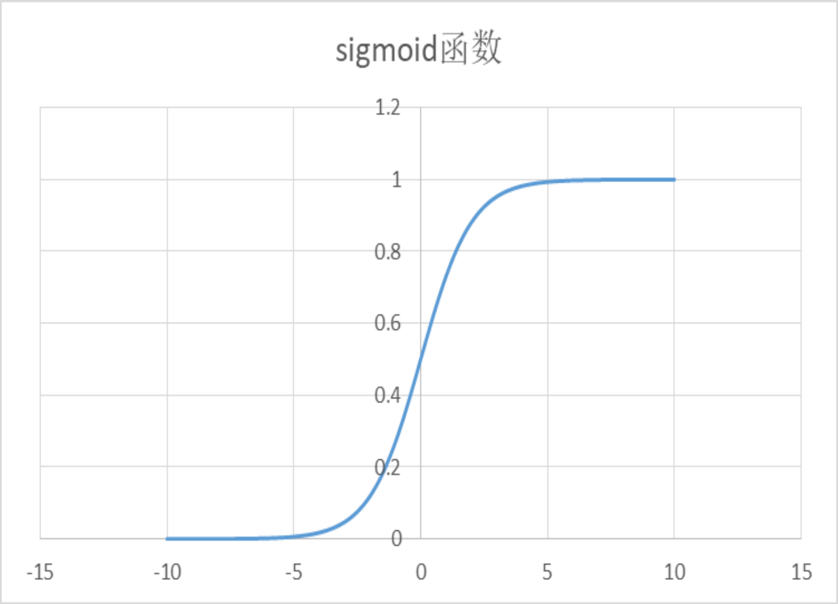
RNN的一般结构

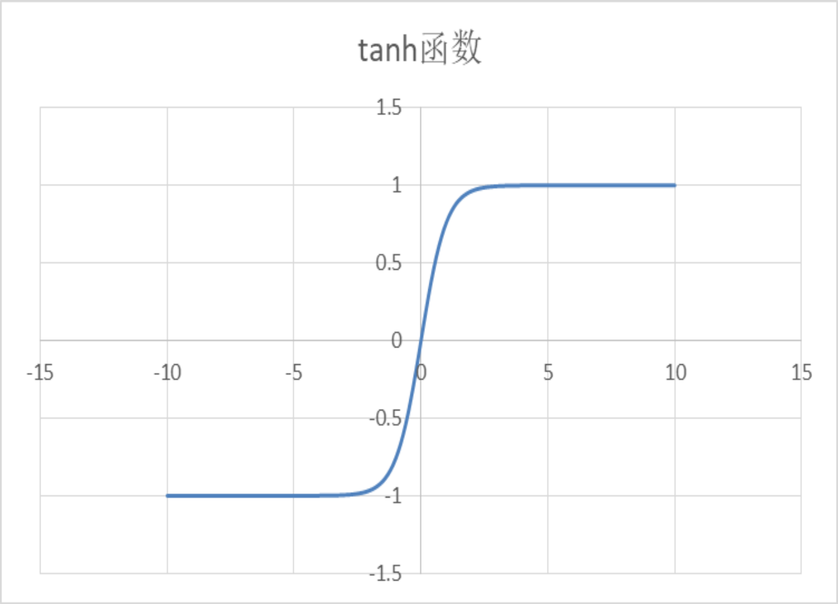


$$s_t = \sigma(Ux_t + Ws_{t-1} + b_s)$$
$$o_t = \varphi(Vs_t + b_o)$$

- x_t是t时刻的输入
- s_t是t时刻的记忆
- o_t是t时刻的输出
- U、V、W是RNN的连接权重
- b_s、b_o是RNN的偏置
- σ、φ是激活函数，σ通常选用tanh或Sigmoid，φ通常选用Softmax

RNN的一般结构





$$\mathrm{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

RNN的一般结构

❑ Softmax函数

- 用于分类问题的概率计算。本质上将一个K维的任意实数向量压缩（映射）成另一个K维的实数向量，其中向量中的每个元素取值都介于（0，1）之间

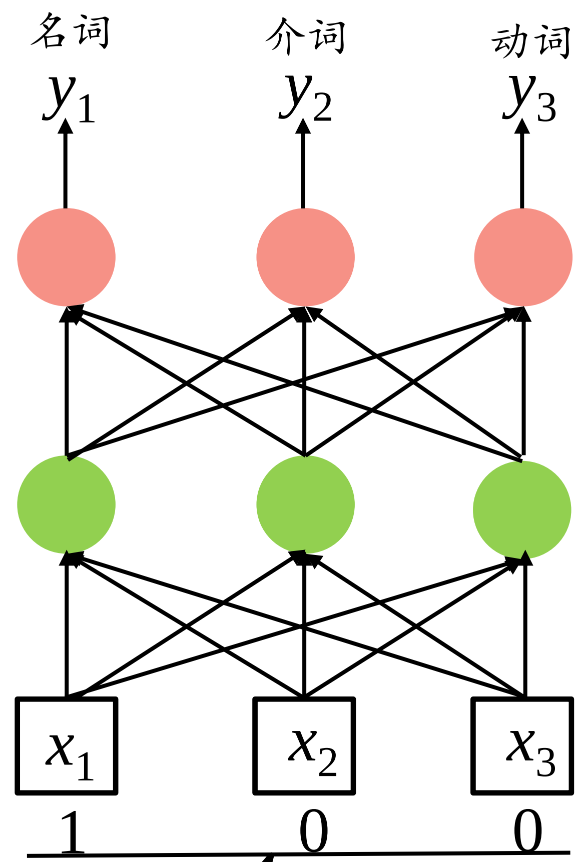
$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

RNN示例

- 词性标注
  - 我/n,爱/v 购物/n,
  - 我/n在/pre华联/n购物/v

Word Embedding：自然语言处理（NLP）中的一组语言建模和特征学习技术的统称，其中来自词汇表的单词或短语被映射到实数的向量

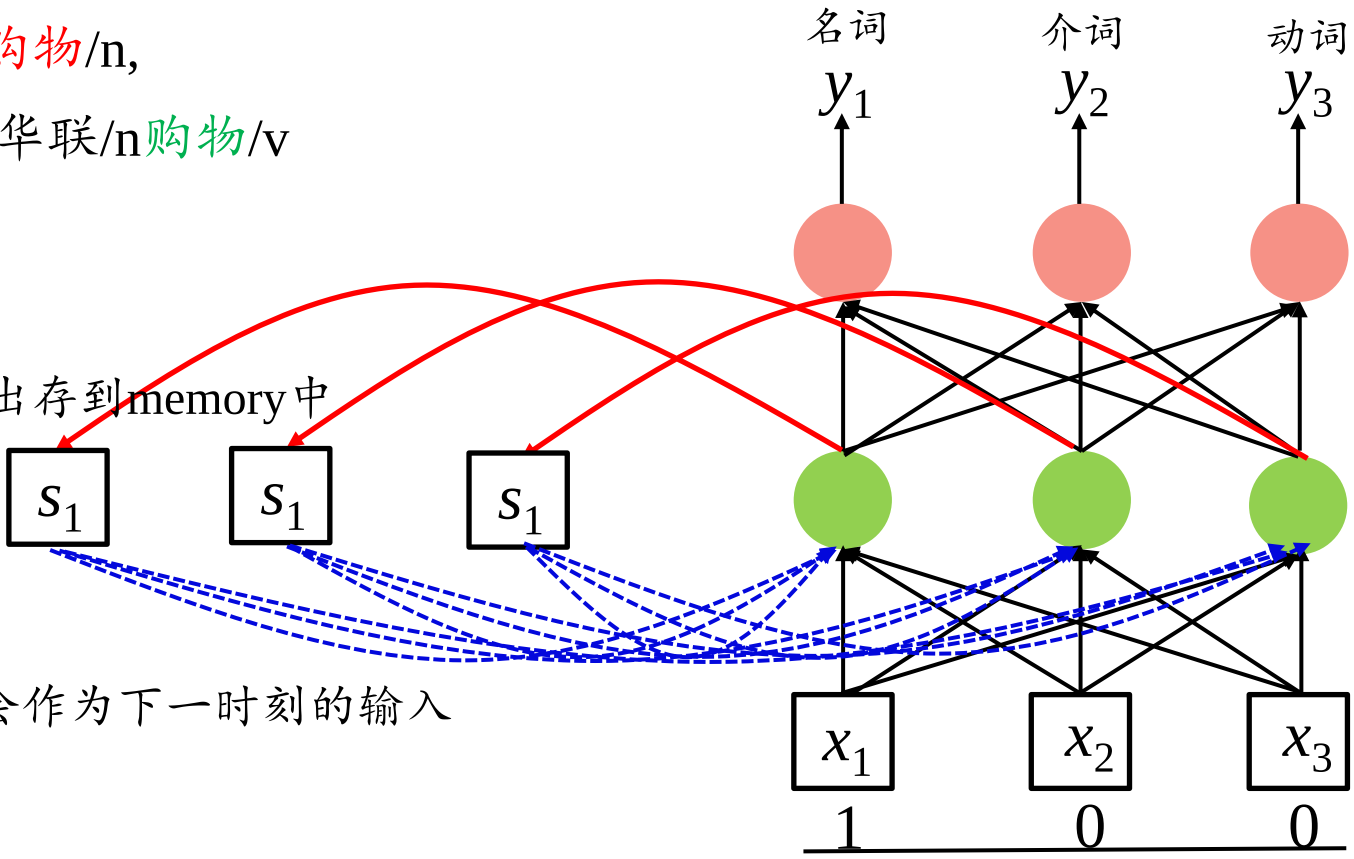
$$\begin{bmatrix}1\\0\\0\end{bmatrix} \quad \begin{bmatrix}0\\1\\0\end{bmatrix} \quad \begin{bmatrix}0\\0\\1\end{bmatrix}$$



