厦门大学信息学院 本科选修课
2021-2022 第二学期

模 式 识 别
Pattern Recognition

主 讲：王 程

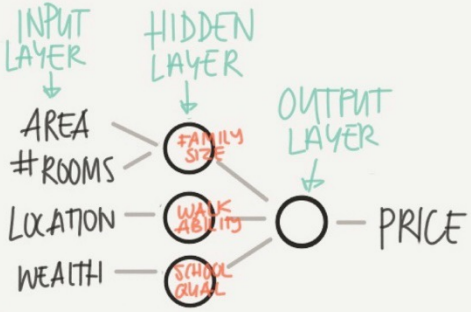
第九章

循环神经网络（RNN）

陈龙彪
longbiaochen@xmu.edu.cn

厦门大学信息学院
2022年4月

INTRO TO DEEP LEARNING



SUPERVISED LEARNING

INPUT: X	OUTPUT: y	NN TYPE
HOME FEATURES	PRICE	STANDARD NN
AD + USER INFO	WILL CLICK ON AD (0/1)	
IMAGE	OBJECT (1...1000)	CONV. NN (CNN)
AUDIO	TEXT TRANSCRIPT	RECURRENT NN (RNN)
ENGLISH	CHINESE	
IMAGE/RADAR	POS OF OTHER CARS	CUSTOM/HYBRID

NNs CAN DEAL WITH BOTH STRUCTURED & UNSTRUCTURED DATA

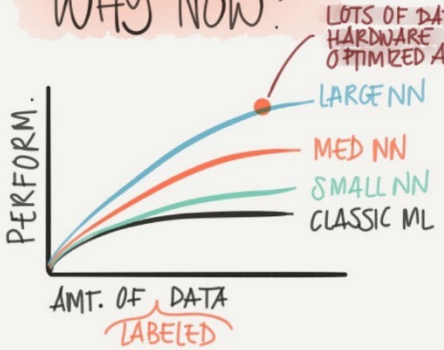
STRUCTURED

"THE QUICK BROWN FOX"

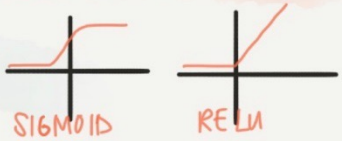
UNSTRUCTURED

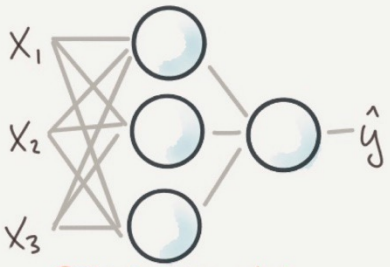
HUMANS ARE GOOD AT THIS

WHY NOW?



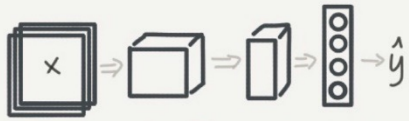
ONE OF THE BIG BREAKTHROUGHS HAS BEEN MOVING FROM SIGMOID TO RELU FOR FASTER GRADIENT DESCENT



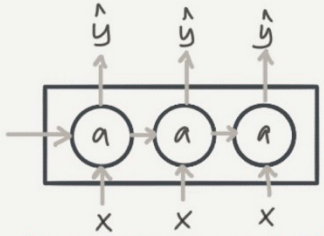


STANDARD NN

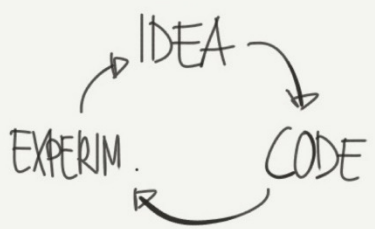
NETWORK ARCHITECTURES



CONVOLUTIONAL NN



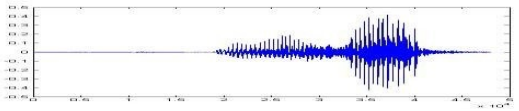
RECURRENT NN



FASTER COMPUTATION IS IMPORTANT TO SPEED UP THE ITERATIVE PROCESS

机器学习≈寻找一个函数

- 语音识别

$f($  $)=$ “How are you”

- 图片识别

$f($  $)=$ “Cat”

- 下围棋

$f($  $)=$ “5-5” (下一步棋的结果)

- 对话系统

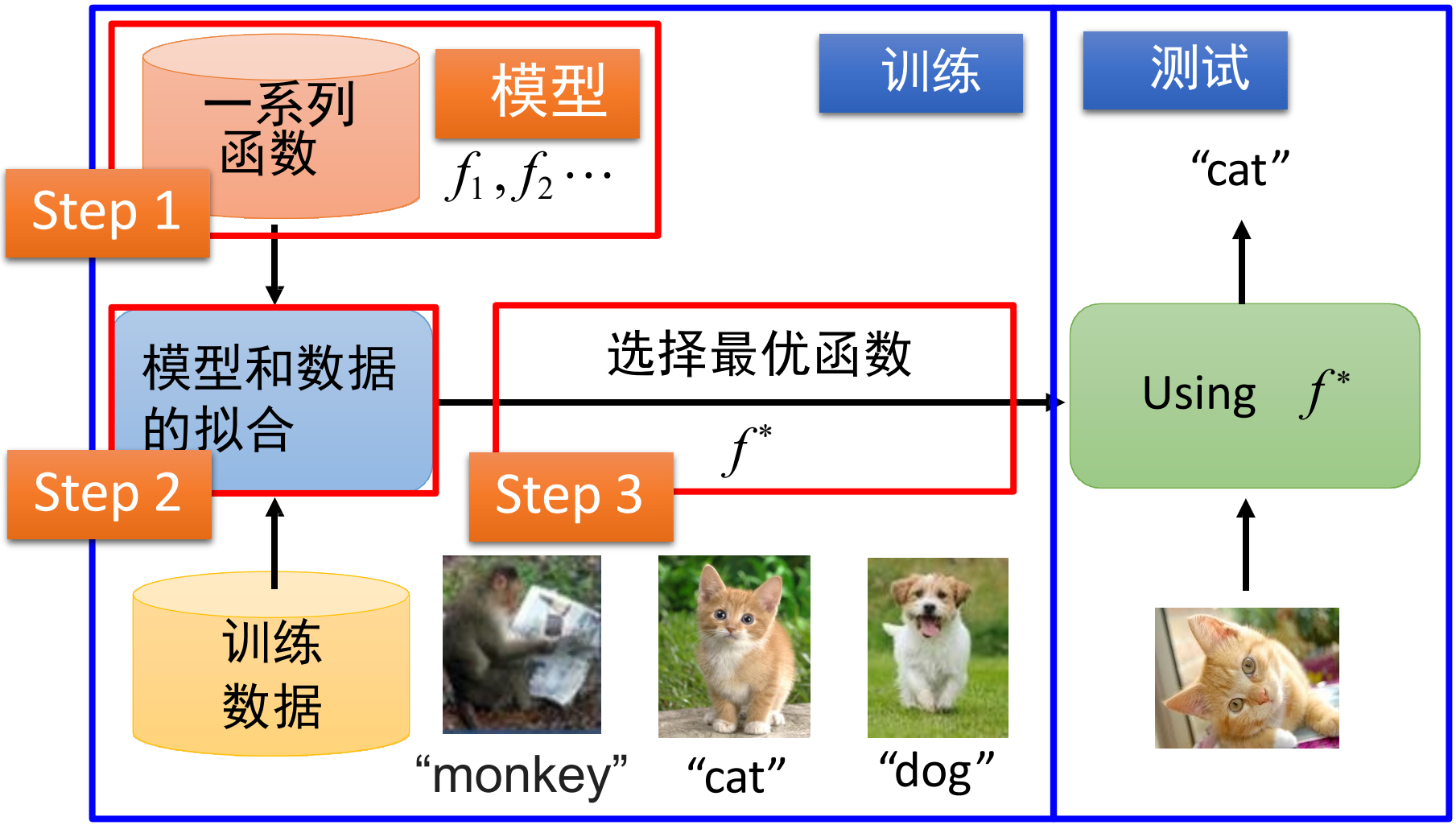
$f($“导航回家”$)=$ “你说啥？”

(你说的内容) (系统的回答)

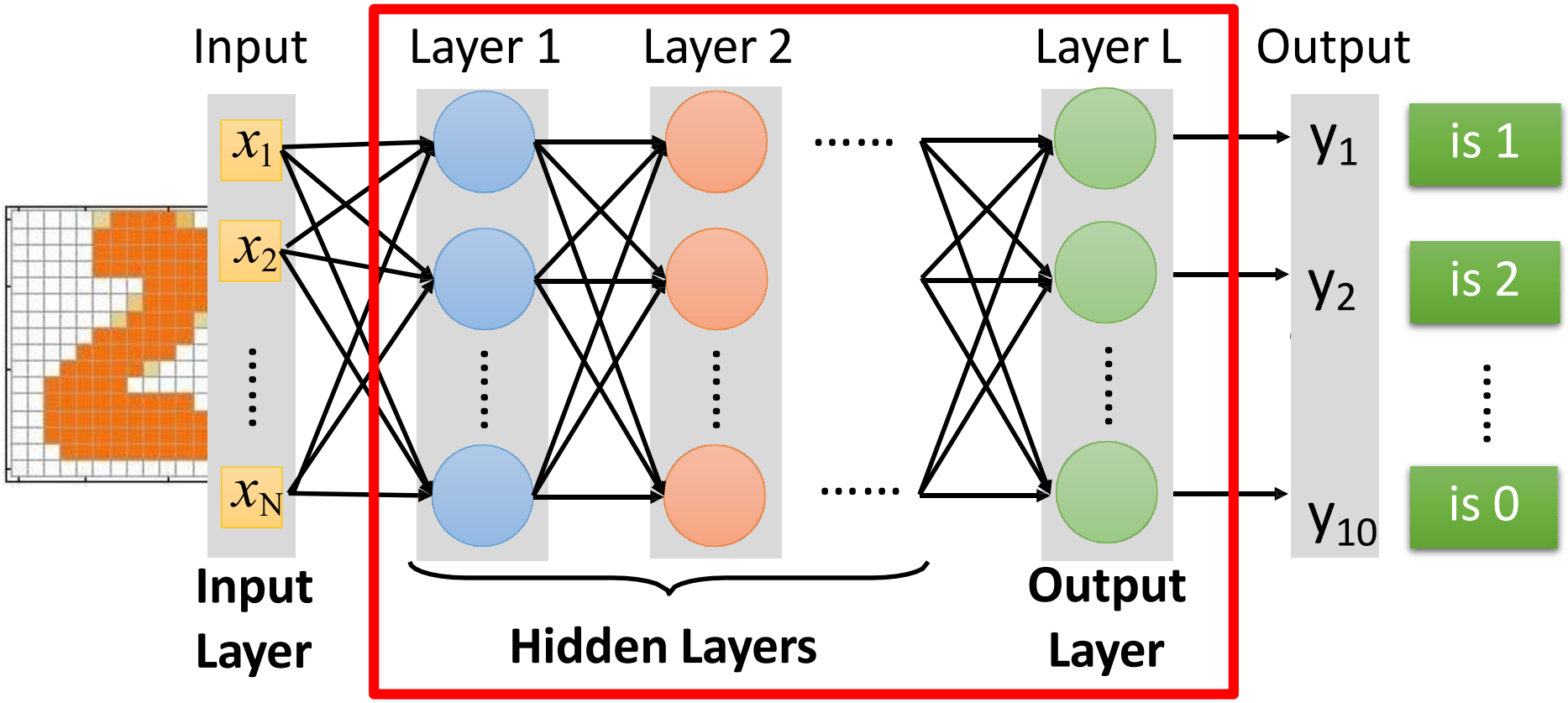
机器学习框架

图像识别:

$f($  $)=$ “cat”



深度神经网络结构



你需要决定网络结构，让你的函数有好的功能

神经网络的变体

卷积神经网络
(CNN)

循环神经网络（RNN)

具有记忆的神经网络

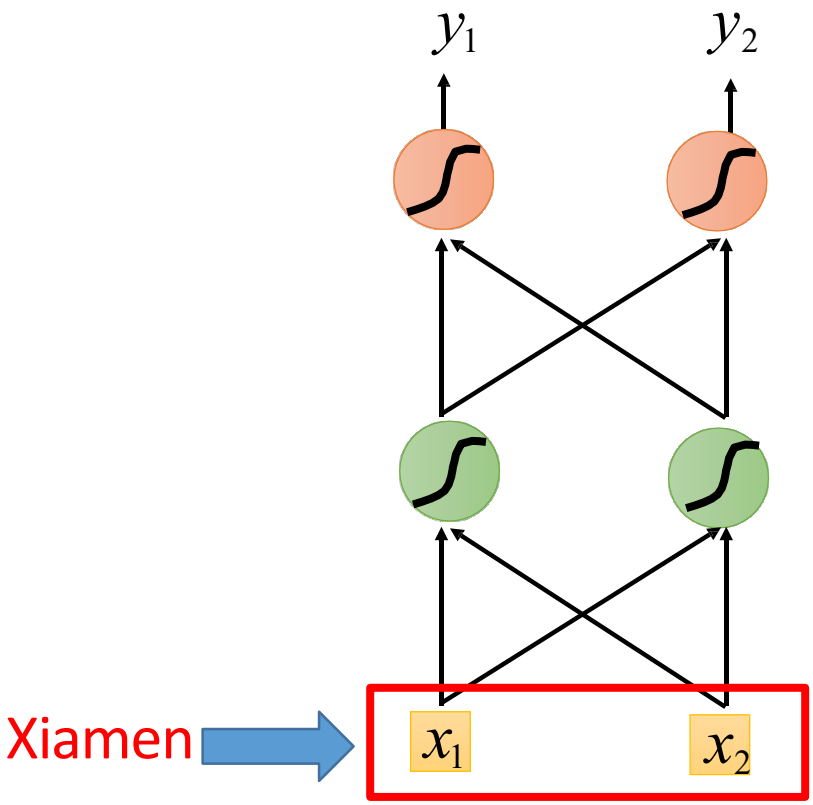
示例应用程序

- 填空问题


I would like to leave Xiamen on November 2nd.
智能订票系统
空位
地点:
Xiamen
时间:
November 2nd

示例应用程序

- 通过前馈神经网络解决填空问题?
  - 输入: a word (每个单词都被表示成一个向量)



1-of-N encoding

- 如何将每个单词表示成一个词向量?

