人工智能

深度学习

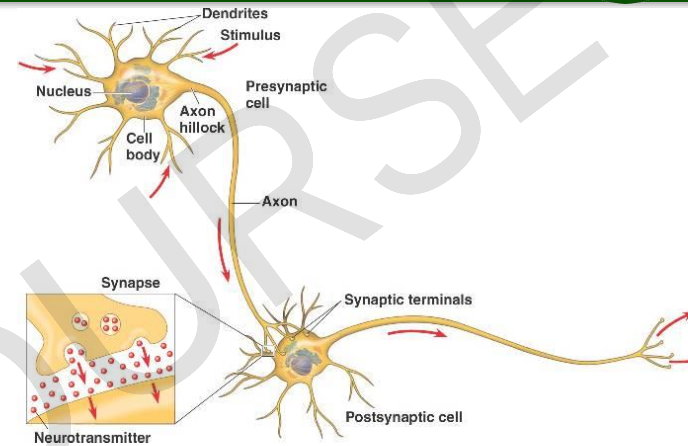
Deep Learning

中山大學
SUN YAT-SEN UNIVERSITY

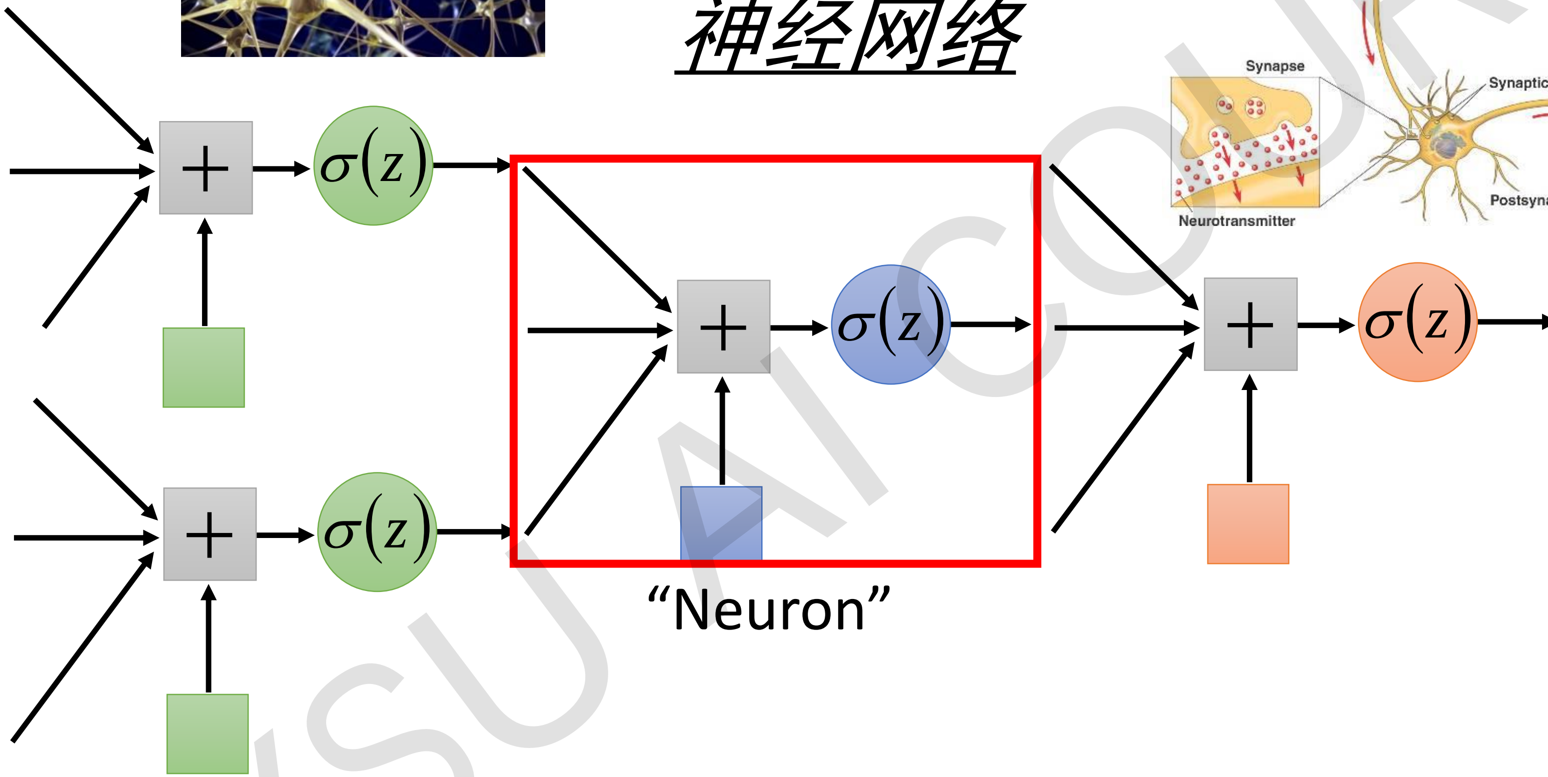
深度学习

- Recap: 神经网络
- 卷积神经网络 (Convolution Neural Network, CNN)
- 循环神经网络 (Recurrent Neural Network, RNN)
- 深度学习的可解释性

Recap: 神经网络

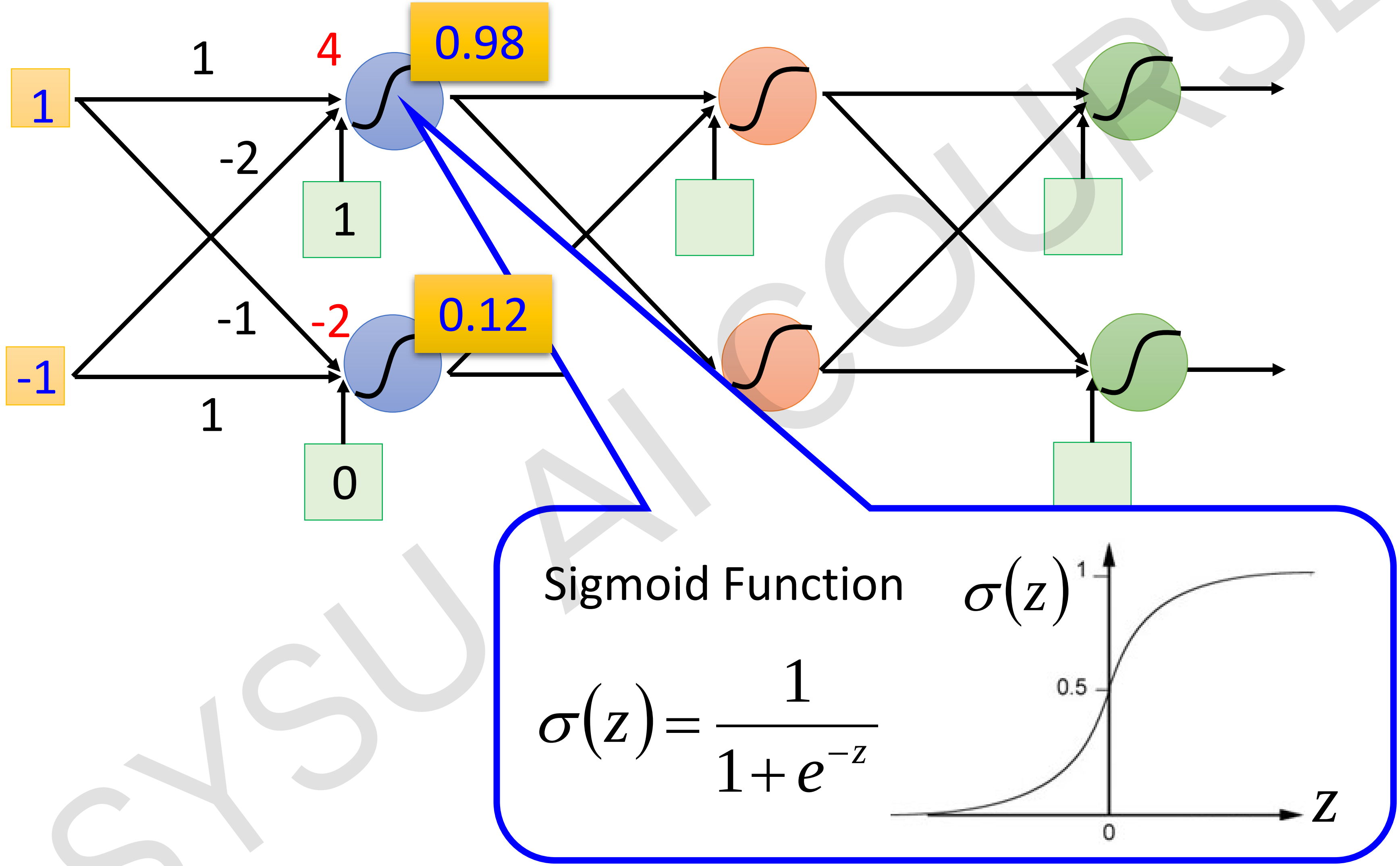


<u>神经网络</u>



网络参数 θ：“神经元”中的权重(weights)和偏置(biases)

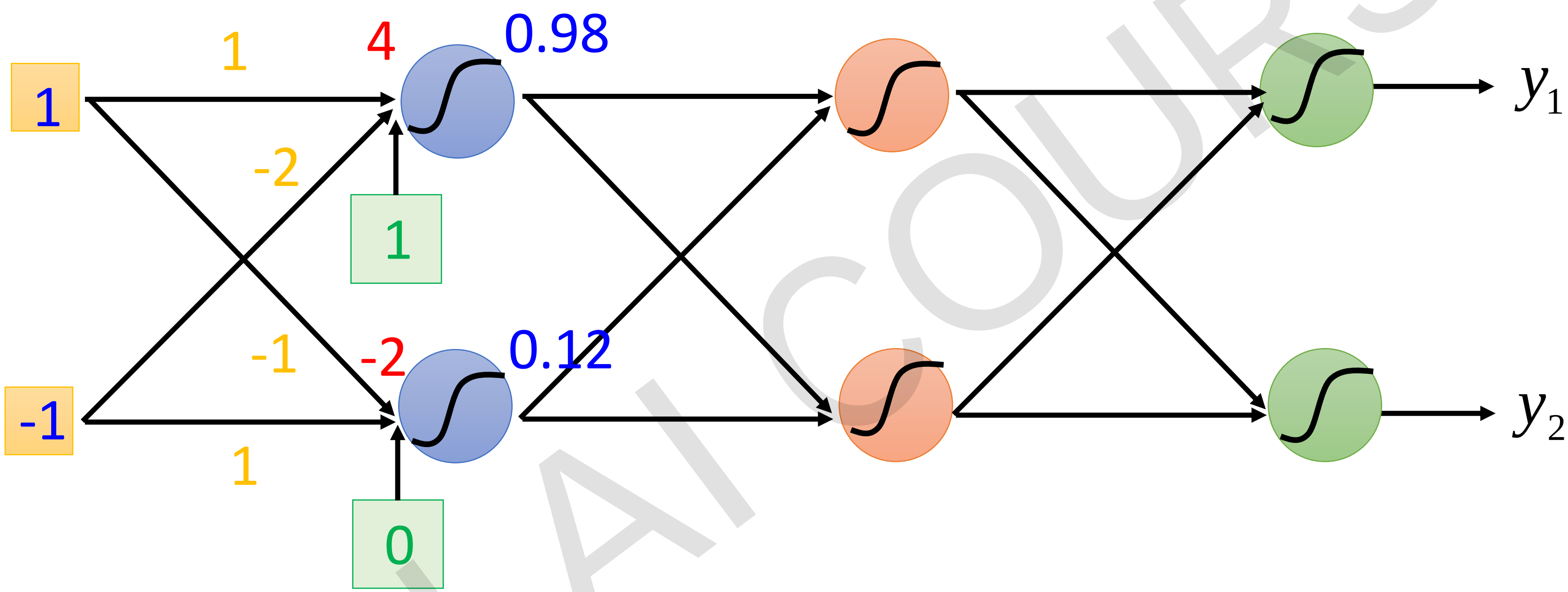
Recap: 神经网络



$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Recap: 神经网络

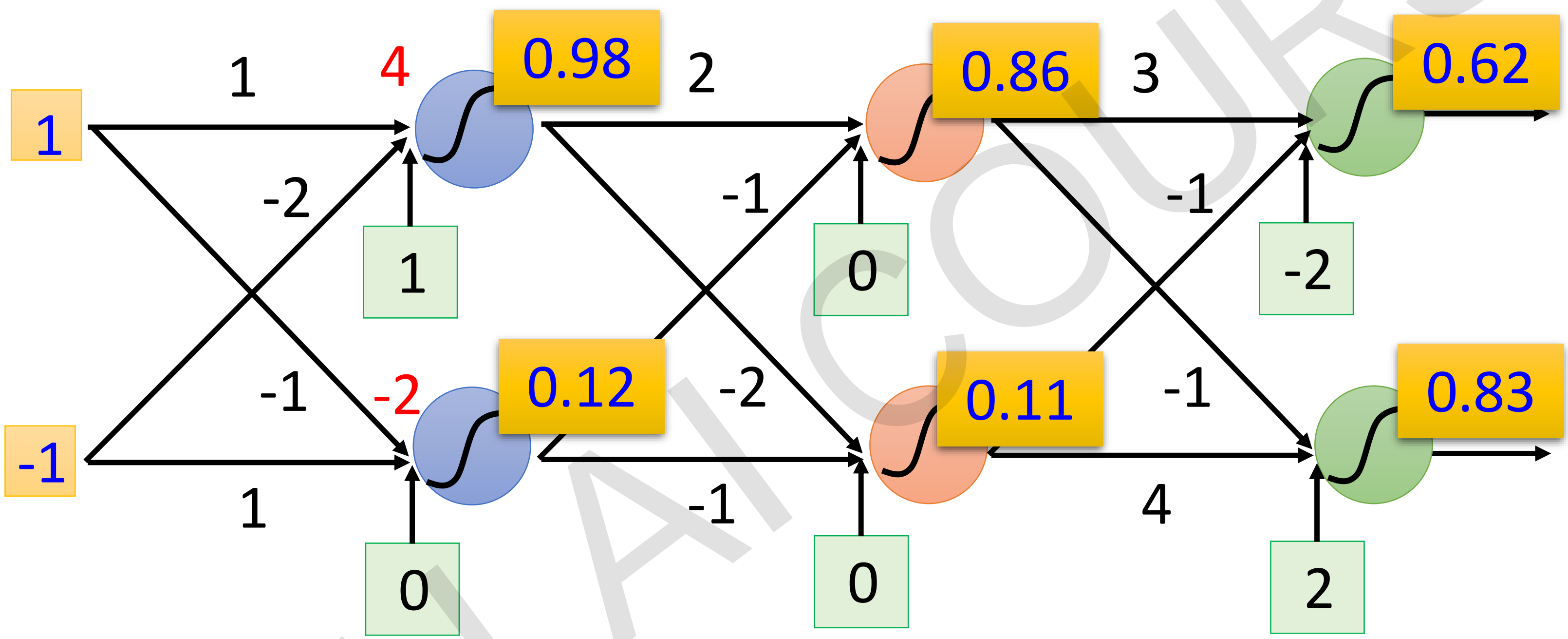
矩阵操作



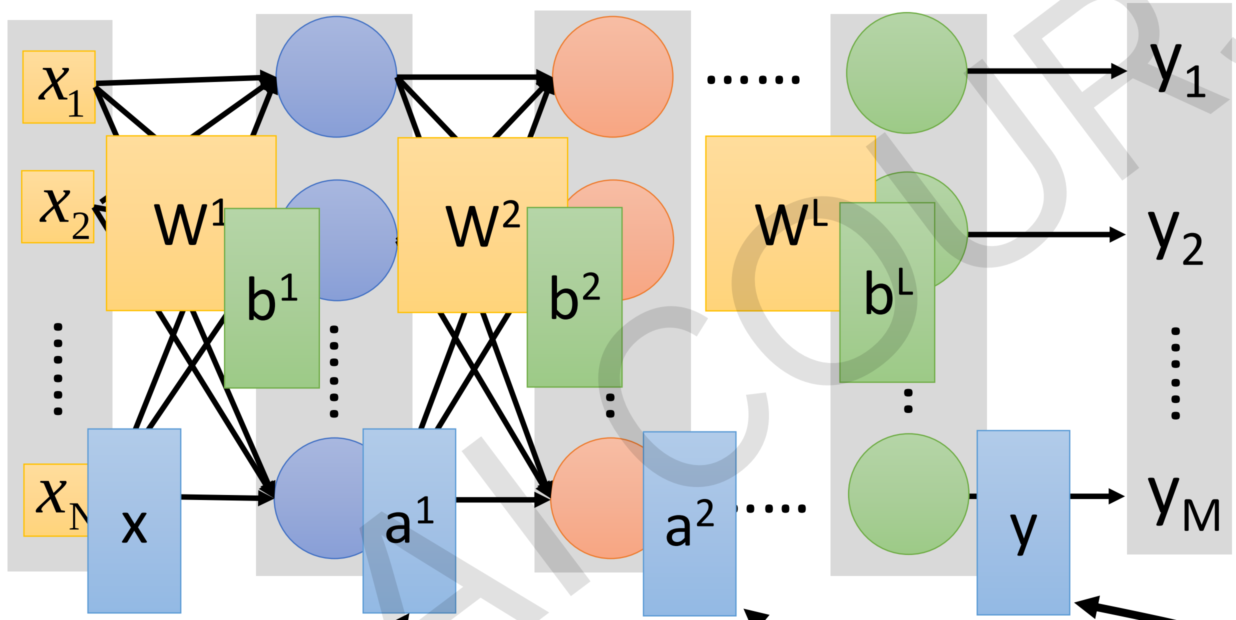
$$\sigma\left(\underbrace{\begin{bmatrix} 1 & -2 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\begin{bmatrix} 4 \\ -2 \end{bmatrix}}\right) = \begin{bmatrix} 0.98 \\ 0.12 \end{bmatrix}$$

Recap: 神经网络

Try it!