RNN示例

- **词性标注**
  - 我/n,爱/v 购物/n,
  - 我/n在/pre华联/n购物/v

将神经元的输出存到memory中

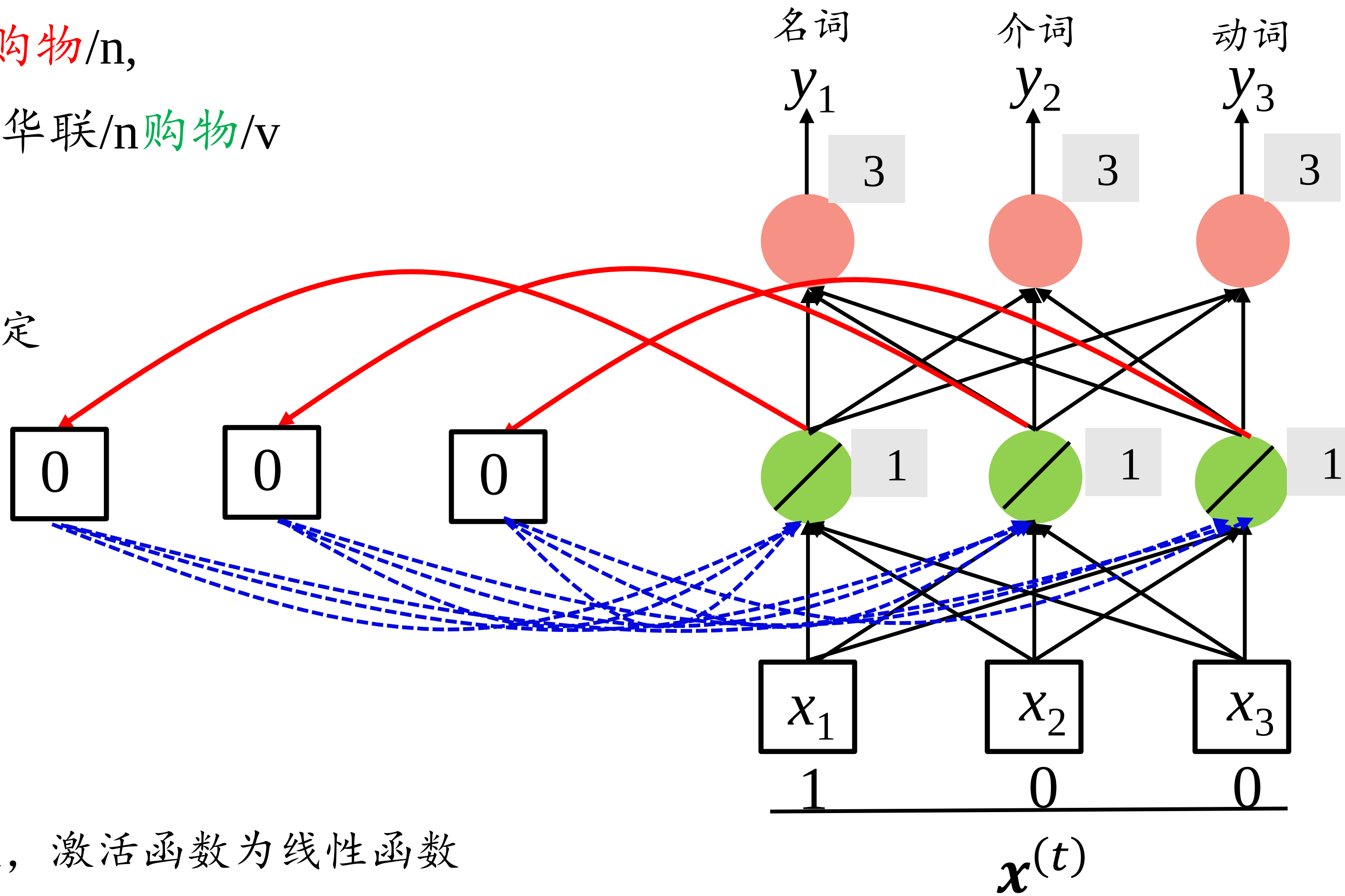


memory中值会作为下一时刻的输入

RNN示例

- 词性标注
  - 我/n,爱/v 购物/n,
  - 我/n在/pre华联/n购物/v

在最开始时刻，给定memory初始值为0

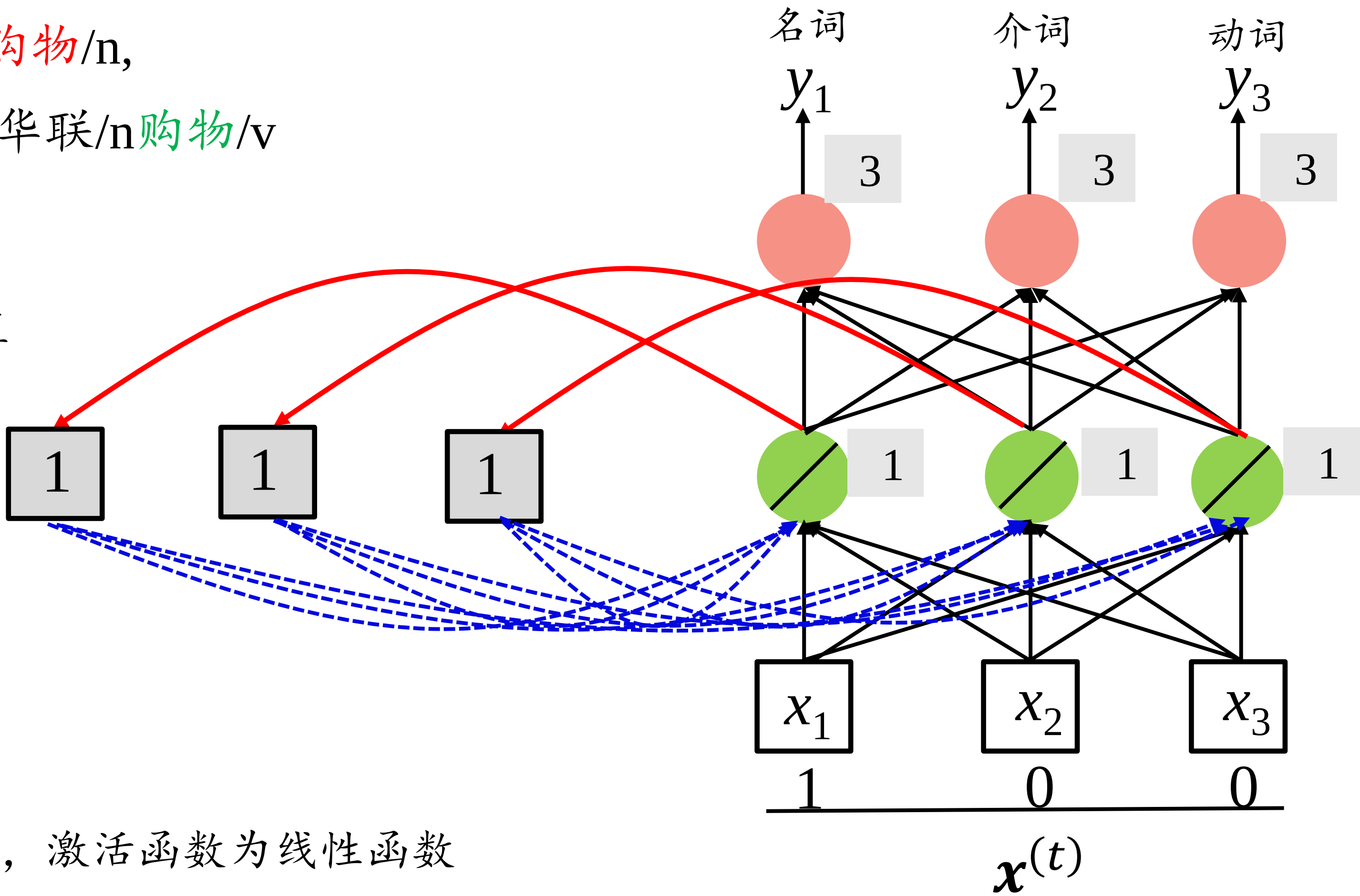


假设所有权重均为1，激活函数为线性函数

RNN示例

- 词性标注
  - 我/n,爱/v 购物/n,
  - 我/n在/pre华联/n购物/v

更新memory中的值



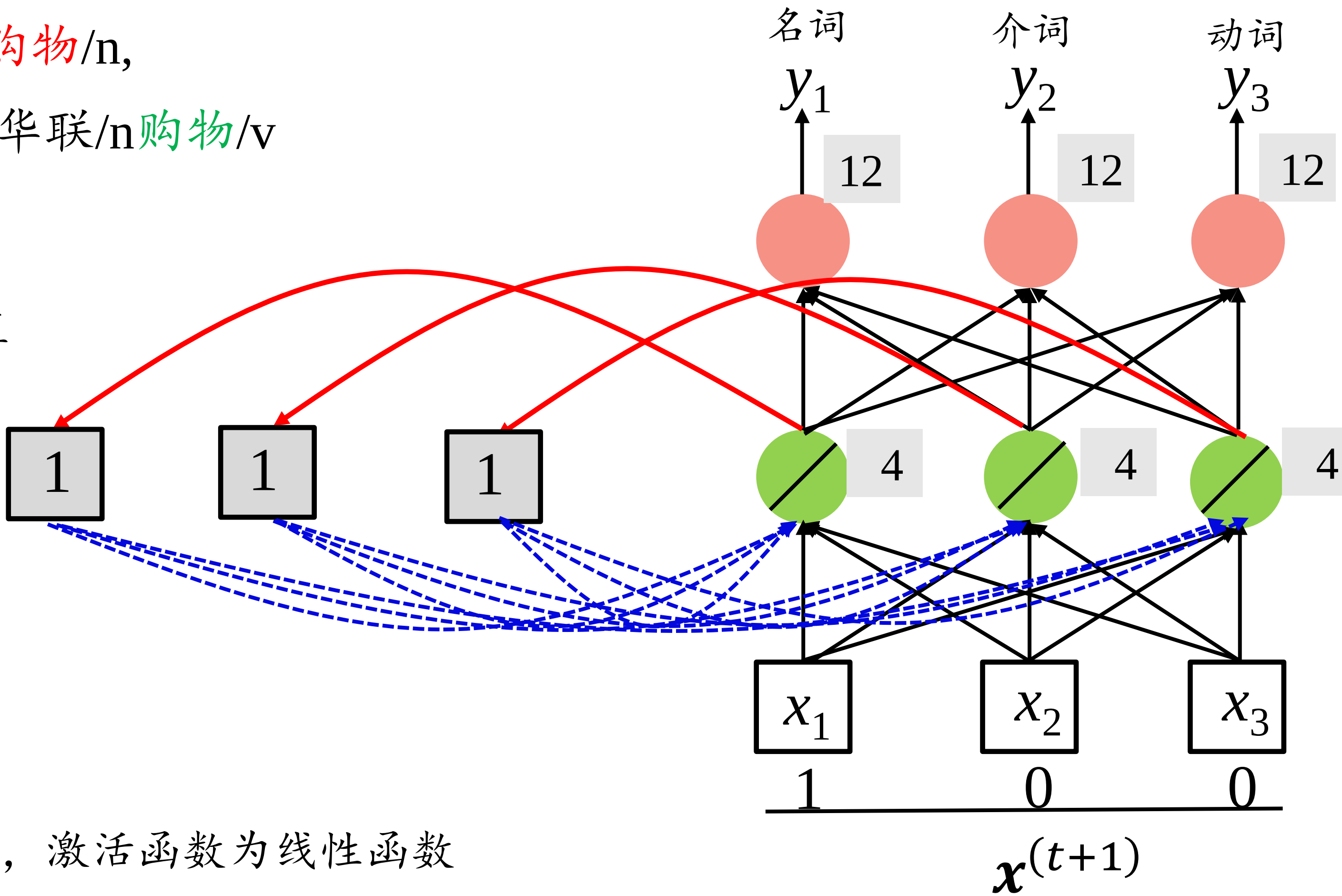
假设所有权重均为1，激活函数为线性函数

RNN示例

❑ 词性标注

- 我/n,爱/v 购物/n,
- 我/n在/pre华联/n购物/v

更新memory中的值

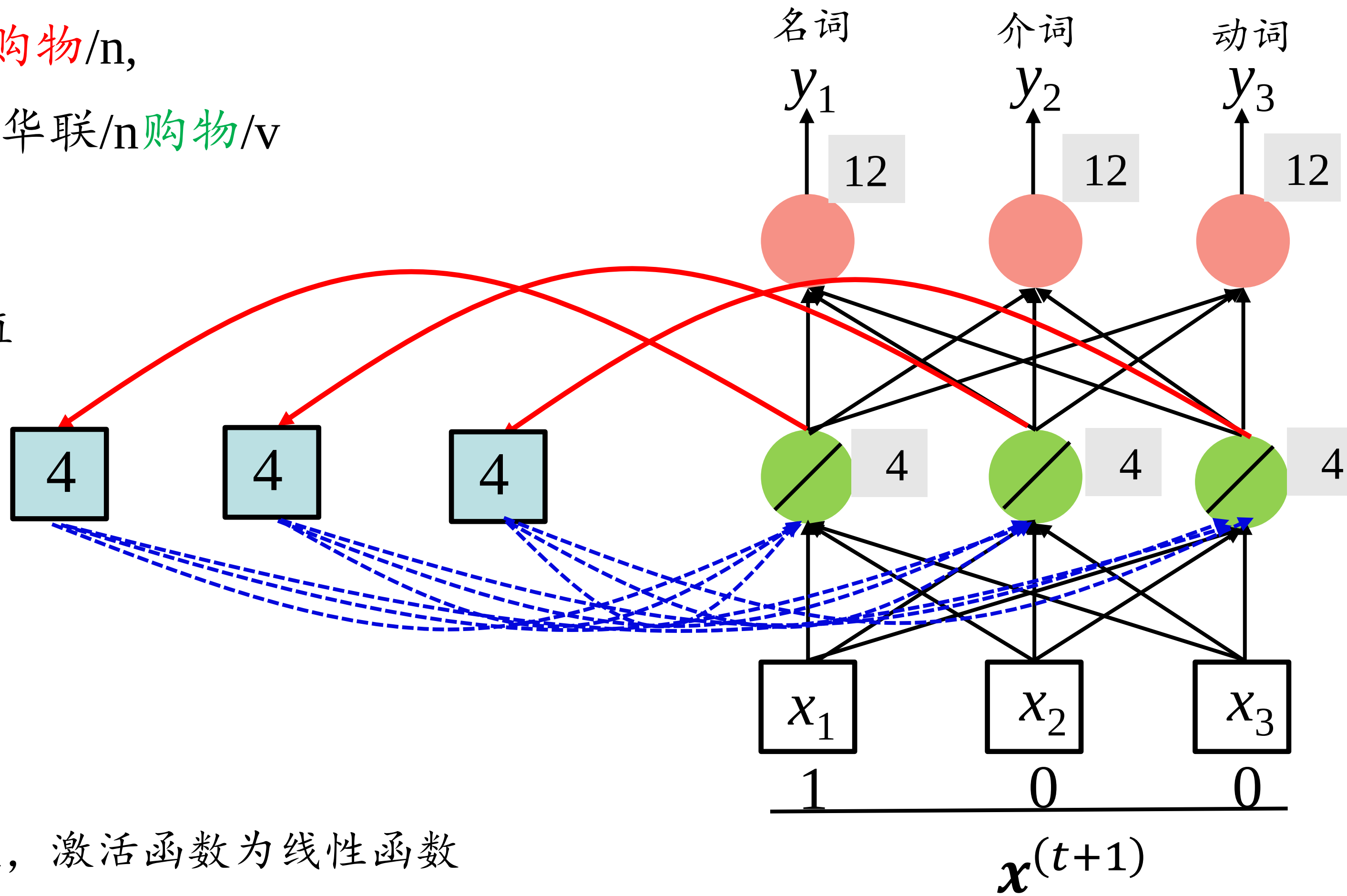


假设所有权重均为1，激活函数为线性函数

RNN示例

- 词性标注
  - 我/n,爱/v 购物/n,
  - 我/n在/pre华联/n购物/v

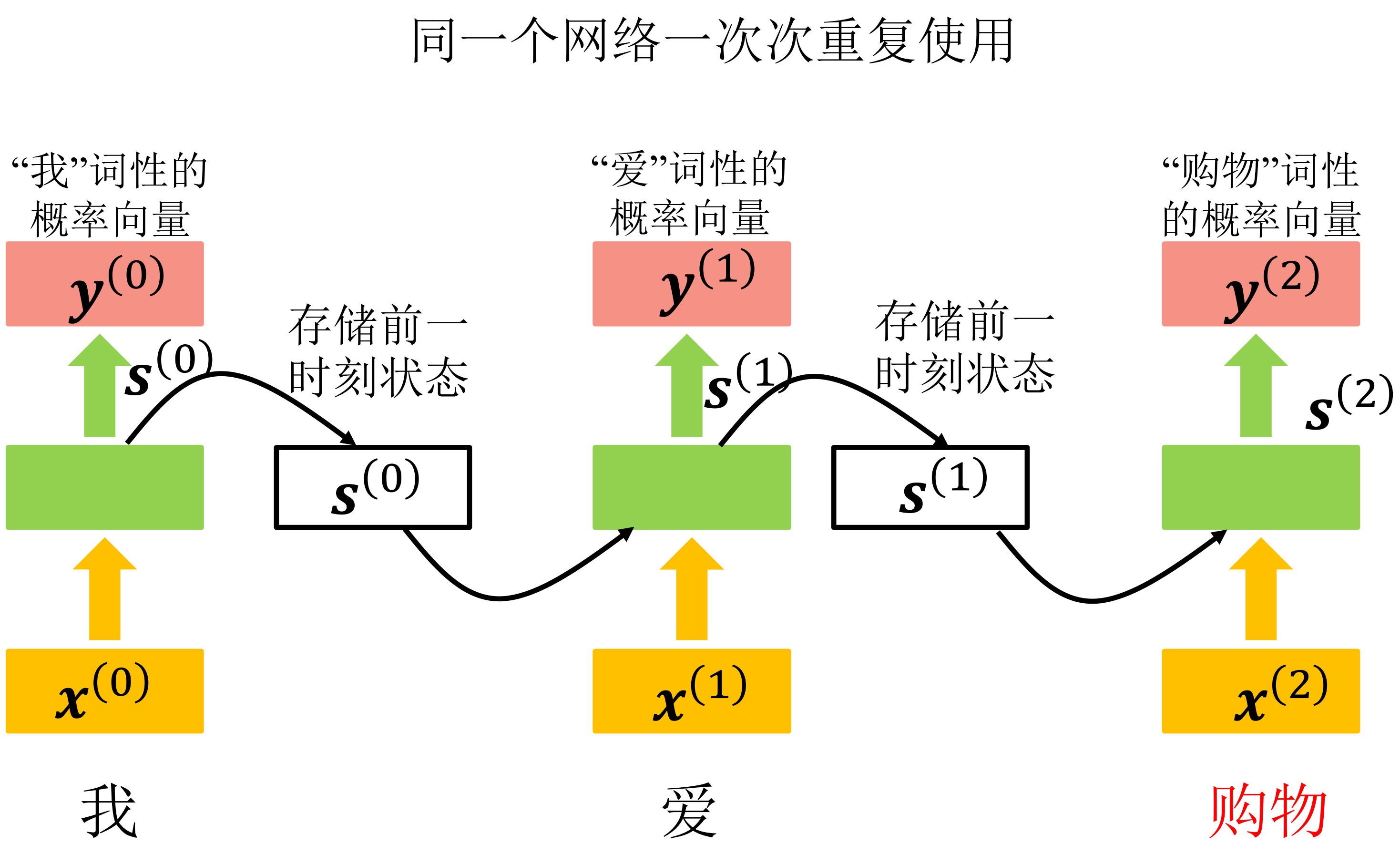
更新memory中的值



假设所有权重均为1，激活函数为线性函数

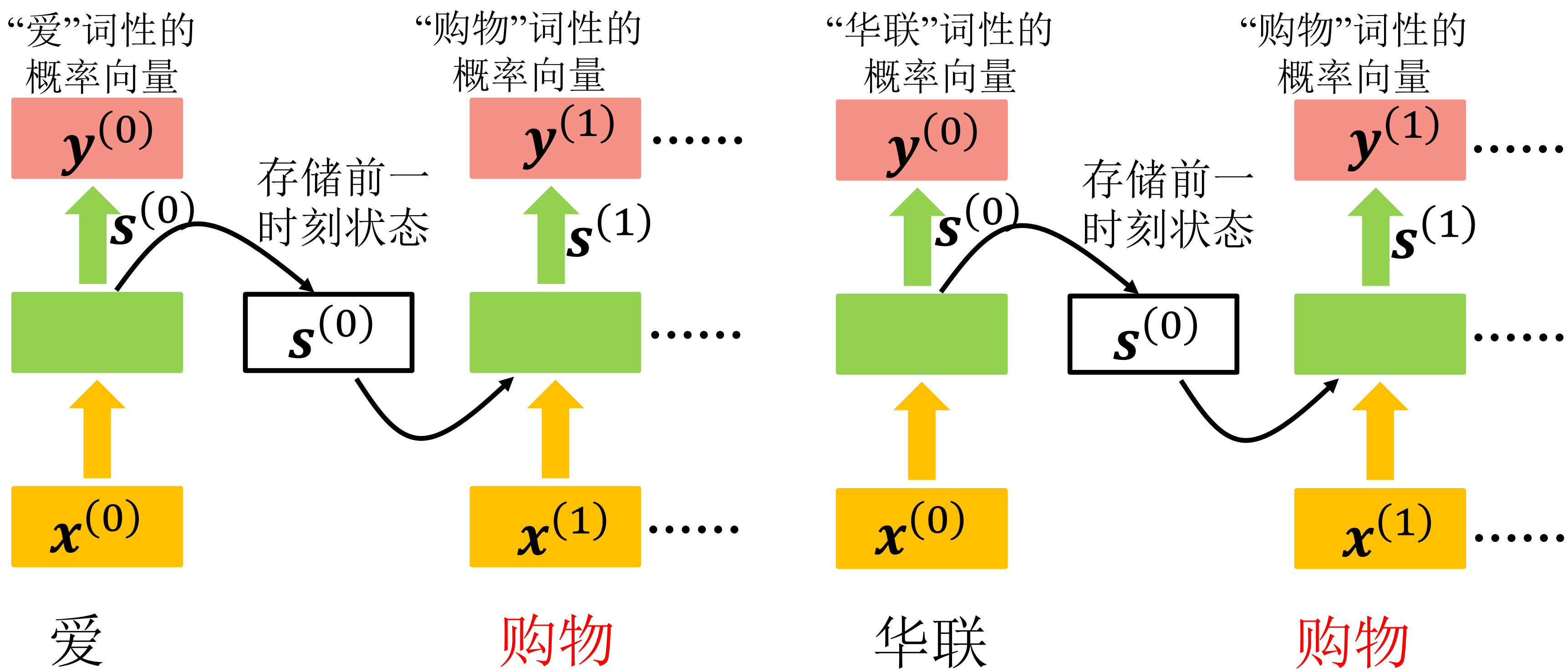
RNN示例

❑ 词性标注


同一个网络一次次重复使用
“我”词性的概率向量
“爱”词性的概率向量
“购物”词性的概率向量
$\boldsymbol{y}^{(0)}$
$\boldsymbol{y}^{(1)}$
$\boldsymbol{y}^{(2)}$
存储前一时刻状态
存储前一时刻状态
$\boldsymbol{s}^{(0)}$
$\boldsymbol{s}^{(1)}$
$\boldsymbol{s}^{(2)}$
$\boldsymbol{s}^{(0)}$
$\boldsymbol{s}^{(1)}$
$\boldsymbol{x}^{(0)}$
$\boldsymbol{x}^{(1)}$
$\boldsymbol{x}^{(2)}$
我
爱
购物