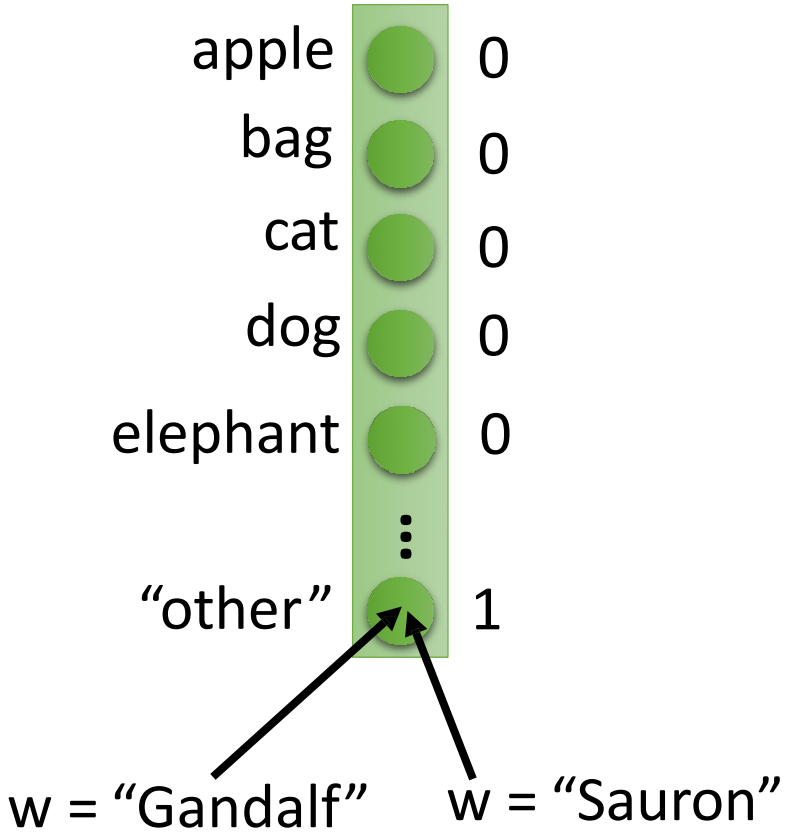
1-of-N Encoding 词典 = {apple, bag, cat, dog, elephant}

- 向量长度是词典大小
- 每个维度对应于词典中的一个单词
- 这个词的维度是1，其他是0

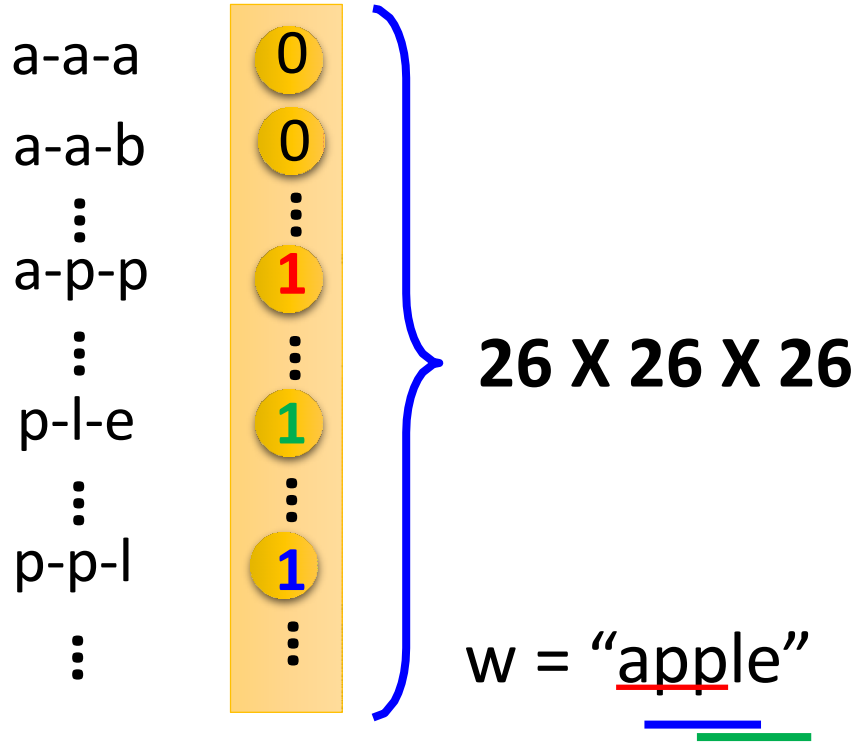
apple = [1 0 0 0 0]
bag = [0 1 0 0 0]
cat = [0 0 1 0 0]
dog = [0 0 0 1 0]
elephant = [0 0 0 0 1]

Beyond 1-of-N encoding

Dimension for "Other"

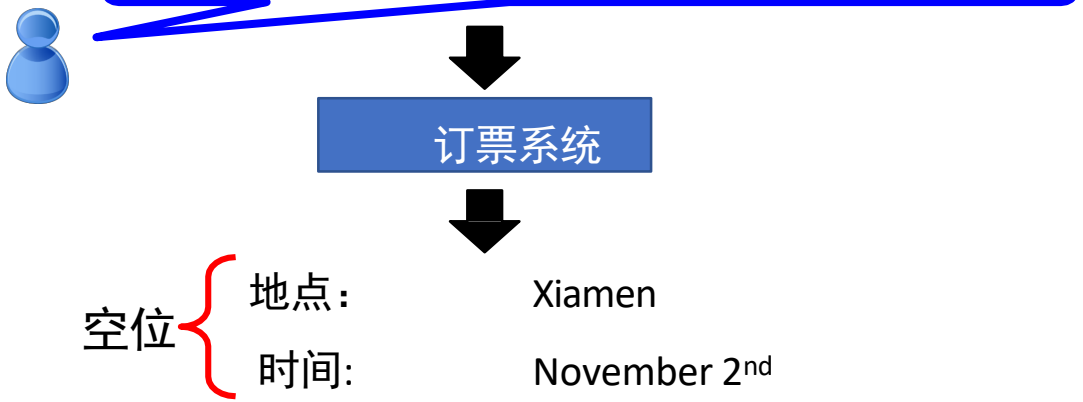


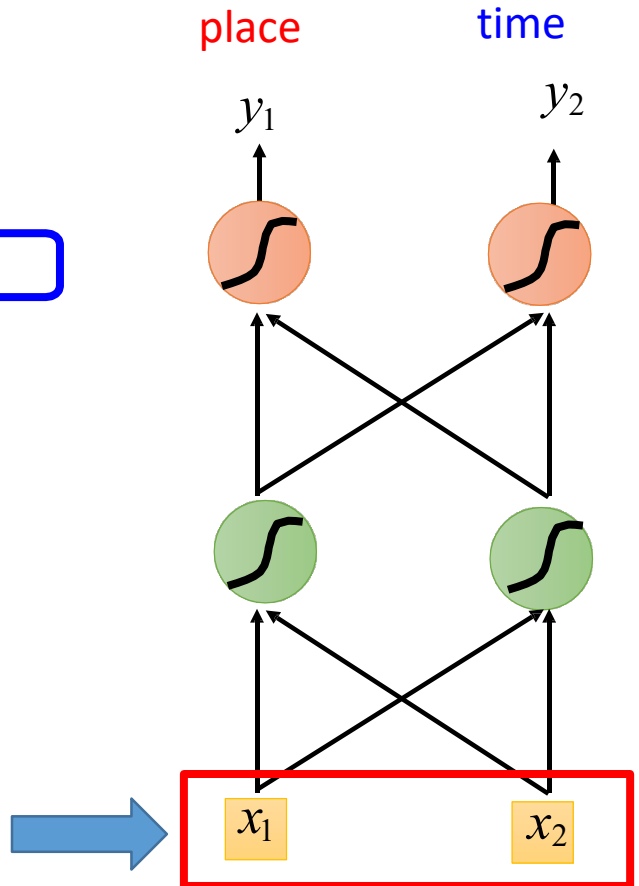
Word hashing



示例应用

- 通过前馈神经网络解决填空问题?
  - 输入：a word (每个单词都被表示成一个向量)
  - 输出：输入单词属于空位的概率分布





示例应用程序

leave Xiamen on November 2nd

other place other time time

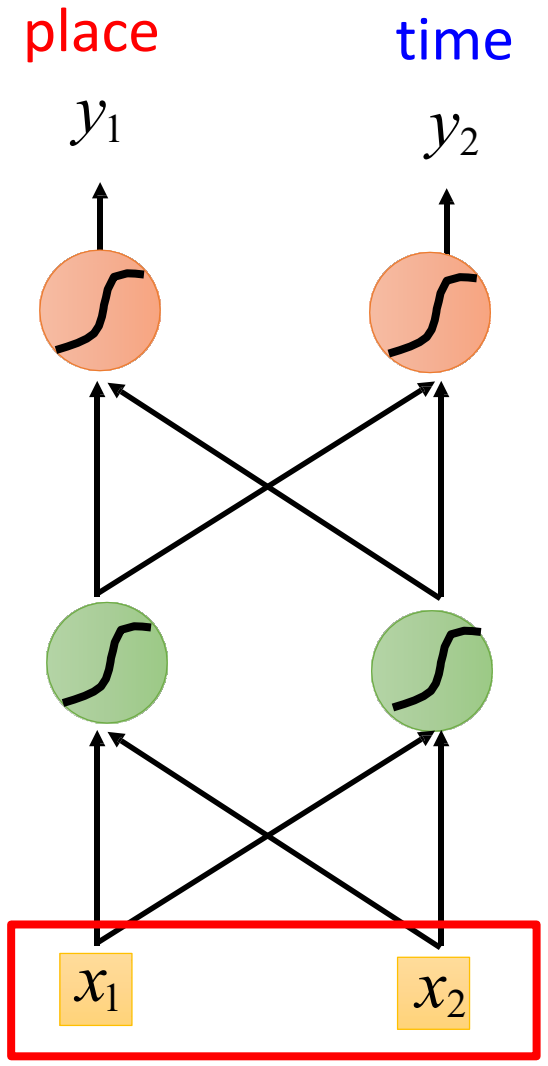
Problem?