Recap: 神经网络

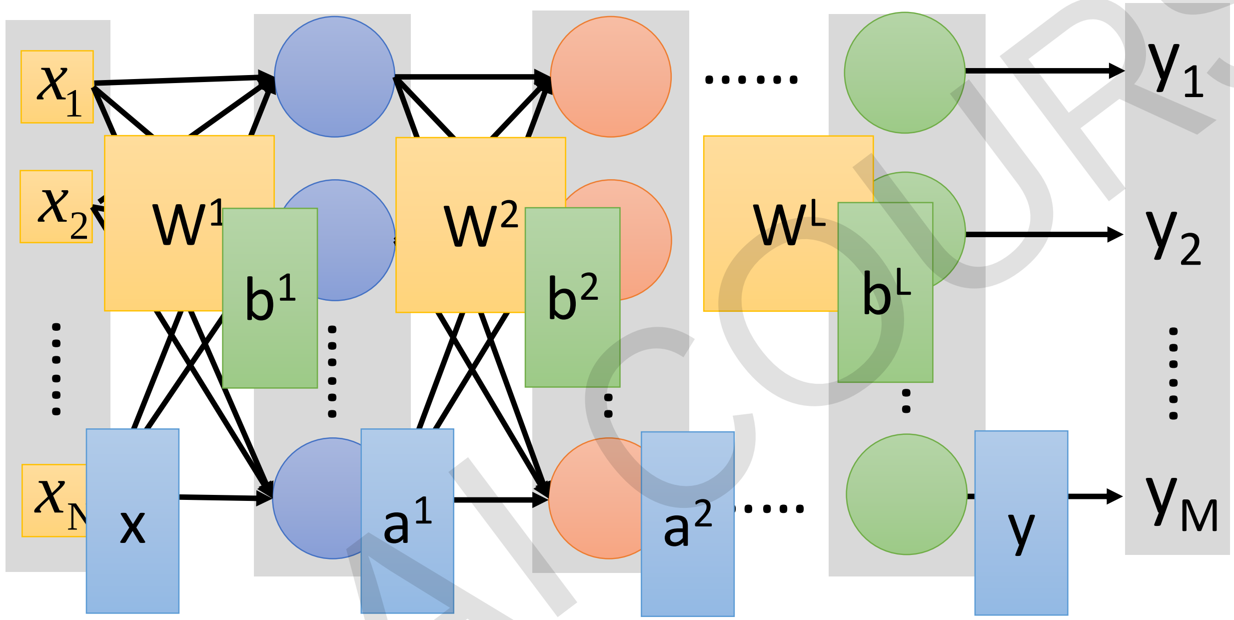


$\sigma($ W¹ x + b¹ $)$

$\sigma($ W² a¹ + b² $)$

$\sigma($ Wᴸ aᴸ⁻¹ + bᴸ $)$

Recap: 神经网络



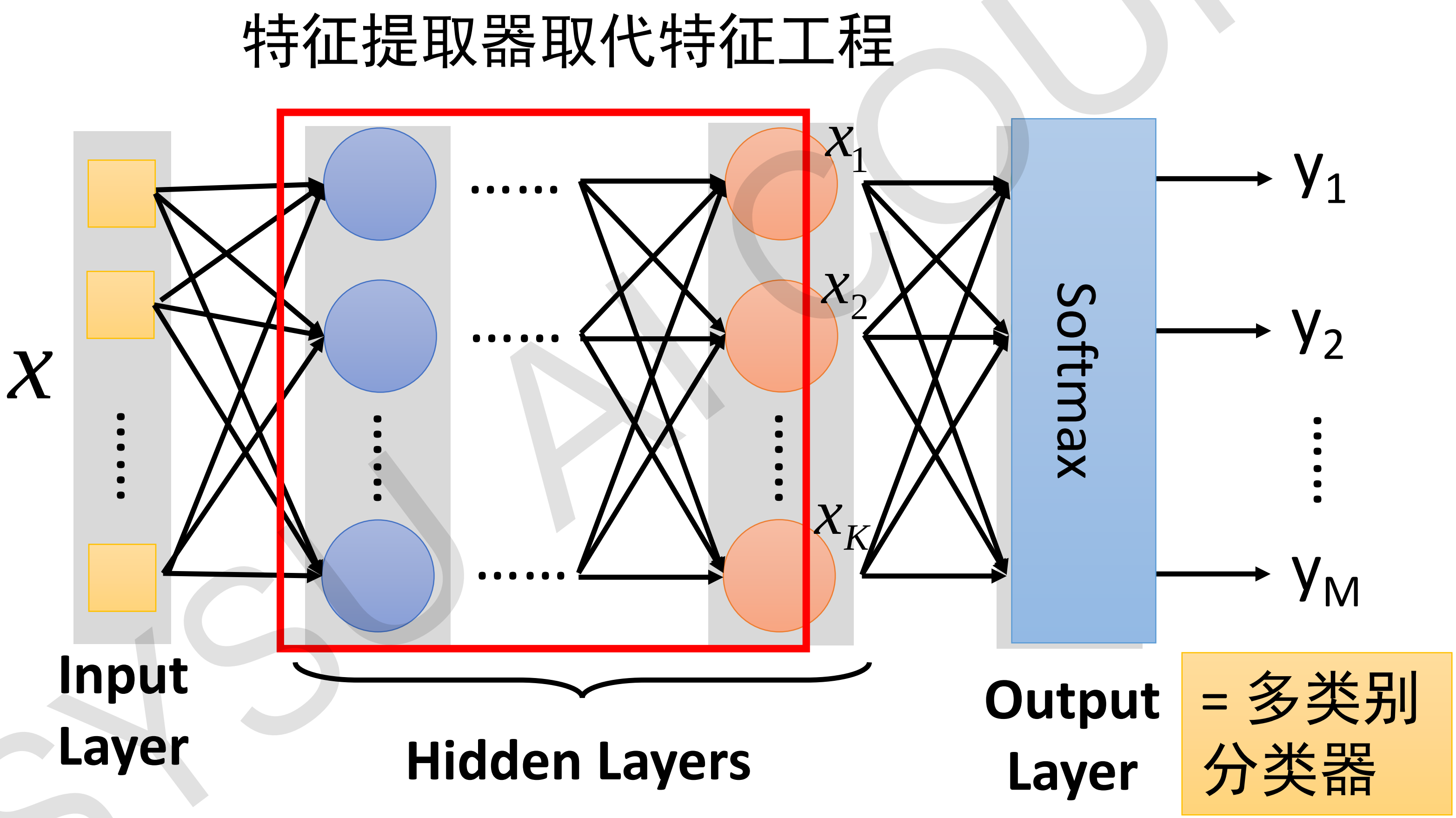
$$y = f(x)$$

使用**并行计算**技术来加速矩阵操作

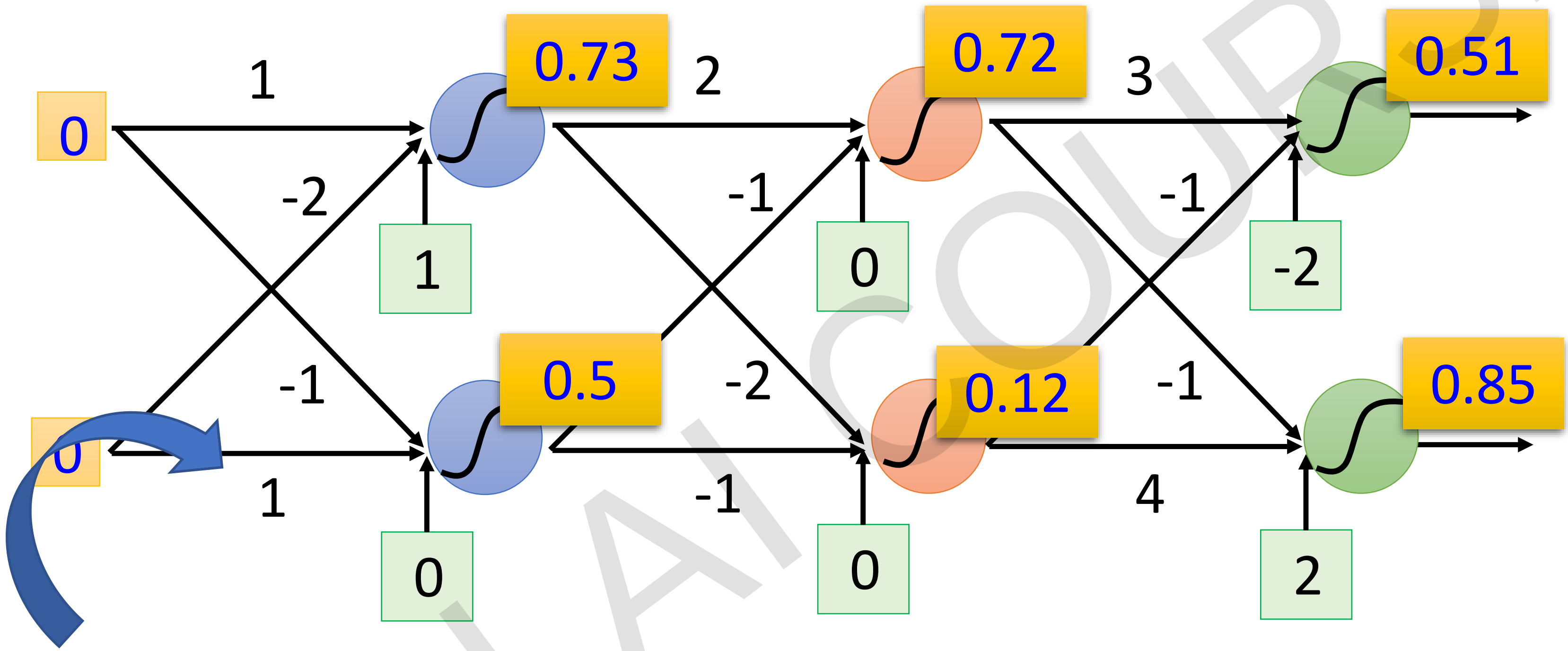
$$= \sigma(W^L \cdots \sigma(W^2 \; \sigma(W^1 \; x + b^1) + b^2) \cdots + b^L)$$

Recap: 神经网络

输出层作为多类别分类器



Recap: 神经网络



相当于函数功能

$$f\left(\begin{bmatrix}1\\-1\end{bmatrix}\right)=\begin{bmatrix}0.62\\0.83\end{bmatrix} \quad f\left(\begin{bmatrix}0\\0\end{bmatrix}\right)=\begin{bmatrix}0.51\\0.85\end{bmatrix}$$

输入一个向量，输出另一个向量

网络结构相当于定义了函数集空间

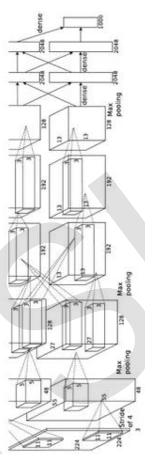
深度学习

Deep = Many hidden layers

http://cs231n.stanford.edu/slides/winter1516_lecture8.pdf

8 layers

16.4%



AlexNet (2012)

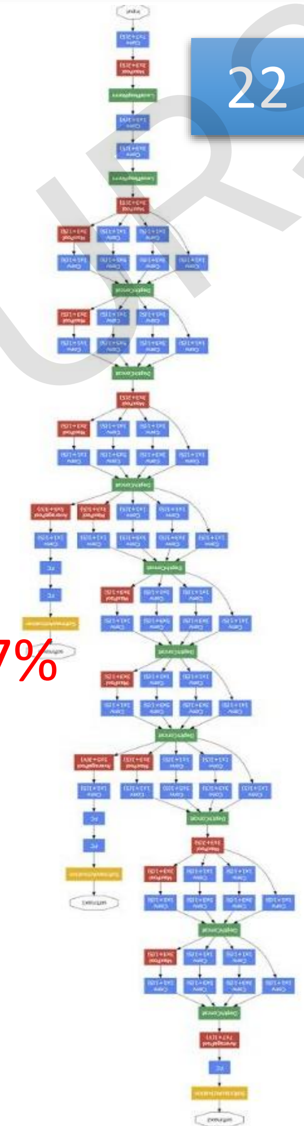
19 layers

7.3%

VGG (2014)

22 layers

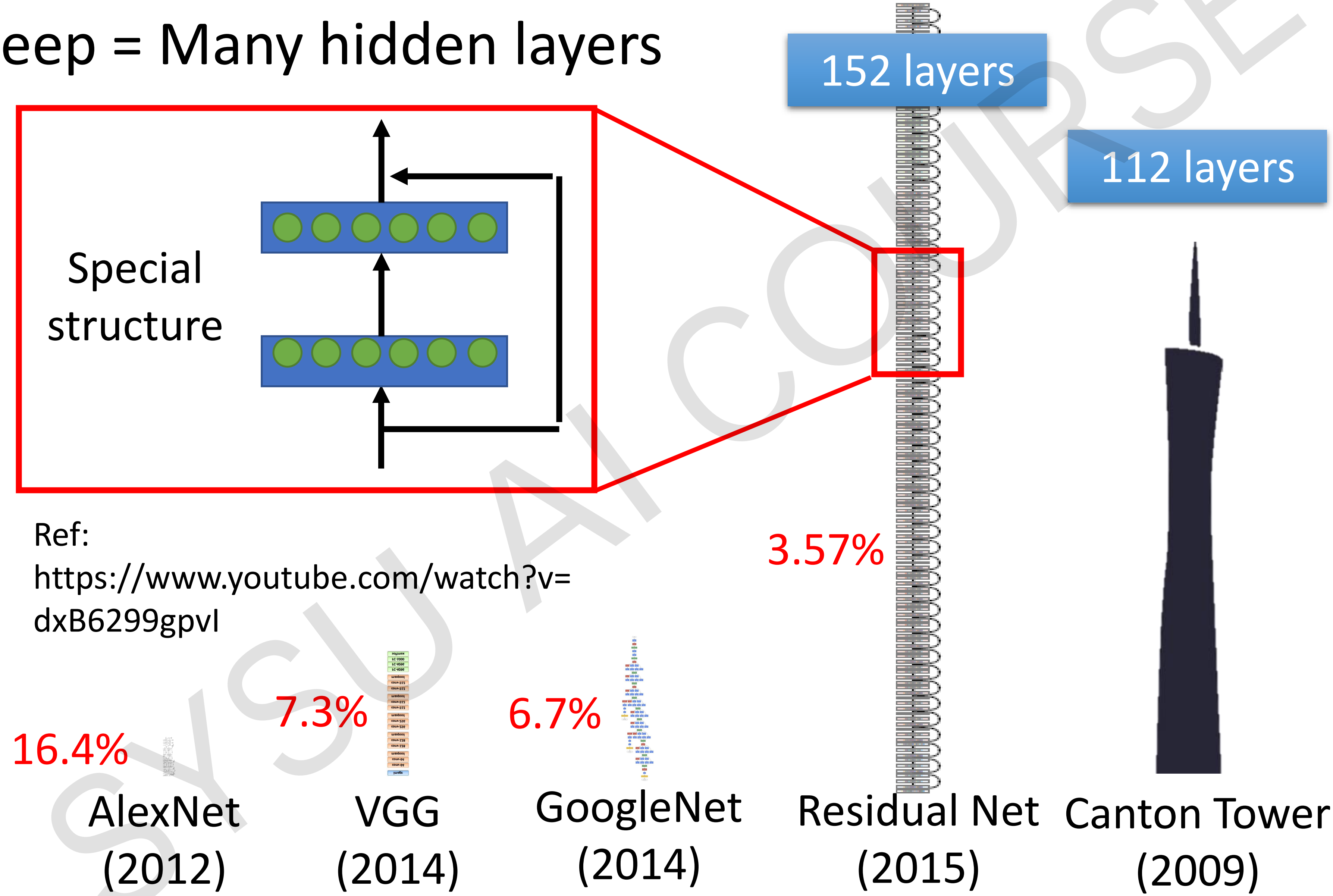
6.7%



GoogleNet (2014)

深度学习

Deep = Many hidden layers



Ref: https://www.youtube.com/watch?v=dxB6299gpvI

卷积神经网络 (CNN)

图像的特点与启发

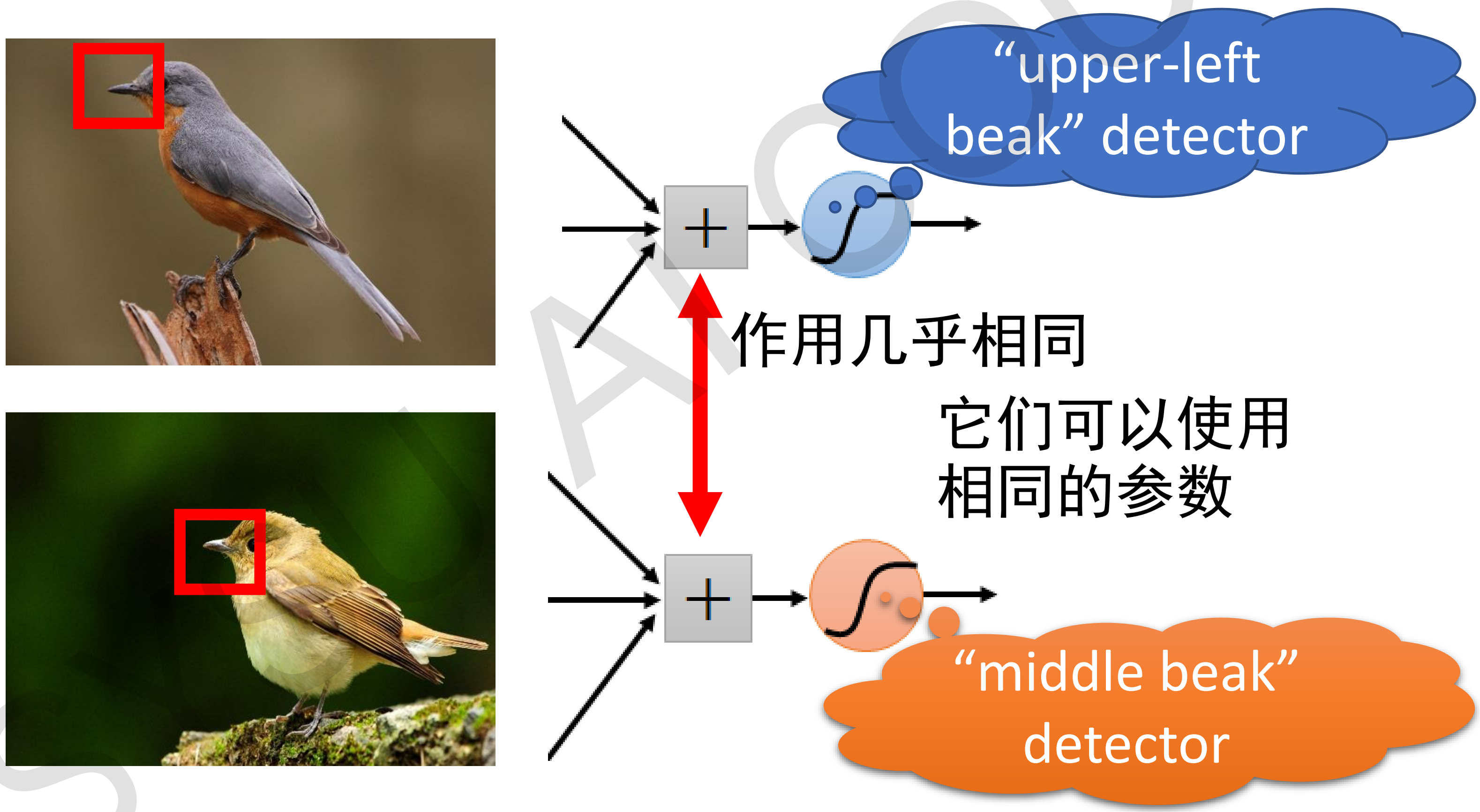
一些局部特征要比整张图片小得多

神经网络不必看完整张图片来识别

卷积神经网络 (CNN)

图像的特点与启发

相同特征会出现在不同区域



卷积神经网络 (CNN)

对像素进行二次采样不会改变识别对象信息

bird

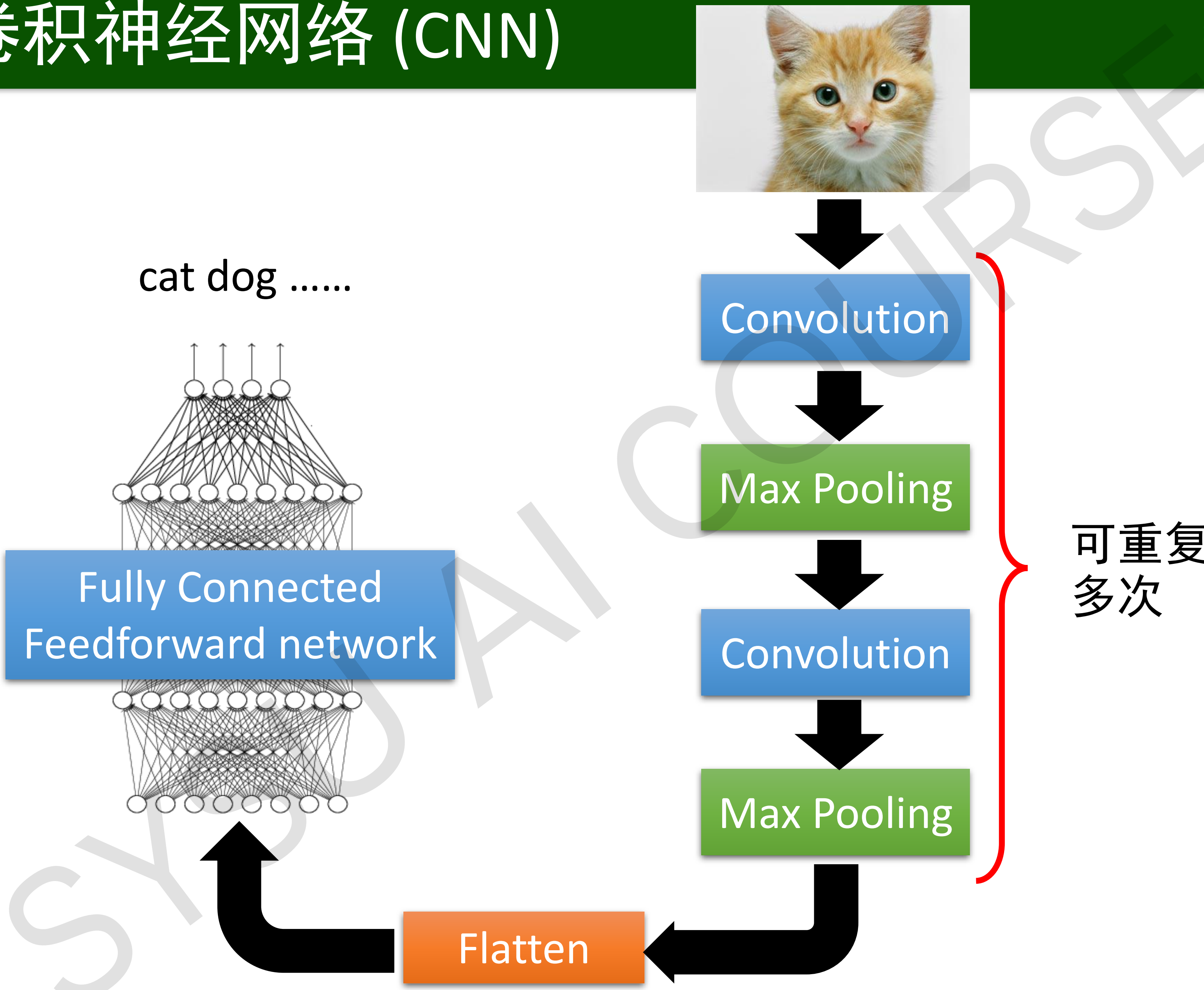
subsampling

bird

可以使用二次采样使图片更小

➡ 网络可以使用更少的参数来处理图片

卷积神经网络 (CNN)



cat dog ……
Fully Connected Feedforward network
Convolution
Max Pooling
Convolution
Max Pooling
可重复多次
Flatten