RNN示例

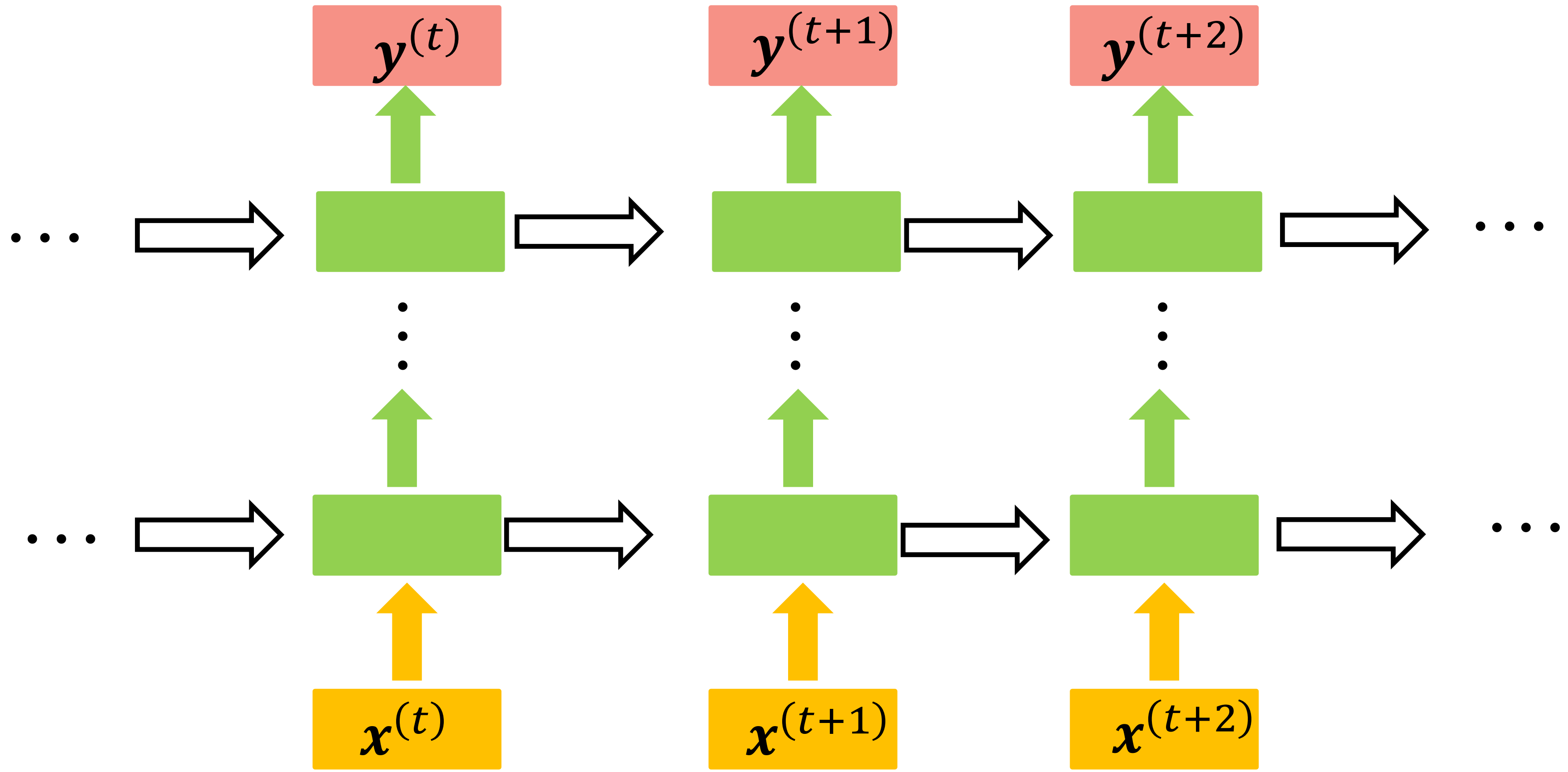
❑ 词性标注



“爱”词性的
概率向量
$y^{(0)}$
$s^{(0)}$
存储前一
时刻状态
$s^{(0)}$
$x^{(0)}$
爱
“购物”词性的
概率向量
$y^{(1)}$
$s^{(1)}$
$x^{(1)}$
购物
……
“华联”词性的
概率向量
$y^{(0)}$
$s^{(0)}$
存储前一
时刻状态
$s^{(0)}$
$x^{(0)}$
华联
“购物”词性的
概率向量
$y^{(1)}$
$s^{(1)}$
$x^{(1)}$
购物
……