leave Hangzhou on November 2nd

place

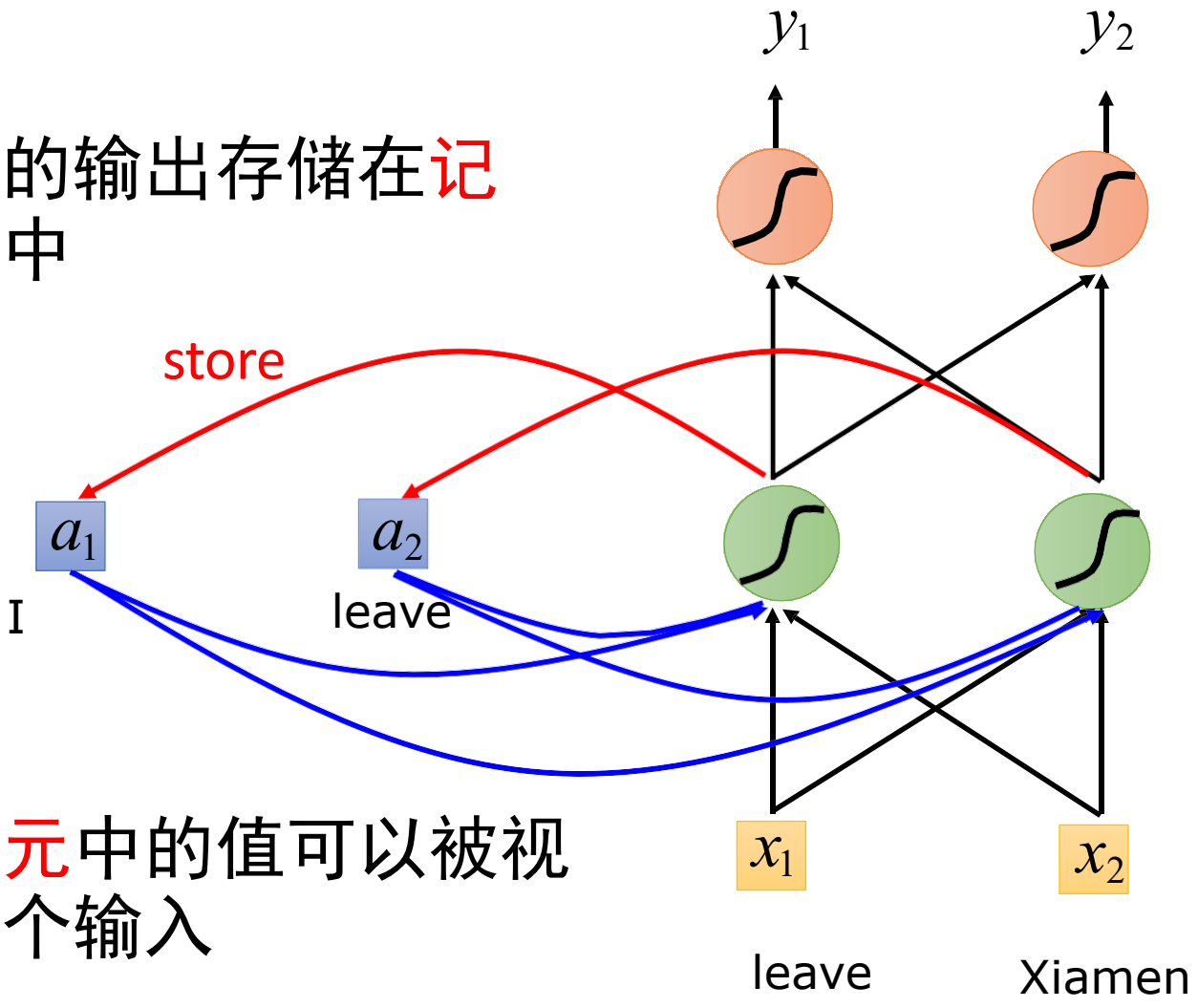
神经网络需要记忆

Xiamen



循环神经网络（RNN）

隐藏层的输出存储在记忆单元中



记忆单元中的值可以被视为另一个输入

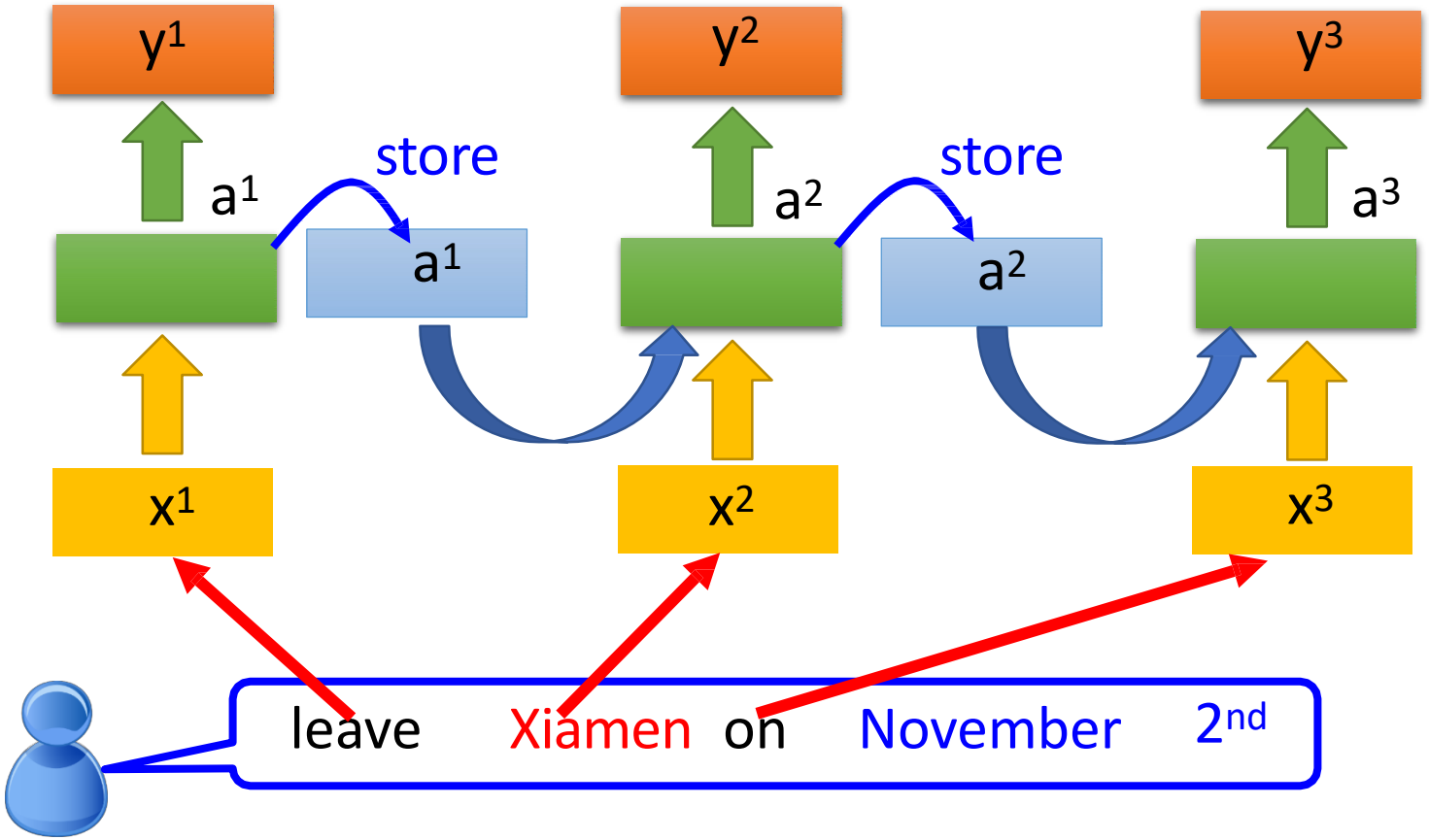
循环神经网络（RNN）

同样的网络多次使用

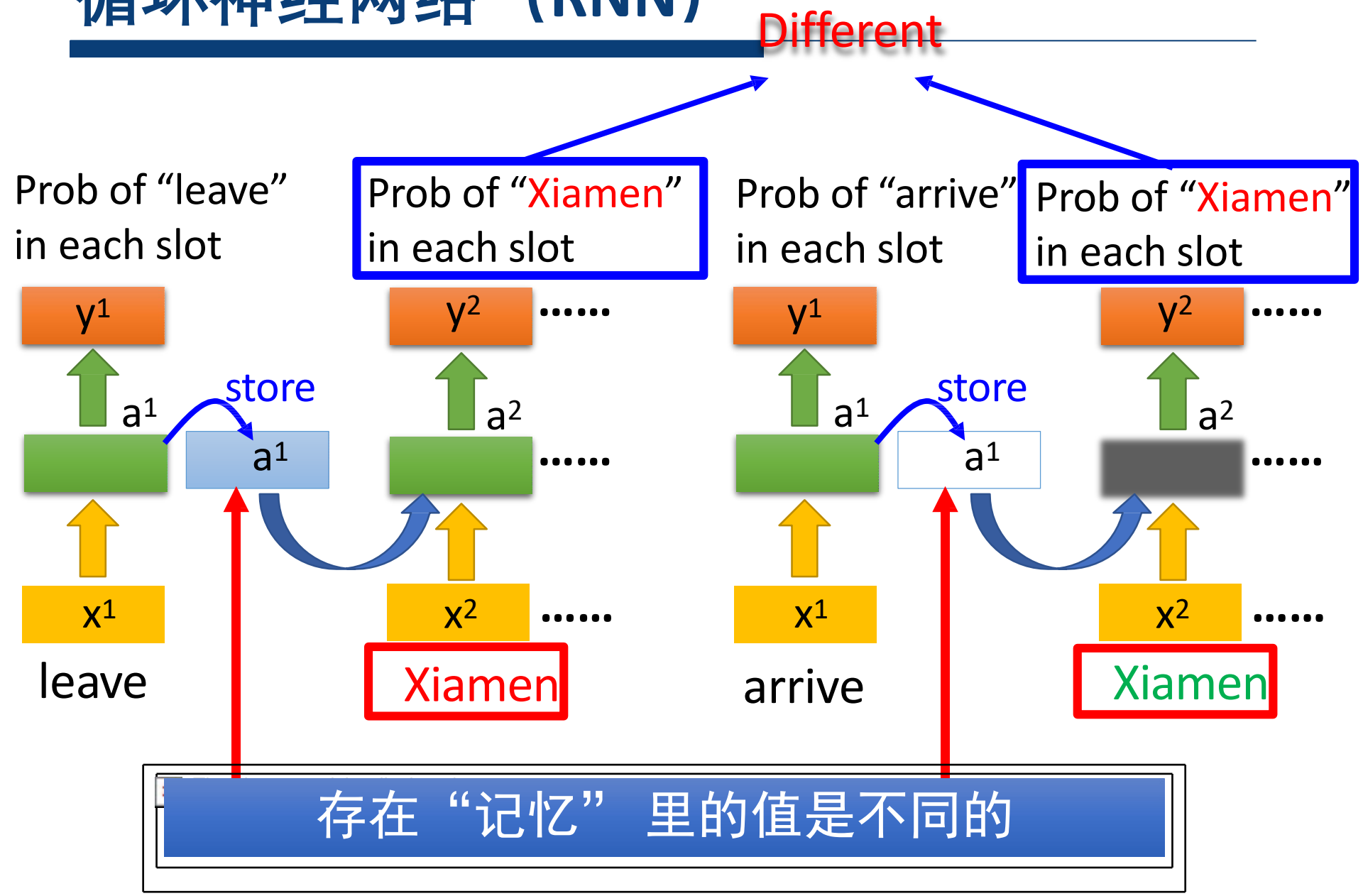
Probability of "arrive" in each slot

Probability of "Xiamen" in each slot

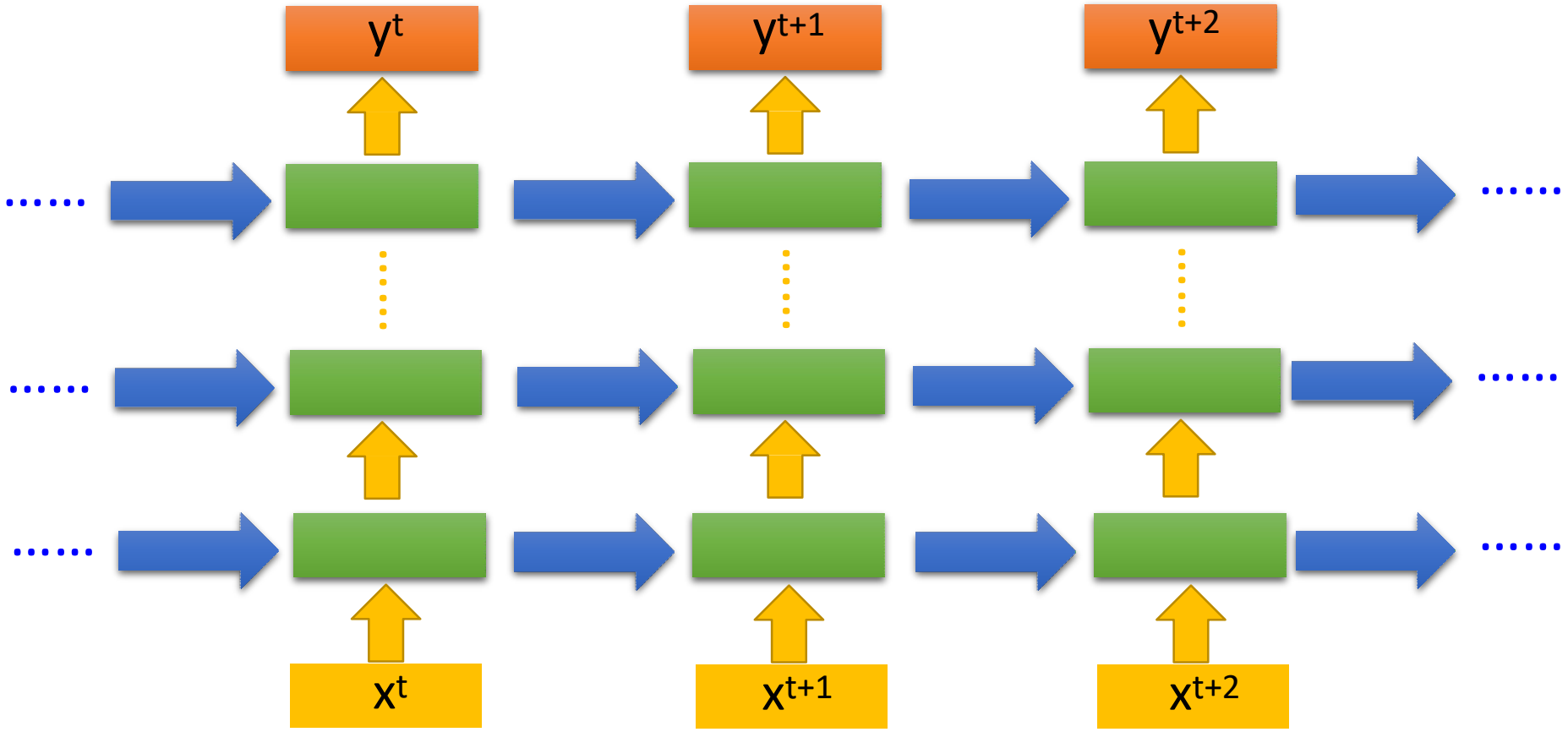
Probability of "on" in each slot



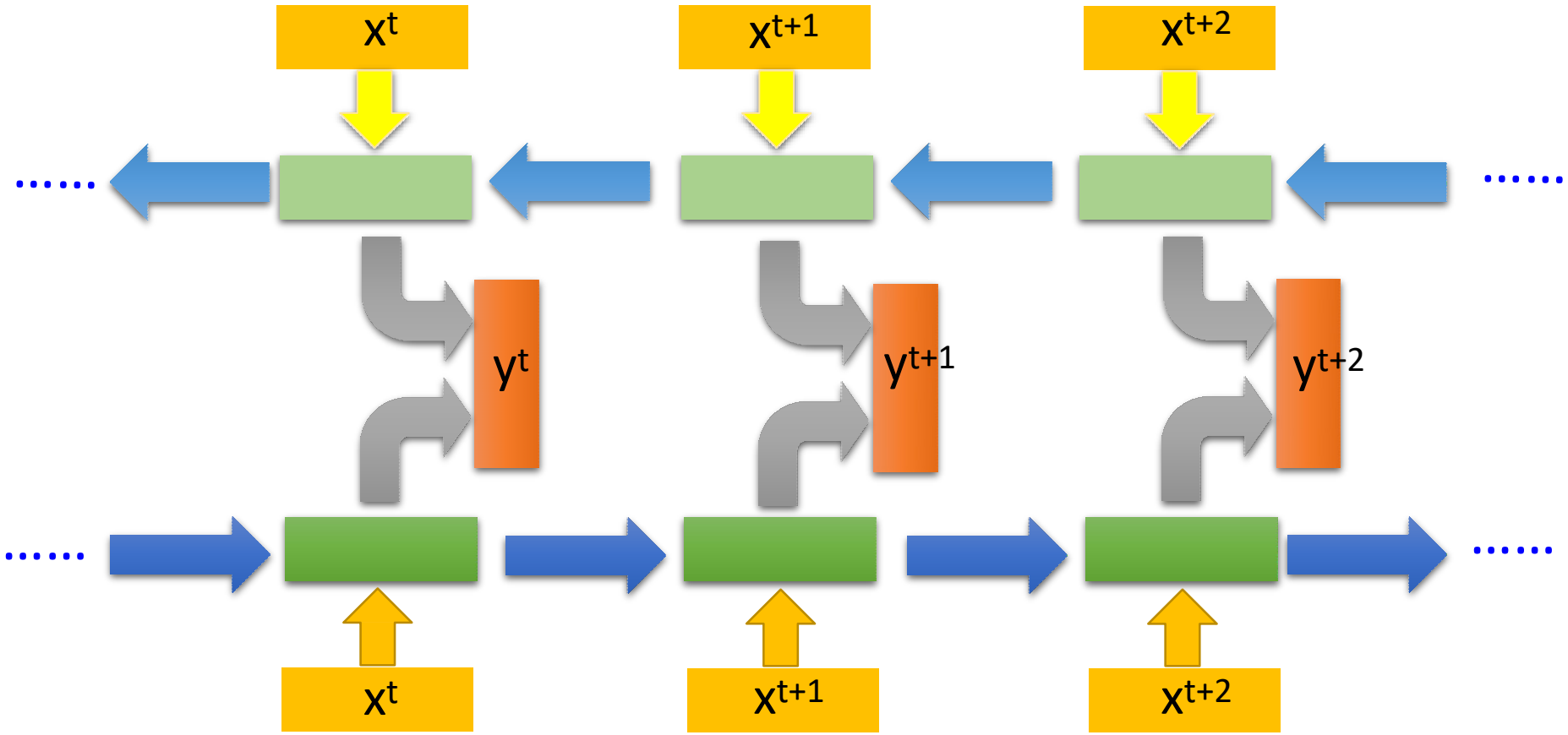
循环神经网络（RNN）


Different
Prob of “leave” in each slot
Prob of “Xiamen” in each slot
Prob of “arrive” in each slot
Prob of “Xiamen” in each slot
y1
y2
a1
a2
store
x1
x2
leave
Xiamen
arrive
Xiamen
存在“记忆”里的值是不同的