卷积神经网络 (CNN)

Property 1

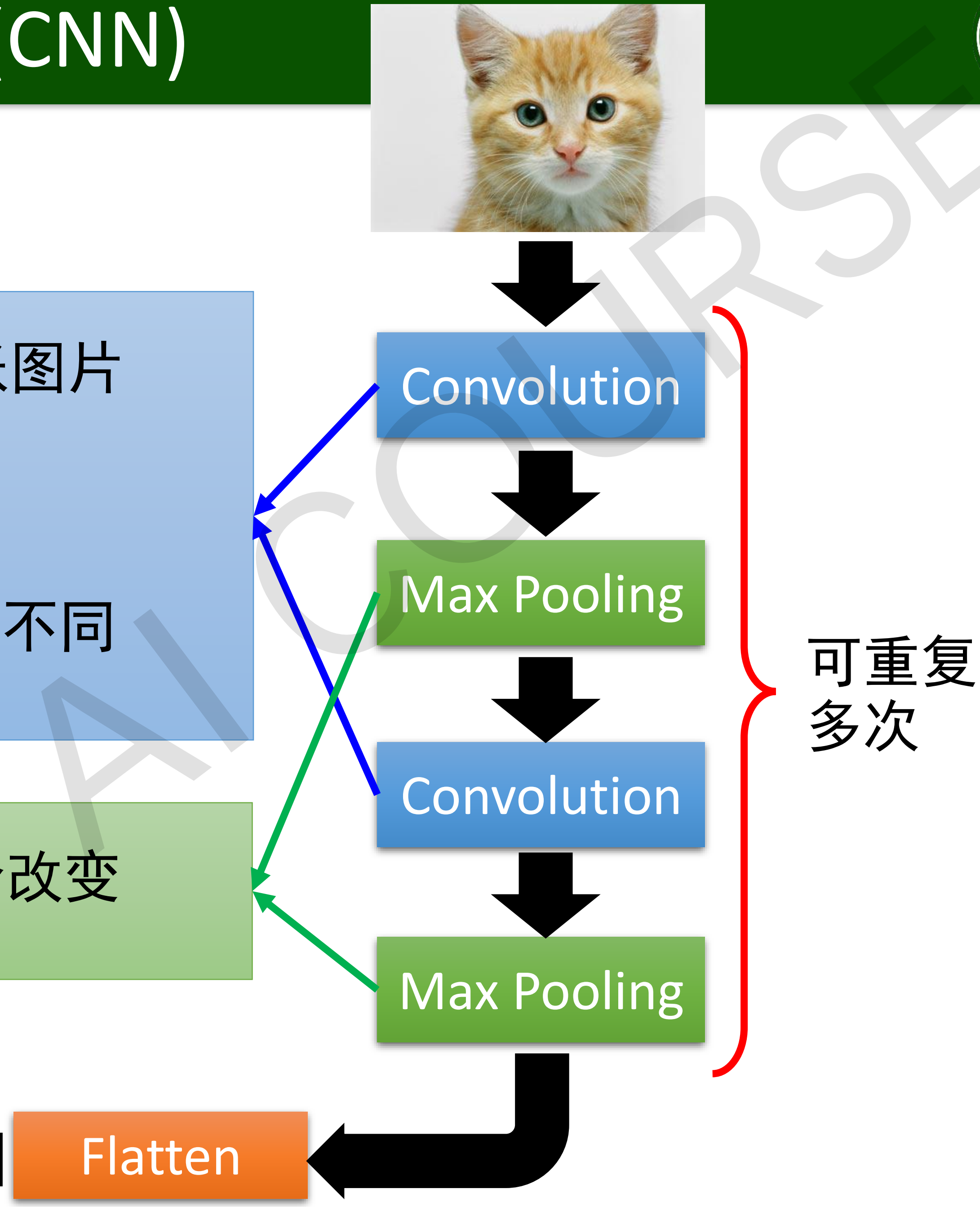
- 一些特征会比整张图片小很多

Property 2

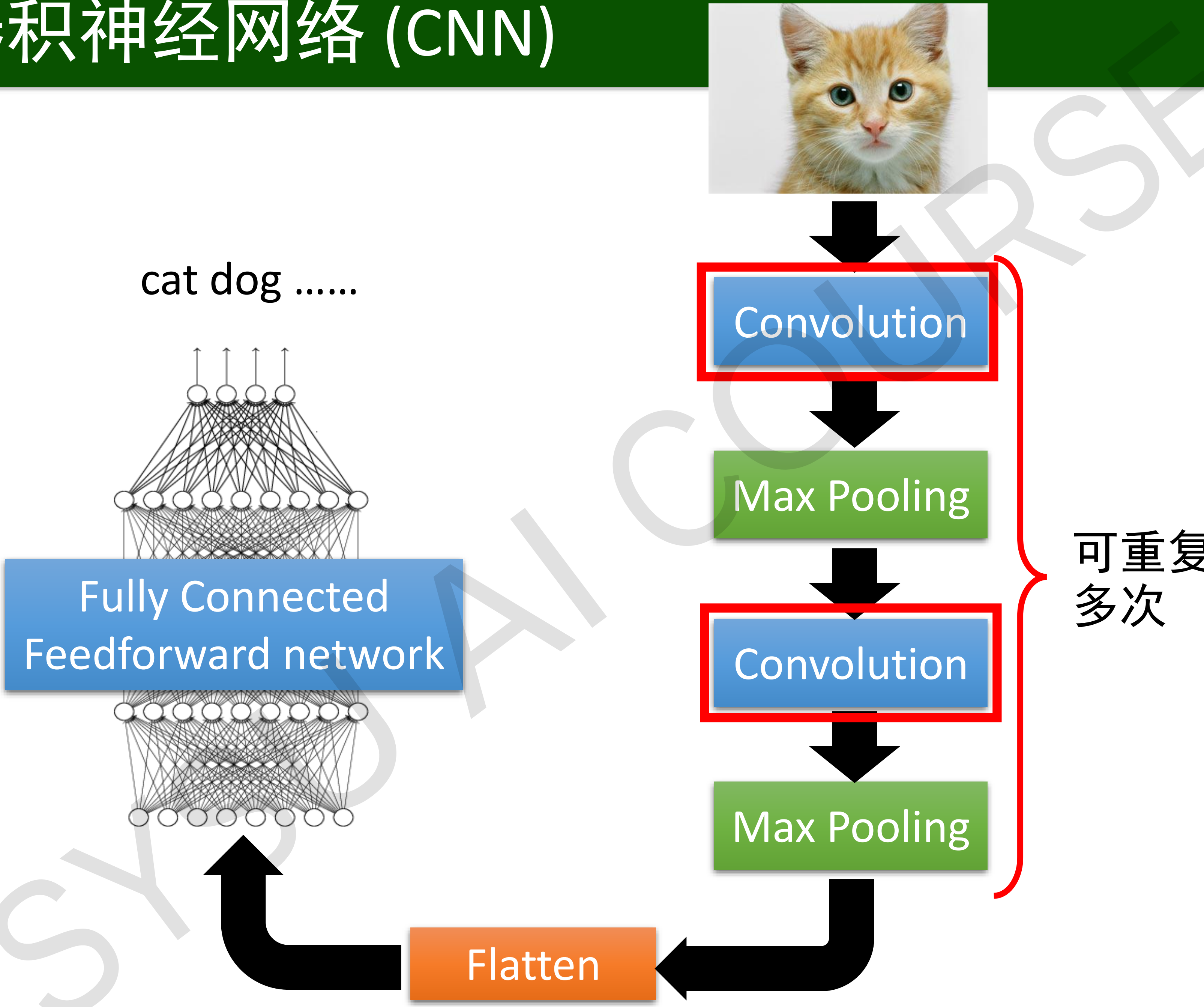
- 相同特征会出现在不同区域

Property 3

- 像素二次采样不会改变识别的目标信息



卷积神经网络 (CNN)



cat dog ......
Fully Connected
Feedforward network
Convolution
Max Pooling
Convolution
Max Pooling
可重复
多次
Flatten

卷积神经网络 (CNN)

Convolution(卷积)

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

需要学习的网络参数

1	-1	-1
-1	1	-1
-1	-1	1

卷积核 1

Matrix

-1	1	-1
-1	1	-1
-1	1	-1

卷积核 2

Matrix

⋮

Property 1

每个卷积核可以识别区域特征(3 x 3).

卷积神经网络 (CNN)

Convolution(卷积)

stride(步长)=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

卷积核 1

3 -1

Convolution(卷积)

If stride(步长)=2

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

卷积核 1

3 -3

在下文继续使用 stride=1

卷积神经网络 (CNN)

Convolution(卷积)

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

卷积核 1

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

Property 2 相同特征出现在不同区域

卷积神经网络 (CNN)

Try it!

stride=1

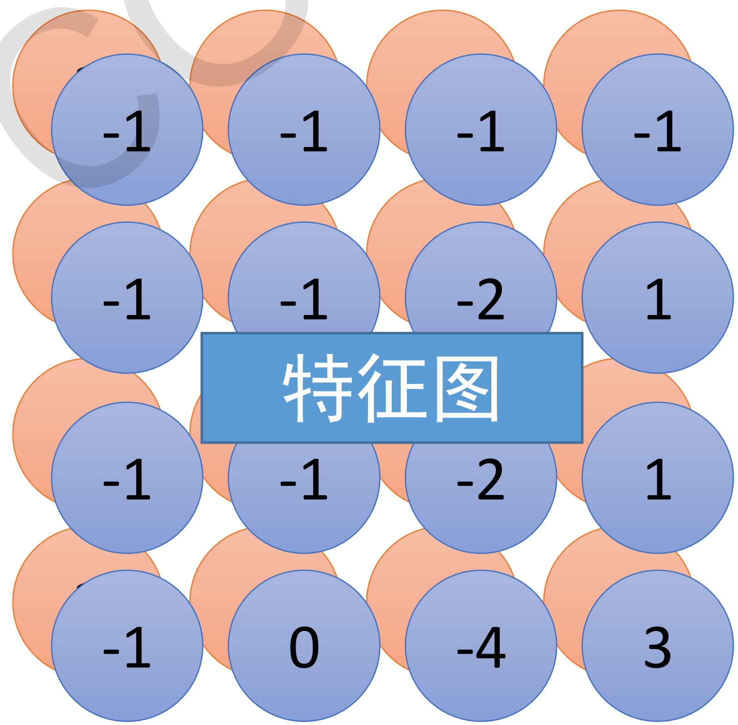
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

-1	1	-1
-1	1	-1
-1	1	-1

卷积核 2

对每个卷积核做相同的运算



4 x 4 image

卷积神经网络 (CNN)

Convolution(卷积)

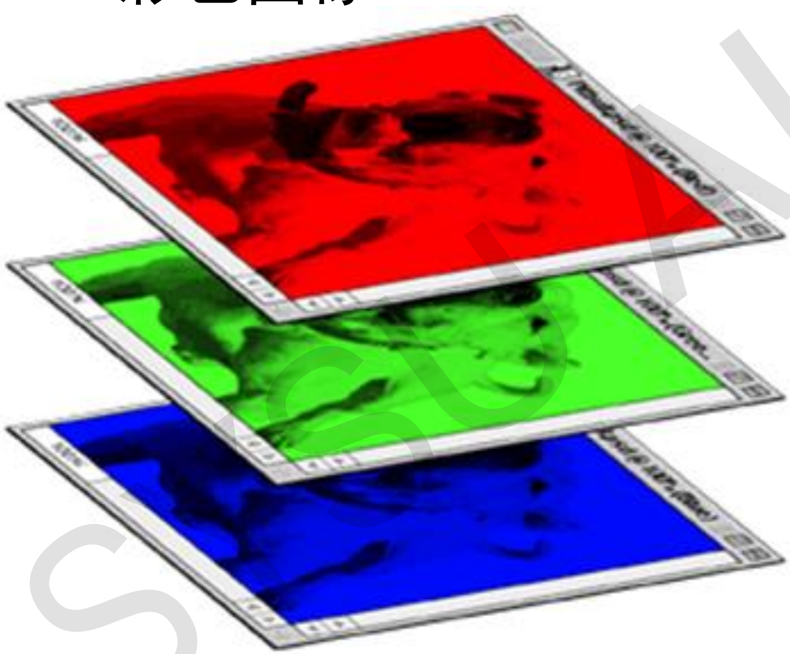
卷积核 1

1	-1	-1
-1	1	-1
-1	-1	1

卷积核 2

-1	1	-1
-1	1	-1
-1	1	-1

RGB 彩色图像



1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

卷积神经网络 (CNN)

卷积

v.s.

全连接

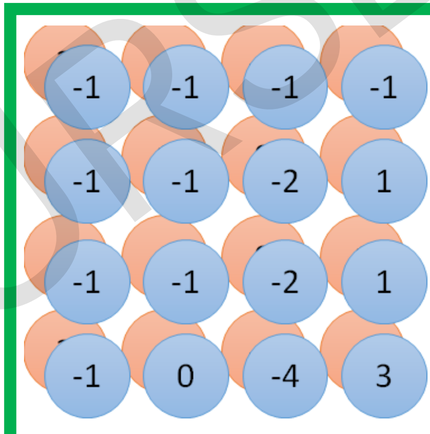
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

image

1	-1	-1
-1	1	-1
-1	-1	1

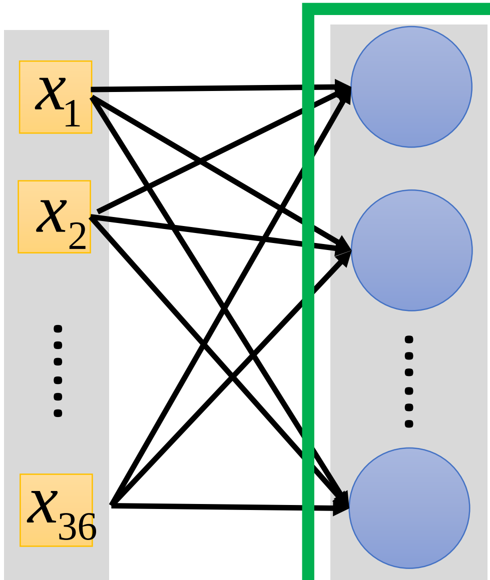
-1	1	-1
-1	1	-1
-1	1	-1

卷积

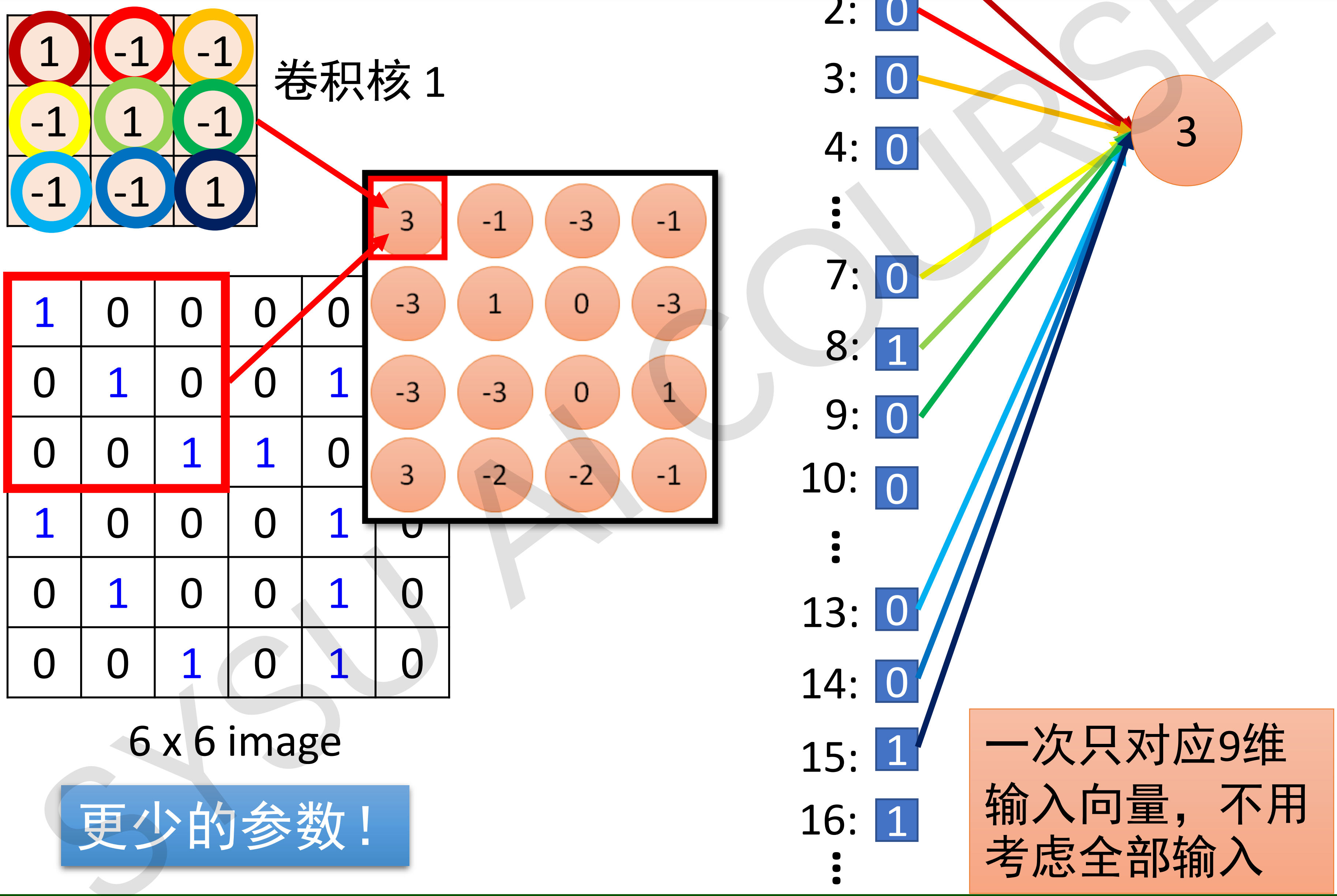


全连接

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

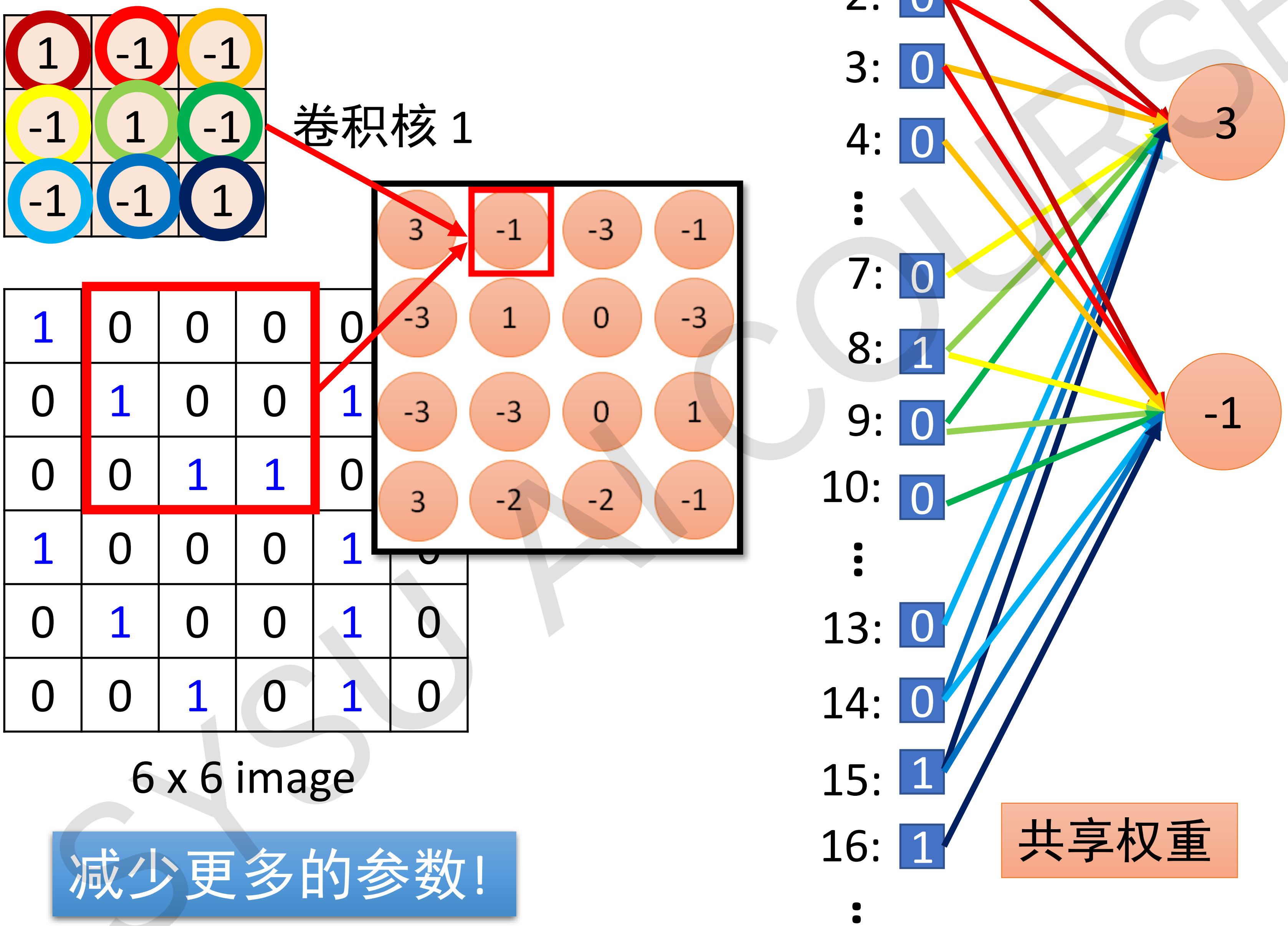


卷积神经网络 (CNN)