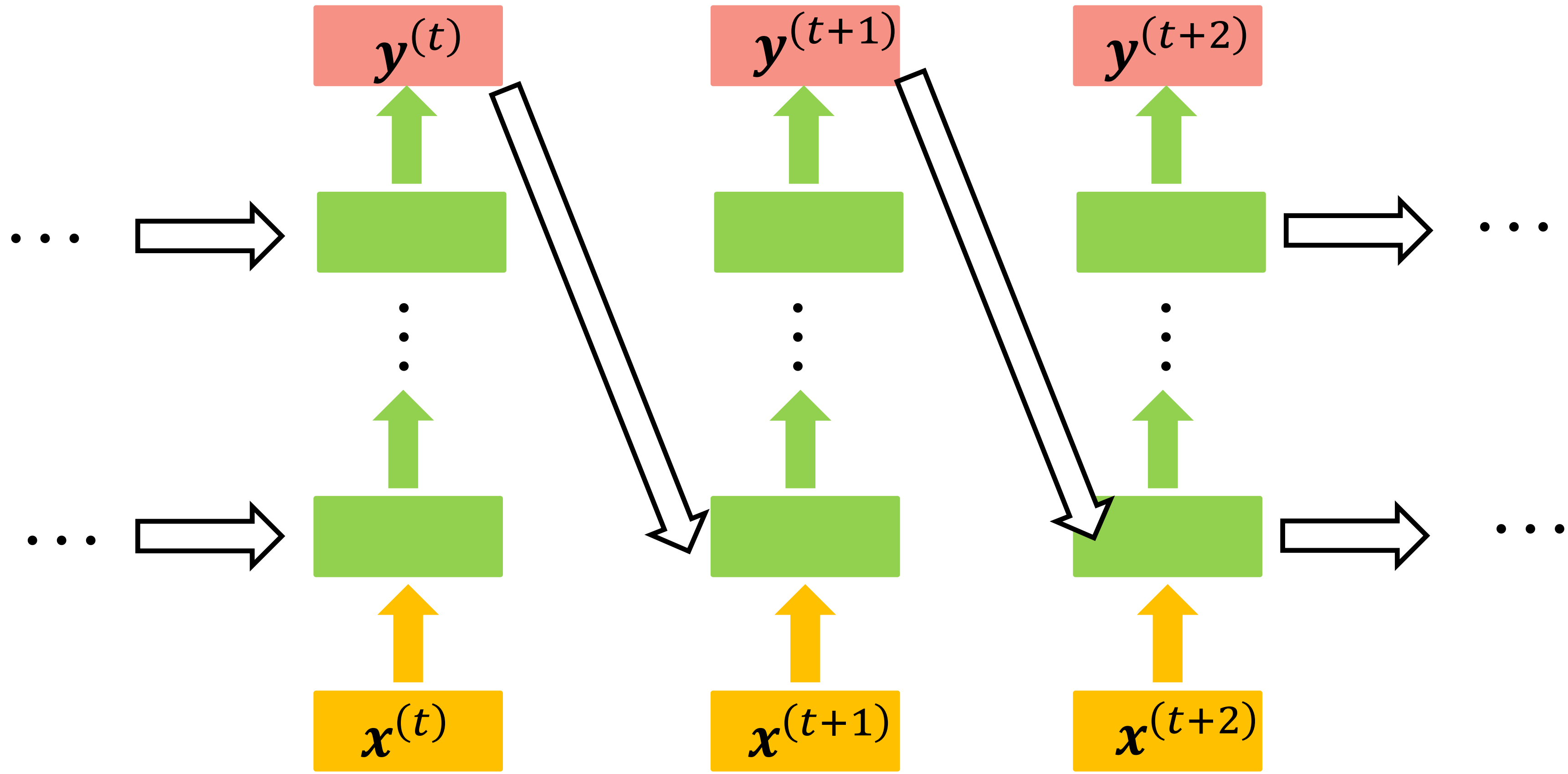
RNN的一般结构

❑ Elman Network



RNN的一般结构

- Jordan Network

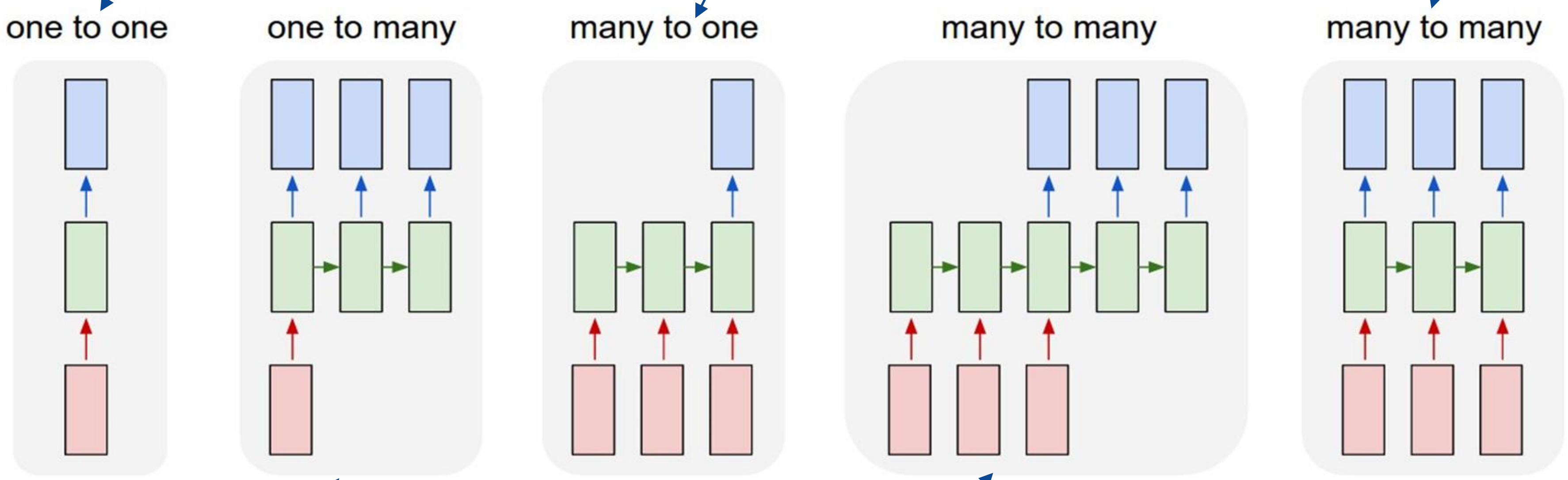


RNN的不同结构

Vanilla RNN

e.g. Sentiment Classification
sequence of words -> sentiment

e.g. Video classification
on frame level



e.g. Image Captioning
image -> sequence of words

e.g. Machine Translation
sequence of words -> sequence of words

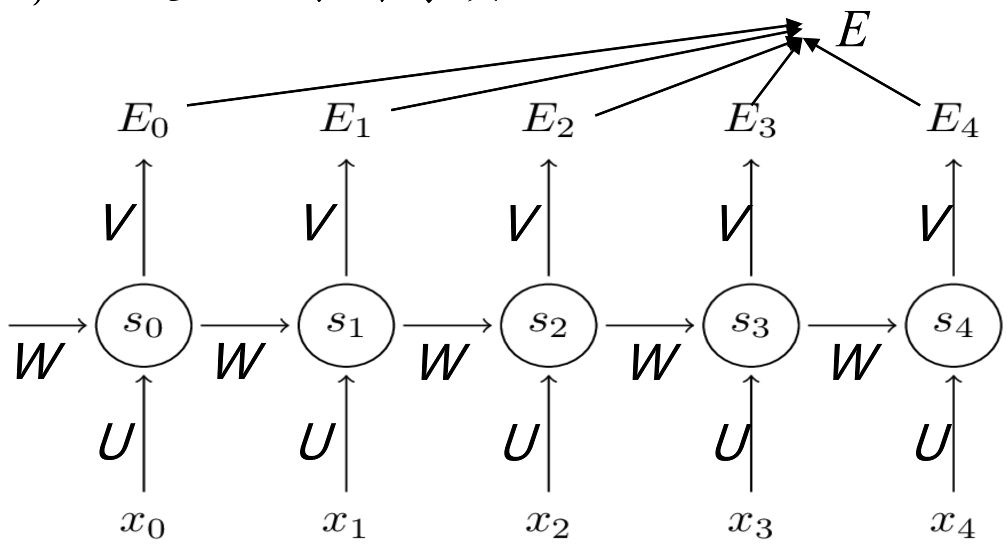
RNN训练算法-BPTT

❑ BP算法回顾

– 定义损失函数E来表示输出$\hat{y}$和真实标签y的误差，通过链式法则自顶向下求得E对网络权重的偏导。沿梯度的反方向更新权重的值，直到E收敛

❑ RNN训练算法（BP Through Time, BPTT）

– 和BP类似，就是加上了时序演化



RNN训练算法-BPTT

- 定义输出函数

$$s_t = tanh(Ux_t + Ws_{t-1})$$

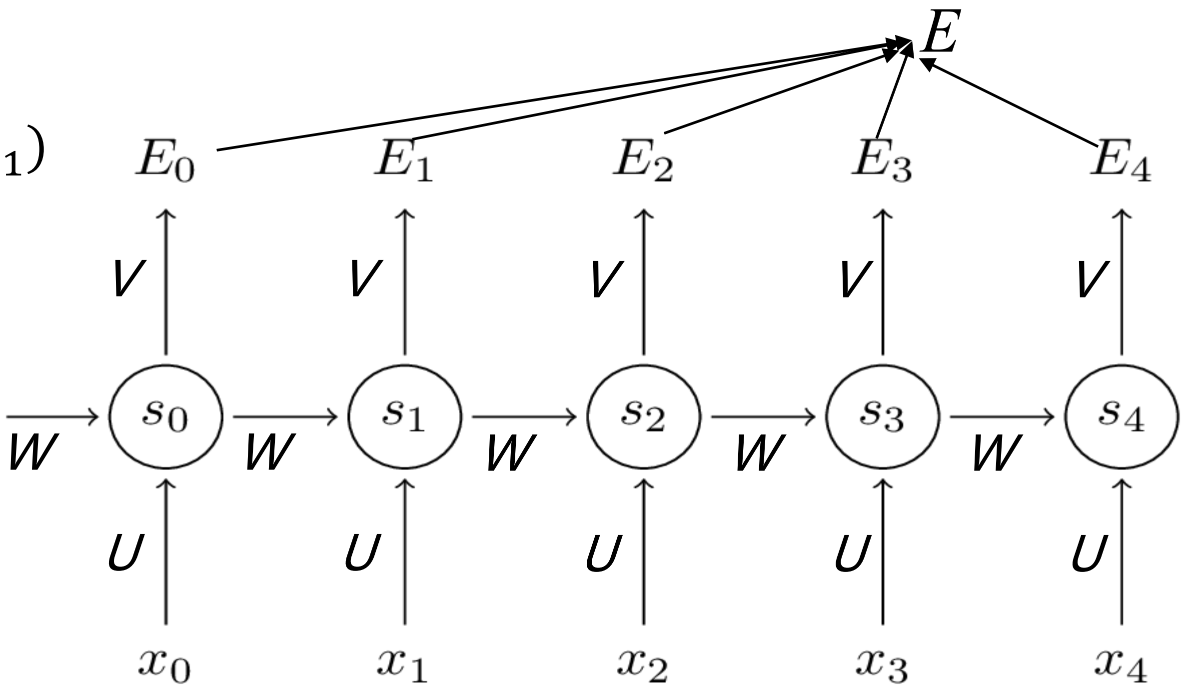
$$\hat{y}_t = softmax(Vs_t)$$

- 定义损失函数

$$E_t(y_t, \hat{y}_t) = -y_t \log \hat{y}_t$$

$$E(y, \hat{y}) = \sum_t E_t(y_t, \hat{y}_t)$$

$$= -\sum_t y_t \log \hat{y}_t$$



http://www.wildml.com/2015/10/recurrent-neural-networks-tutorial-part-3-backpropagation-through-time-and-vanishing-gradients/

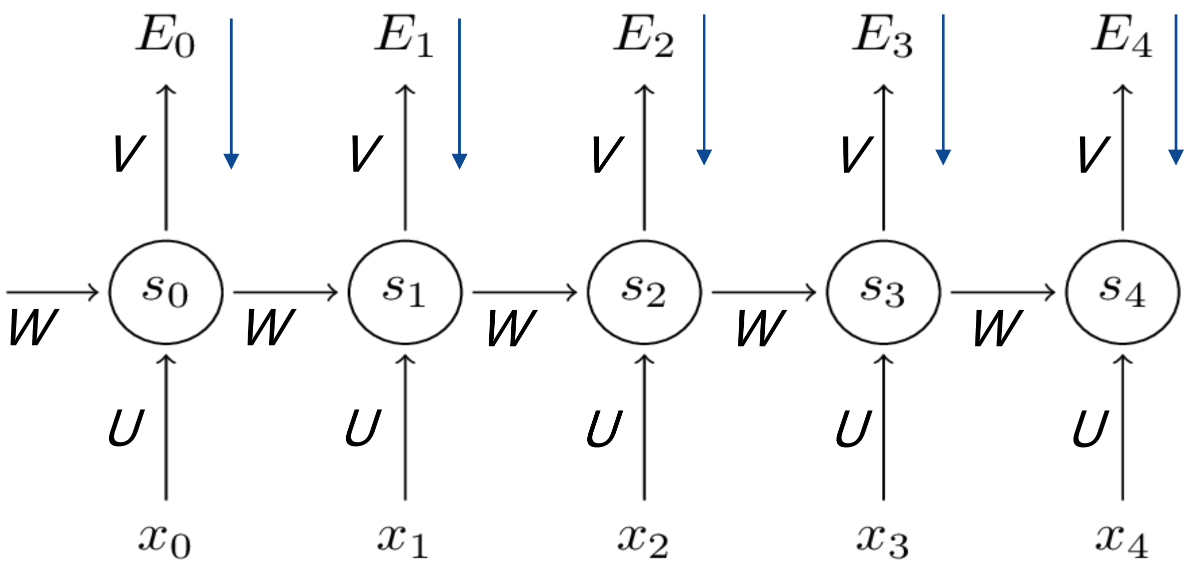
RNN训练算法-BPTT

- 求 E 对U，V，W 的梯度

$$\frac{\partial E}{\partial V}=\sum_t \frac{\partial E_t}{\partial V}$$

$$\frac{\partial E}{\partial W}=\sum_t \frac{\partial E_t}{\partial W}$$

$$\frac{\partial E}{\partial U}=\sum_t \frac{\partial E_t}{\partial U}$$

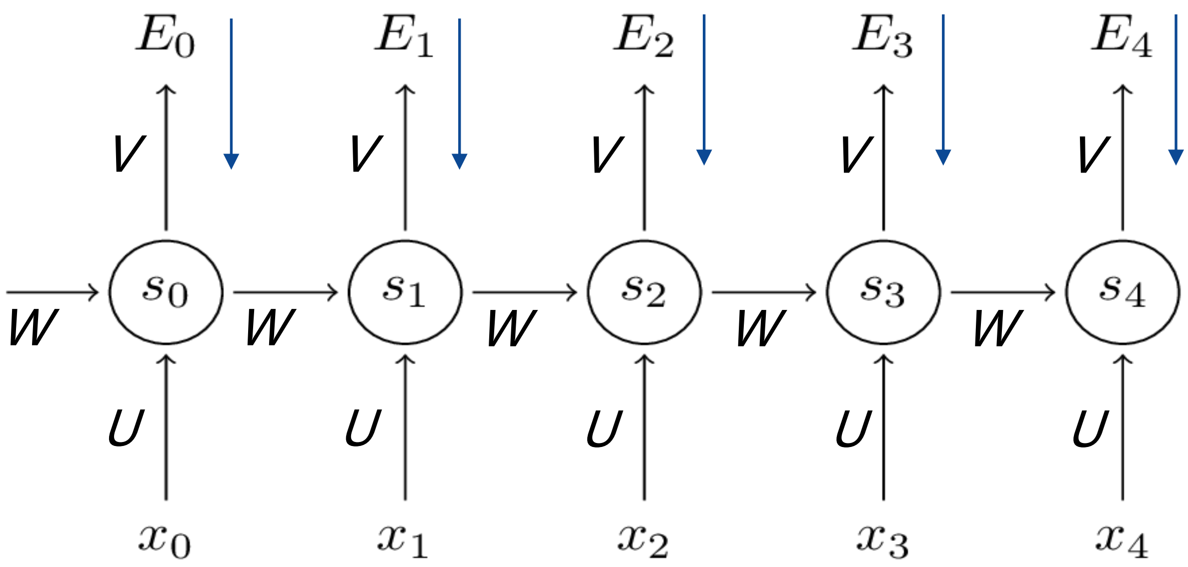


RNN训练算法-BPTT

- 求 E 对于V的梯度，先求 E_3对于V的梯度

$$\frac{\partial E_3}{\partial V} = \frac{\partial E_3}{\partial \hat{y}_3}\frac{\partial \hat{y}_3}{\partial V}$$

$$= \frac{\partial E_3}{\partial \hat{y}_3}\frac{\partial \hat{y}_3}{\partial z_3}\frac{\partial z_3}{\partial V}$$



其中： $z_3 = V s_3$

求和可得 $\frac{\partial E}{\partial V}$

RNN训练算法-BPTT

- 求 E 对于W的梯度，先求 E_3 对于W的梯度

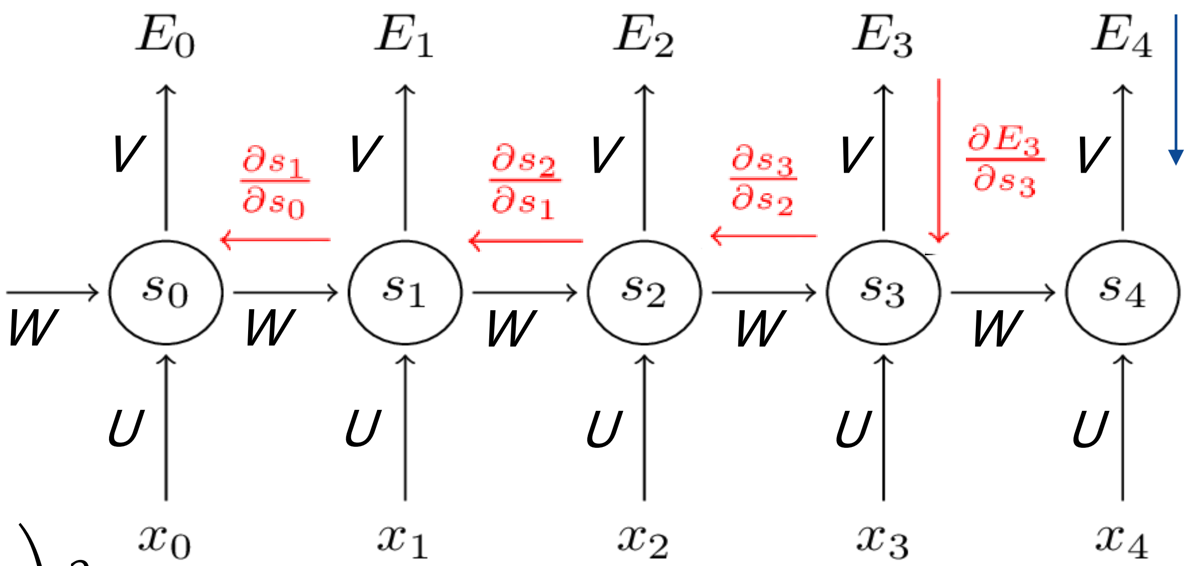
$$\frac{\partial E_3}{\partial W}=\frac{\partial E_3}{\partial \hat{y}_3}\frac{\partial \hat{y}_3}{\partial s_3}\frac{\partial s_3}{\partial W}$$

$$s_3=tanh(Ux_3+Ws_2)$$

$$\frac{\partial E_3}{\partial W}=\sum_{k=0}^{3}\frac{\partial E_3}{\partial \hat{y}_3}\frac{\partial \hat{y}_3}{\partial s_3}\frac{\partial s_3}{\partial s_k}\frac{\partial s_k}{\partial W}$$

$$\frac{\partial E_3}{\partial W}=\sum_{k=0}^{3}\frac{\partial E_3}{\partial \hat{y}_3}\frac{\partial \hat{y}_3}{\partial s_3}\left(\prod_{j=k+1}^{3}\frac{\partial s_j}{\partial s_{j-1}}\right)\frac{\partial s_k}{\partial W}$$

求和可得 $\frac{\partial E}{\partial W}$



其中：s_3依赖于s_2，而s_2又依赖于s_1和W ，依赖关系一直传递到t = 0的时刻。**因此，当我们计算对于*W*的偏导数时，不能把s_2看作是常数项！**

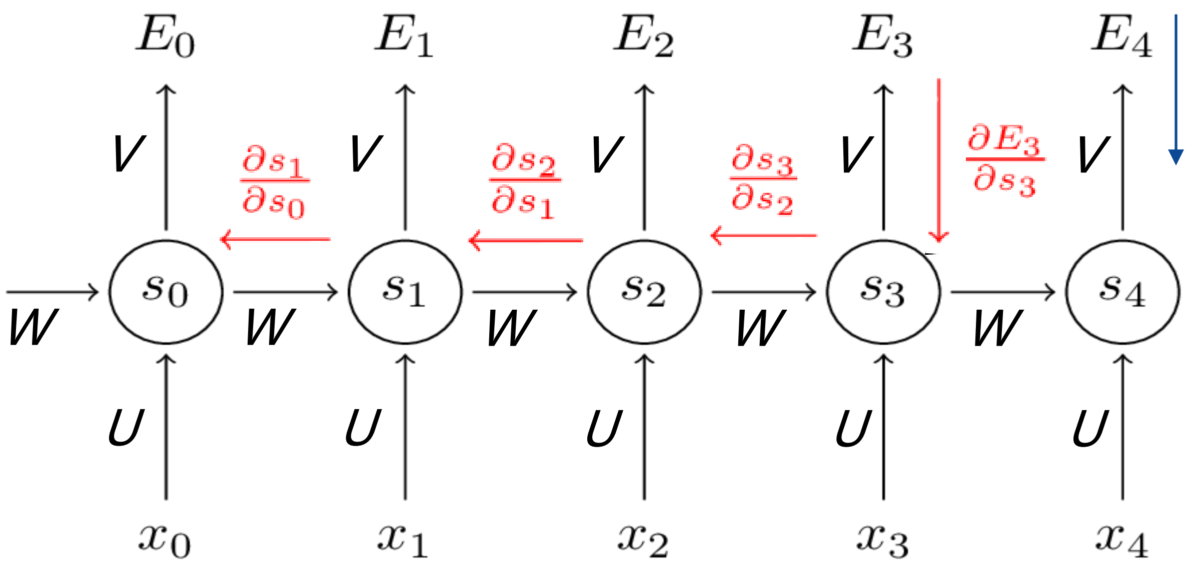
RNN训练算法-BPTT

- 求 E 对于U的梯度，先求 E_3 对于U的梯度

$$\frac{\partial E_3}{\partial W} = \frac{\partial E_3}{\partial \hat{y}_3}\frac{\partial \hat{y}_3}{\partial s_3}\frac{\partial s_3}{\partial U}$$

$$s_3 = tanh(Ux_3 + Ws_2)$$

$$\frac{\partial E_3}{\partial U} = \sum_{k=0}^{3}\frac{\partial E_3}{\partial \hat{y}_3}\frac{\partial \hat{y}_3}{\partial s_3}\frac{\partial s_3}{\partial s_k}\frac{\partial s_k}{\partial U}$$



求和可得 $\frac{\partial E}{\partial U}$

其中：s_3依赖于s_2，而s_2又依赖于s_1和U ，依赖关系一直传递到t = 0的时刻。**因此，当我们计算对于U的偏导数时，也不能把s_2看作是常数项！**

3 长短时记忆网络

RNN的梯度消失问题

- 不能有效解决长时依赖问题
- 梯度消失的原因
  - BPTT算法
  - 激活函数Tanh

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \left(\prod_{j=k+1}^{3} \frac{\partial s_j}{\partial s_{j-1}} \right) \frac{\partial s_k}{\partial W}$$

- 解决方案
  - ReLU函数
  - 门控RNN（LSTM）

The cat, which already ate a bunch of food, was full.

The cats, which already ate a bunch of food, were full.