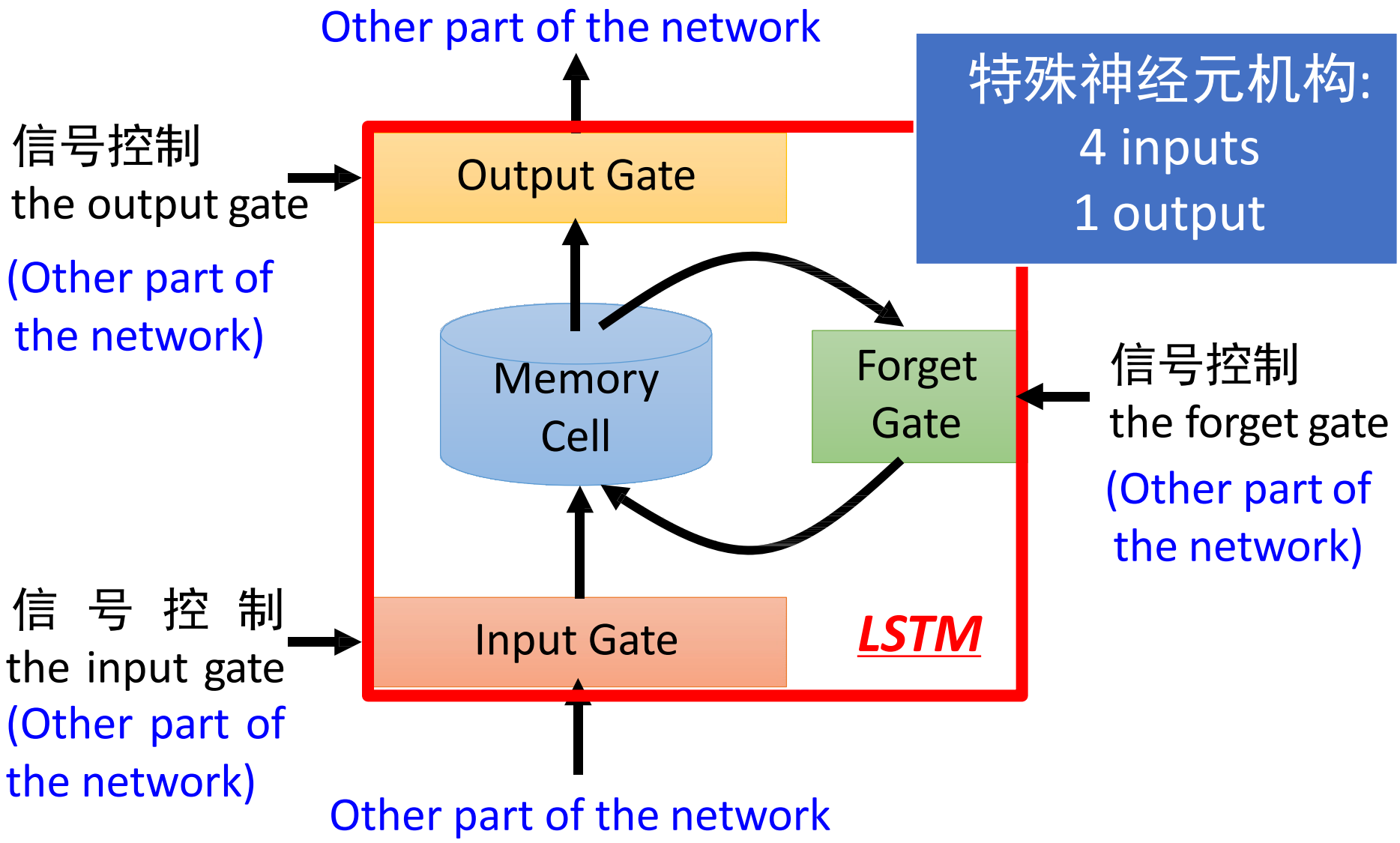
当然，它可以是深层网络 …



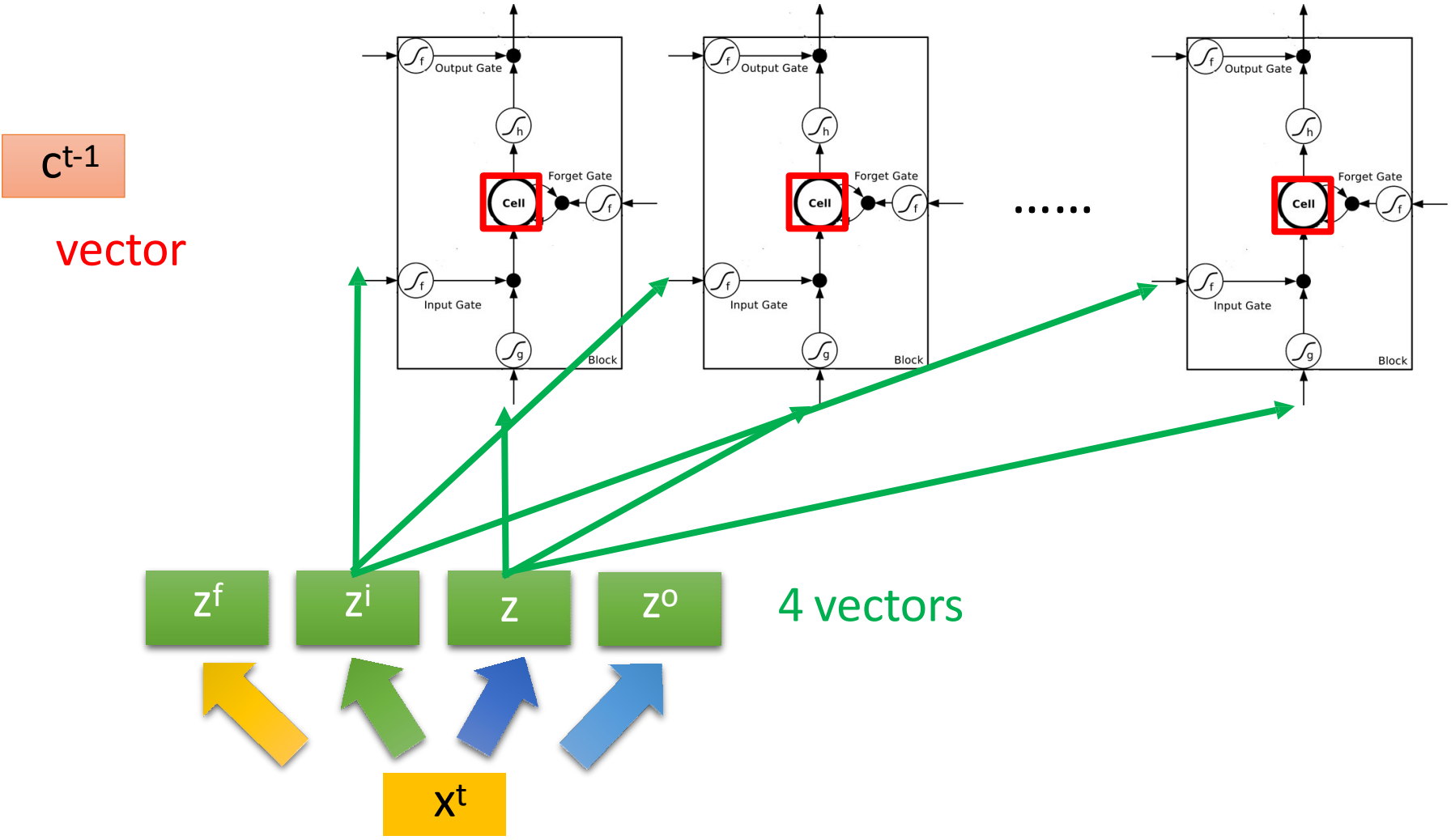
双向 RNN



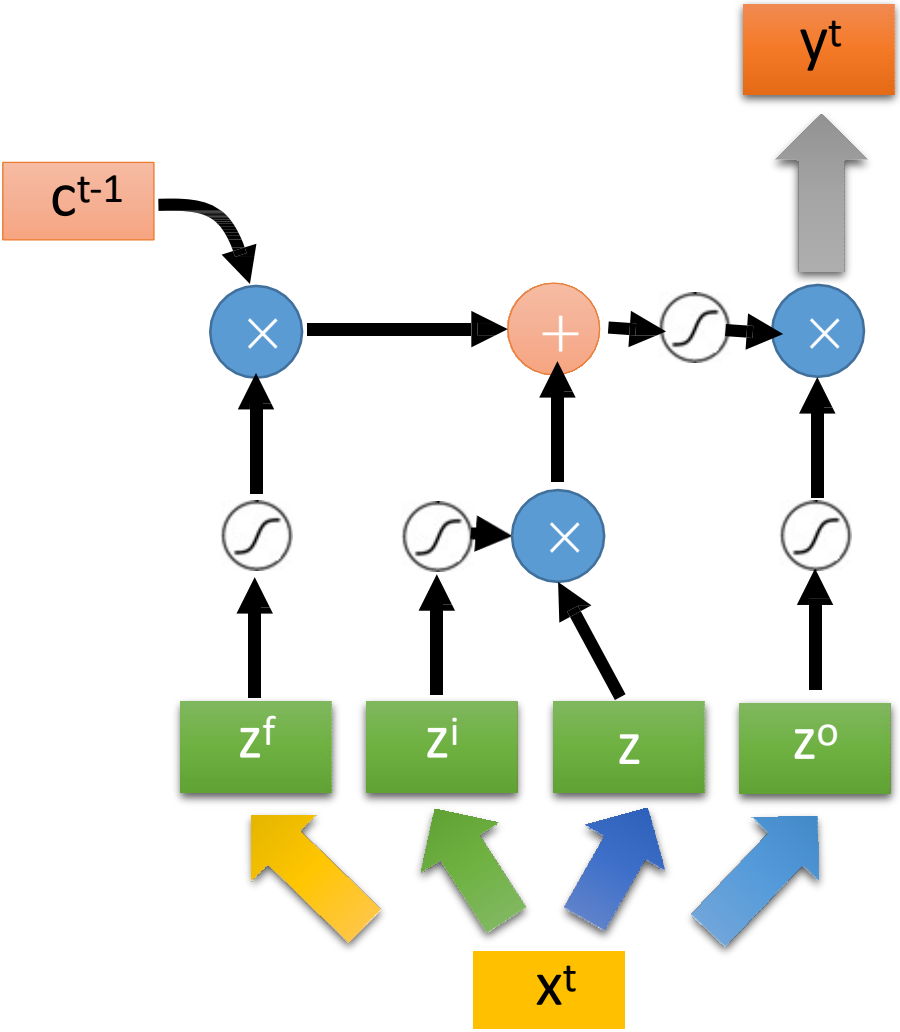
Long Short-term Memory (LSTM)

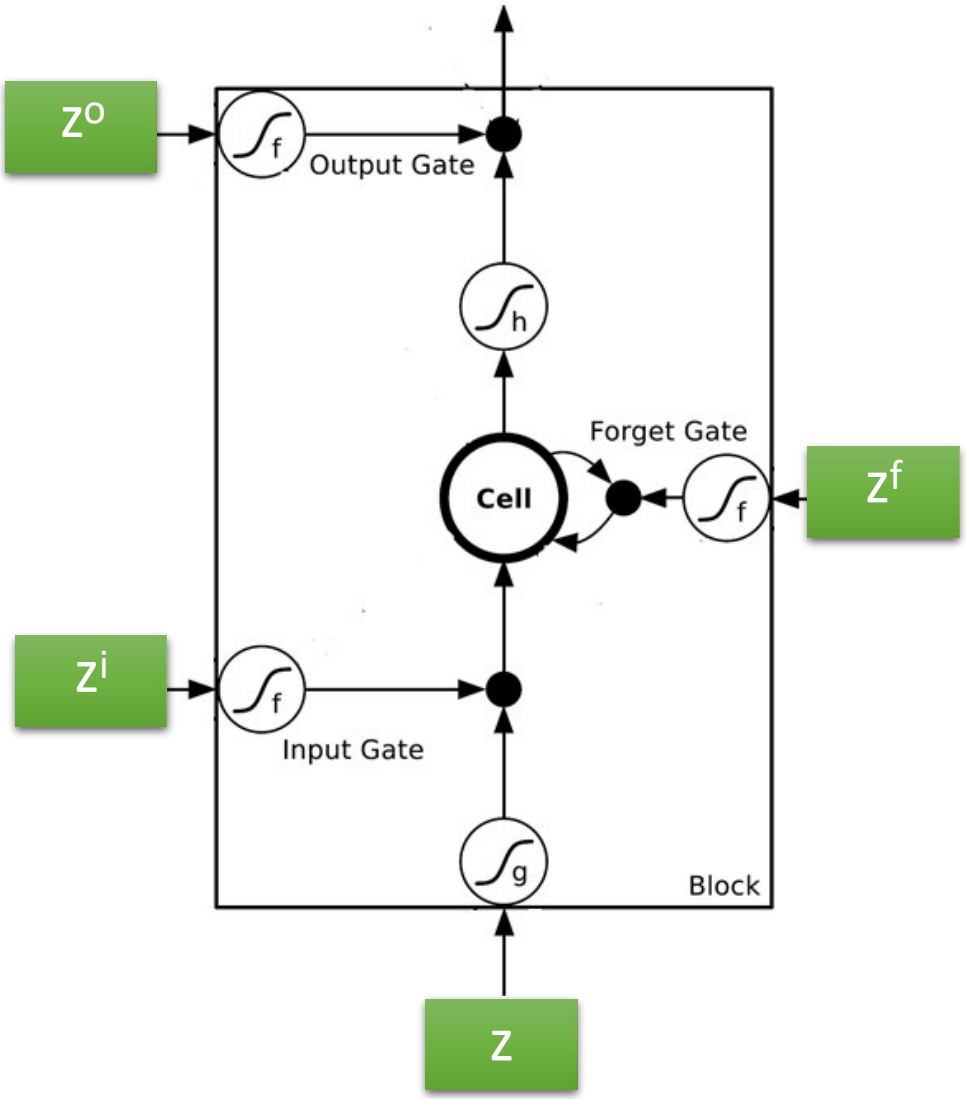


LSTM

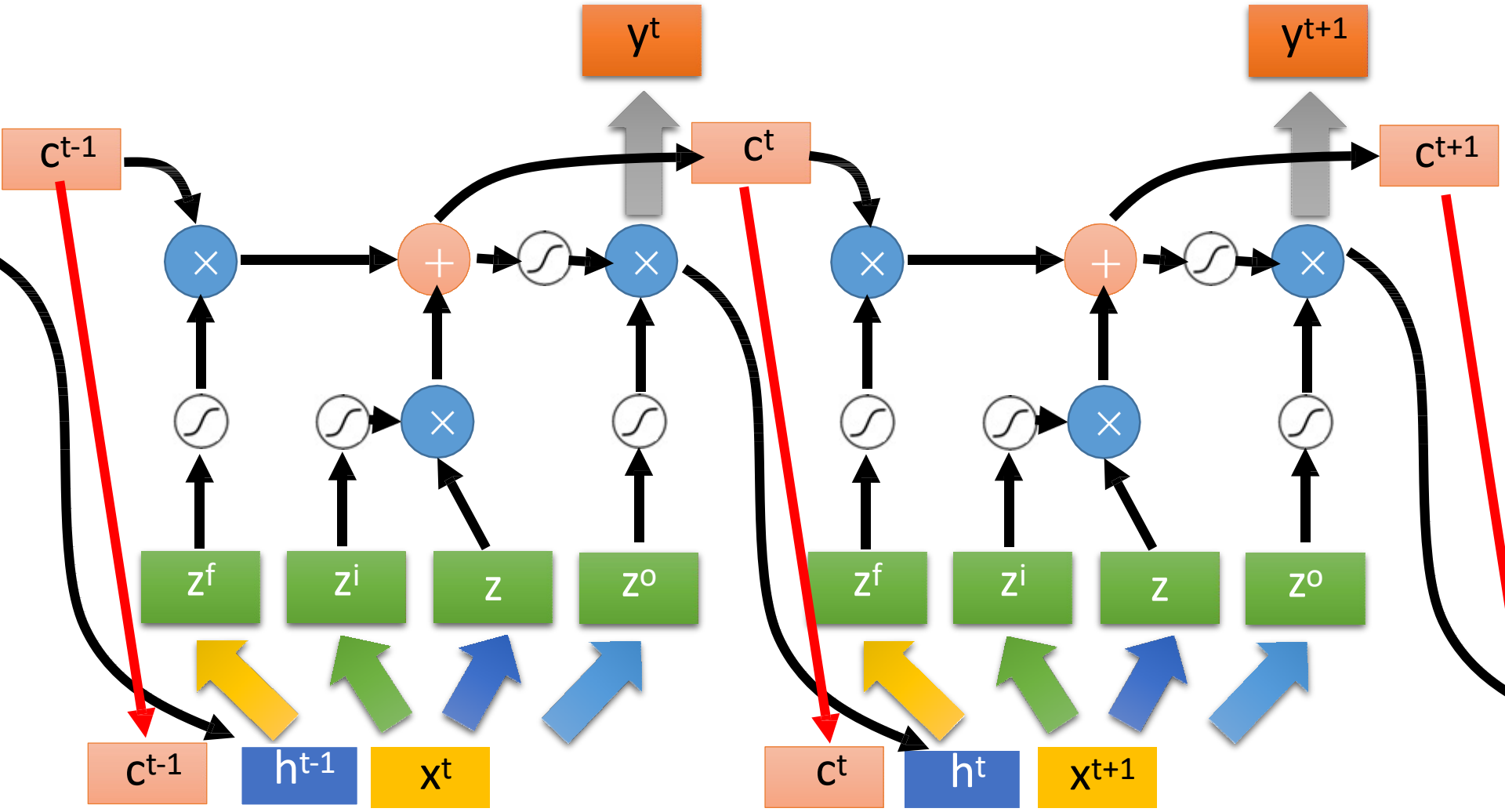

c^{t-1}
vector
Output Gate
Forget Gate
Cell
Input Gate
Block
.....
z^f
z^i
z
z^o
4 vectors
x^t