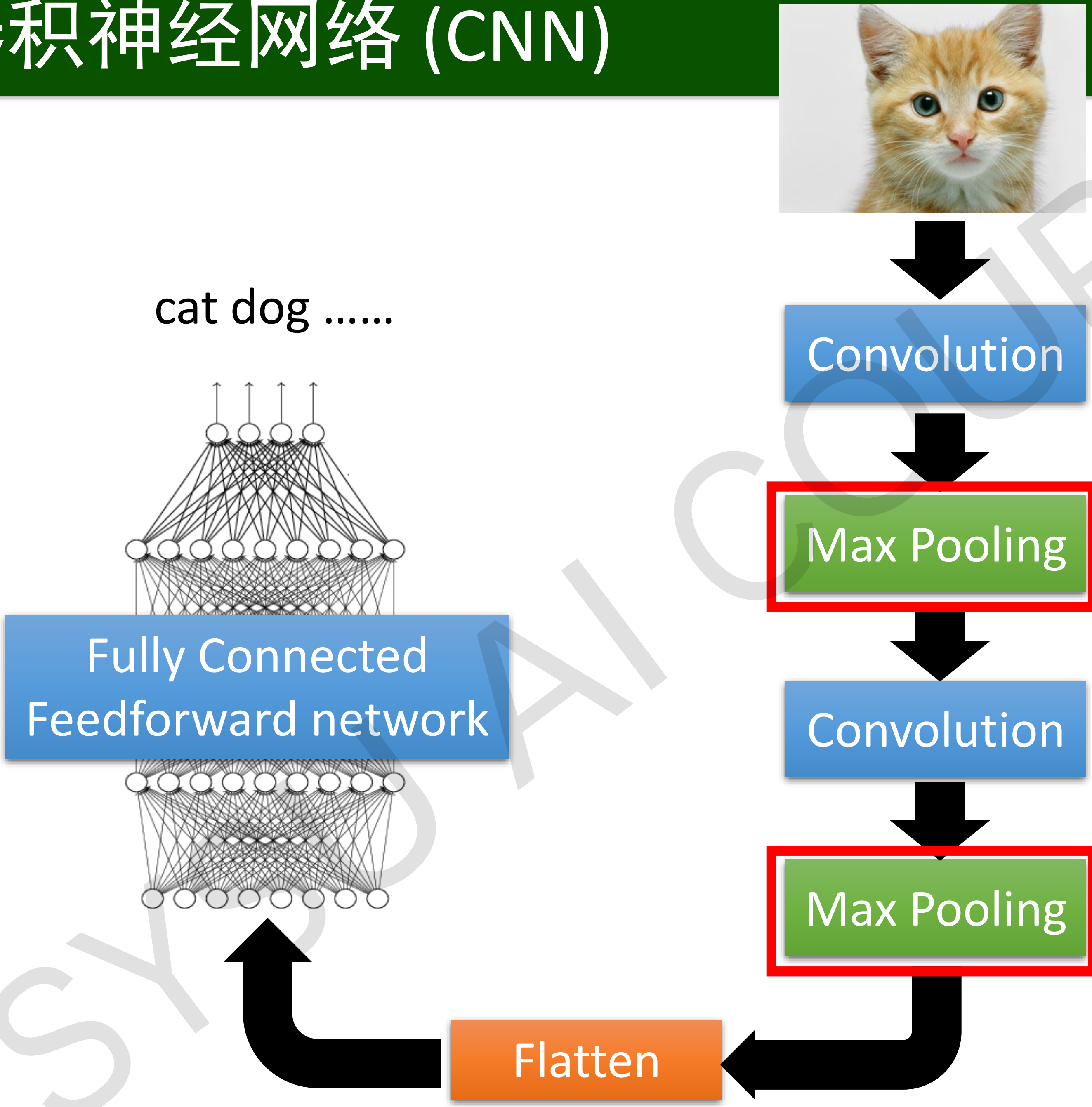
卷积核 1
1 -1 -1
-1 1 -1
-1 -1 1
1 0 0 0 0
0 1 0 0 1
0 0 1 1 0
1 0 0 0 1 0
0 1 0 0 1 0
0 0 1 0 1 0
3 -1 -3 -1
-3 1 0 -3
-3 -3 0 1
3 -2 -2 -1
6 x 6 image
更少的参数！
1: 1
2: 0
3: 0
4: 0
7: 0
8: 1
9: 0
10: 0
13: 0
14: 0
15: 1
16: 1
3
一次只对应9维输入向量，不用考虑全部输入

卷积神经网络 (CNN)



1 -1 -1
-1 1 -1
-1 -1 1
卷积核 1
1 0 0 0 0 1
0 1 0 0 1 0
0 0 1 1 0 0
1 0 0 0 1 0
0 1 0 0 1 0
0 0 1 0 1 0
6 x 6 image
3 -1 -3 -1
-3 1 0 -3
-3 -3 0 1
3 -2 -2 -1
1: 1
2: 0
3: 0
4: 0
7: 0
8: 1
9: 0
10: 0
13: 0
14: 0
15: 1
16: 1
3
-1
共享权重
减少更多的参数!

卷积神经网络 (CNN)



cat dog ……
Convolution
Max Pooling
Convolution
Max Pooling
Flatten
Fully Connected
Feedforward network

卷积神经网络 (CNN)

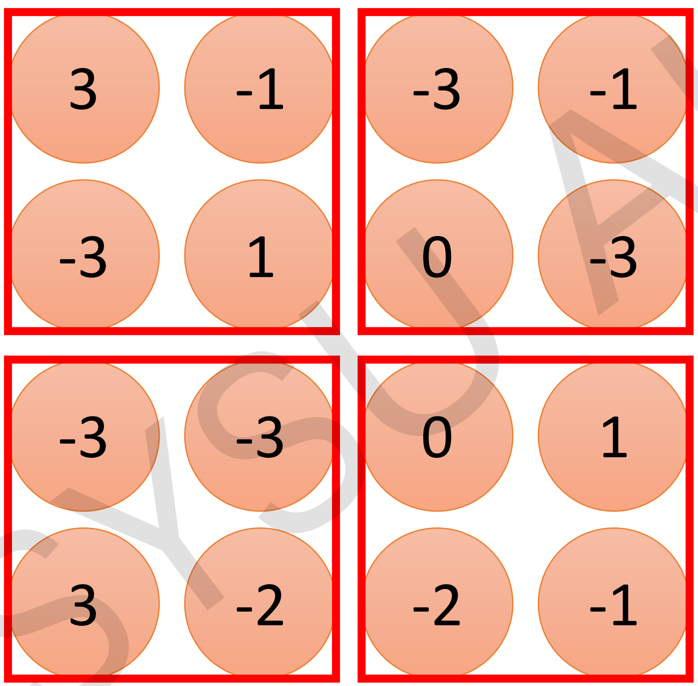
Max Pooling (最大池化)

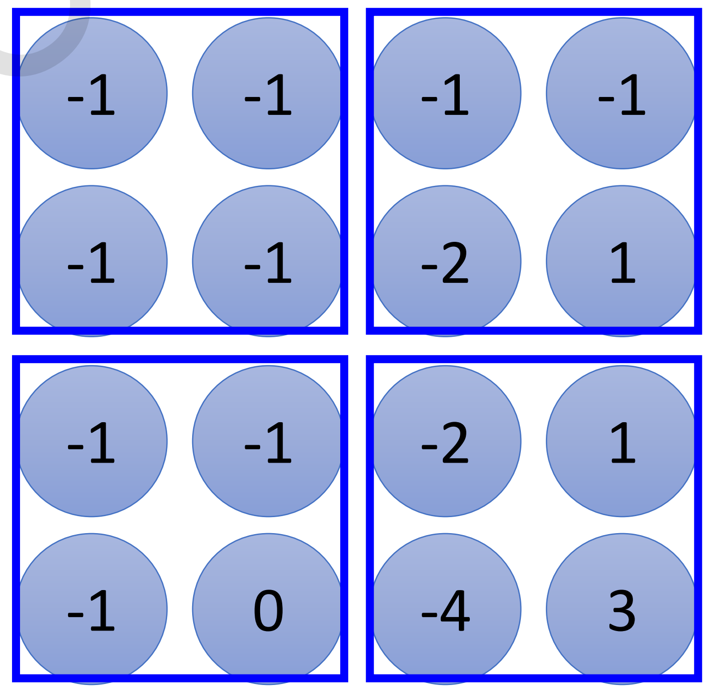
1	-1	-1
-1	1	-1
-1	-1	1

卷积核 1

-1	1	-1
-1	1	-1
-1	1	-1

卷积核 2



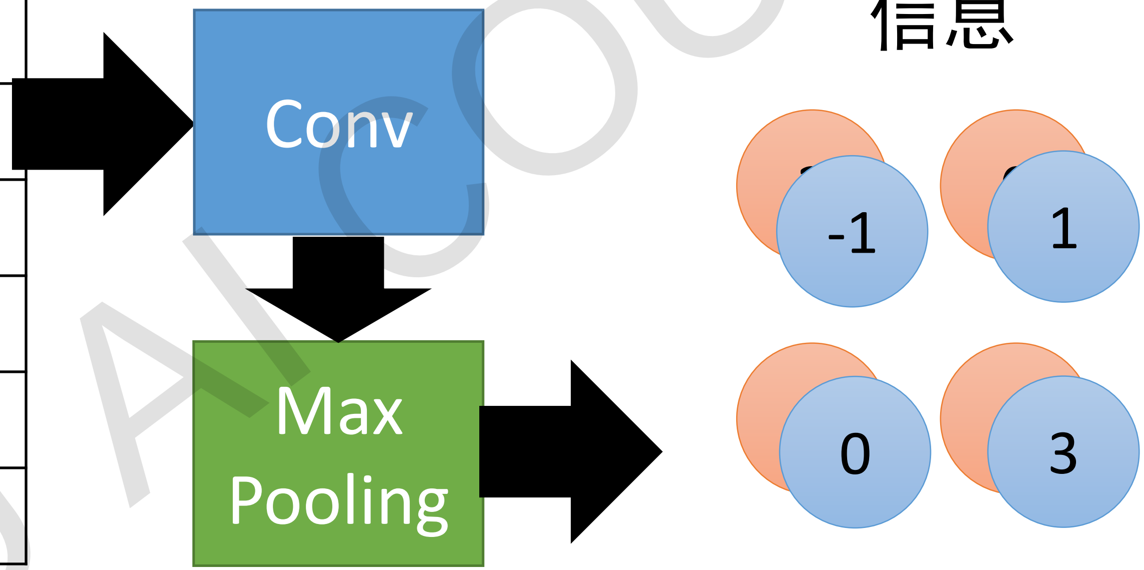


卷积神经网络 (CNN)

Max Pooling (最大池化)

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

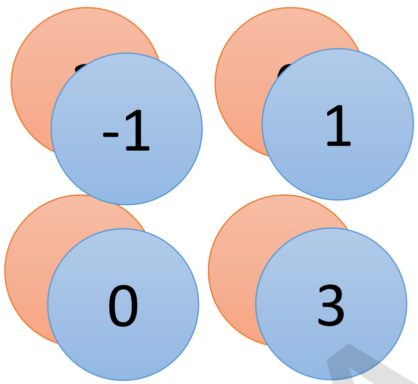


更少的特征信息

2 x 2 image

每个卷积核都会生成一个通道(channel)

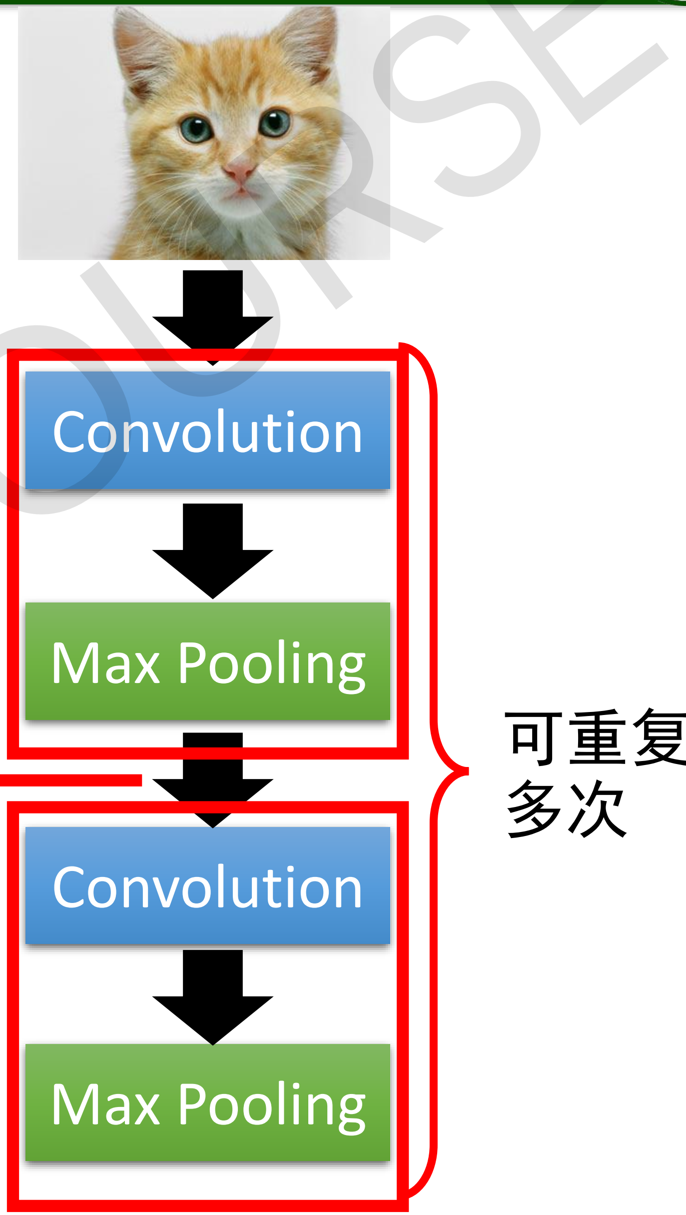
卷积神经网络 (CNN)