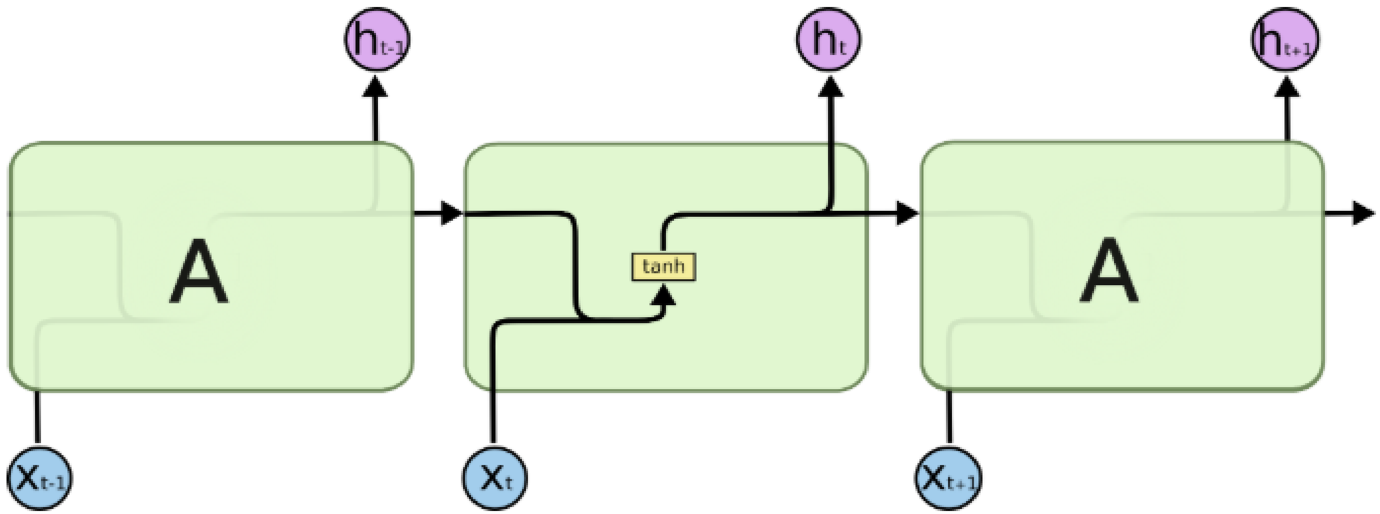
LSTM的介绍

❑ Long short-term memory (LSTM)

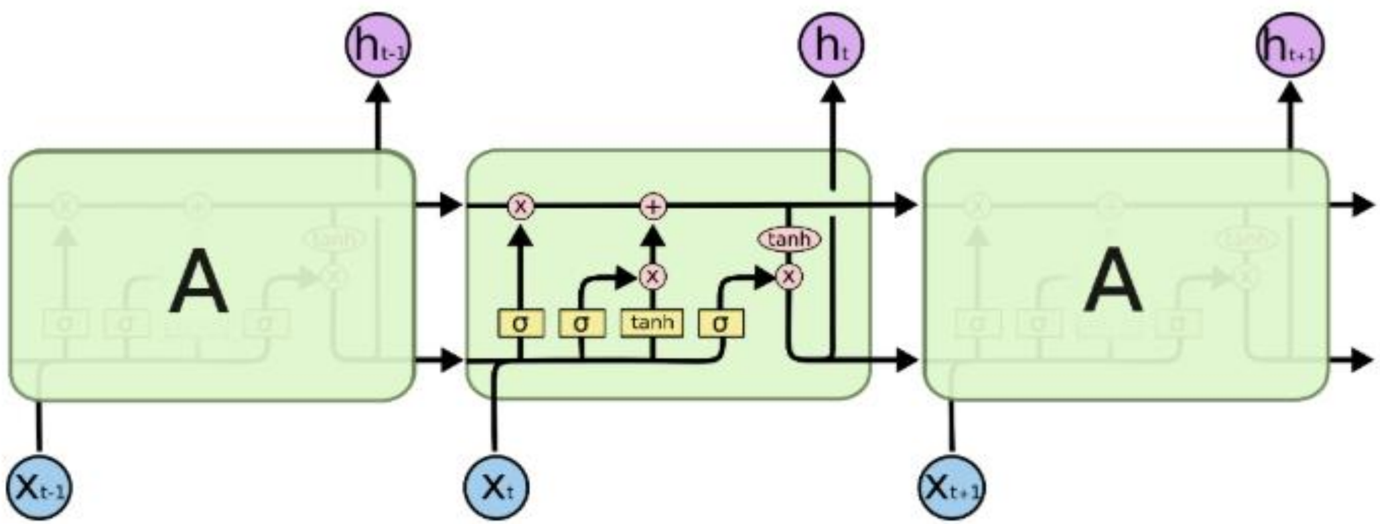
- Proposed in 1997 by Sepp Hochreiter and Jürgen Schmidhuber
- LSTM is an artificial recurrent neural network (RNN) architecture used in the field of deep learning
- A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell

LSTM的基本结构

一般RNN

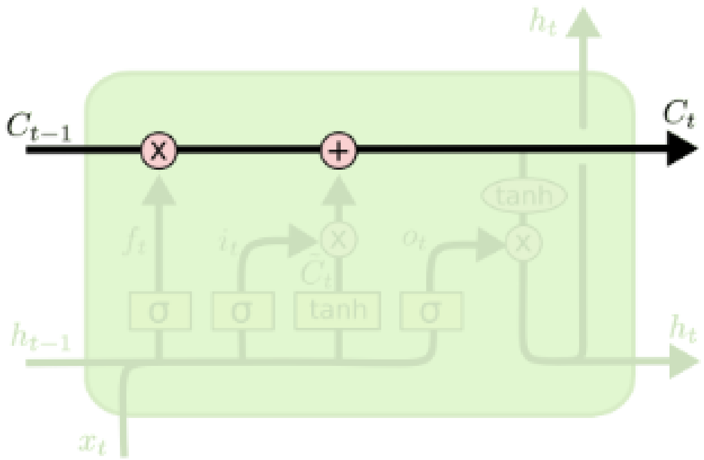


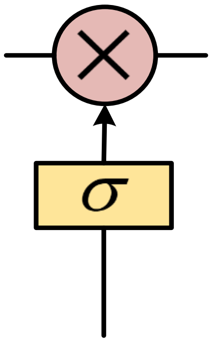
LSTM



LSTM的基本结构

- LSTM依靠贯穿隐藏层的细胞状态实现隐藏单元之间的信息传递，其中只有少量的线性操作
- LSTM引入了“门”机制对细胞状态信息进行添加或删除，由此实现长程记忆
- “门”机制由一个Sigmoid激活函数层和一个向量点乘操作组成，Sigmoid层的输出控制了信息传递的比例





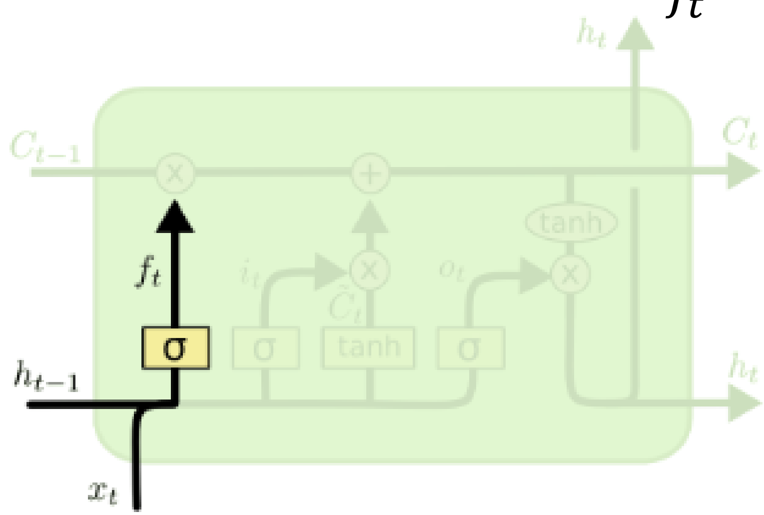
LSTM的基本结构

❑ 遗忘门

– LSTM通过遗忘门（forget gate）实现对细胞状态信息遗忘程度的控制，输出当前状态的遗忘权重，取决于h_{t-1}和x_t

$$f_t = \sigma(W_f \cdot [\, h_{t-1}, x_t] + b_f)$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right)$$