LSTM

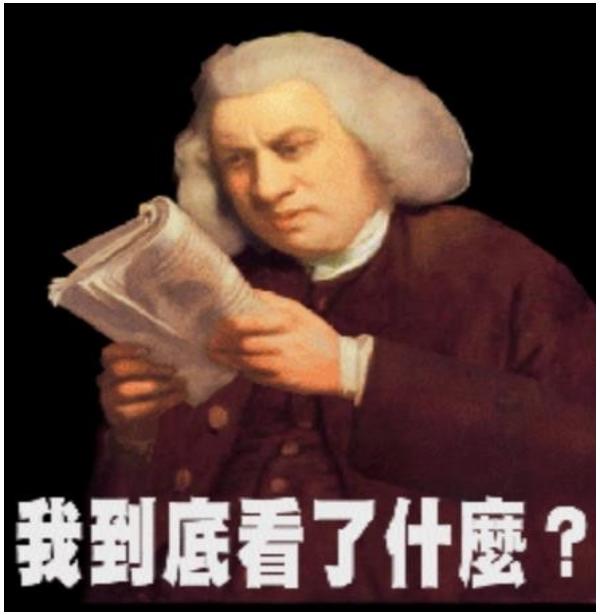

y^t
c^{t-1}
z^f
z^i
z
z^o
x^t

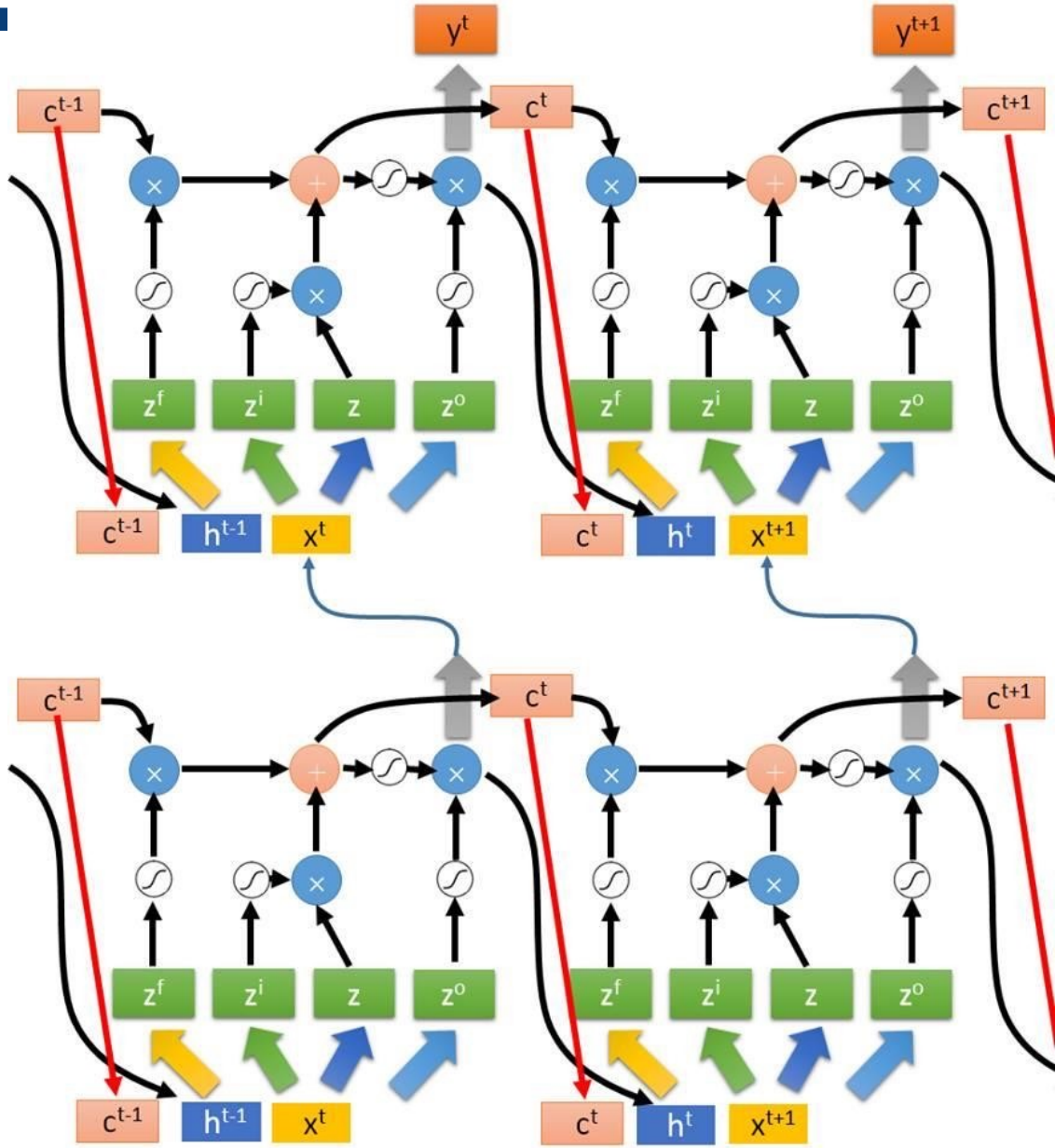

z^o
Output Gate
h
Forget Gate
Cell
z^f
z^i
Input Gate
g
Block
z

LSTM

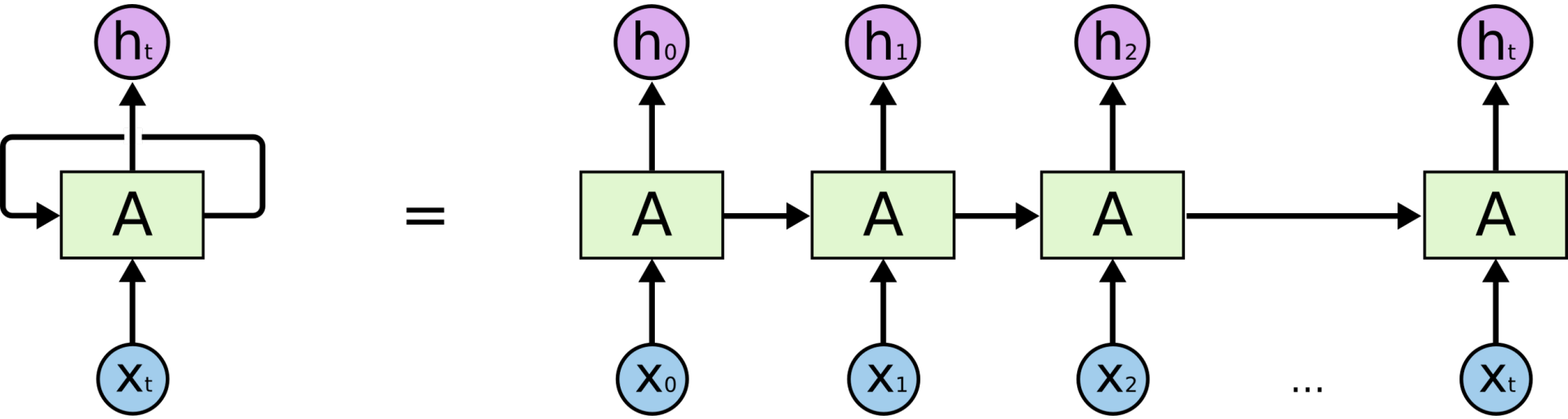


Multiple-layer LSTM

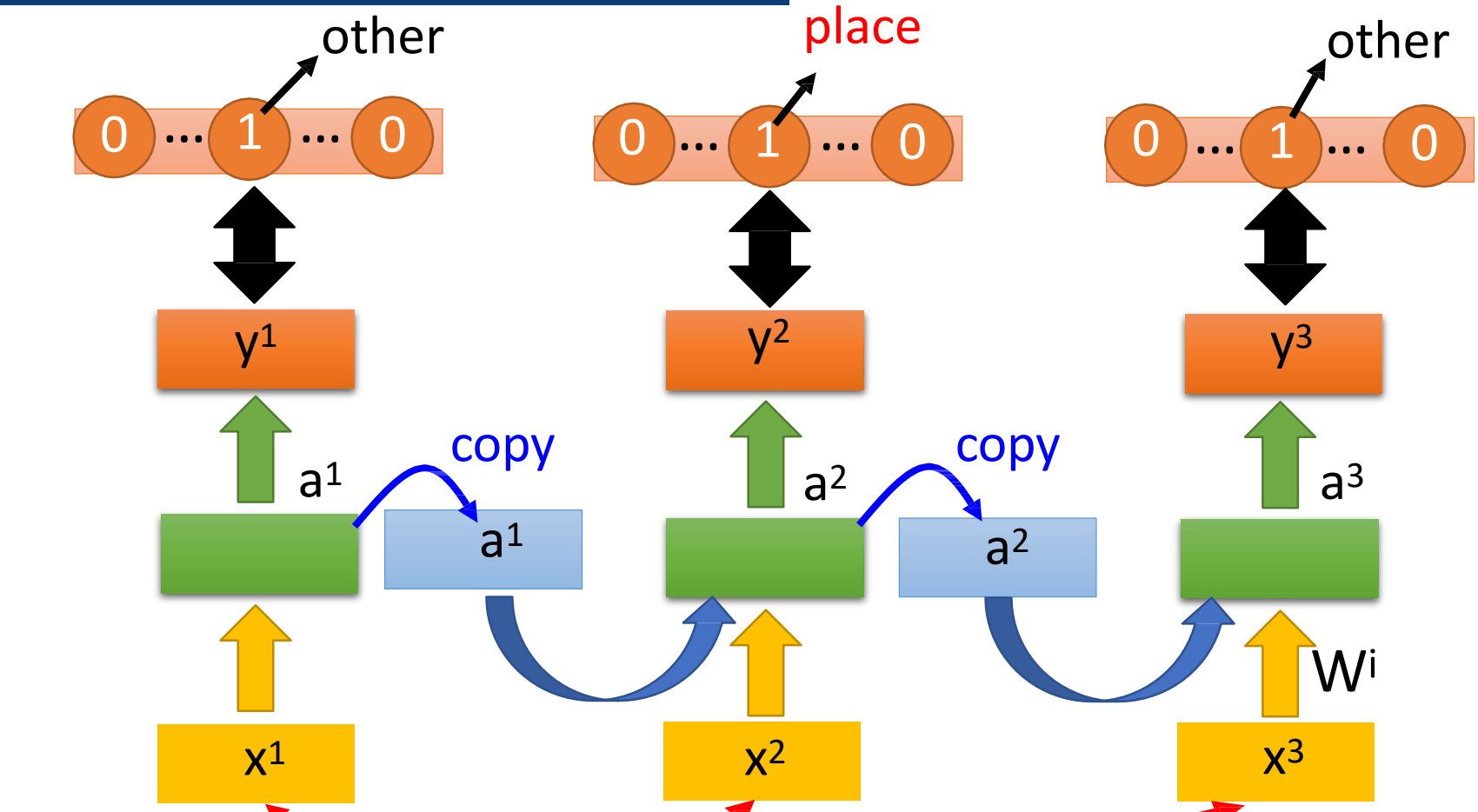




RNN结构

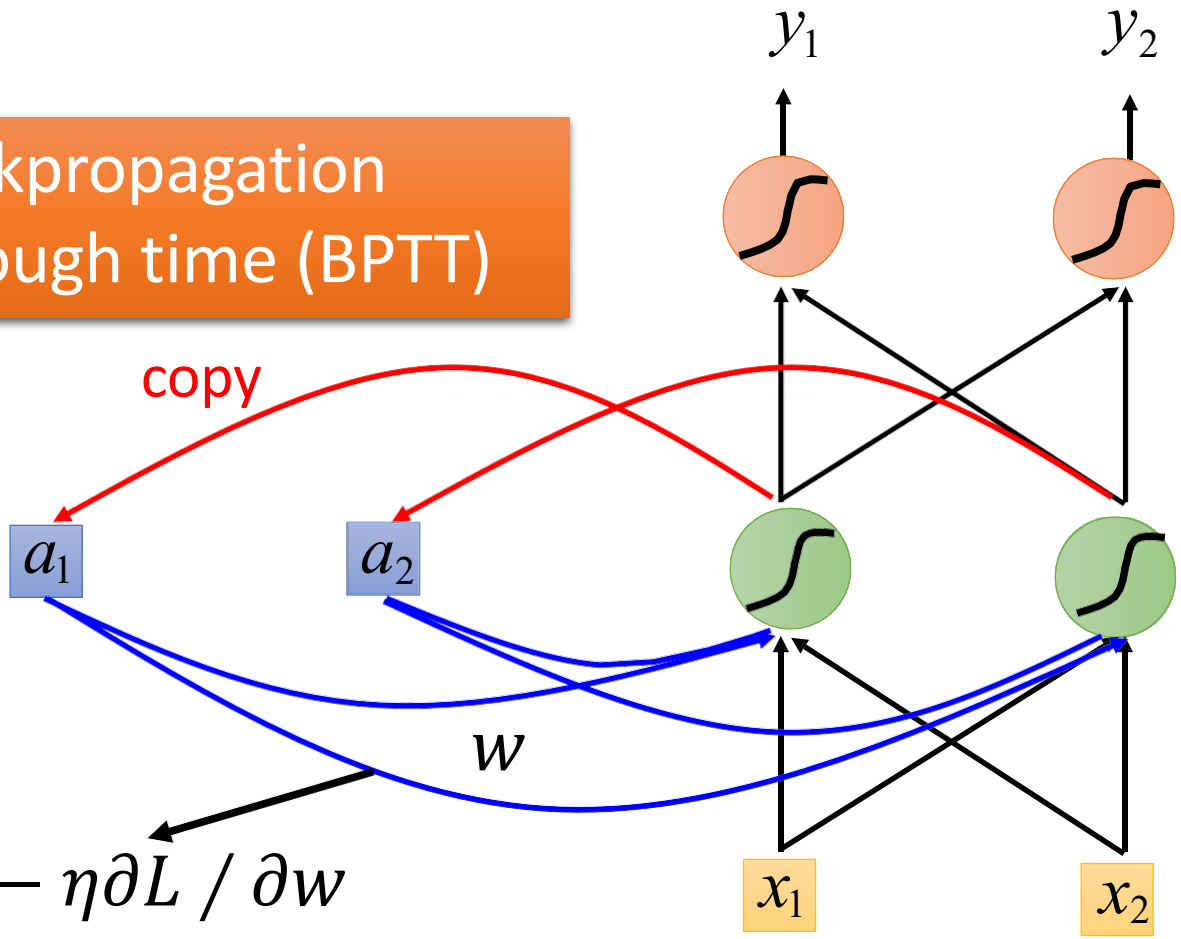


Learning Target



Learning

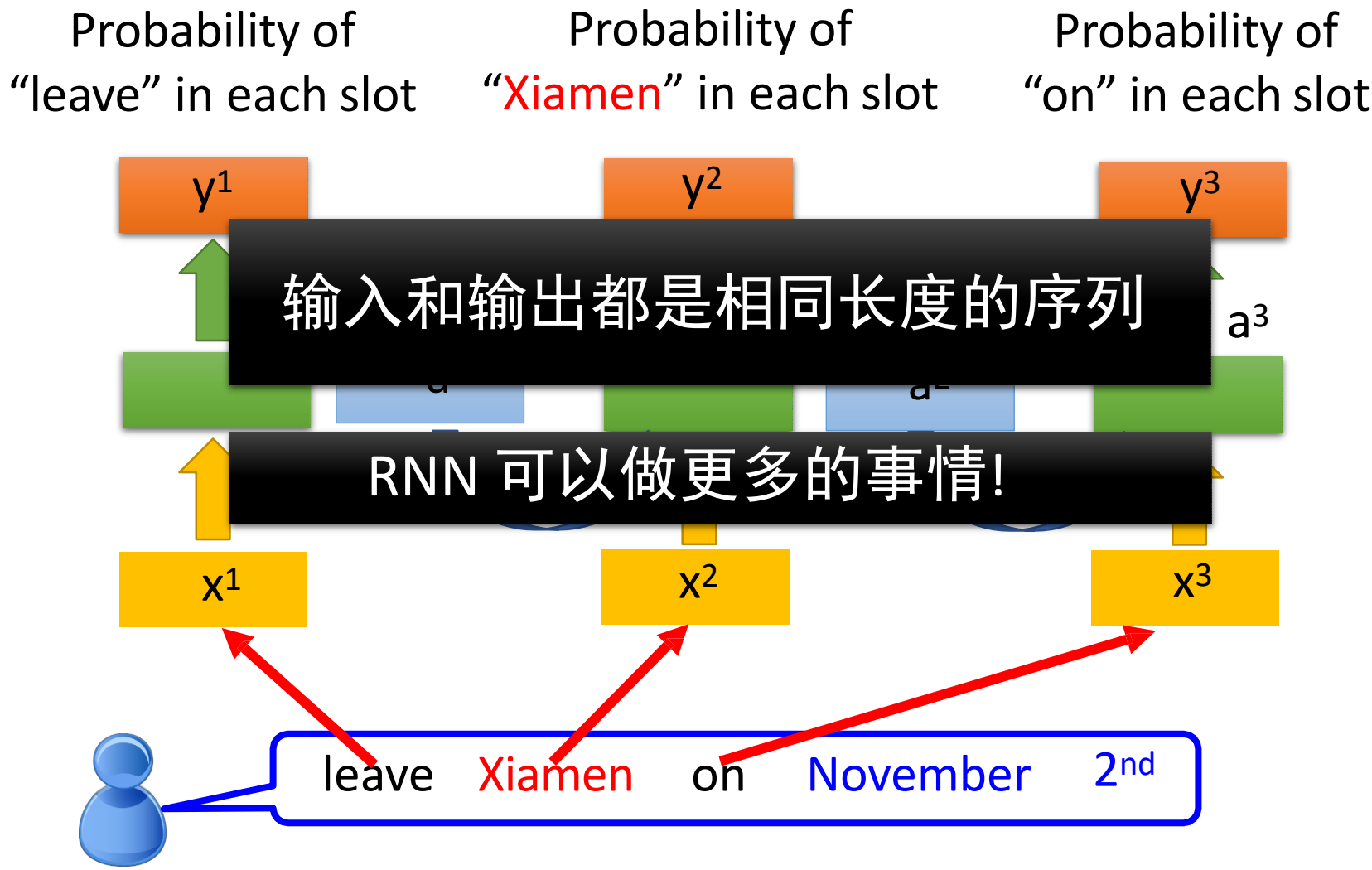
Backpropagation through time (BPTT)



$$w \leftarrow w - \eta \partial L / \partial w$$