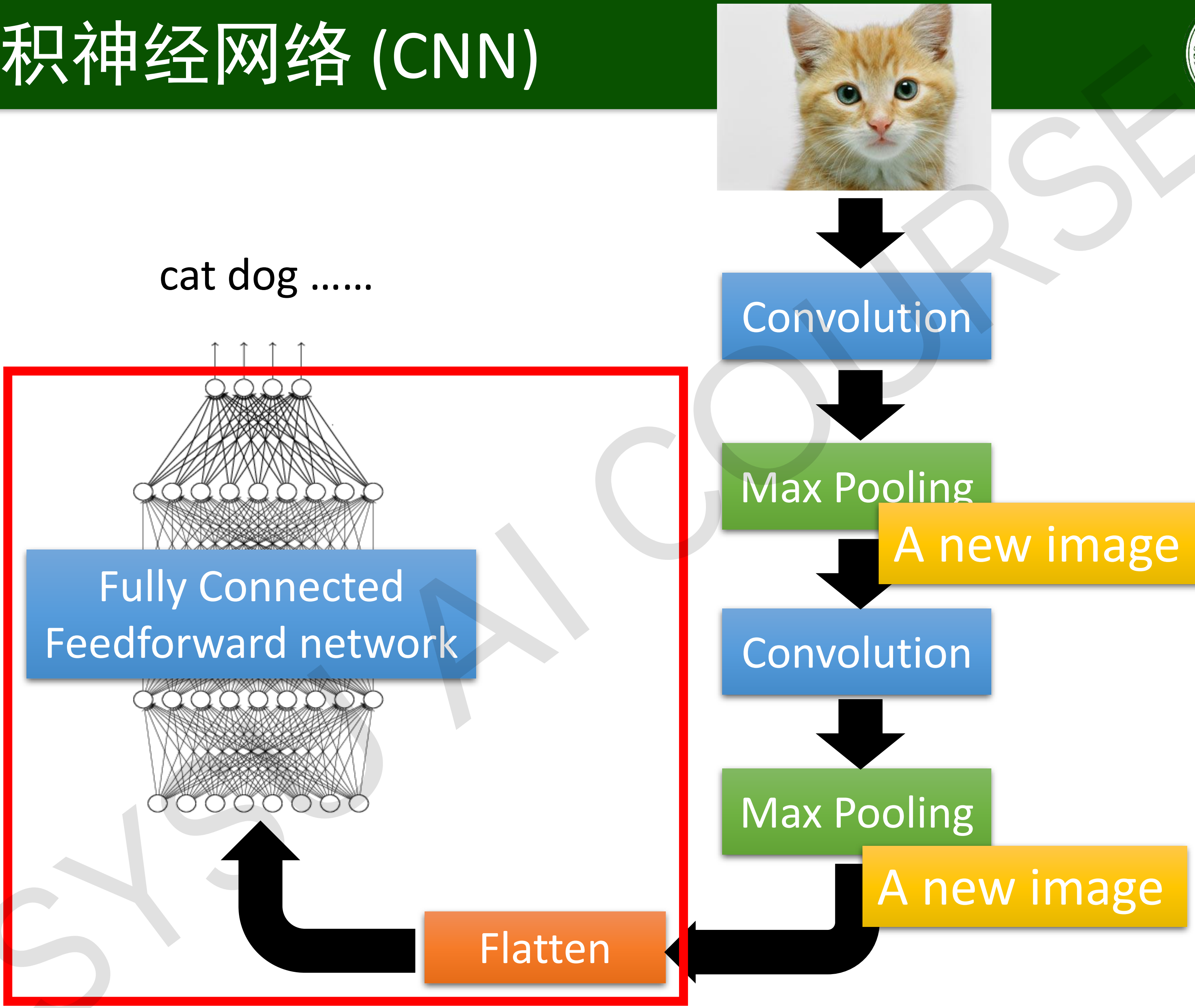
A new image

比原始图像更小

通道数和卷积核数量相同

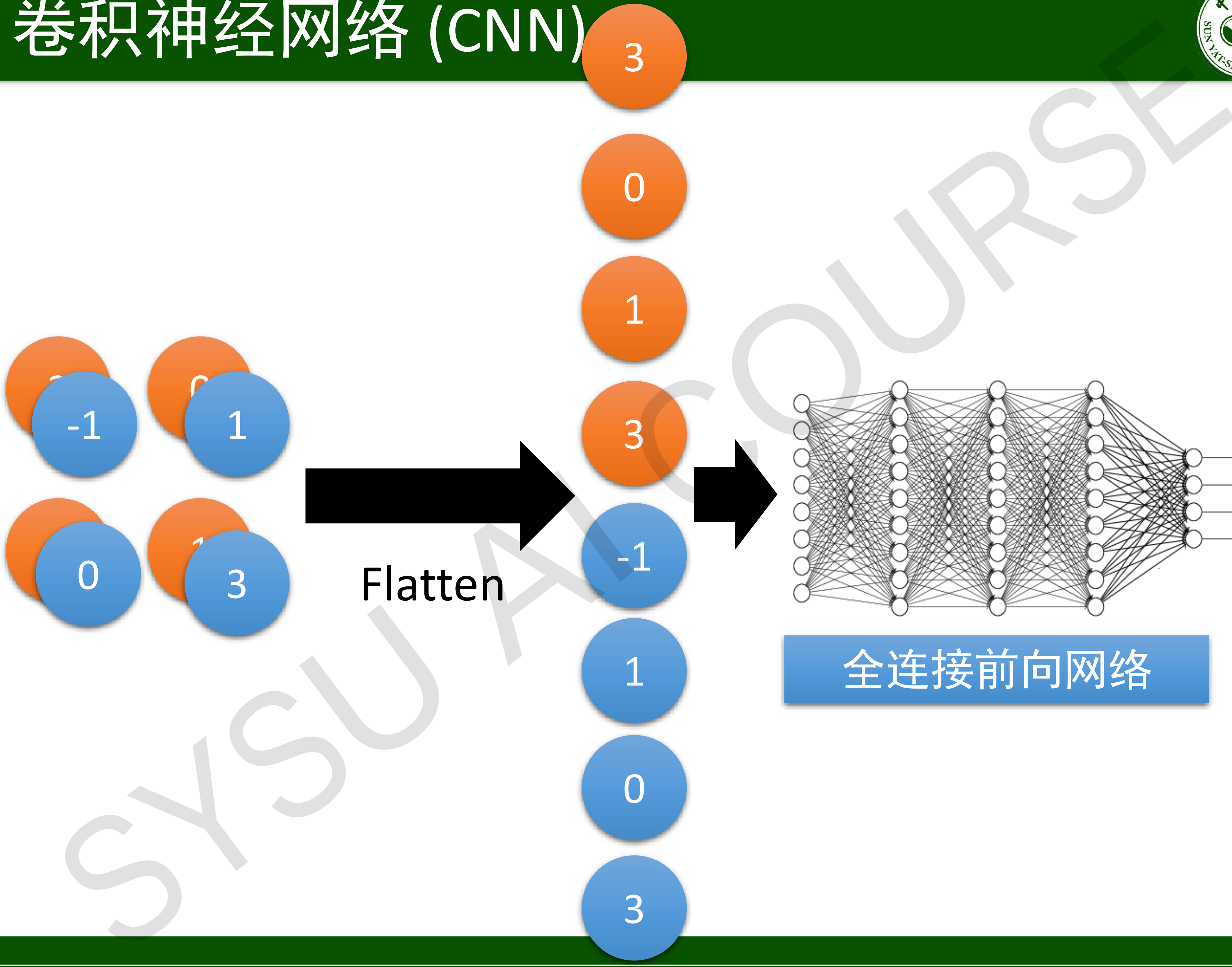


卷积神经网络 (CNN)



cat dog ......
Convolution
Max Pooling
A new image
Convolution
Max Pooling
A new image
Fully Connected
Feedforward network
Flatten

卷积神经网络 (CNN)



卷积神经网络 (CNN) 可视化

https://poloclub.github.io/cnn-explainer/

循环神经网络 (RNN)

为什么使用 Recurrent neural network？
考虑下面的 Slot Filling(槽填充)问题：

在订票系统中设置几个槽位（Slot），希望算法能够将关键词 ‘Guangzhou’ 放入目的地（Destination）槽位，将 May 和 31th 放入到达时间（Time of Arrival）槽位。

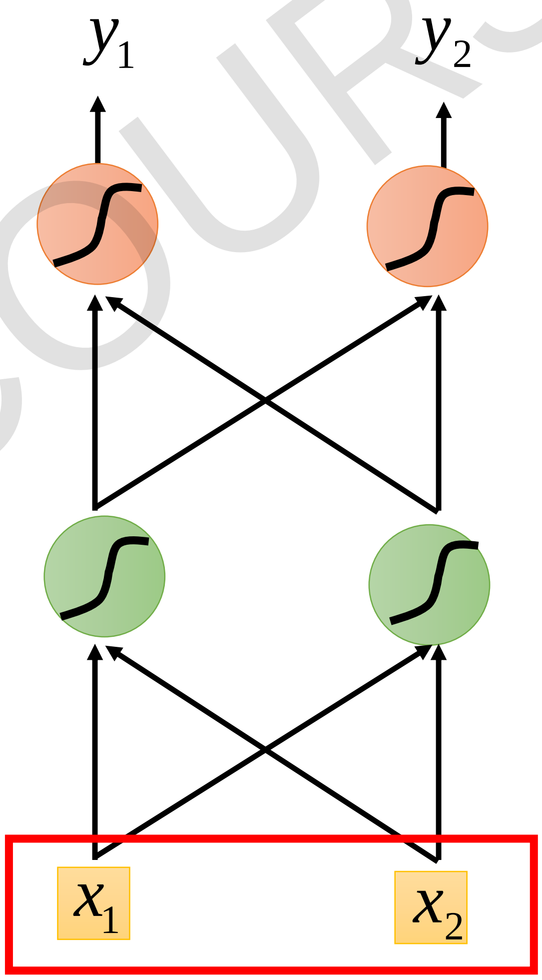
循环神经网络 (RNN)

使用全连接神经网络解决?

输入: 一个单词

(每个单词用一个向量表示)

Guangzhou



循环神经网络 (RNN)

如何用向量表示单词?

One-hot Encoding 词典 = {apple, bag, cat, dog, elephant}

向量和词典大小相同

每个维度对应一个单词

对应维度的单词为1，其他位置为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]

循环神经网络 (RNN)

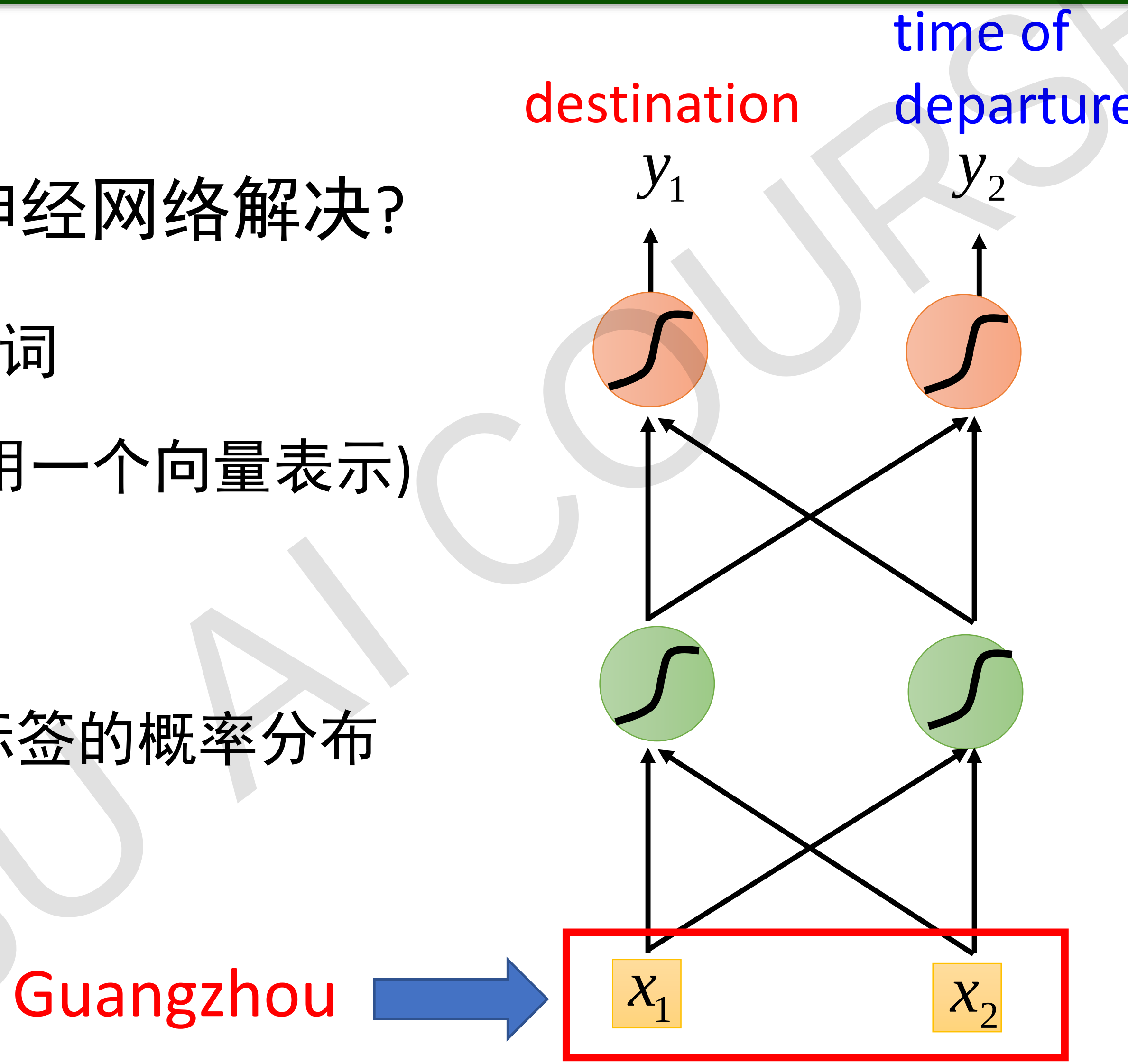
使用全连接神经网络解决?

输入: 一个单词

(每个单词用一个向量表示)

输出:

分类标签的概率分布



循环神经网络 (RNN)

arrive Guangzhou on May 31th

other dest other time time

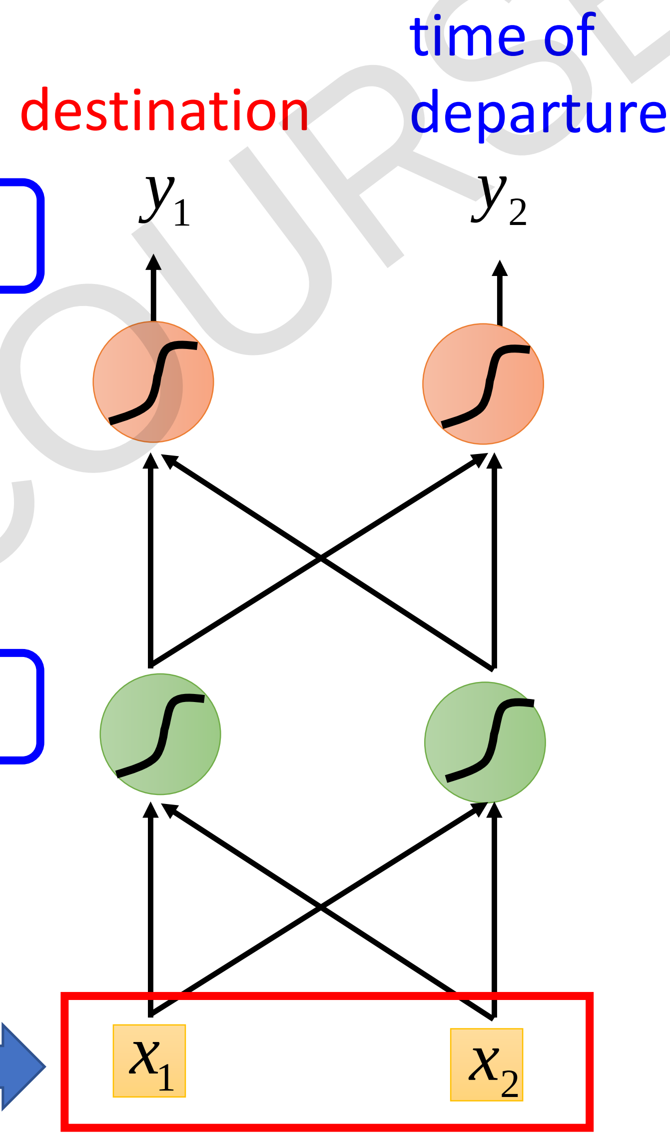
存在什么问题?

leave Guangzhou on May 31th

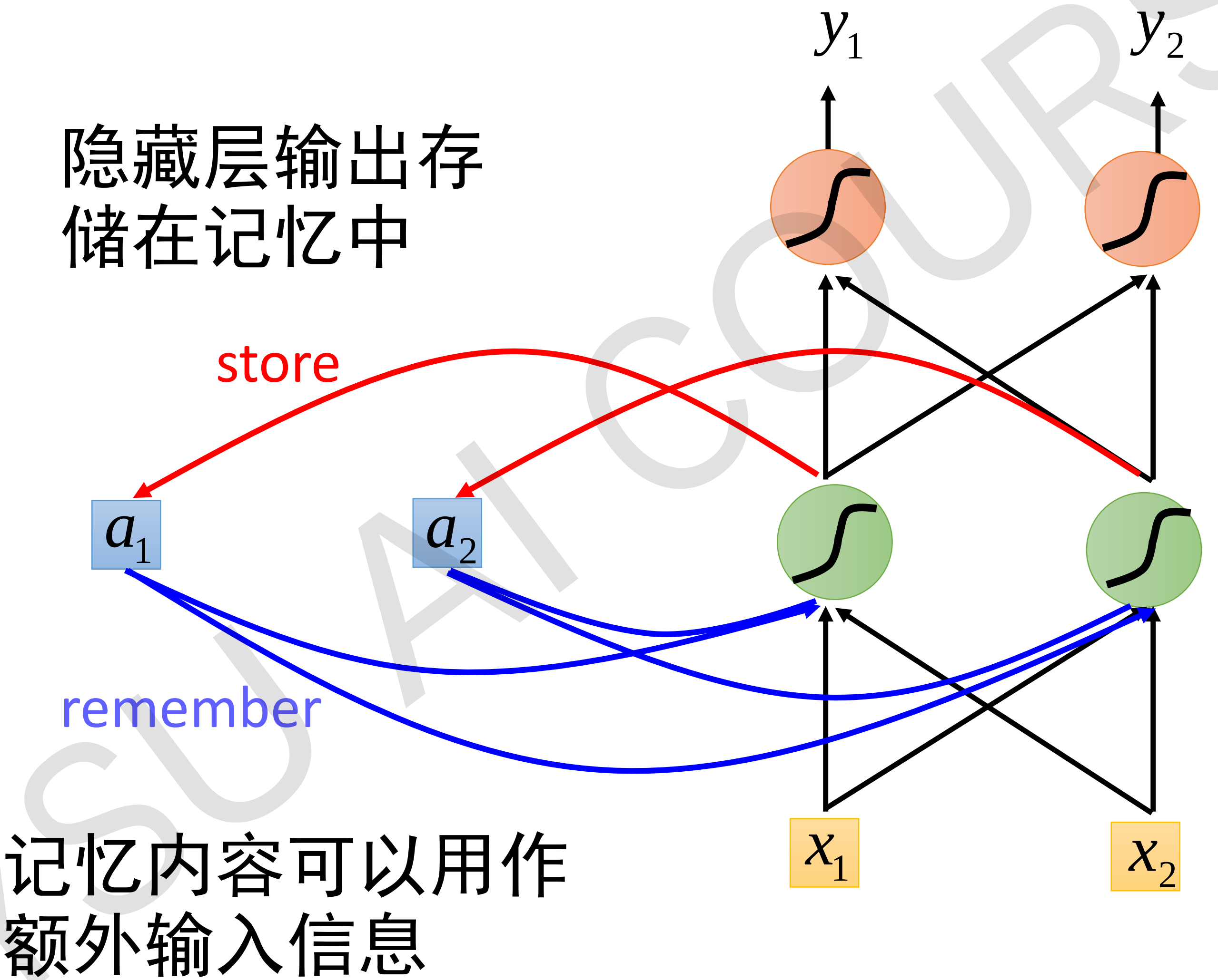
place of departure

神经网络需要记忆功能!

Guangzhou



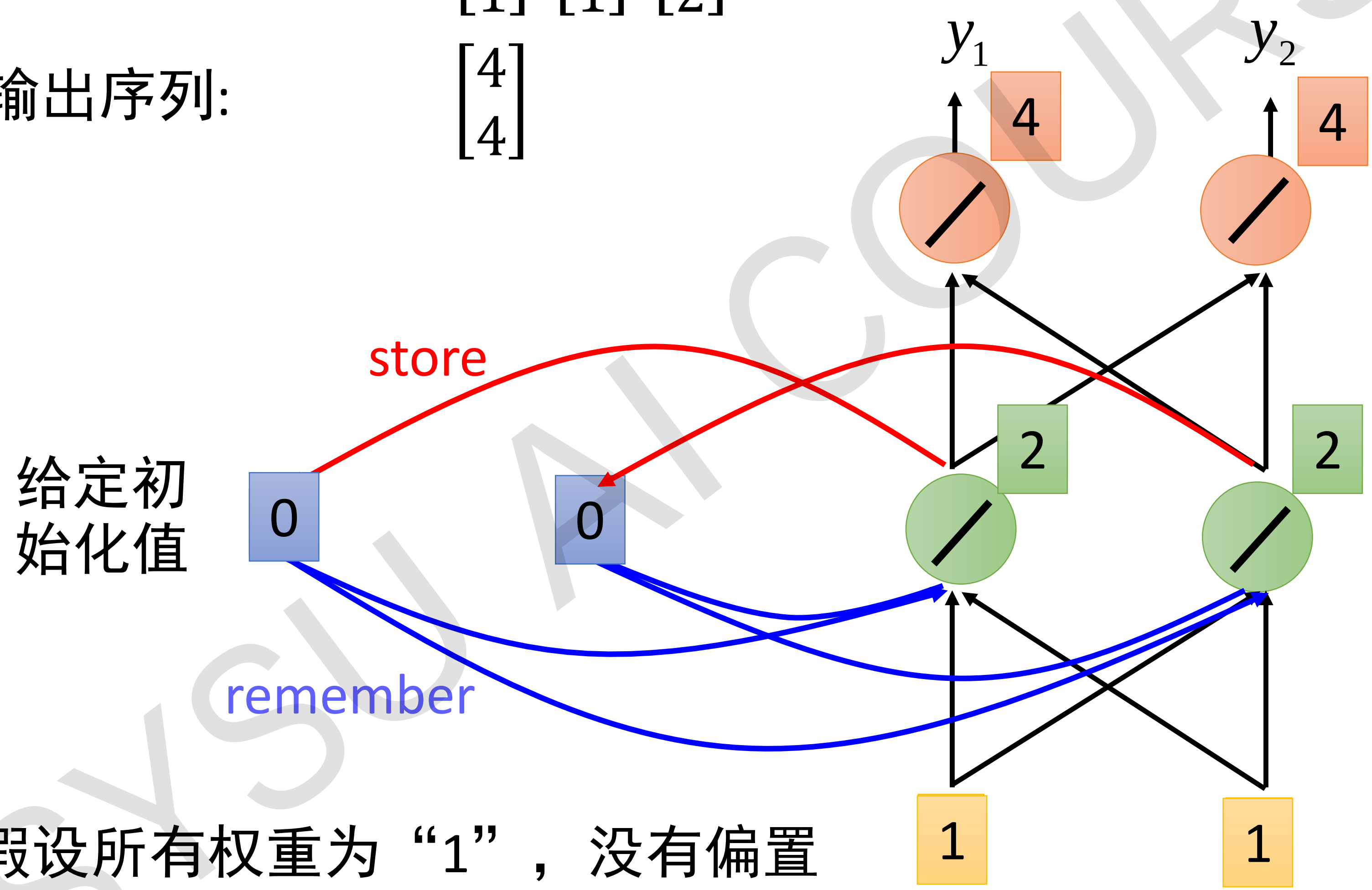
循环神经网络 (RNN)