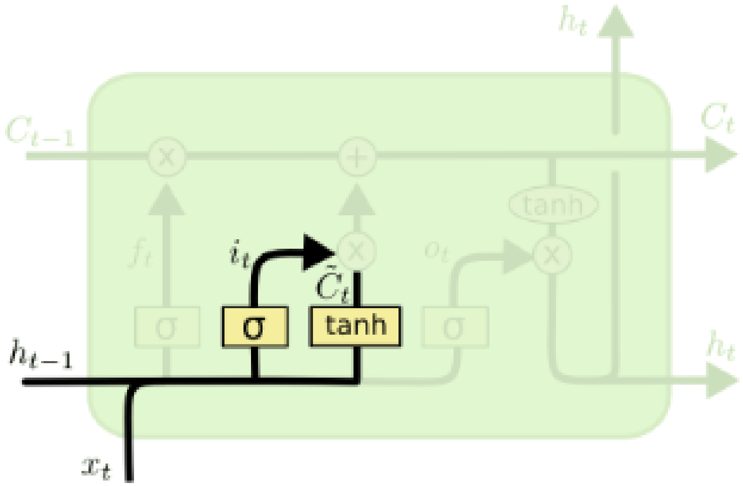
LSTM的基本结构

❑ 输入门

– LSTM通过输入门（input gate）实现对细胞状态输入接收程度的控制，输出当前输入信息的接受权重，取决于h_{t-1}和x_t

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

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

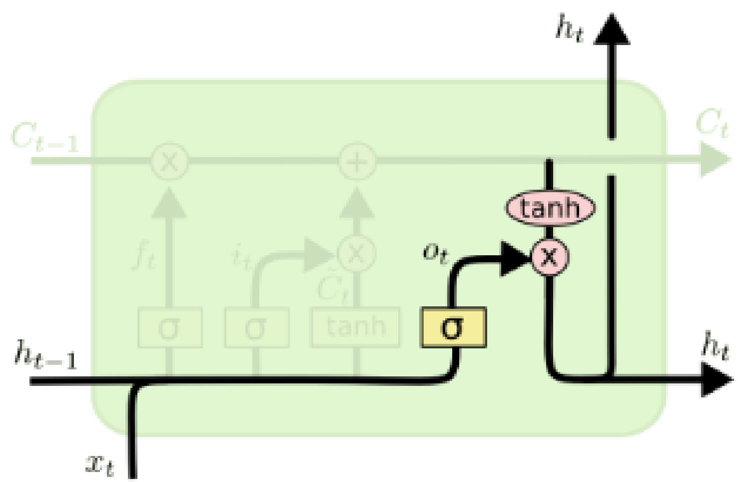


LSTM的基本结构

❑ 输出门

– LSTM通过输出门（output gate）实现对细胞状态输出认可程度的控制，输出当前输出信息的认可权重，取决于h_{t-1}和x_t

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$



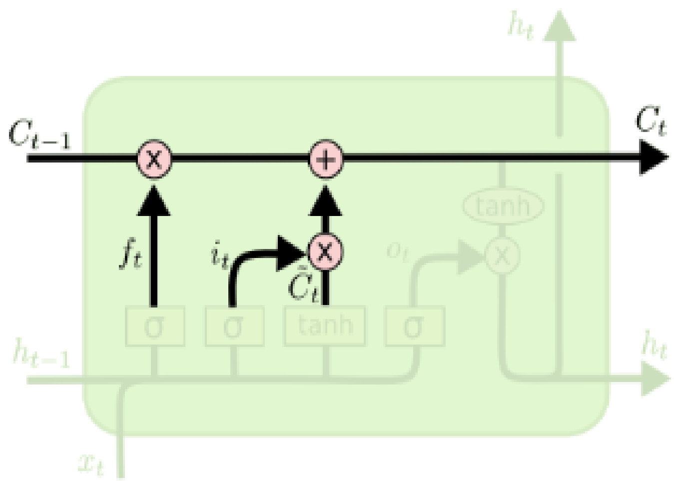
LSTM的基本结构

❑ 状态更新

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$h_t = o_t * tanh(C_t)$$

"门"机制对细胞状态信息进行添加或删除，由此实现长程记忆



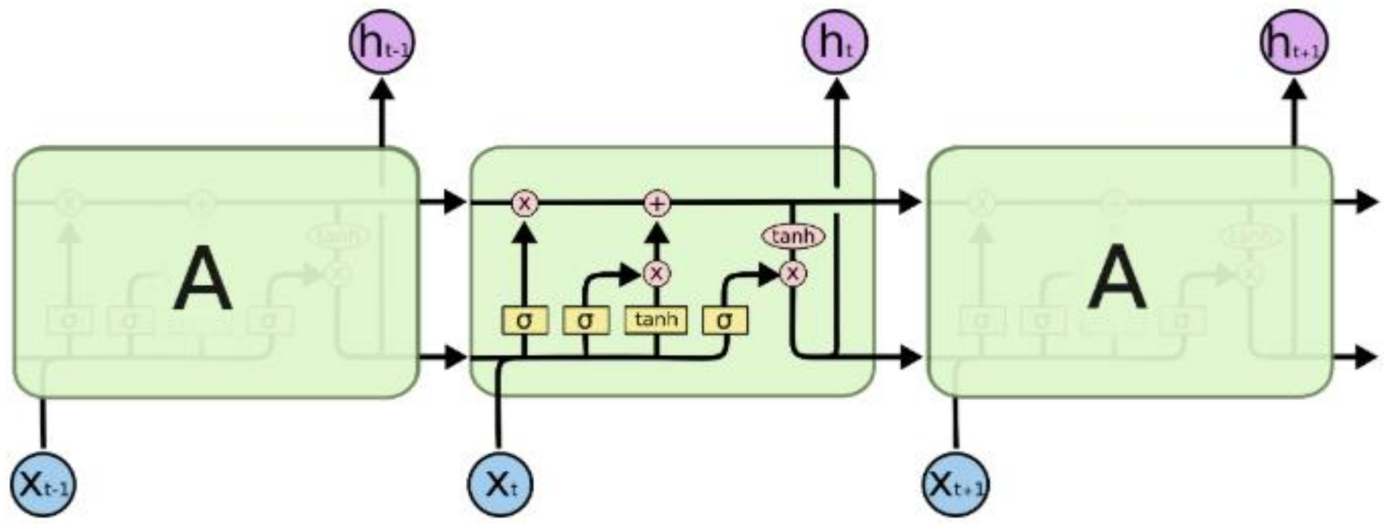
LSTM的基本结构

❑ 状态更新

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

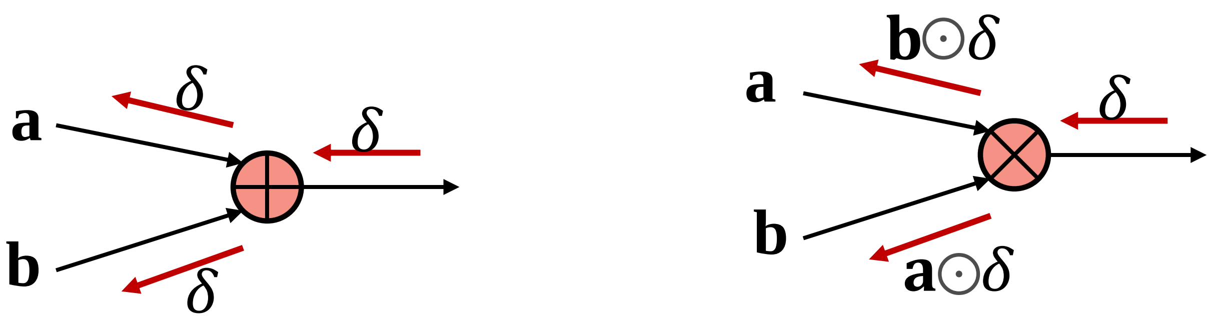
$$h_t = o_t * tanh(C_t)$$

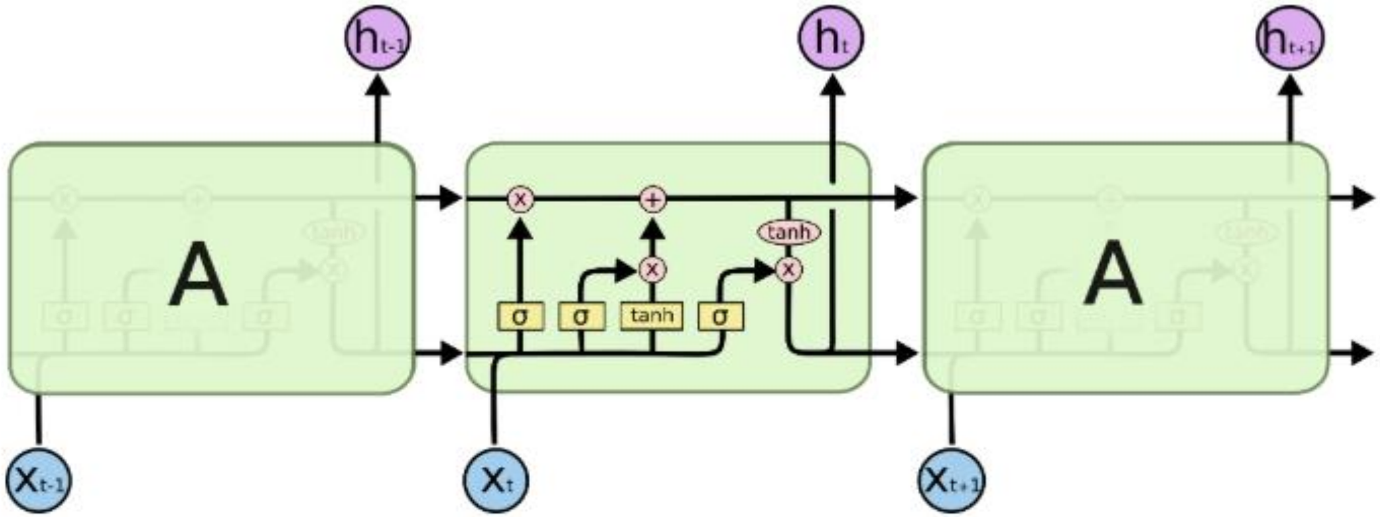
"门"机制对细胞状态信息进行添加或删除，由此实现长程记忆



LSTM的基本结构

- 状态更新





标准化的RNN

```python
# 构造RNN网络，x的维度5，隐层的维度10,网络的层数2
rnn_seq = nn.RNN(5, 10,2)
# 构造一个输入序列，长为 6，batch 是 3， 特征是 5
x = V(torch.randn(6, 3, 5))
#out,ht = rnn_seq(x, h0) # h0可以指定或者不指定
out,ht = rnn_seq(x)
# q1:这里out、ht的size是多少呢？ out:6*3*10, ht:2*3*10
```

```python
# 输入维度 50，隐层100维，两层
lstm_seq = nn.LSTM(50, 100, num_layers=2)
# 输入序列seq= 10，batch =3，输入维度=50
lstm_input = torch.randn(10, 3, 50)
out, (h, c) = lstm_seq(lstm_input) # 使用默认的全 0 隐藏状态
```

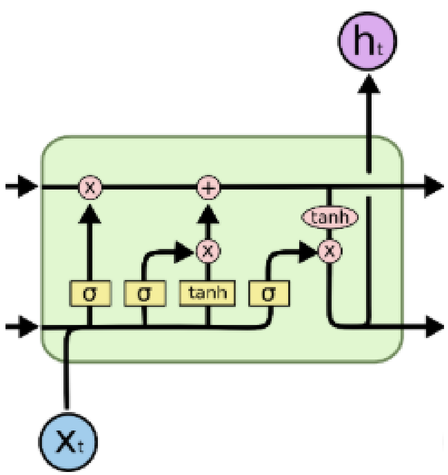
4 其他典型循环神经网络

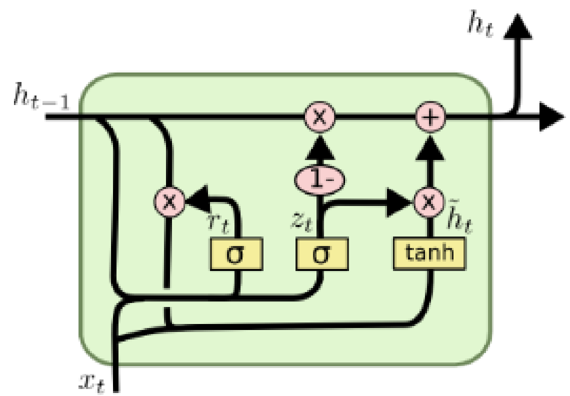
其他典型循环神经网络

- **Gated Recurrent Unit （GRU）**
- **Peephole LSTM**
- **Bi-directional RNN（双向RNN）**
- **Continuous time RNN**

GRU

- Gated Recurrent Unit (GRU), 2014年提出,可认为是LSTM的变种
  - 细胞状态与隐状态合并，在计算当前时刻新信息的方法和LSTM有所不同
  - GRU只包含重置门和更新门
  - 在音乐建模与语音信号建模领域与LSTM具有相似的性能，但是参数更少，只有两个门控





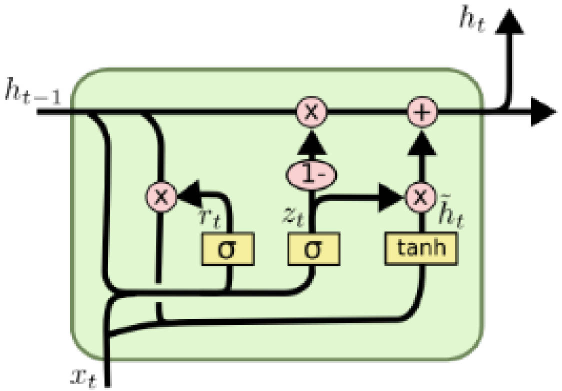
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

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

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

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

GRU



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

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

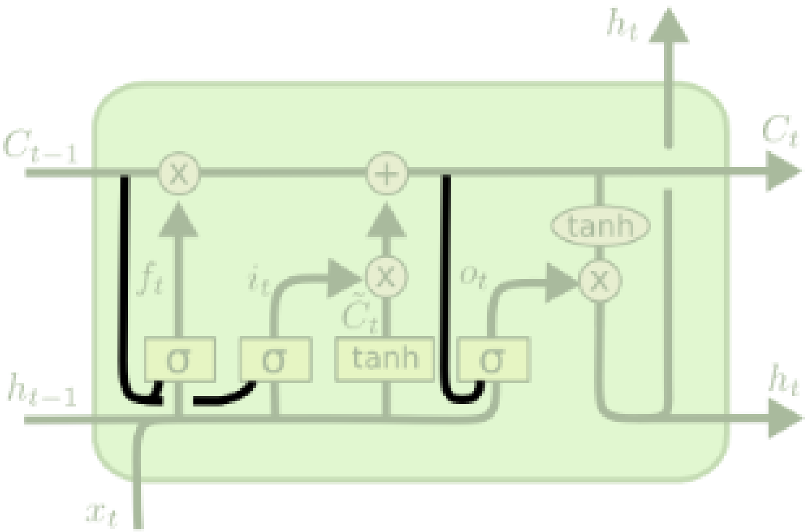
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

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

- x_t: input vector（输入向量）
- h_t : output vector（输出向量）
- z_t : update gate vector （更新门向量）
- r_t : reset gate vector （重置门向量）
- W : parameter matrices and vector (参数矩阵
- σ: 一般选用Sigmoid函数

Peephole LSTM

让门层也接受细胞状态的输入，同时考虑隐层信息的输入



$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_f\right)$$

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

$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \; + \; b_o\right)$$

不同LSTM变体

❑ 多种LSTM变体性能差异不显著

- 8 LSTM variants on three representative tasks: speech recognition, handwriting recognition, and polyphonic music modeling
- None of the variants can improve upon the standard LSTM architecture significantly
- The forget gate and the output activation function to be its most critical components

VI. Conclusion

This paper reports the results of a large scale study on variants of the LSTM architecture. We conclude that the most commonly used LSTM architecture (vanilla LSTM) performs reasonably well on various datasets. None of the eight investigated modifications significantly improves performance. However, certain modifications such as coupling the input and forget gates (CIFG) or removing peephole connections (NP) simplified LSTMs in our experiments without significantly decreasing performance. These two variants are also attractive because they reduce the number of parameters and the computational cost of the LSTM.

K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, "LSTM: A Search Space Odyssey," in IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222-2232, Oct. 2017.

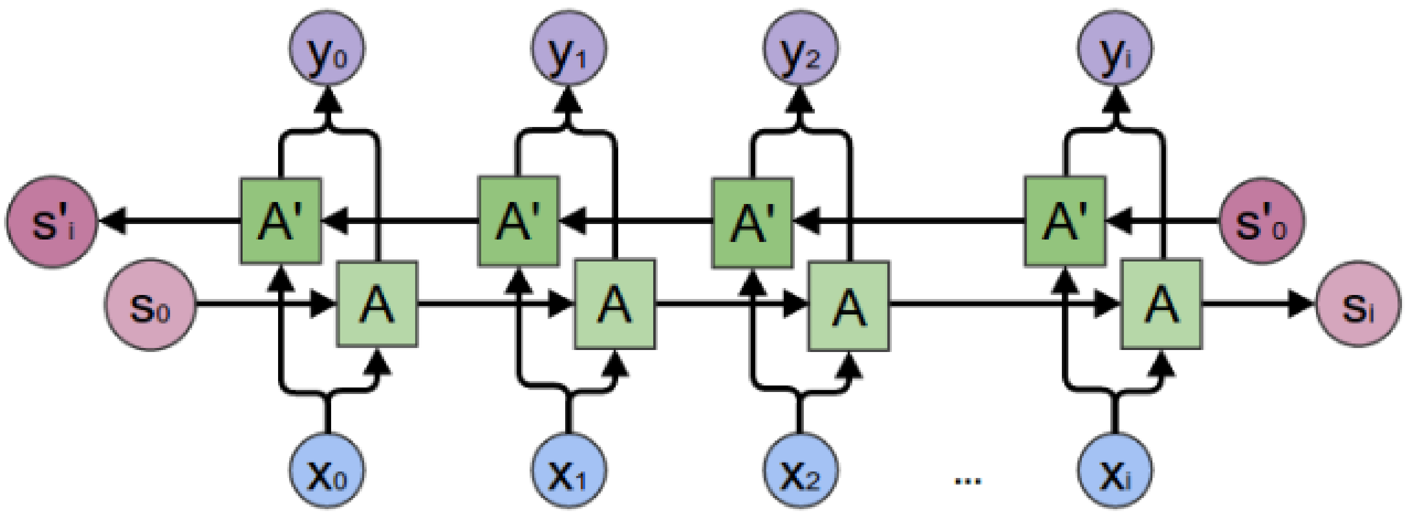
双向RNN

- Bidirectional RNN(双向RNN)假设**当前t的输出不仅仅和之前的序列有关，并且还与之后的序列有关**，例如:完形填空
- Bidirectional RNN由**两个RNNs上下叠加在一起组成**，输出由这两个RNNs的隐藏层的状态决定

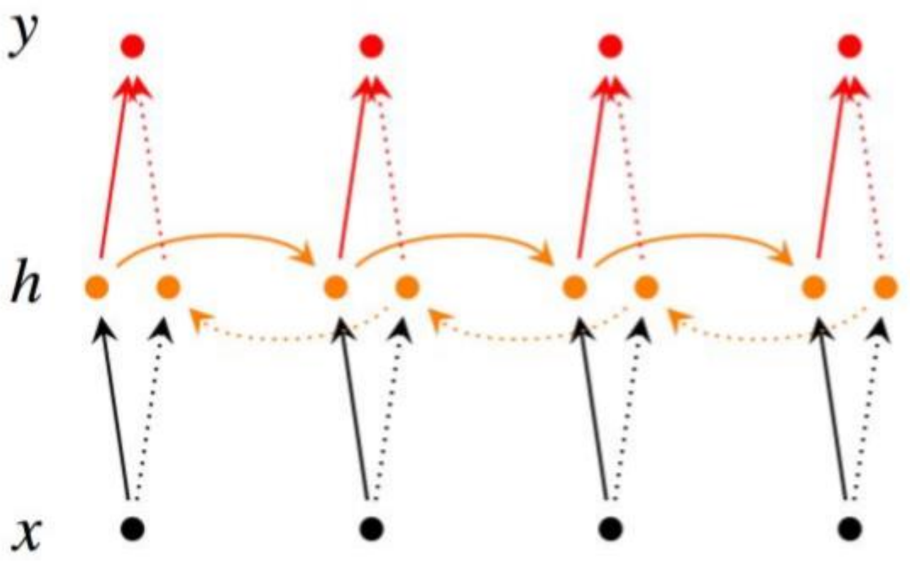
I am ______

I am ______ very hungry

“happy” and “not”



双向RNN

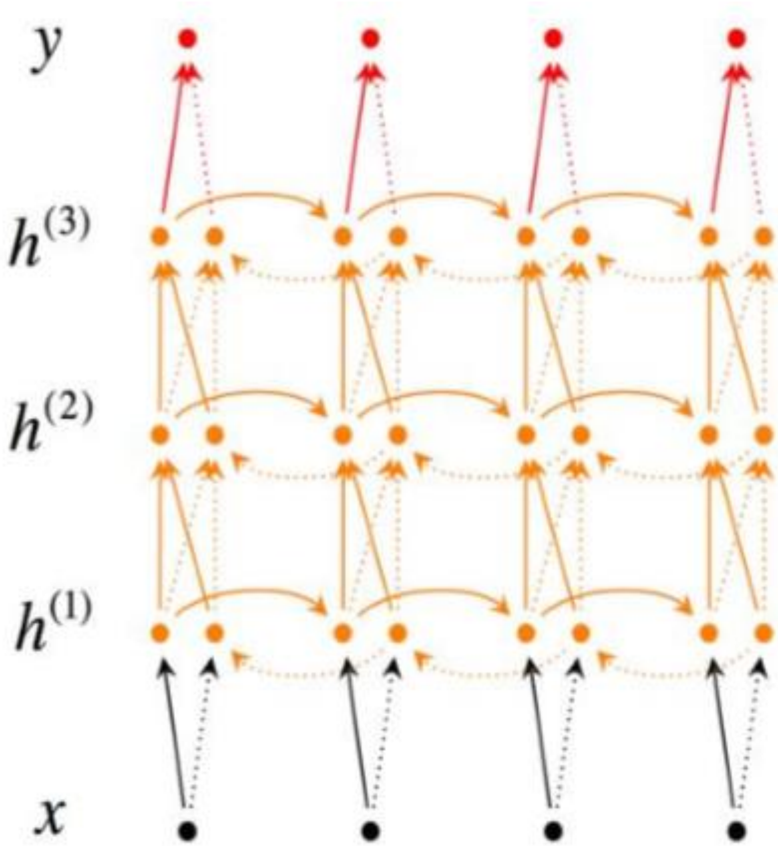


$$\overrightarrow{h}_t = f(\overrightarrow{W}x_t + \overrightarrow{V}\overrightarrow{h}_{t-1} + \overrightarrow{b})$$

$$\overleftarrow{h}_t = f(\overleftarrow{W}x_t + \overleftarrow{V}\overleftarrow{h}_{t+1} + \overleftarrow{b})$$

$$y_t = g(U[\overrightarrow{h}_t;\overleftarrow{h}_t] + c)$$

双向RNN



$$\overrightarrow{h}_t^{(i)} = f(\overrightarrow{W}^{(i)} h_t^{(i-1)} + \overrightarrow{V}^{(i)} \overrightarrow{h}_{t-1}^{(i)} + \overrightarrow{b}^{(i)})$$

$$\overleftarrow{h}_t^{(i)} = f(\overleftarrow{W}^{(i)} h_t^{(i-1)} + \overleftarrow{V}^{(i)} \overleftarrow{h}_{t+1}^{(i)} + \overleftarrow{b}^{(i)})$$

$$y_t = g(U[\overrightarrow{h}_t^{(L)}; \overleftarrow{h}_t^{(L)}] + c)$$

Continuous time RNN

- CTRNN利用常微分方程系统对输入脉冲序列神经元的影响进行建模。
- CTRNN被应用到进化机器人中，用于解决视觉、协作和最小认知行为等问题

$$\tau_i \dot{y}_i = -y_i + \sum^{n} w_{ji} \sigma(y_j - \Theta_j) + I_i(t)$$

- τ_i : Time constant of postsynaptic node
- y_i : Activation of postsynaptic node
- $\dot{y}_i$: Rate of change of activation of postsynaptic node
- w_{ji} : Weight of connection from pre to postsynaptic node
- $\sigma(x)$: Sigmoid of x e.g. $\sigma(x) = 1/(1+e^{-x})$.
- y_j : Activation of presynaptic node
- Θ_j : Bias of presynaptic node
- $I_i(t)$: Input (if any) to node