RNN的学习在实践中是非常困难的

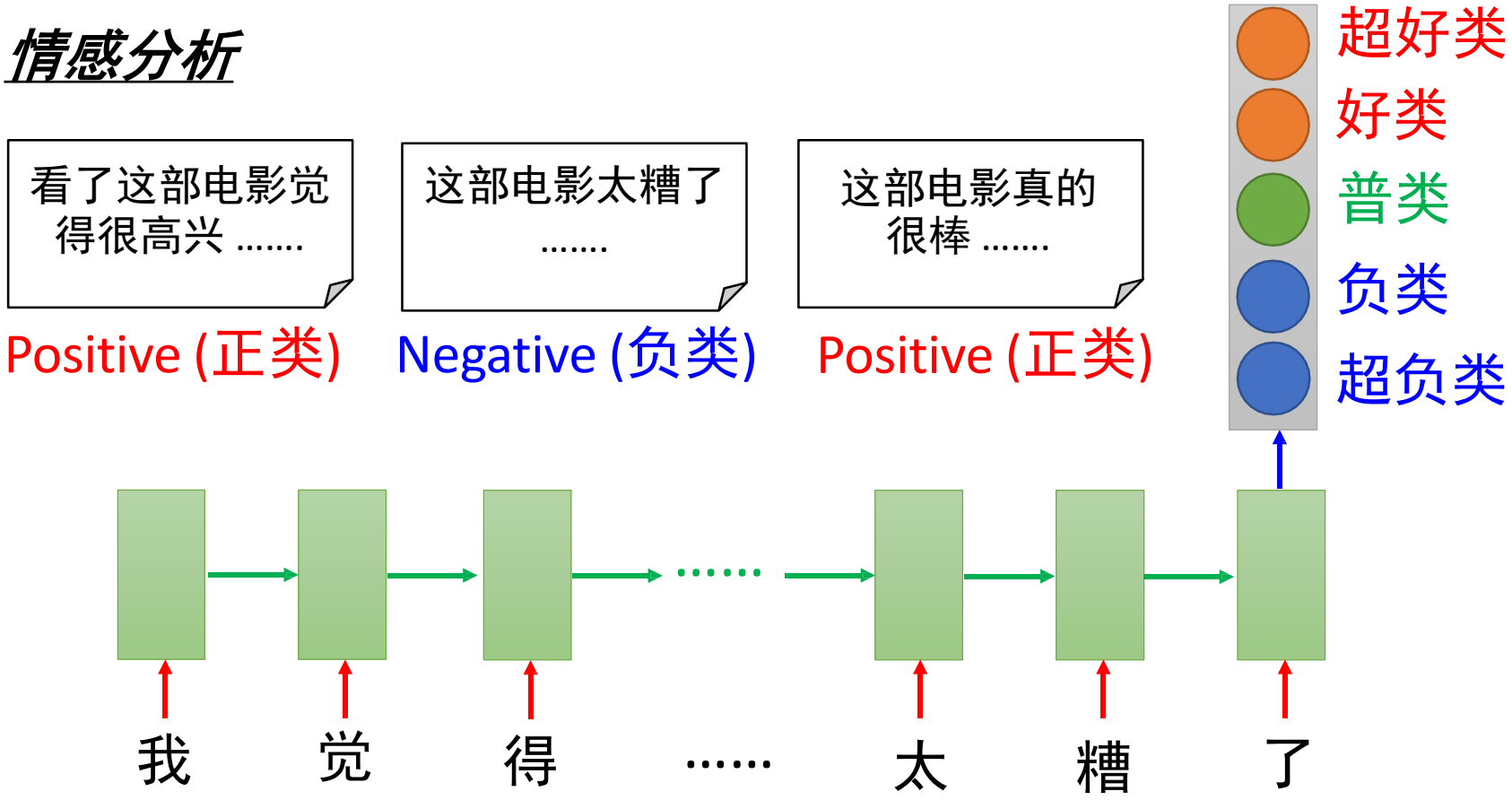
更多应用


Probability of "leave" in each slot
Probability of "Xiamen" in each slot
Probability of "on" in each slot
y1
y2
y3
a3
输入和输出都是相同长度的序列
RNN 可以做更多的事情!
x1
x2
x3
leave
Xiamen
on
November
2nd

Many to one

- 输入是一个向量序列，但输出只是一个向量

情感分析



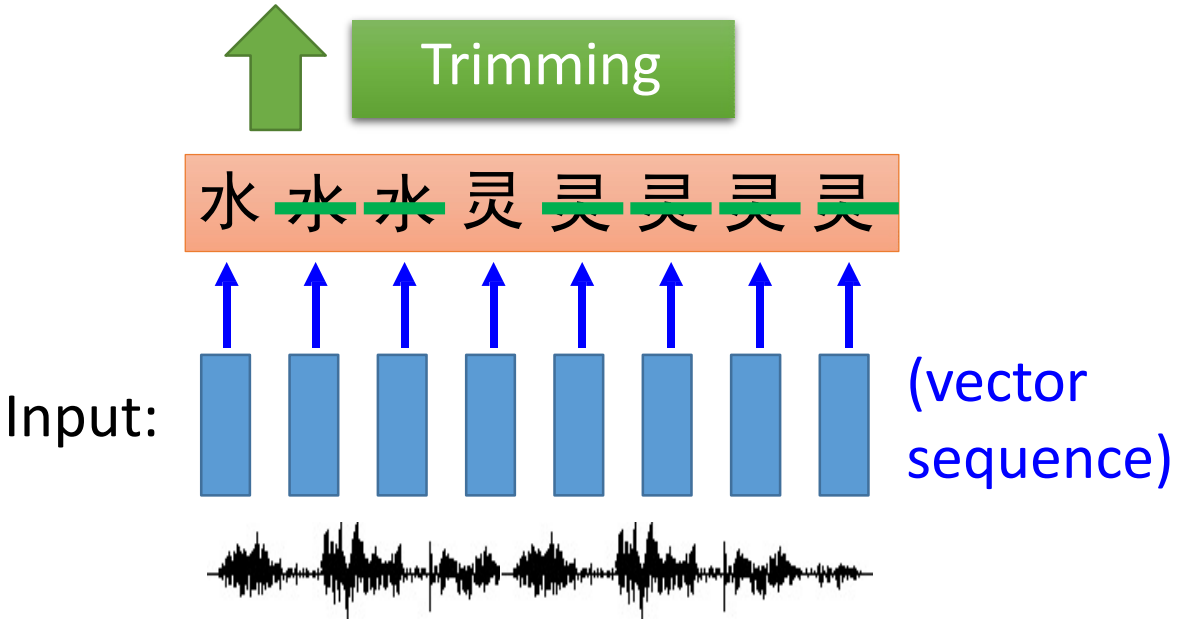
Many to Many (输出较短)

- 输入和输出都是序列, ***但输出比较短***
  - E.g. 语音识别

Output: “水灵” (character sequence)

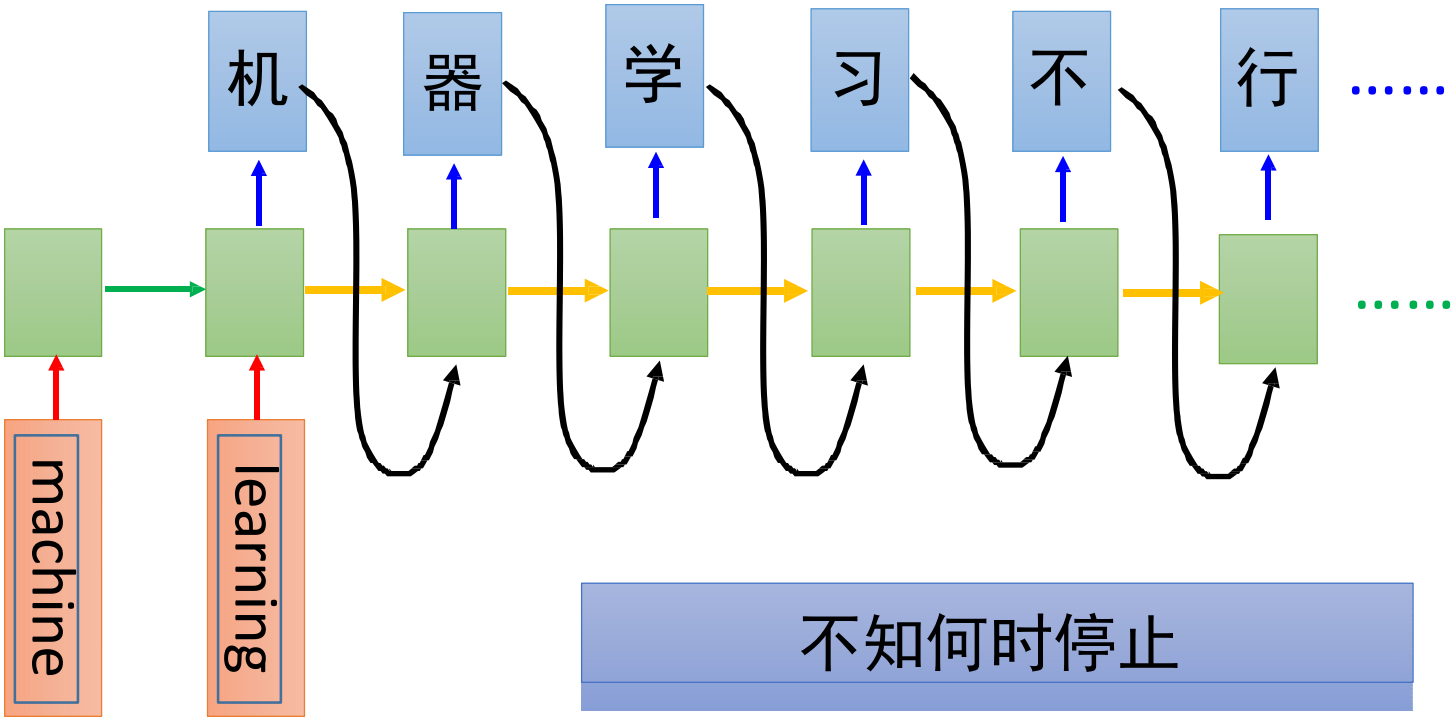
Problem?