隐藏层输出存
储在记忆中
store
y_1
y_2
a_1
a_2
remember
x_1
x_2
记忆内容可以用作
额外输入信息

循环神经网络 (RNN)

输入序列: $\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}2\\2\end{bmatrix}$ …… ……

输出序列: $\begin{bmatrix}4\\4\end{bmatrix}$

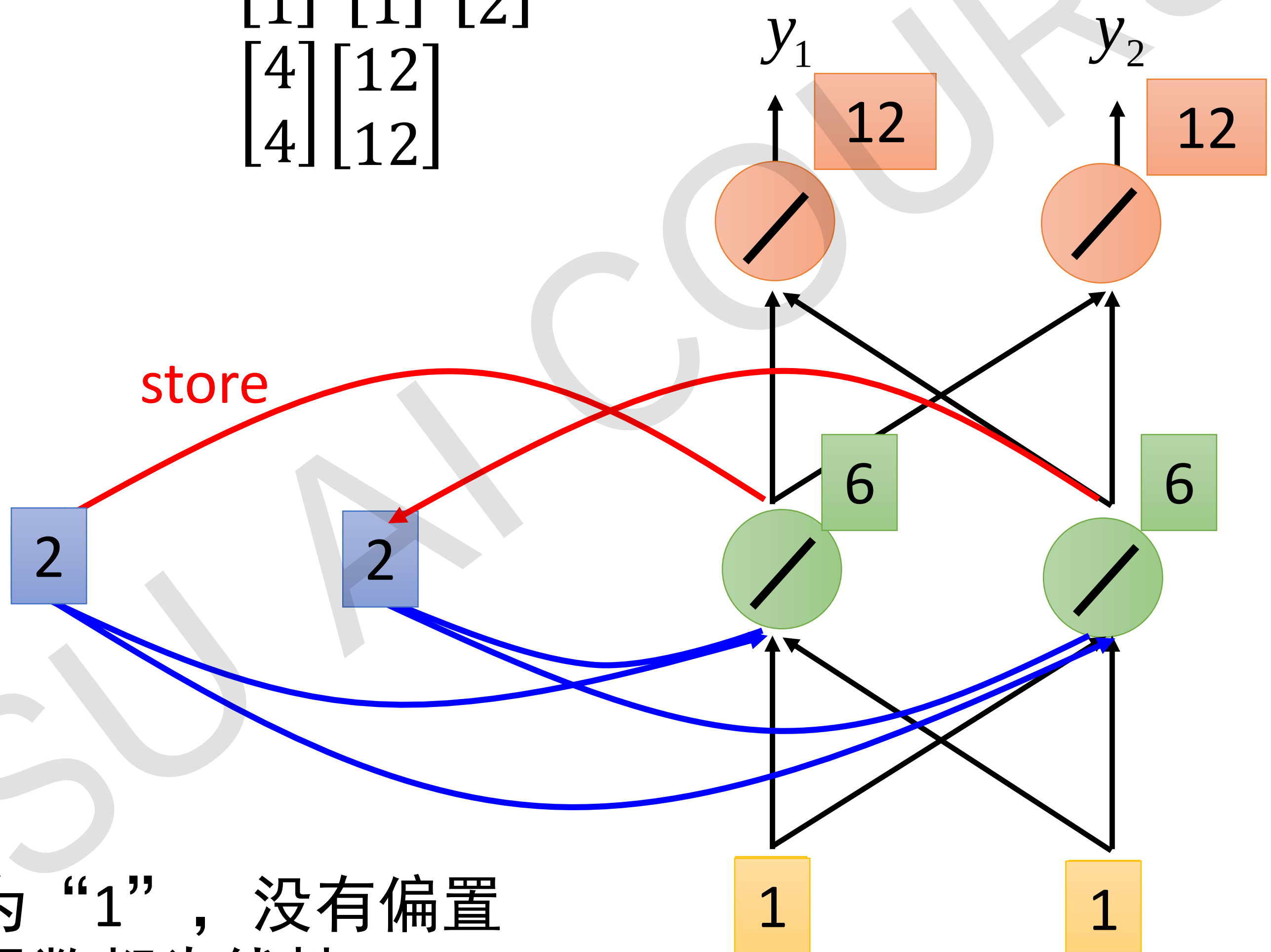


假设所有权重为“1”，没有偏置
且所有激活函数都为线性

循环神经网络 (RNN)

输入序列: $\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}2\\2\end{bmatrix}$ ……

输出序列: $\begin{bmatrix}4\\4\end{bmatrix}\begin{bmatrix}12\\12\end{bmatrix}$



所有权重为“1”，没有偏置
所有激活函数都为线性

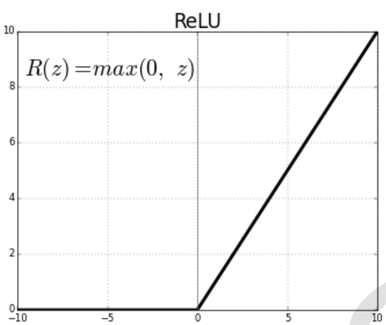
循环神经网络 (RNN)

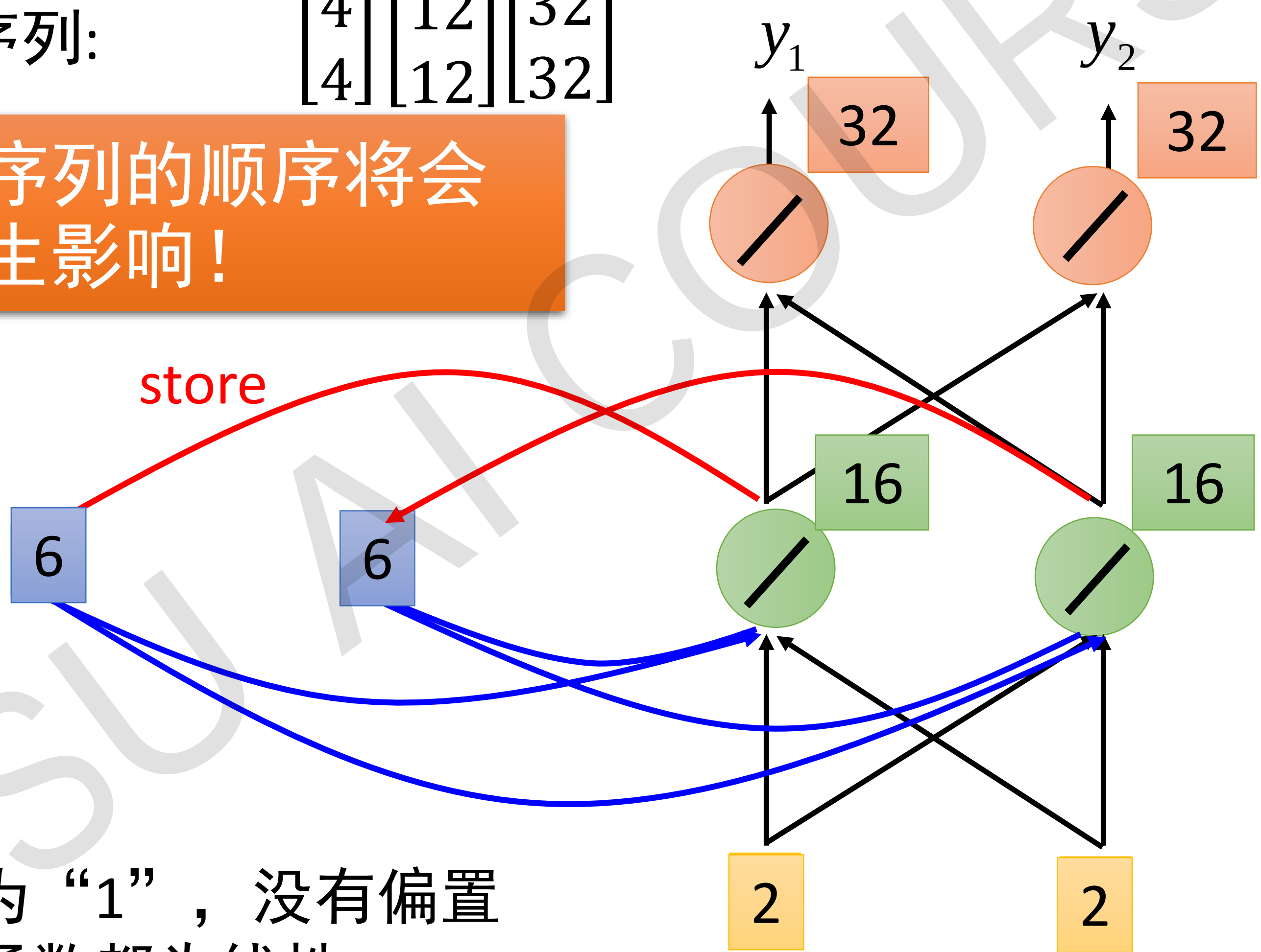
输入序列: $\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}2\\2\end{bmatrix}$ …… ……

输出序列: $\begin{bmatrix}4\\4\end{bmatrix}\begin{bmatrix}12\\12\end{bmatrix}\begin{bmatrix}32\\32\end{bmatrix}$

Try the following sequence With ReLU! $\begin{bmatrix}2\\2\end{bmatrix}\begin{bmatrix}-1\\-1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}$ …… ……

改变输入序列的顺序将会对输出产生影响！





所有权重为“1”，没有偏置
所有激活函数都为线性

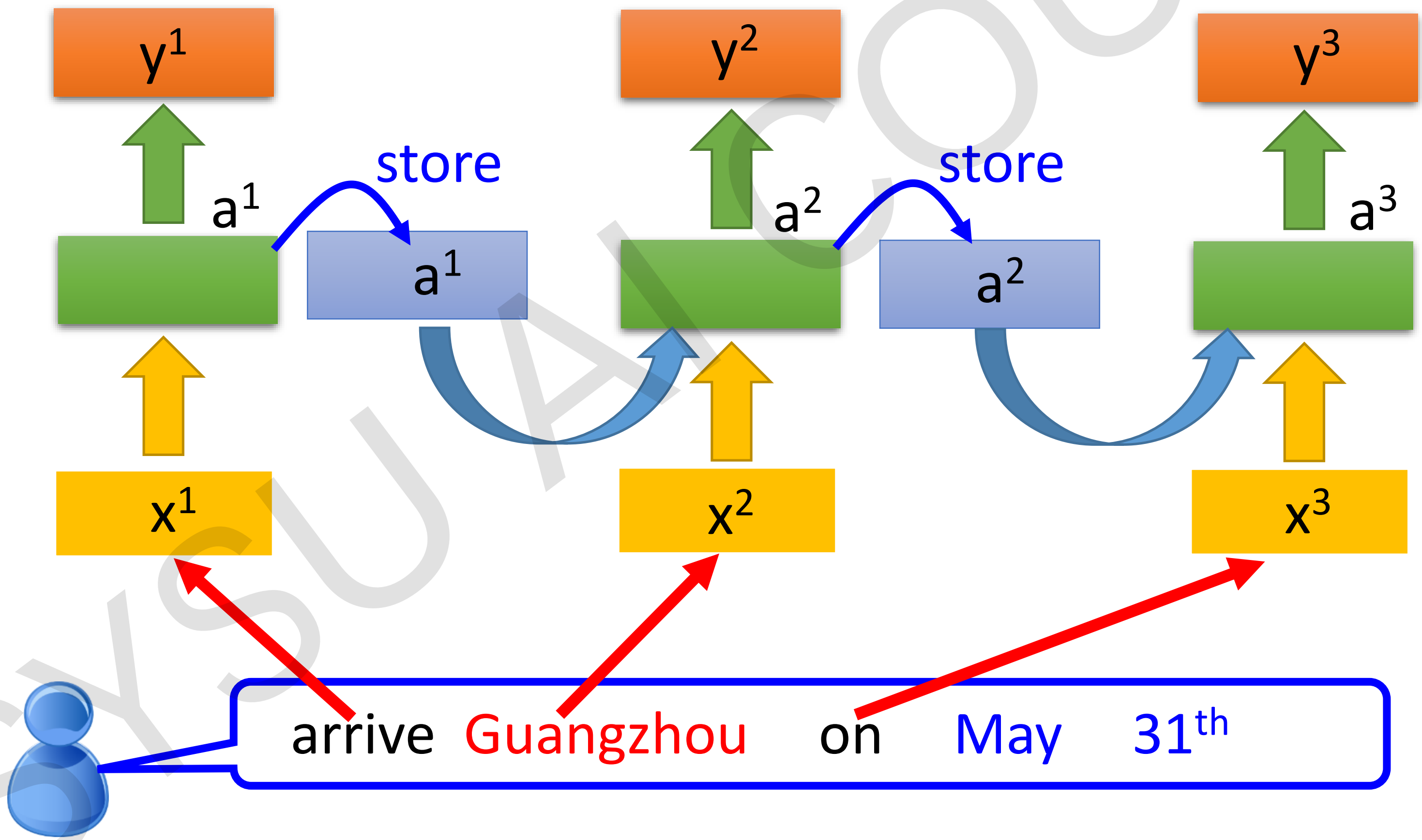
循环神经网络 (RNN)

相同的网络结构多次使用

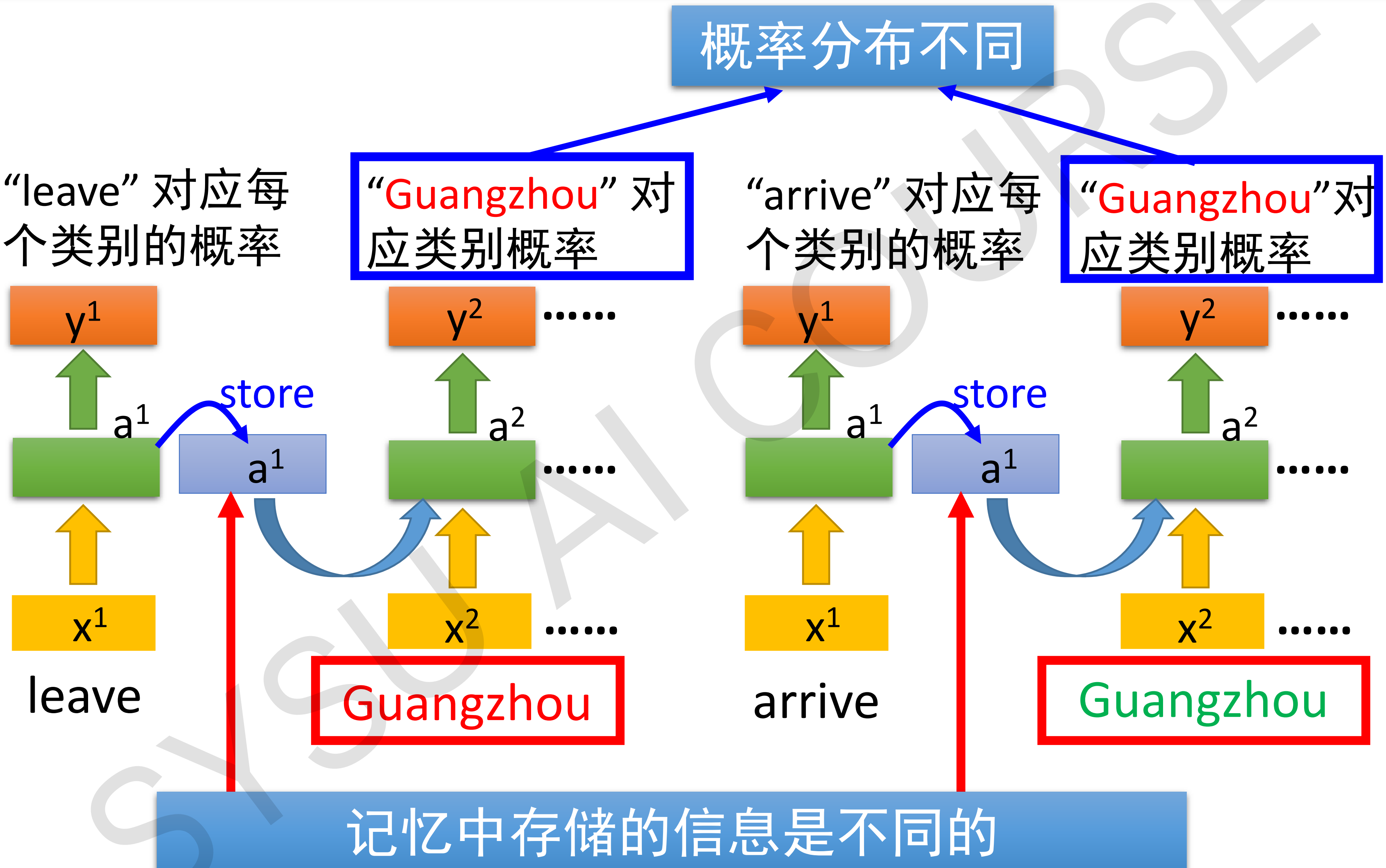
"arrive" 对应每个类别的概率

"Guangzhou" 对应每个类别的概率

"on" 对应每个类别的概率



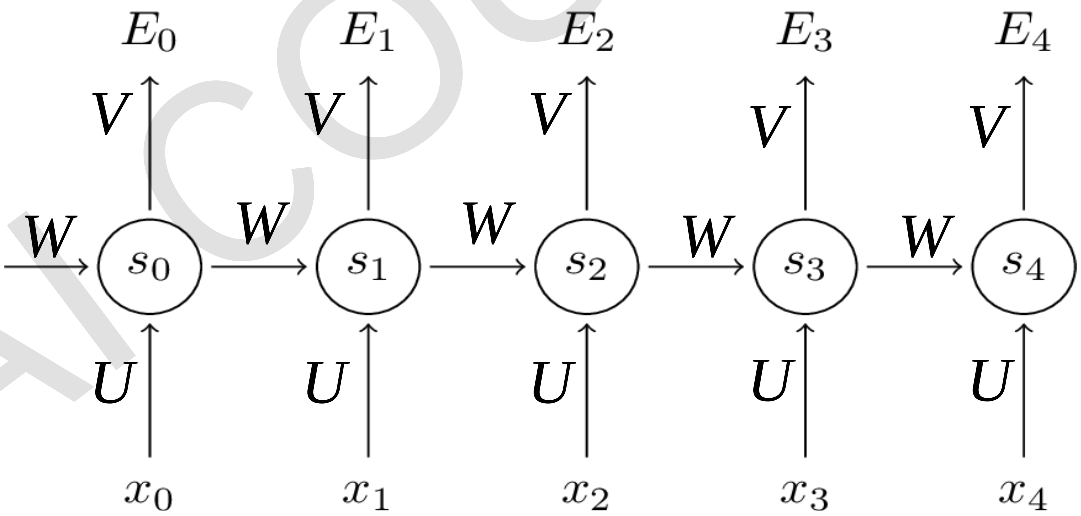
循环神经网络 (RNN)


概率分布不同
"leave" 对应每个类别的概率
"Guangzhou" 对应类别概率
"arrive" 对应每个类别的概率
"Guangzhou"对应类别概率
y1
y2
......
a1
store
a1
a2
x1
x2
leave
Guangzhou
arrive
Guangzhou
记忆中存储的信息是不同的

RNN forward pass

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

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



循环神经网络 (RNN)

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

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

Backpropagation Through Time（BPTT）

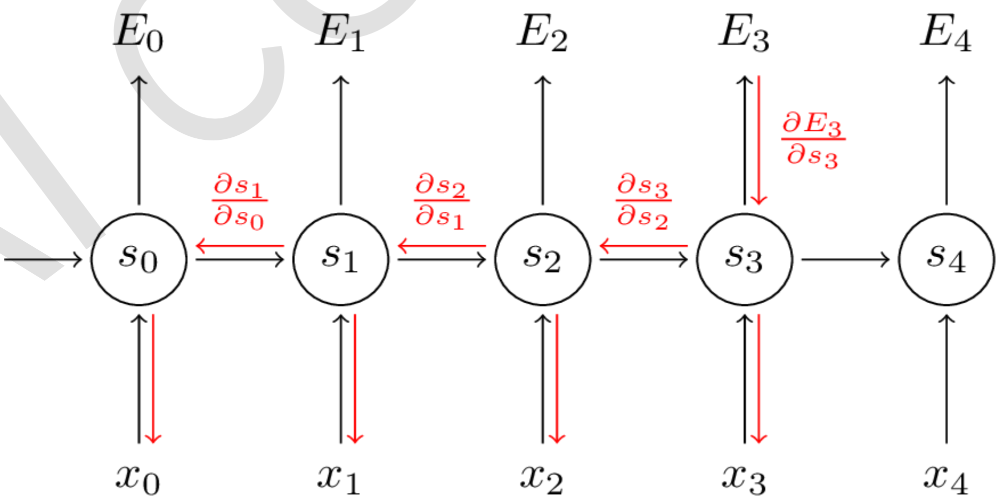
$$\frac{\partial E}{\partial W} = \sum_t \frac{\partial E_t}{\partial W}$$

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

But $s_3 = \tanh(Ux_t + Ws_2)$

S_3 depends on s_2, which depends on W and s_1, and so on.