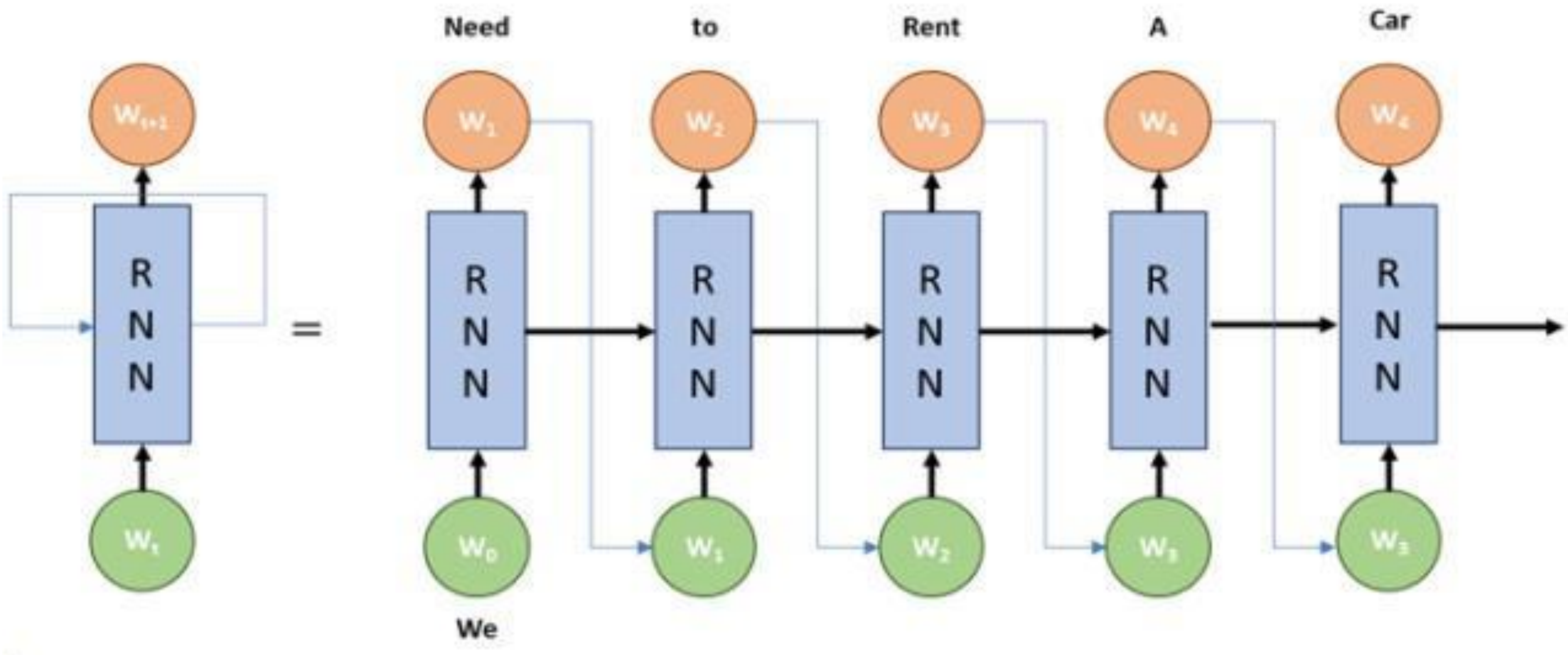
5 循环神经网络的主要应用

循环神经网络的主要应用

- ❑ 语言模型
- ❑ 语音识别
- ❑ 自动作曲
- ❑ 机器翻译
- ❑ 自动摘要
- ❑ 自动写作
- ❑ 图像描述

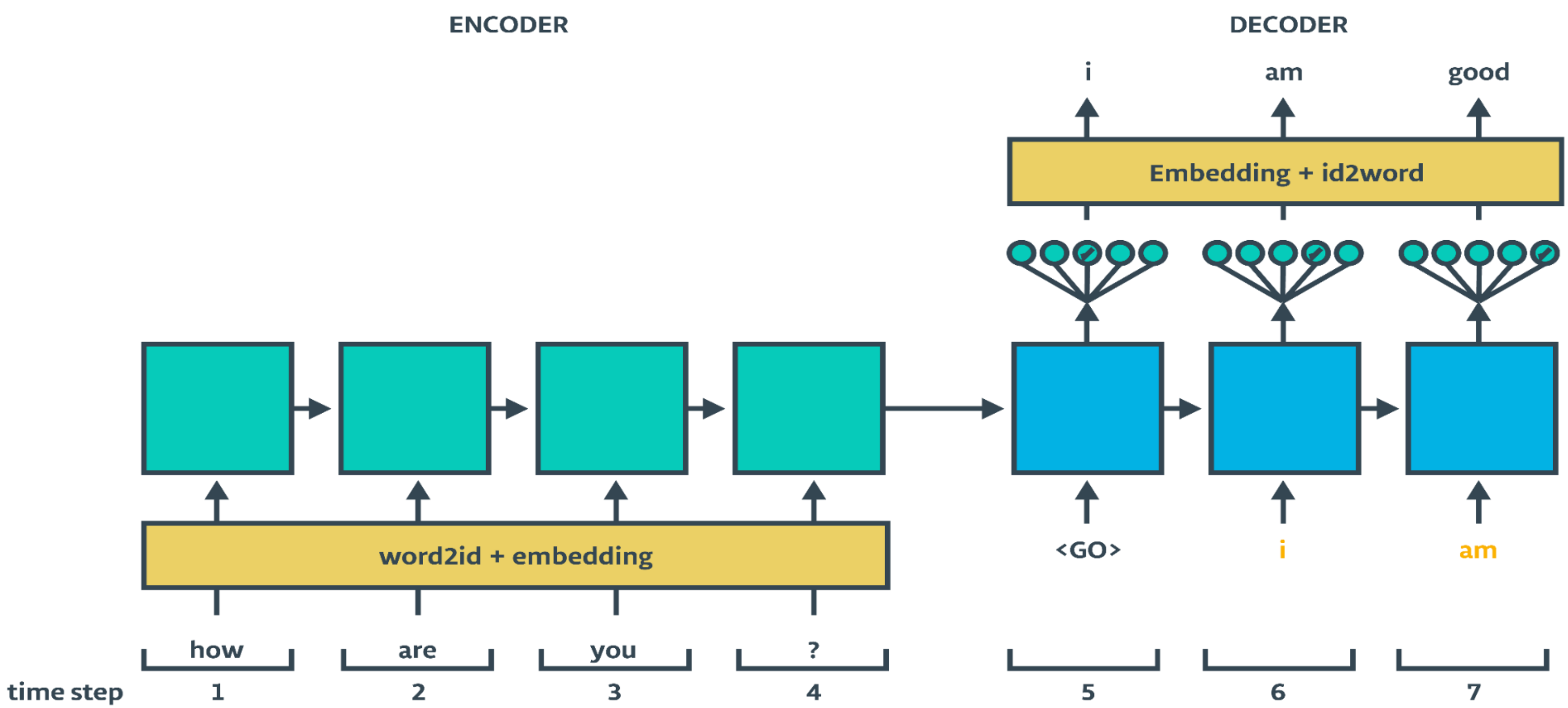
语言模型

- ❑ 根据之前和当前词预测下一个单词或者字母



语言模型

- 问答系统


ENCODER
DECODER
i
am
good
Embedding + id2word
word2id + embedding
<GO>
i
am
how
are
you
?
time step
1
2
3
4
5
6
7

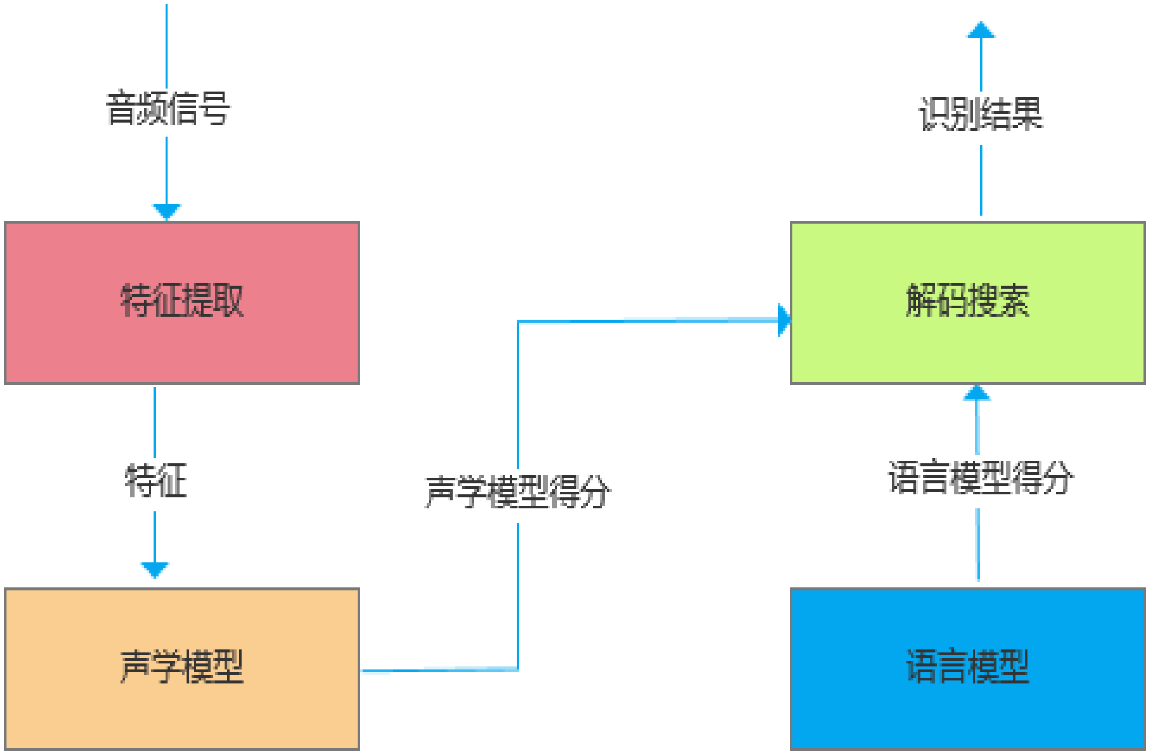
语音识别

❑ 将语音识别成文字

– 例子: 给定语音的拼音串“ta shi yan jiu sheng wu de”

可能的汉字串：

“踏实研究生物的”
“他实验救生物的”
“他使烟酒生物的”
“他是研究生物的”



自动作曲

- 根据选定风格自动作曲

自动作曲

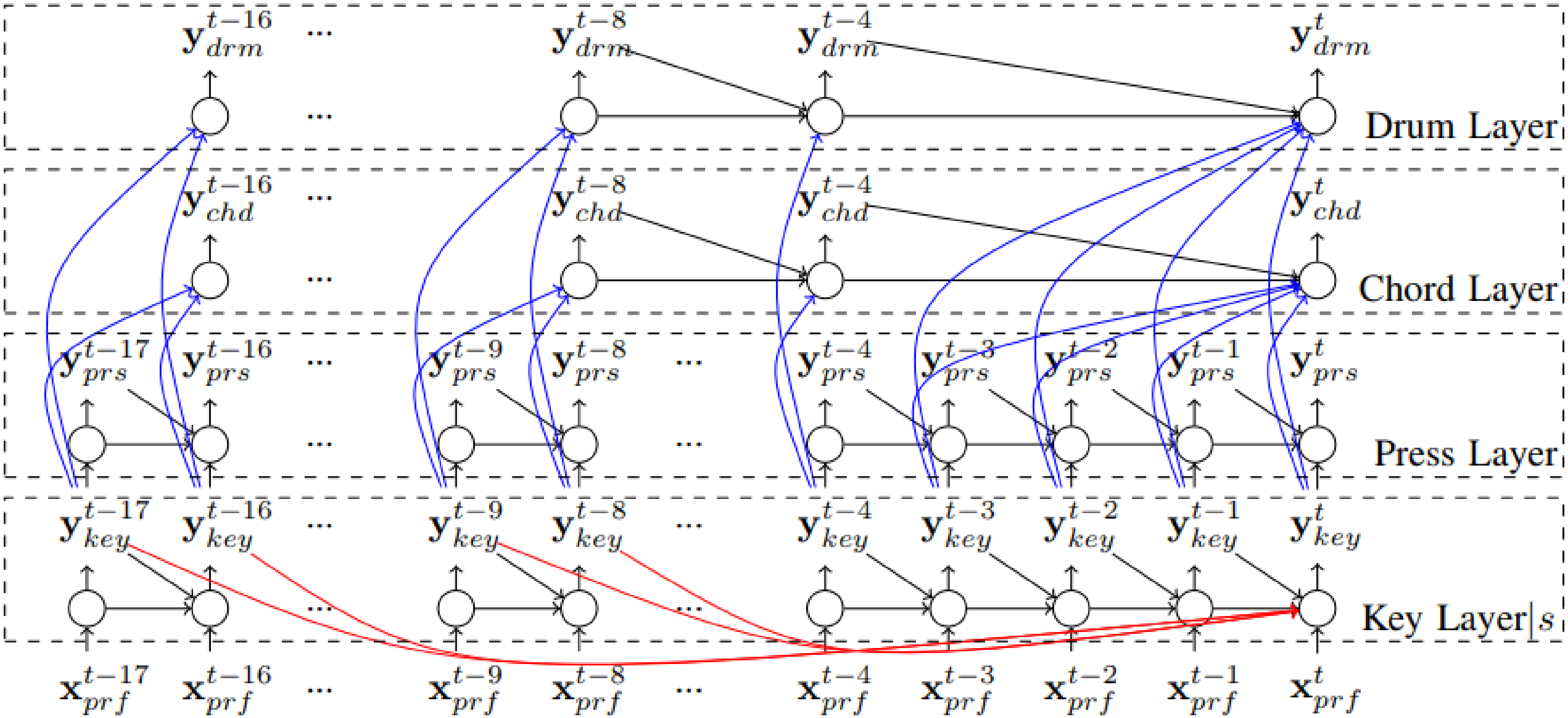


Figure 1: Overview of our framework. Only skip connections for the current time step t are plotted.