Why can’t it be “水灵灵”



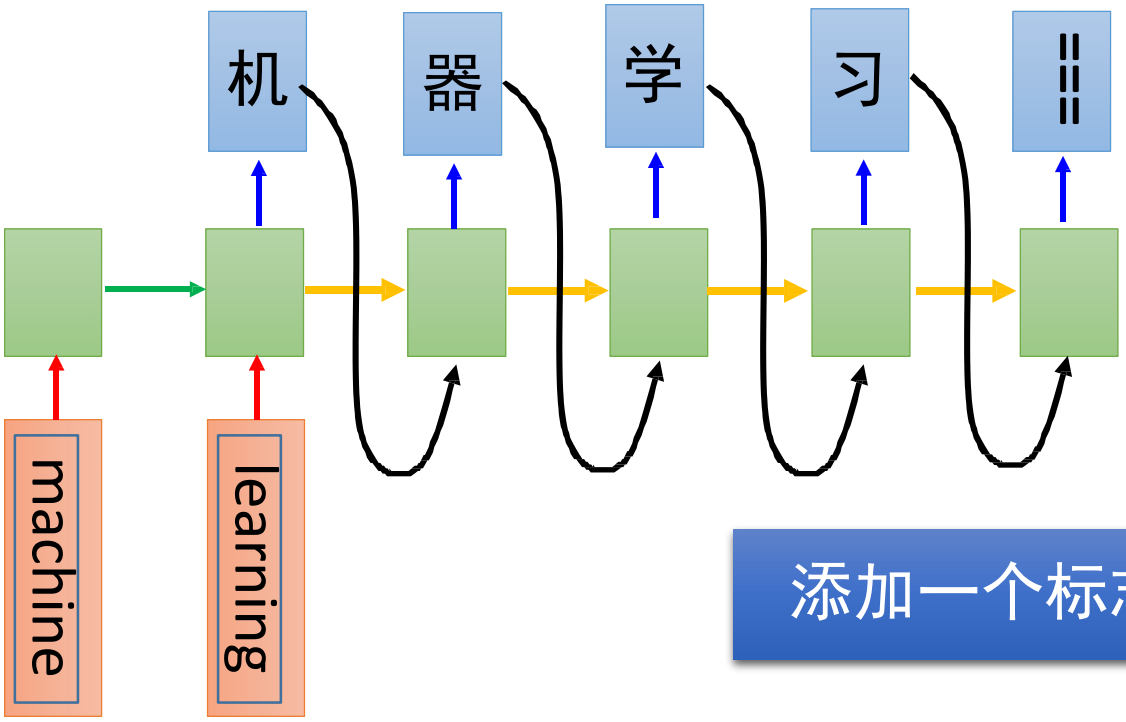
Many to Many (No Limitation)

- 输入和输出都是不同长度的序列 → ***Sequence to sequence learning***
  - E.g. ***机器翻译*** (machine learning→机器学习)



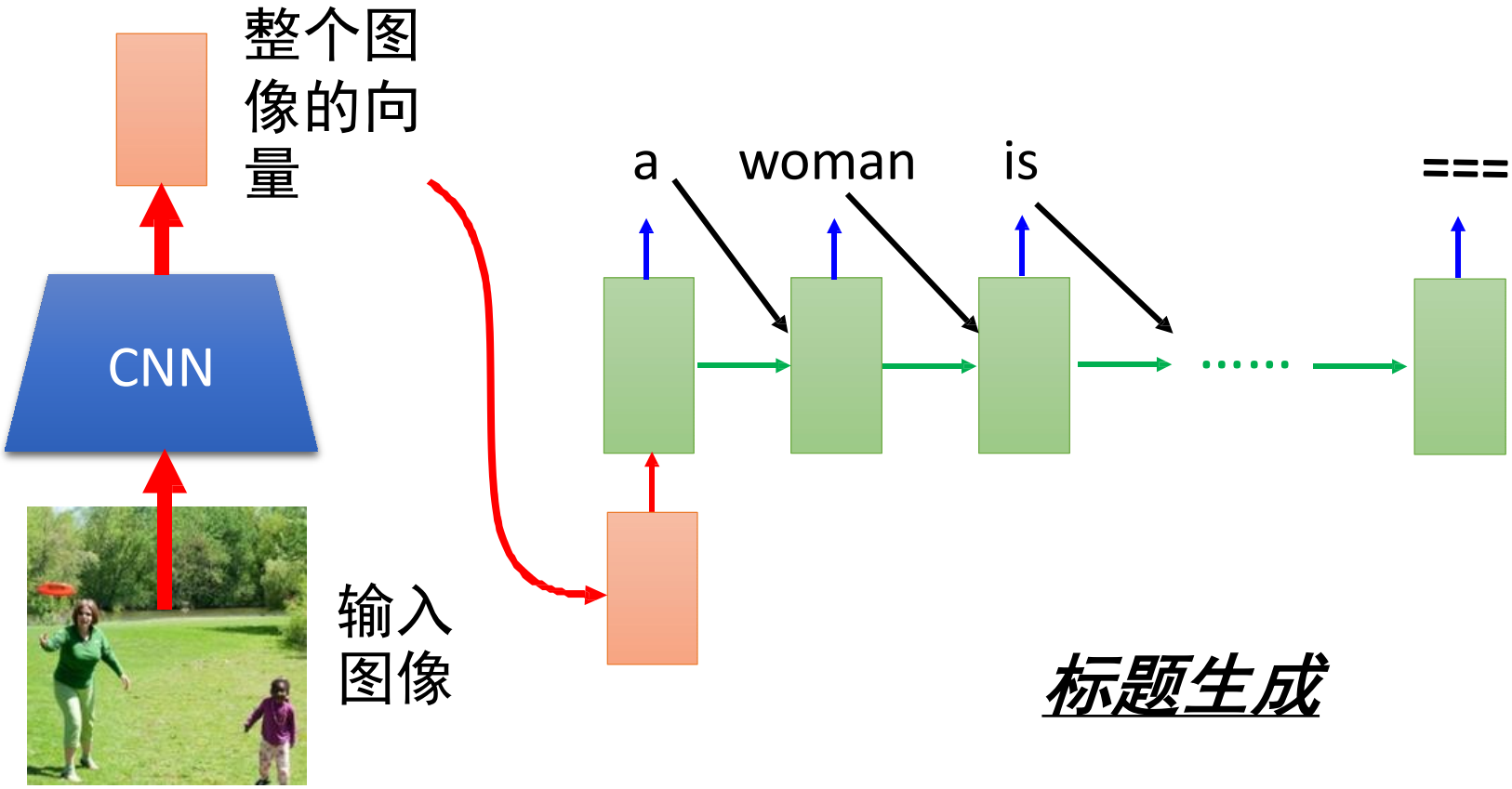
Many to Many (没有限制)

- 输入和输出都是不同长度的序列 → ***Sequence to sequence learning***
  - E.g. ***机器翻译*** (machine learning→机器学习)



One to Many

- 输入图像，但输出一系列单词



Demo

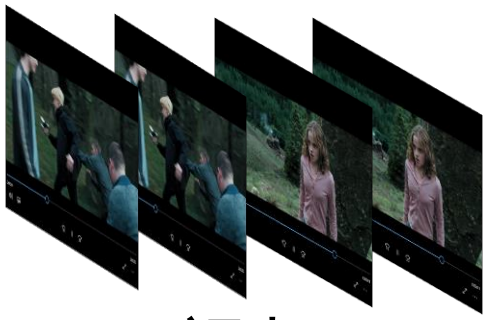
- **Image Captioning**
  - **https://www.captionbot.ai**
    - Prof. Cheng WANG
    - 皮卡丘
    - 皮卡
    - Selfie Lisa !

Microsoft

CaptionBot

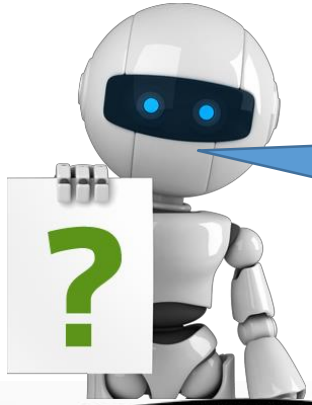
I am not really confident, but I think it's Mona Lisa et al. taking a selfie.

应用：视频字幕生成



视频

一群人被一棵树撞到



一个女孩子正在奔跑

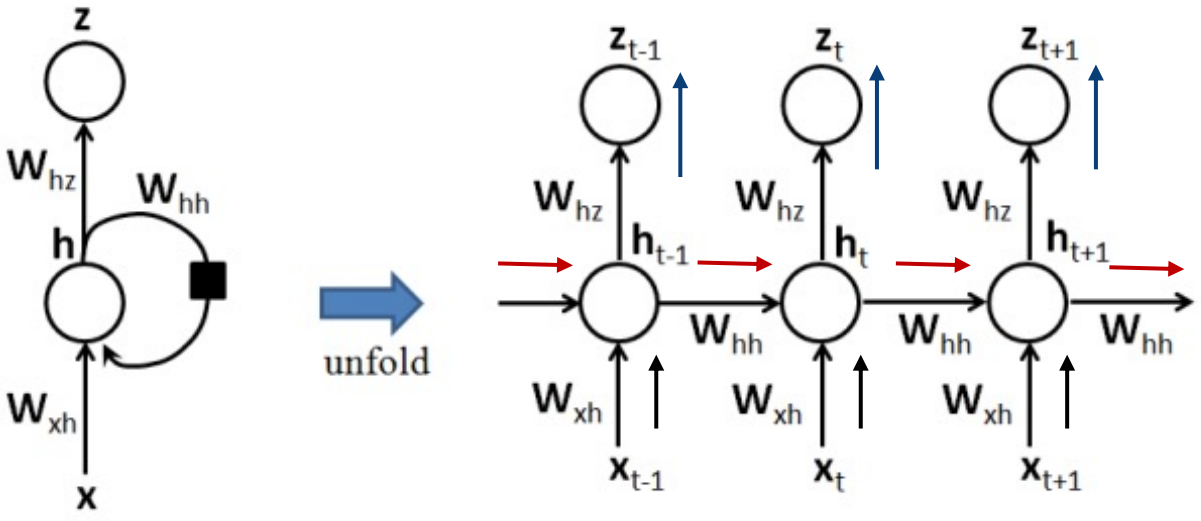
一群人走在森林里

Demo

- Sketch-RNN
  - https://magenta.tensorflow.org/assets/sketch_rnn_demo/index.html
  - Cat and Fish

循环神经网络模型

时序扩展



$$\mathbf{h}_t = \tanh(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h)$$

$$\mathbf{z}_t = \text{softmax}(\mathbf{W}_{hz}\mathbf{h}_t + \mathbf{b}_z)$$

Recurrent Neural Network

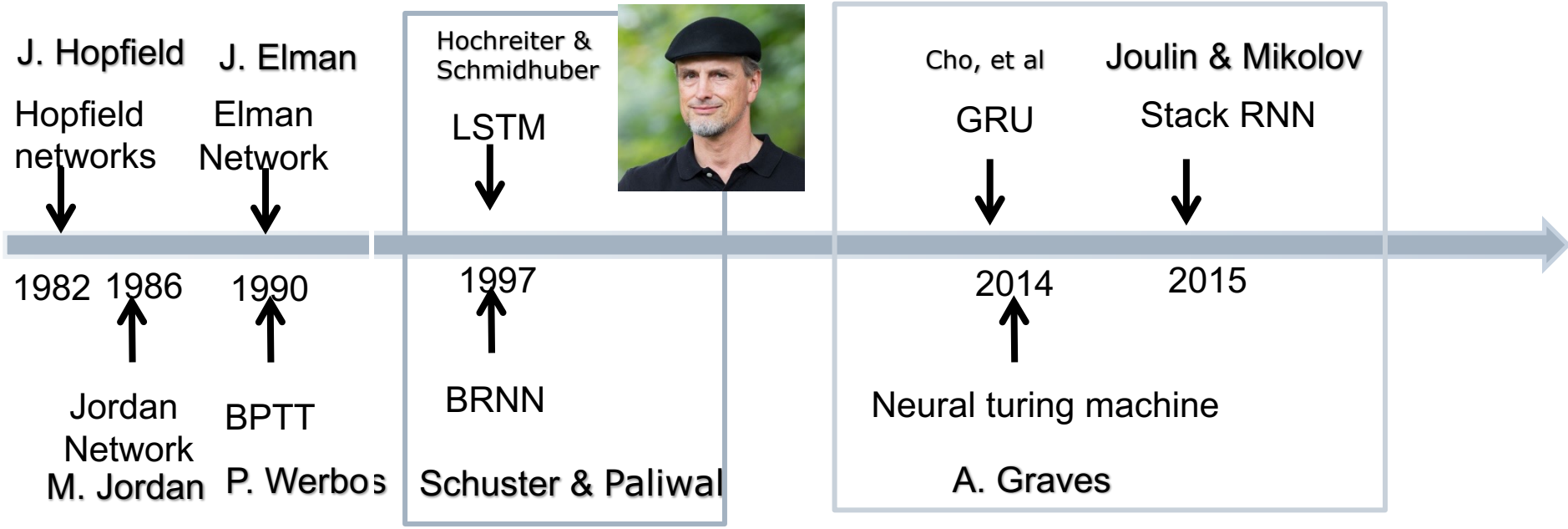
RNN是一类扩展的人工神经网络，它是为了对序列数据进行建模而产生的。

- 针对对象：序列数据。例如文本，是字母和词汇的序列；语音，是音节的序列；视频，是图像的序列；气象观测数据，股票交易数据等等，也都是序列数据。

- 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。

$$h_t = \mathcal{H}\left(W_{ih}x_t + W_{hh}h_{t-1} + b_h\right)$$

RNN发展历史



早期（80、90年代）
主要思想：重新使用参数和计算

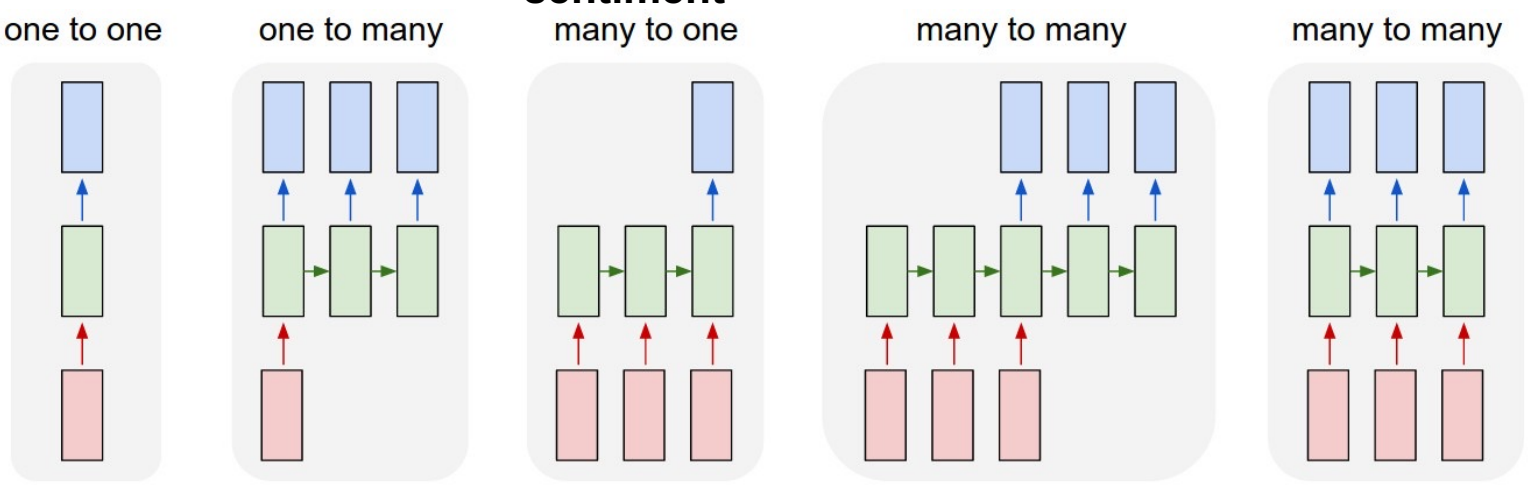
中期（90-2010）
除LSTM以外，RNN基本从主流研究中消失了。

当前（2010 - ）应用广泛：
自然语言应用
视频建模，手写识别，用户意图预测

开源工具包：
Theano
Torch
PyBrain
TensorFlow
,,,

Recurrent Neural Network

RNN基本框架



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
seq of words -> seq of words

Recurrent Neural Network

□ 典型应用：图像标注

"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."