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



The Vanishing Gradient Problem

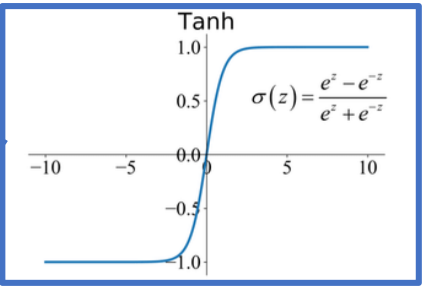
$$\frac{\partial E_3}{\partial W}=\sum_{k=0}^{3}\frac{\partial E_3}{\partial \hat{y}_3}\frac{\partial \hat{y}_3}{\partial s_3}\frac{\partial s_3}{\partial s_k}\frac{\partial s_k}{\partial W}$$

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

Each partial is a Jacobian:

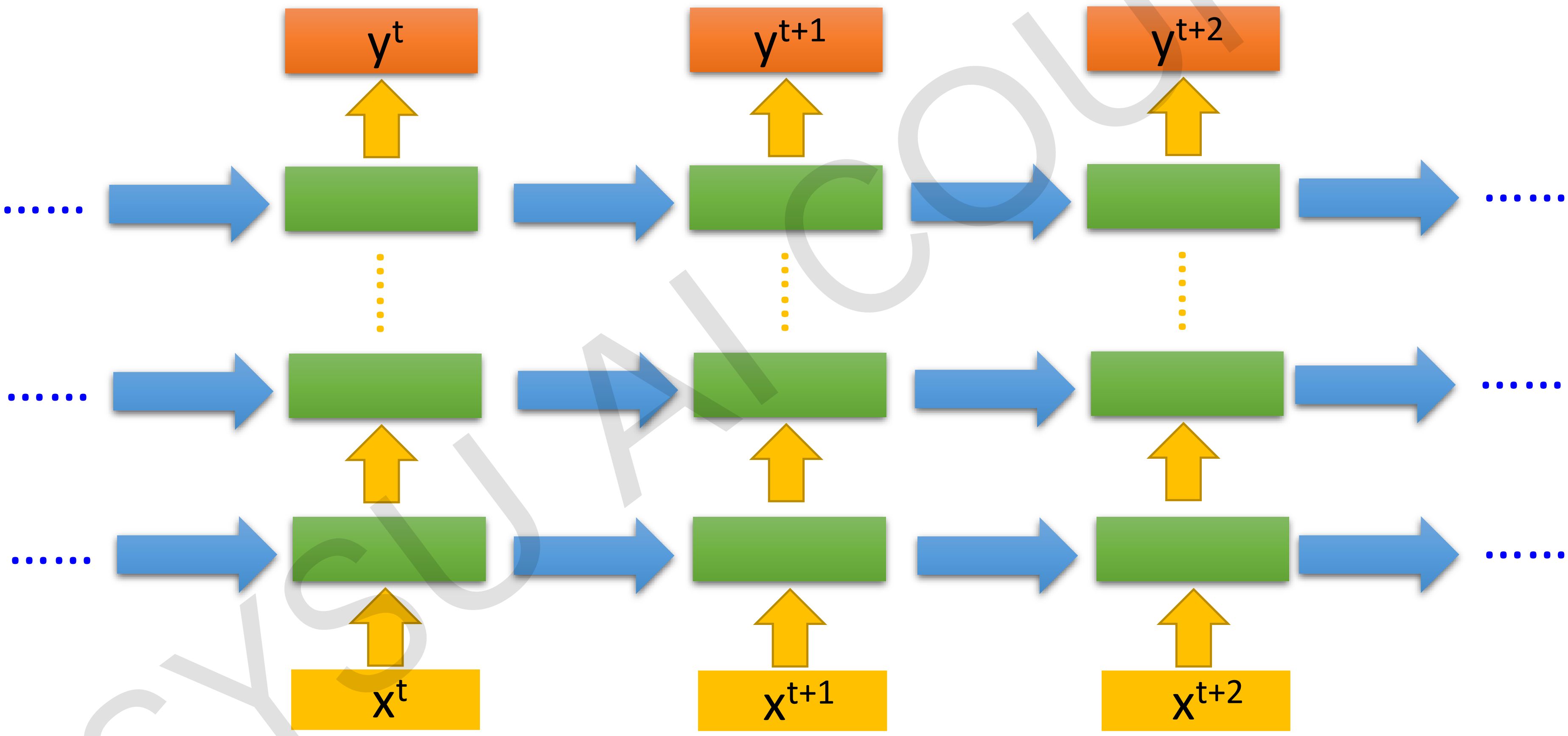
$$\frac{d\mathbf{f}}{d\mathbf{x}}=\left[\frac{\partial \mathbf{f}}{\partial x_1}\ \cdots\ \frac{\partial \mathbf{f}}{\partial x_n}\right]=\begin{bmatrix}\frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n}\\ \vdots & \ddots & \vdots\\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n}\end{bmatrix}$$

- Derivative of a vector w.r.t a vector is a matrix called jacobian
- 2-norm of the above Jacobian matrix has an upper bound of 1
- **tanh** maps all values into a range between -1 and 1, and the **derivative is bounded by 1**
- With multiple matrix multiplications, gradient values **shrink exponentially**
- Gradient contributions from “far away” steps become zero
- Depending on activation functions and network parameters, gradients could **explode** instead of vanishing



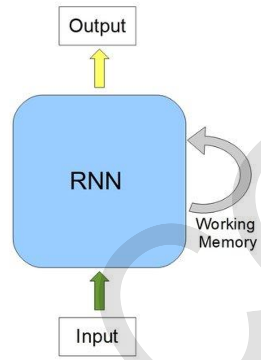
循环神经网络 (RNN)

当然，循环结构可以构造得很深 ...



循环神经网络 (RNN) —— LSTM

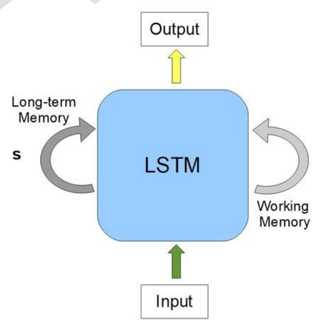
经典循环神经网络 (Recurrent Neural network)



回忆过去,痛苦的相思忘不了…
為何你還來 撥動我心跳…
緣難了 情難了…

长短期记忆网络 (Long Short-term Memory)

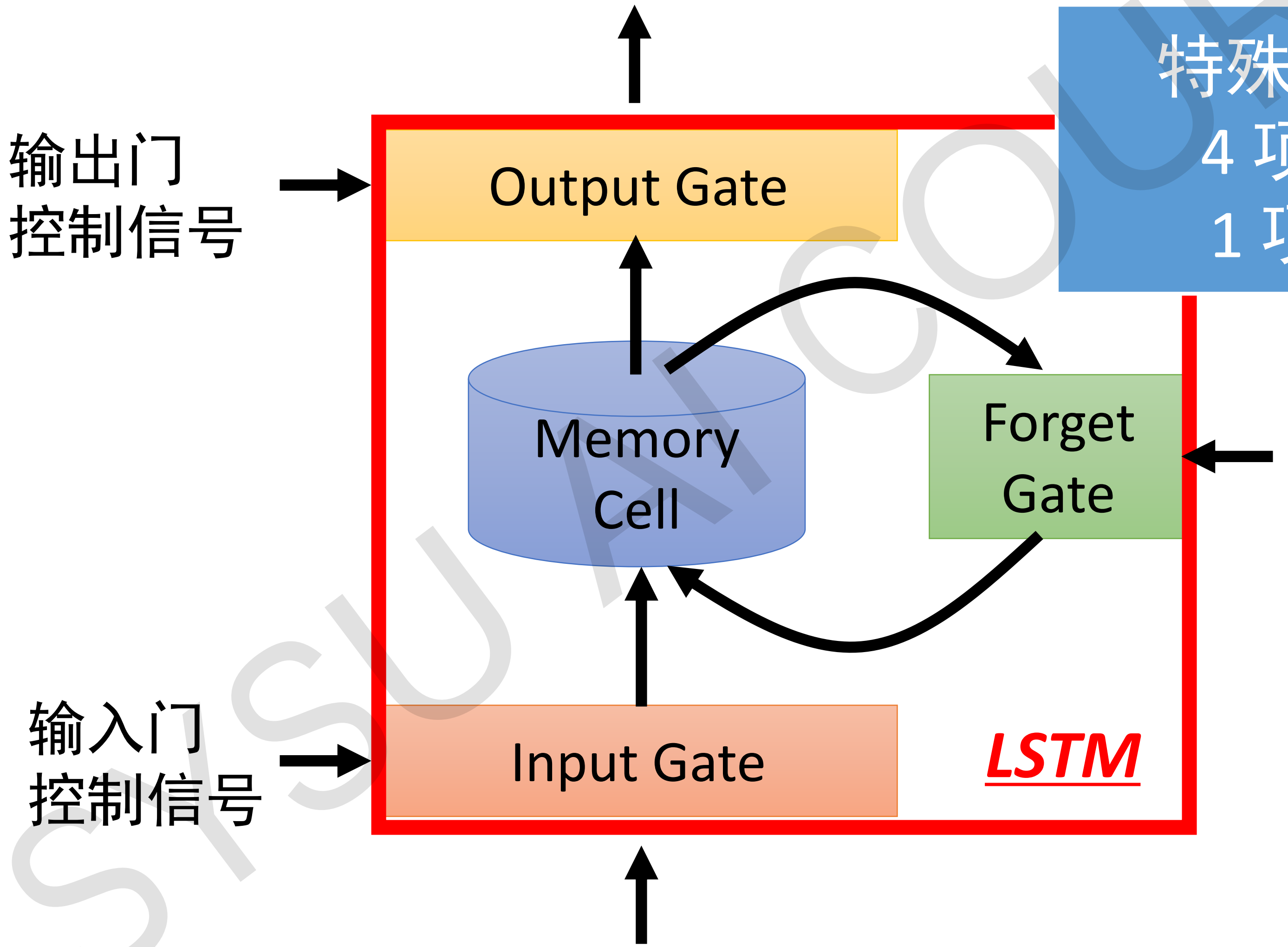
…你的影子無所不在…
落在过去，飘向未来,…
就让往事随风都随风都随风，
心随你动…



引用：1. 萧敬腾，《新不了情》，2010
2. 齐秦，《往事随风》，1995

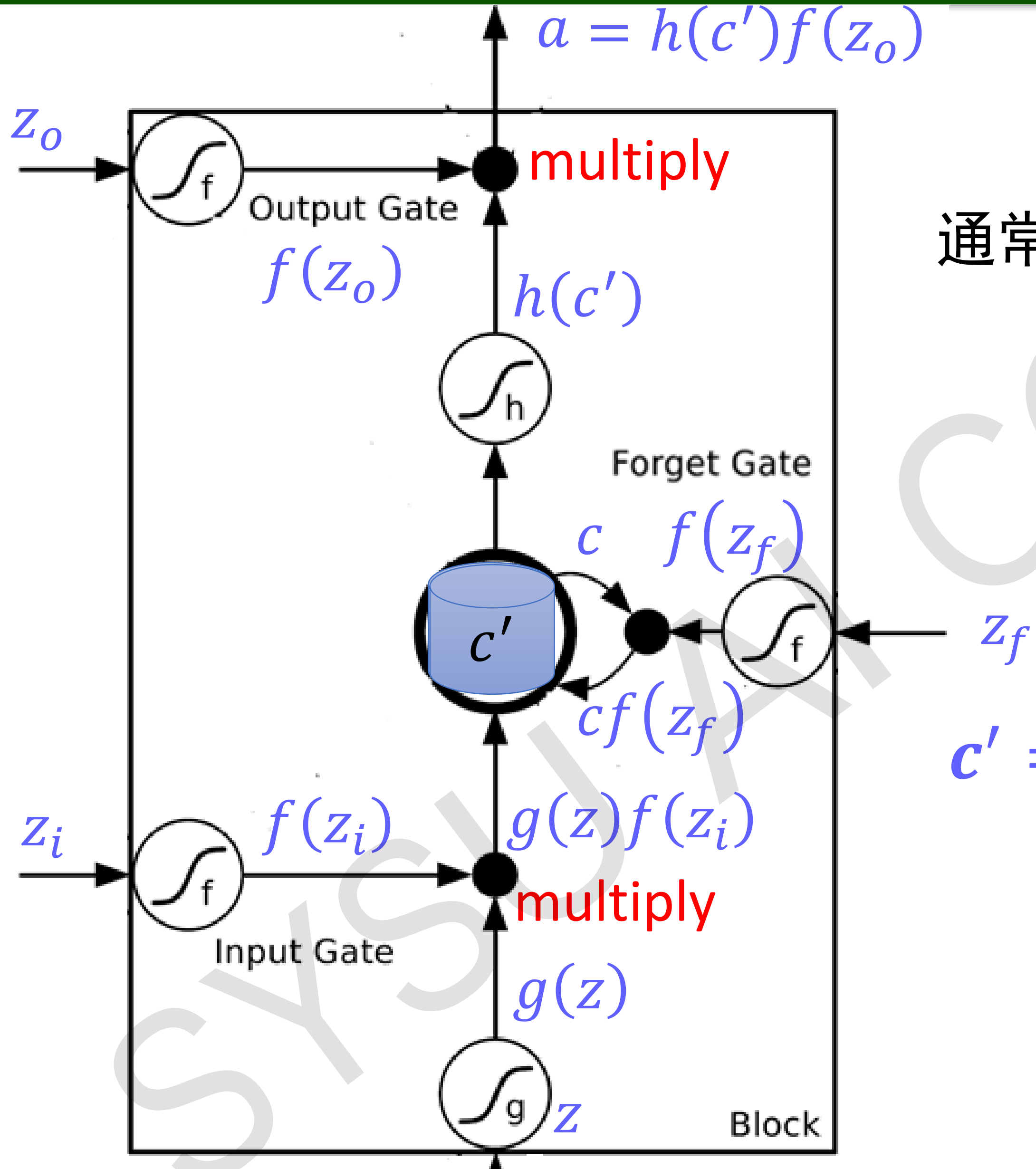
循环神经网络 (RNN) —— LSTM

长短期记忆网络 (Long Short-term Memory)



特殊神经元:
4 项输入,
1 项输出

循环神经网络 (RNN) —— LSTM



通常使用 sigmoid 激活函数

限制在 0 到 1 之间

模拟门的开关

$$\boldsymbol{c' = g(z)f(z_i) + cf(z_f)}$$

LSTM 示例

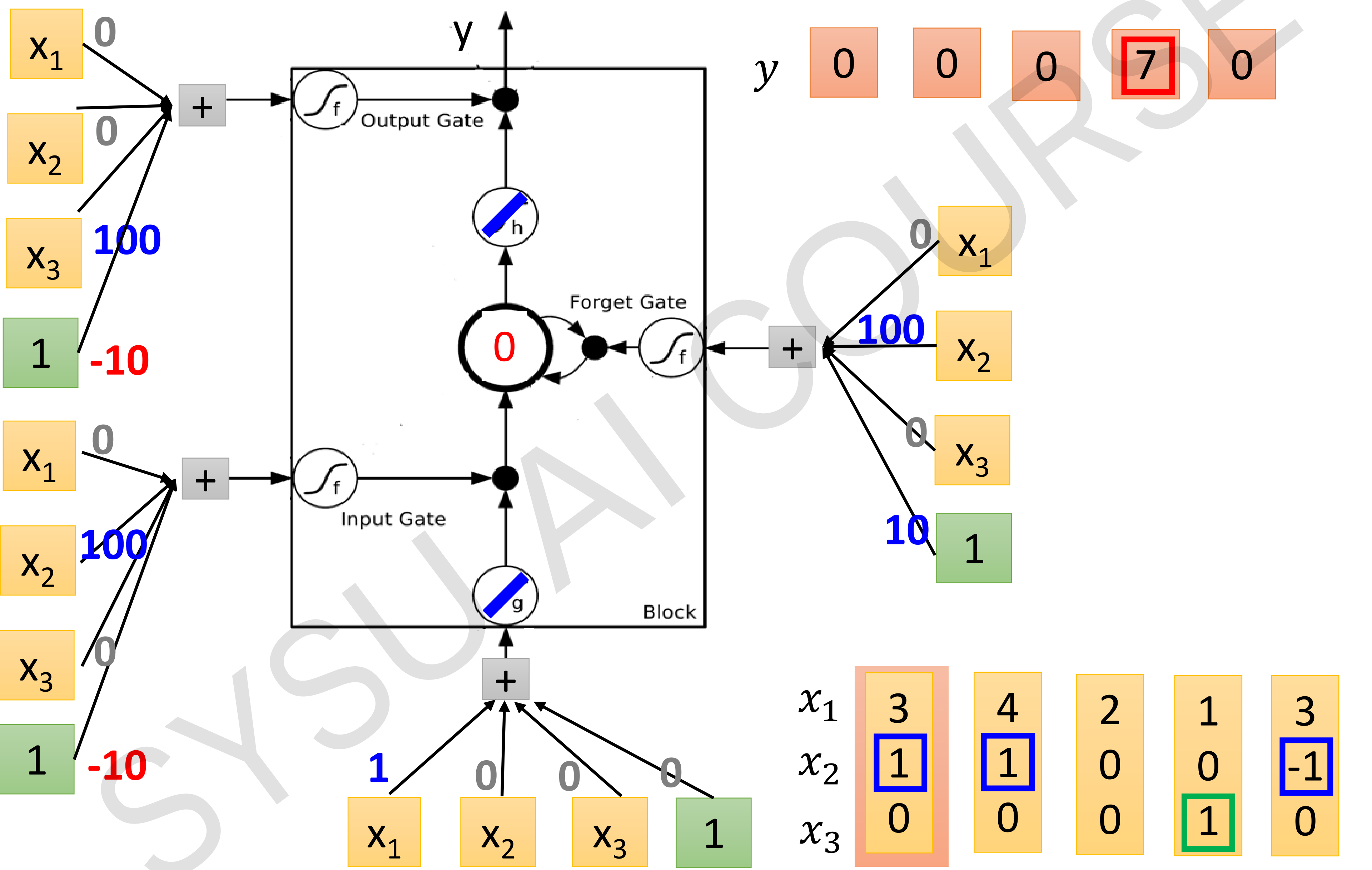
	0	0	3	3	7	7	7	0	6
x_1	1	3	2	4	2	1	3	6	1
x_2	0	1	0	1	0	0	-1	1	0
x_3	0	0	0	0	0	1	0	0	1
y	0	0	0	0	0	7	0	0	6