Hang Chu, Raquel Urtasun, Sanja Fidler. Song From PI: A Musically Plausible Network for Pop Music Generation. CoRR abs/1611.03477 (2016)

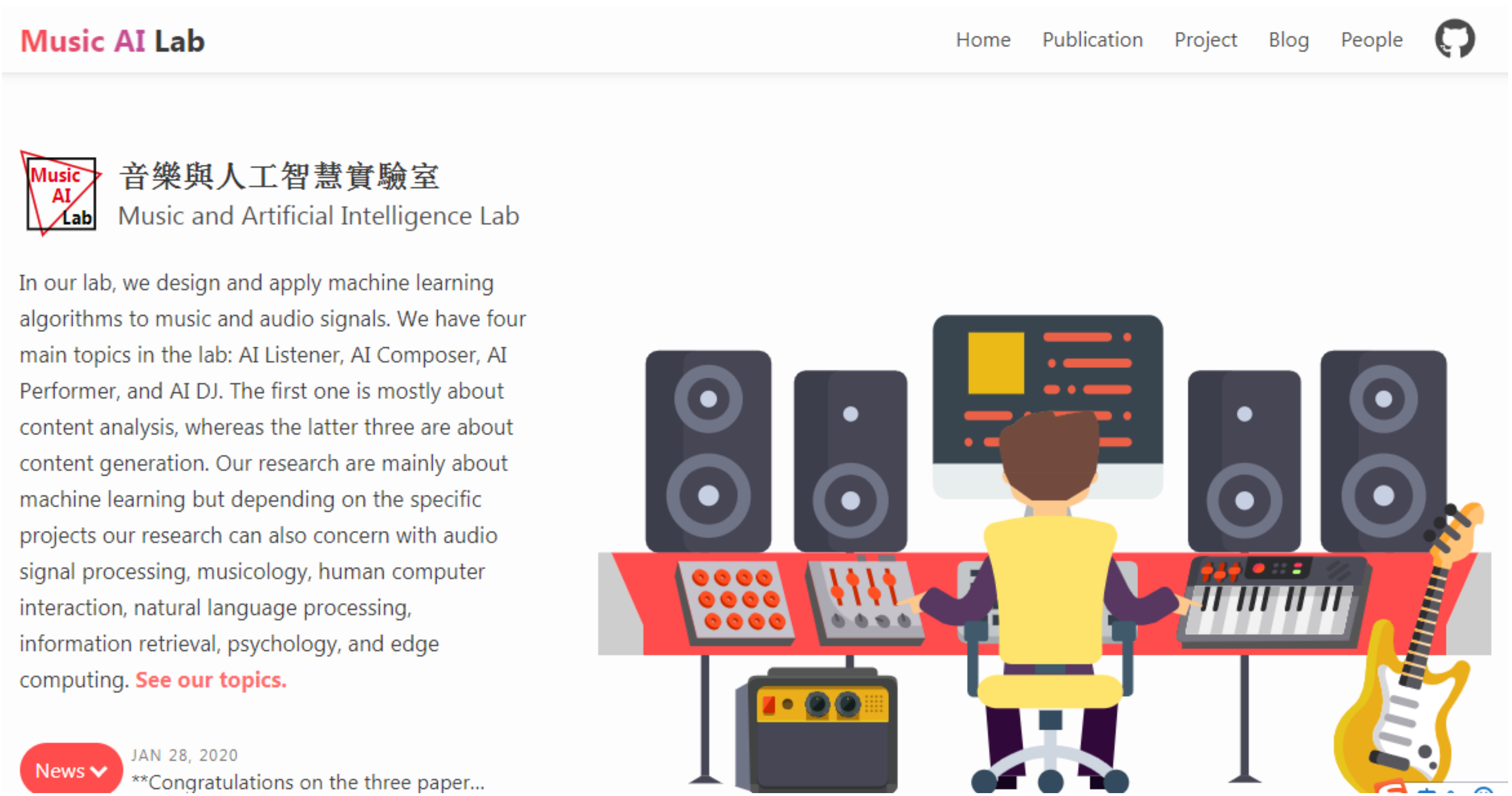
自动作曲

Figure 4: Example of our music generation. From top to bottom: melody, chord and drum respectively.

Hang Chu, Raquel Urtasun, Sanja Fidler. Song From PI: A Musically Plausible Network for Pop Music Generation. CoRR abs/1611.03477 (2016)

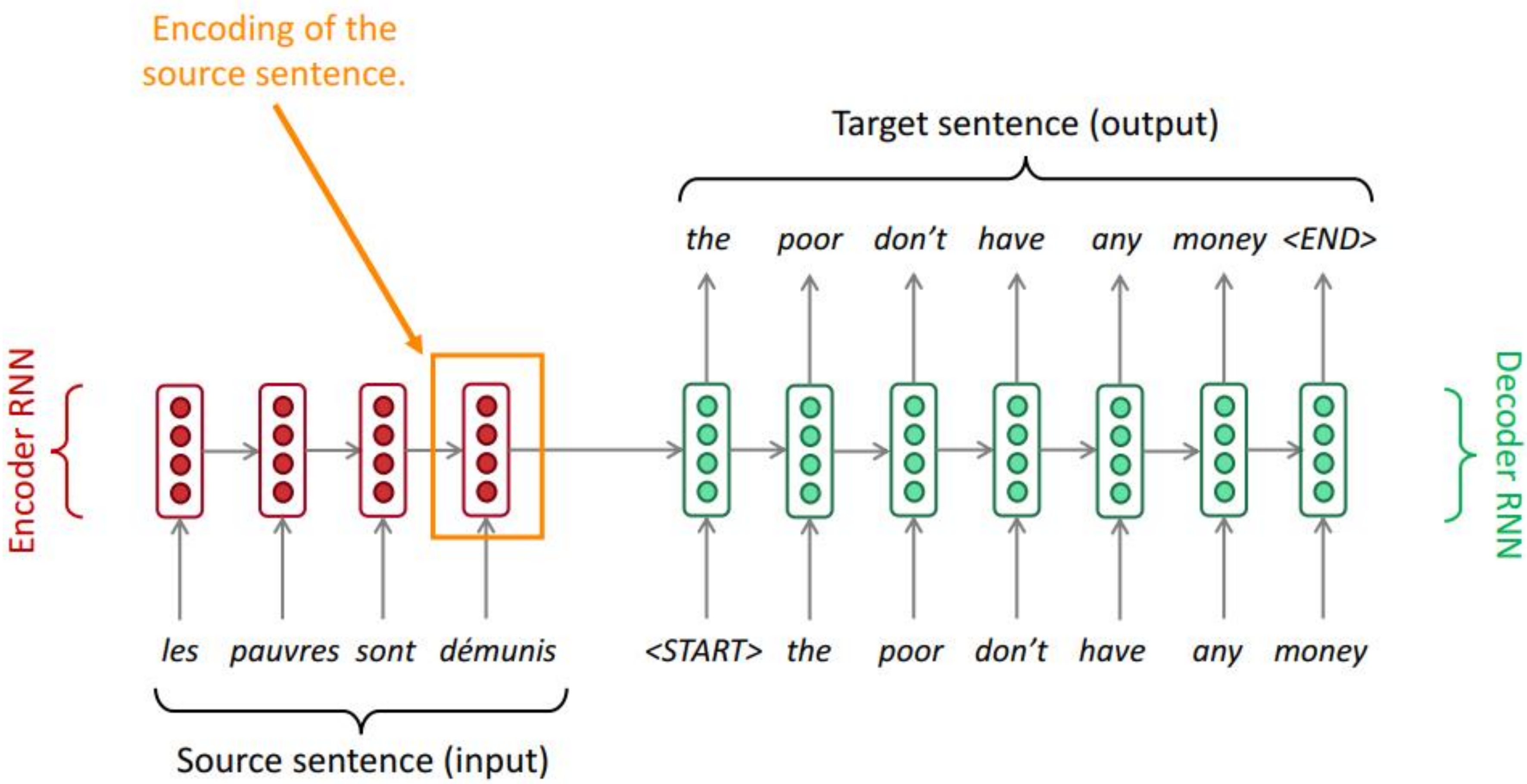
自动作曲

- https://musicai.citi.sinica.edu.tw/



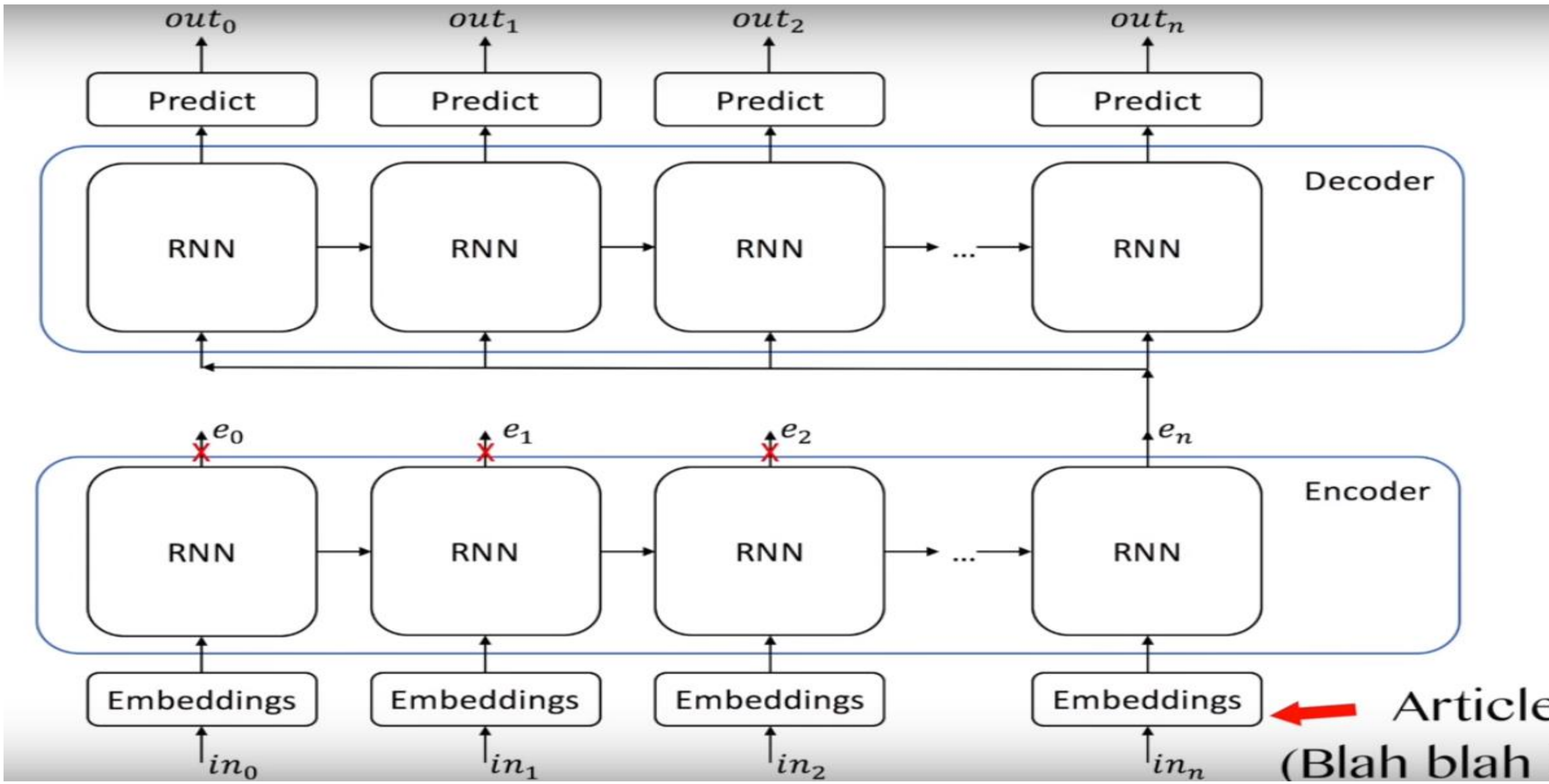
机器翻译

- 将一种语言自动翻译成另一种语言



自动摘要

- 为一篇或者多篇文章自动生成摘要



自动写作

- ❑ **根据现有资料自动写作，当前主要包括新闻写作和诗歌创作**
  - 主要是基于RNN&LSTM的文本生成技术来实现，需要训练大量同类文本，结合模板技术
- ❑ **目前主要产品**
  - 腾讯Dreamwriter写稿机器人
  - 今日头条xiaomingbot
  - 第一财经DT稿王(背后是阿里巴巴)
  - 百度Writing-bots

图像描述

❑ 根据图像形成语言描述

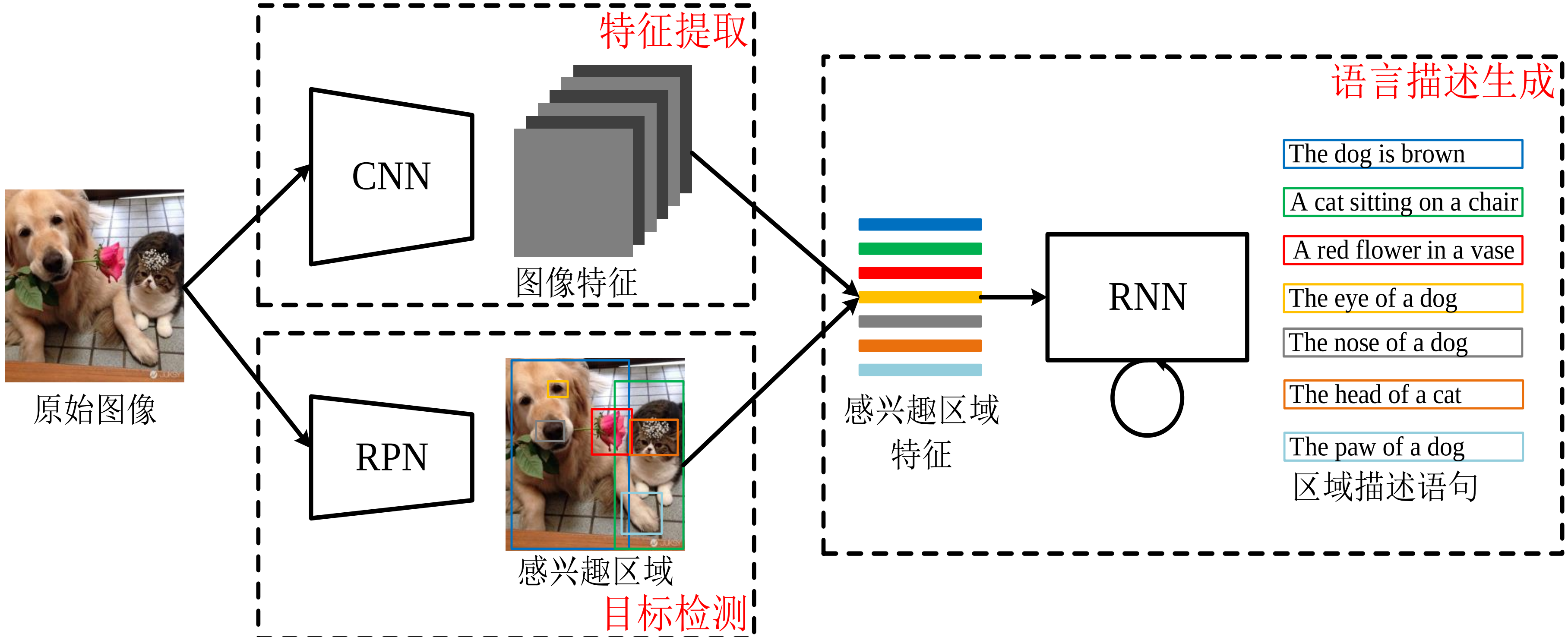
"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

图像描述

❑ 根据图像形成语言描述



6 中英文术语对照

中英文术语对照

- 计算图：**Computational graph**
- 循环神经网络：**Recurrent Neural Network**
- 随时间反向传播算法：**BP Through Time, BPTT**
- 长短时记忆网络：**Long Short-Term Memory**
- 遗忘门：**Forget gate**
- 输入门：**Input gate**
- 输出门：**Output gate**
- 双向RNN：**Bidirectional RNN**

中英文术语对照

- 门控循环单元：**Gated Recurrent Unit （GRU）**
- 窥孔LSTM: **Peephole LSTM**
- 连续时间RNN: **Continuous time RNN**
- 语言模型：**Language model**
- 神经机器翻译: **Neural Machine Translation**
- 图像描述：**Image captioning**
- 自动摘要：**Automatic summarization**
- 自动写作：**Automatic writing**

谢谢！