Recurrent Neural Network

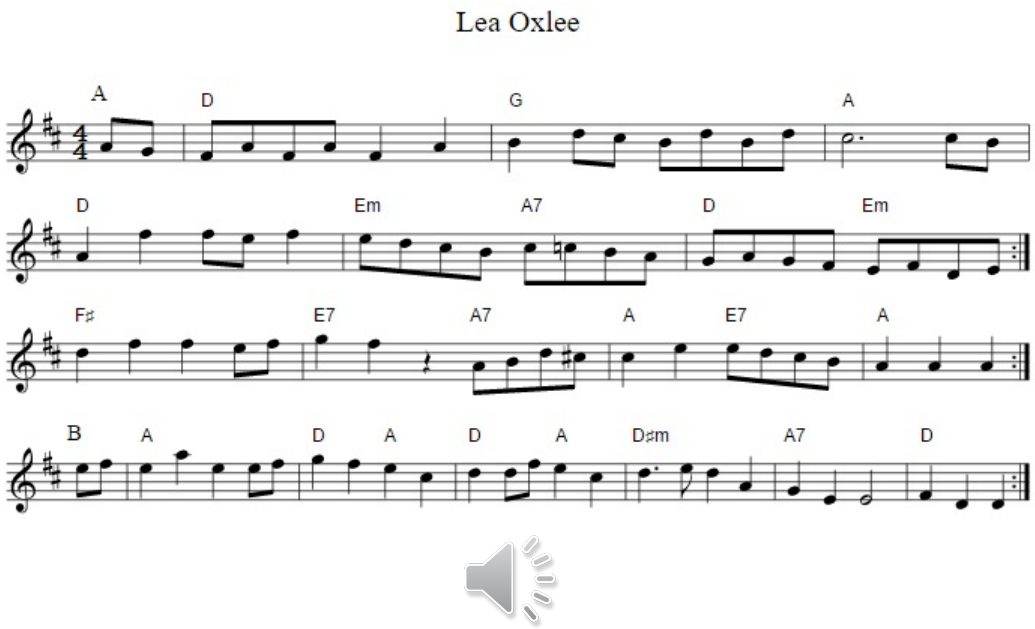
□ 典型应用：语言生成

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

```c
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
    }
  }
  rw->name = "Getjbbregs";
  bprm_self_clearl(&iv->version);
  regs->new = blocks[(BPF_STATS << info->historidac)] | PFMR_CLOBATHINC_SECOND
  return segtable;
}
```

Recurrent Neural Network

□ 典型应用：音乐作曲



循环神经网络模型

softmax

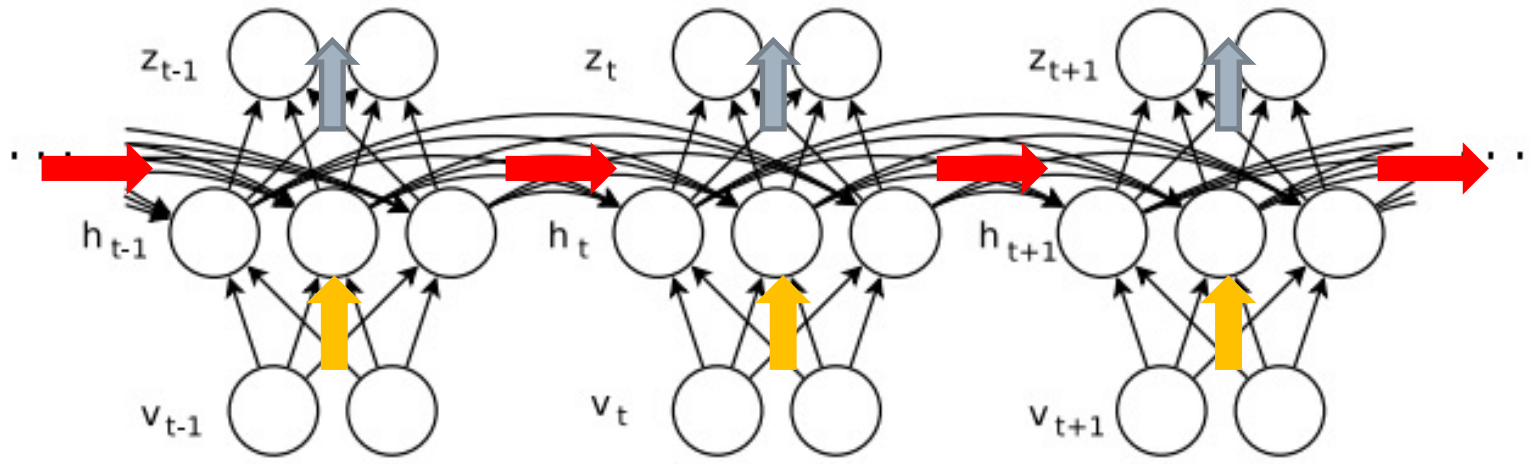
- Softmax函数是sigmoid函数的一个变种，通常我们将其用在多分类任务的输出层，将输入转化成标签的概率。

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)}=1|x^{(i)};\theta) \\ p(y^{(i)}=2|x^{(i)};\theta) \\ \vdots \\ p(y^{(i)}=k|x^{(i)};\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}$$

本质就是将一个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。

循环神经网络模型

简单循环网络SRN



神经元之间的**连接权重**在时域上**不变**。

循环神经网络模型

□ 随时间反向传播算法BPTT

BP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。

BPTT的本质其实和BP很像，就是加上了时序演化。

定义权重*U*，*V*，*W*。

$$s_t = \tanh(Ux_t + Ws_{t-1})$$
$$\hat{y}_t = \mathrm{softmax}(Vs_t)$$

定义损失函数：

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

我们将整个序列作为一次训练，所以需要对每个时刻的误差进行求和。

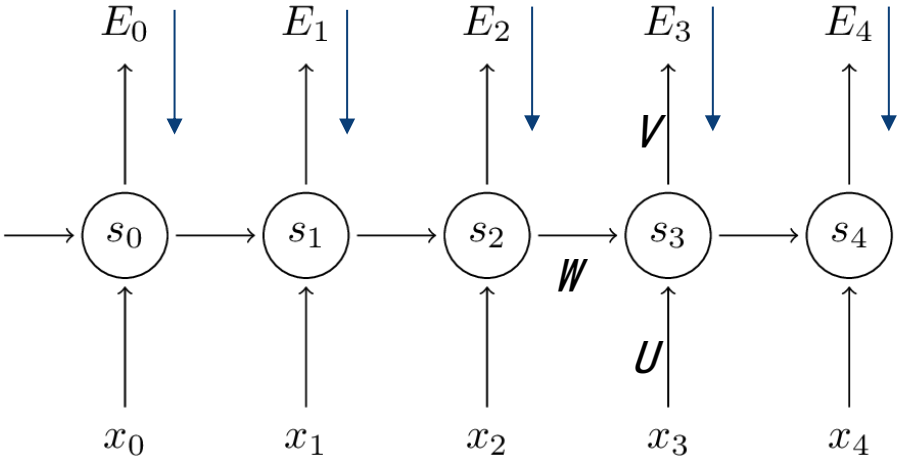
循环神经网络模型

□ 随时间反向传播算法BPTT

目前的任务是求 E 对于*U*，*V*，*W* 的梯度。

定义E对于*W* 的梯度（*U*，*V* 同理）：

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



（1）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：

$$z_3 = V s_3$$

$\frac{\partial E}{\partial V}$ 求和可得。

循环神经网络模型

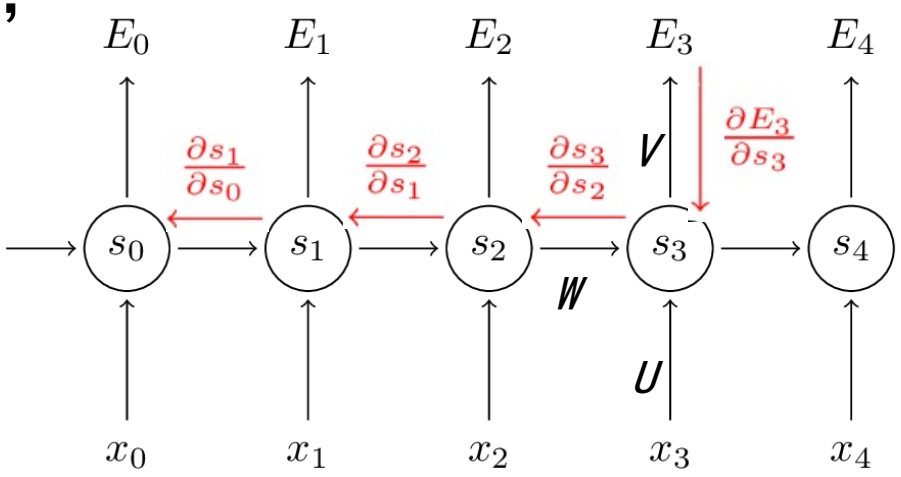
□ 随时间反向传播算法BPTT

$\frac{\partial E}{\partial W}$ 求和可得。

（2）求 E 对于W 的梯度。注意，现在情况开始变得复杂起来。

先求 E3 对于W 的梯度：

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



当我们求 s_3 对于W 的偏导时。

注意到：$s_3 = \tanh(Ux_t + Ws_2)$

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

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！**

循环神经网络模型

□ 随时间反向传播算法BPTT

（3）求 E 对于U 的梯度。

情况与W 类似。

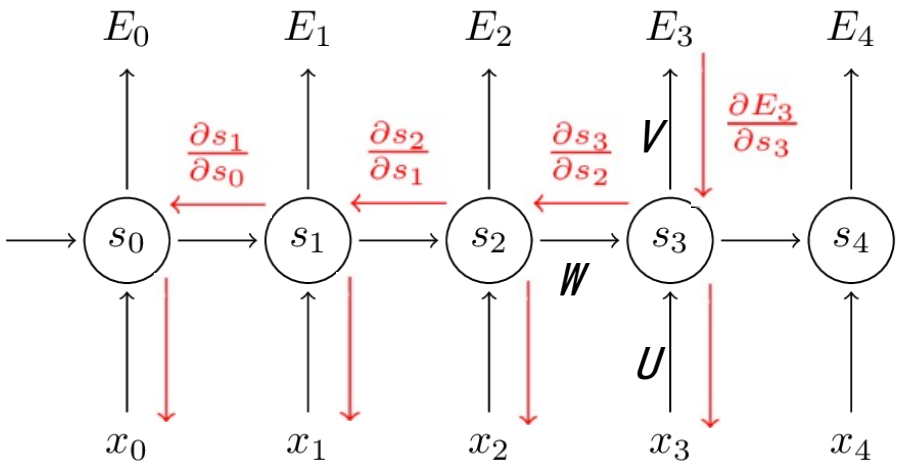
先求 E3 对于U 的梯度：

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

当我们求 s_3 对于W 的偏导时。

注意到：$s_3=\tanh(Ux_t+Ws_2)$

$\frac{\partial E}{\partial U}$ 求和可得。



$$\frac{\partial E_3}{\partial U}=\sum_{k=1}^{3}\frac{\partial E_3}{\partial \hat{y}_3}\frac{\partial \hat{y}_3}{\partial s_3}\frac{\partial s_3}{\partial s_k}\frac{\partial s_k}{\partial U}$$

同样：s_3 依赖于s_2 ，而 s_2 又依赖于 s_1 和U 。

类似求W，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！

循环神经网络模型

□ 随时间反向传播算法BPTT

for t **from** T **downto** 1 **do**

$do_t \leftarrow g'(o_t) \cdot dL(z_t; y_t)/dz_t$

$db_o \leftarrow db_o + do_t$

$dW_{oh} \leftarrow dW_{oh} + do_t h_t^\top$

$dh_t \leftarrow dh_t + W_{oh}^\top do_t$

$dz_t \leftarrow e'(z_t) \cdot dh_t$

$dW_{hv} \leftarrow dW_{hv} + dz_t v_t^\top$

$db_h \leftarrow db_h + dz_t$

$dW_{hh} \leftarrow dW_{hh} + dz_t h_{t-1}^\top$

$dh_{t-1} \leftarrow W_{hh}^\top dz_t$

end for

Return $d\theta = [dW_{hv}, dW_{hh}, dW_{oh}, db_h, db_o, dh_0]$.

参数意义:

W_{hv}:输入层到隐含层的权重参数,

W_{hh}:隐含层到隐含层的权重参数,

W_{oh}:隐含层到输出层的权重参数,

b_h: 隐含层的偏移量,

b_o :输出层的偏移量,

h_0: 起始状态的隐含层的输出,

一般初始为0。

递归神经网络模型

□ 随时间反向传播算法BPTT

面临的问题：

- 梯度消失问题
- 梯度爆炸问题

解决方案：

- 选择其他的激活函数。例如ReLU。
- 引入改进网络结构的机制，例如LSTM，GRU。
- **现在在自然语言处理上应用十分广的的就是LSTM。**

End