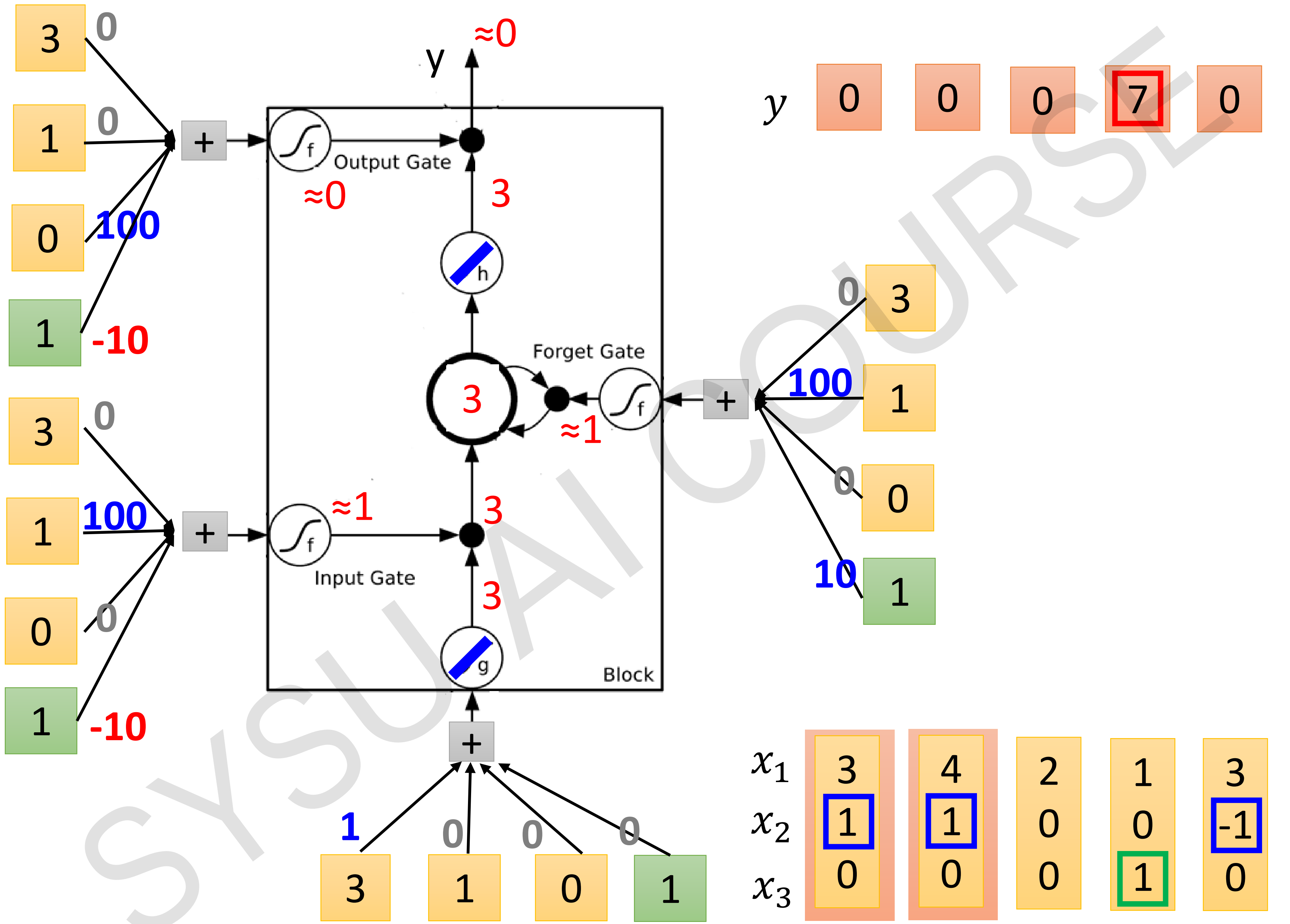
当 $x_2 = 1$, 把 x_1 的值累加到记忆中

当 $x_2 = -1$, 重置记忆

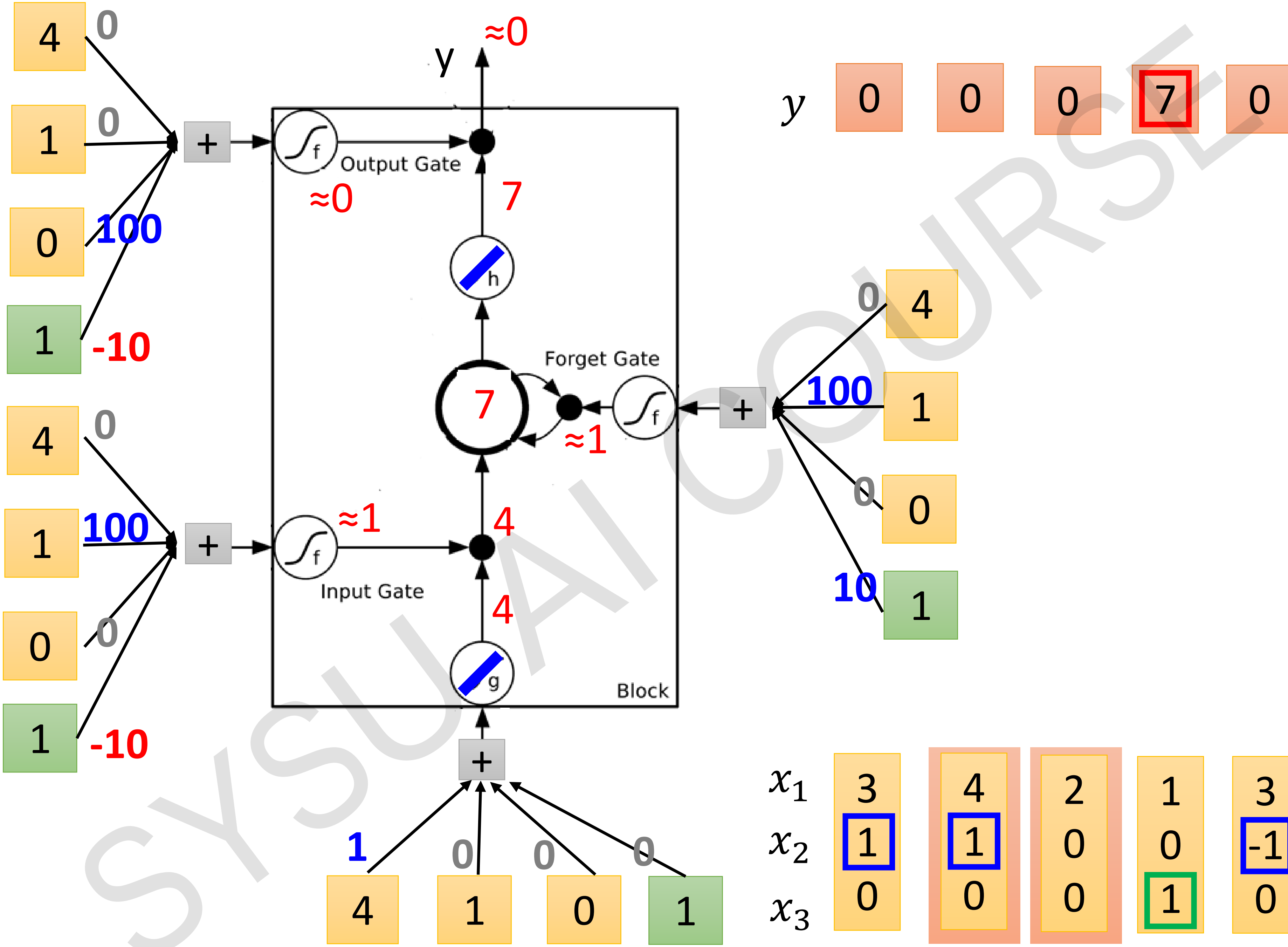
当 $x_3 = 1$, 输出记忆的值

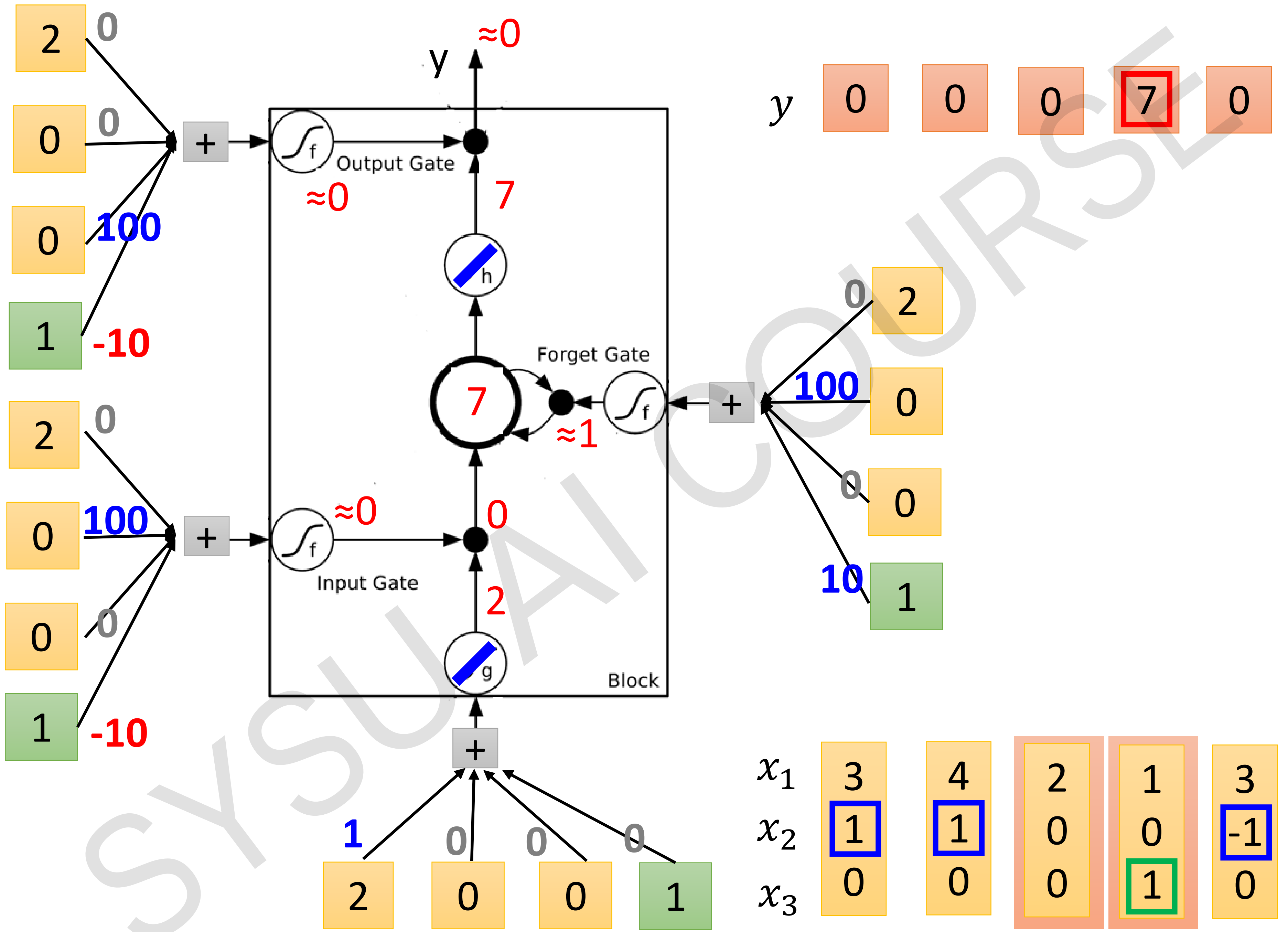
循环神经网络 (RNN) —— LSTM



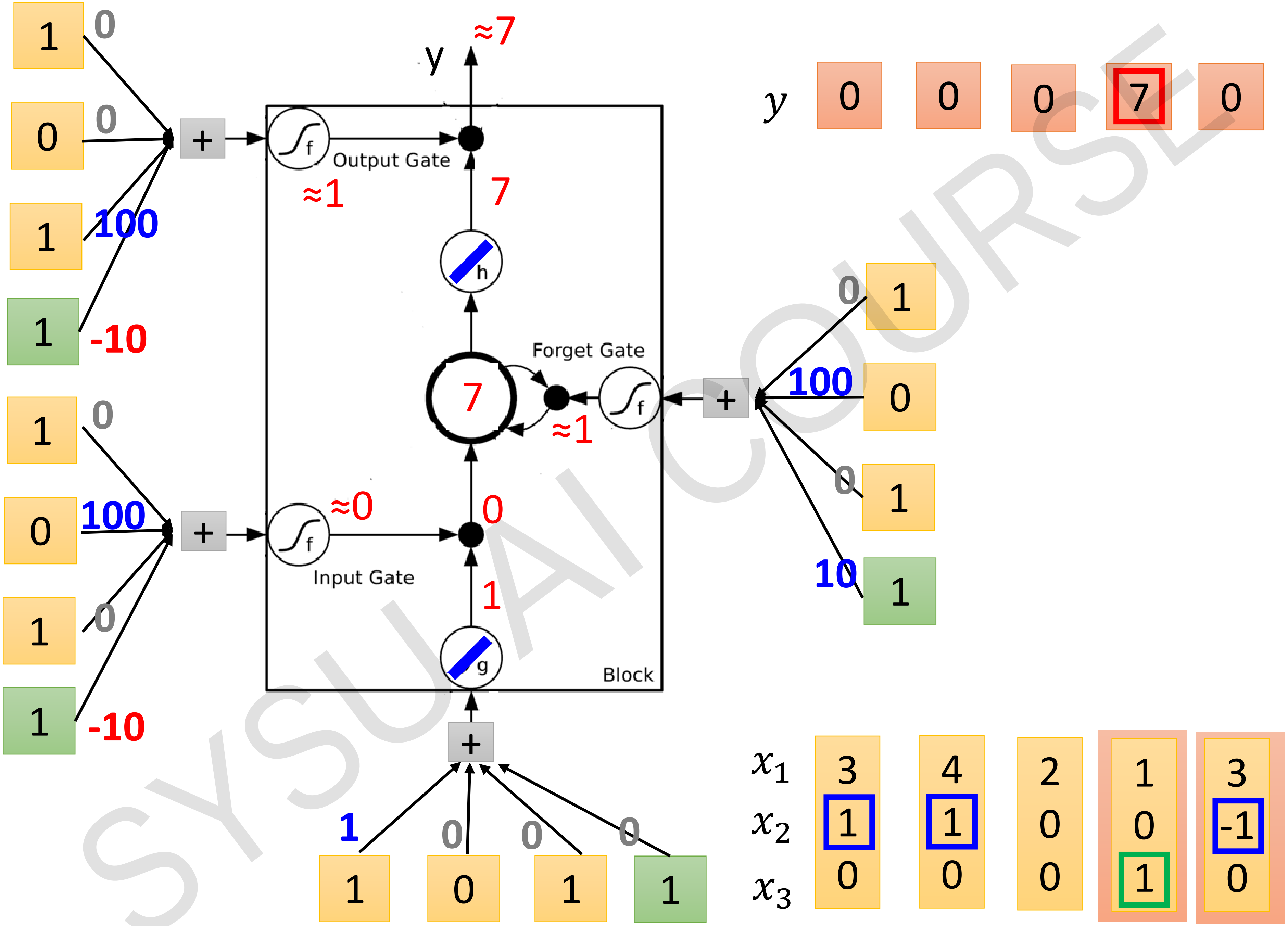


y
≈0
Output Gate
≈0
3
h
Forget Gate
3
≈1
f
Input Gate
≈1
3
3
g
Block
3 0
1 0
0 100
1 -10
3 0
1 100
0 0
1 -10
0 3
100 1
0 0
10 1
1 0 0 0
3 1 0 1
y 0 0 0 7 0
x1 3 4 2 1 3
x2 1 1 0 0 -1
x3 0 0 0 1 0

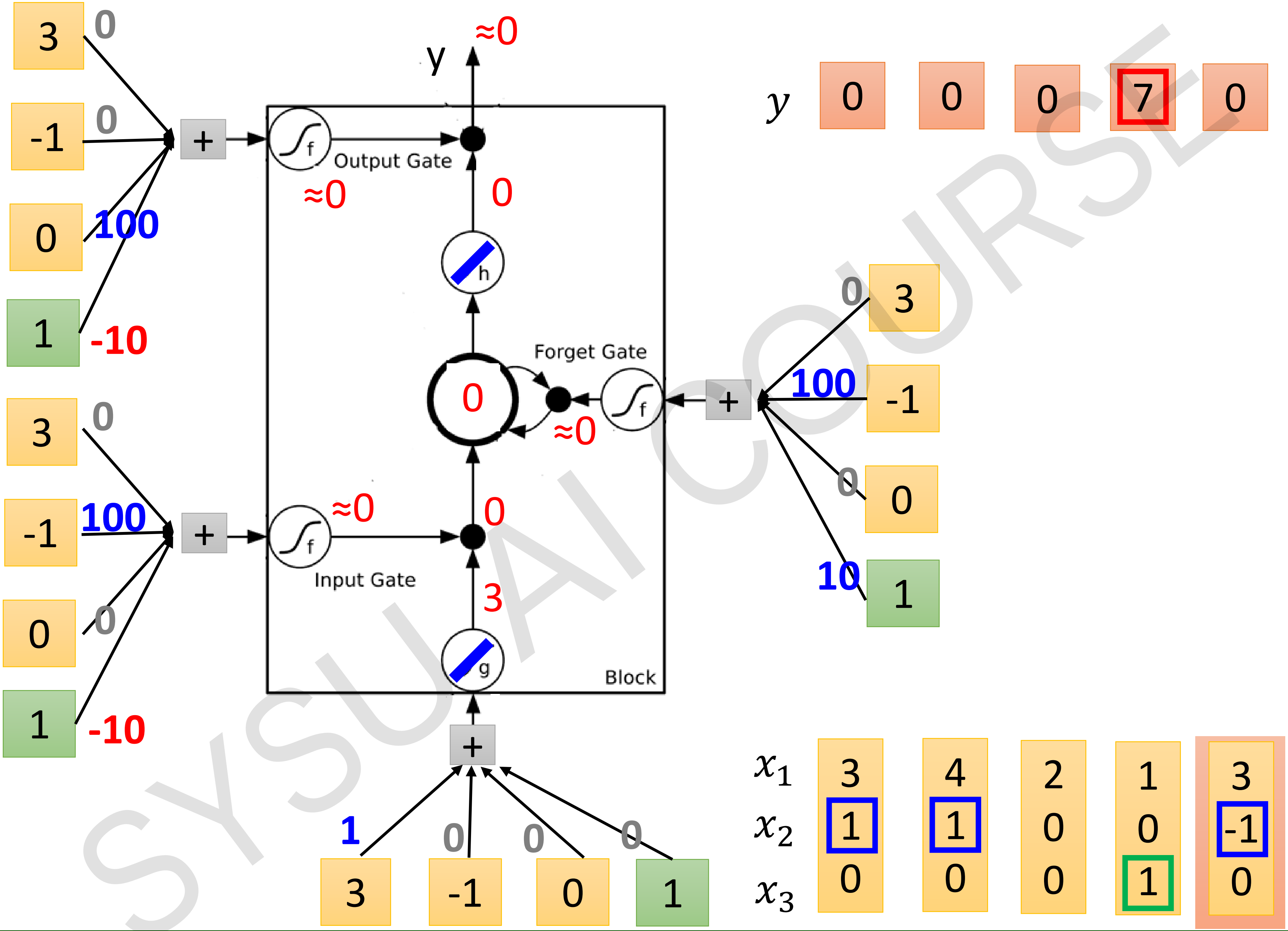

y
≈0
Output Gate
≈0
7
h
7
Forget Gate
≈1
4
Input Gate
≈1
4
g
Block
4
0
1
0
0
100
1
-10
4
0
1
100
0
0
1
-10
0
4
100
1
0
0
10
1
1
4
0
1
0
0
0
1
y
0
0
0
7
0
x_1
x_2
x_3
3
1
0
4
1
0
2
0
0
1
0
1
3
-1
0



y
≈0
Output Gate
≈0
7
h
Forget Gate
7
≈1
≈0
0
Input Gate
2
g
Block
2 0
0 0
0 100
1 -10
2 0
0 100
0 0
1 -10
0 2
100 0
0 0
10 1
1 2
0 0
0 0
0 1
y 0 0 0 7 0
x1 3 4 2 1 3
x2 1 1 0 0 -1
x3 0 0 0 1 0
SYSU AI COURSE



y
≈7
Output Gate
≈1
7
h
Forget Gate
7
≈1
≈0
0
Input Gate
1
g
Block
+
1
0
0
0
1
100
1
-10
1
0
0
100
1
0
1
-10
0
1
100
0
0
1
10
1
1
1
0
0
0
1
0
1
1
y
0 0 0 7 0
x1
3 4 2 1 3
x2
1 1 0 0 -1
x3
0 0 0 1 0

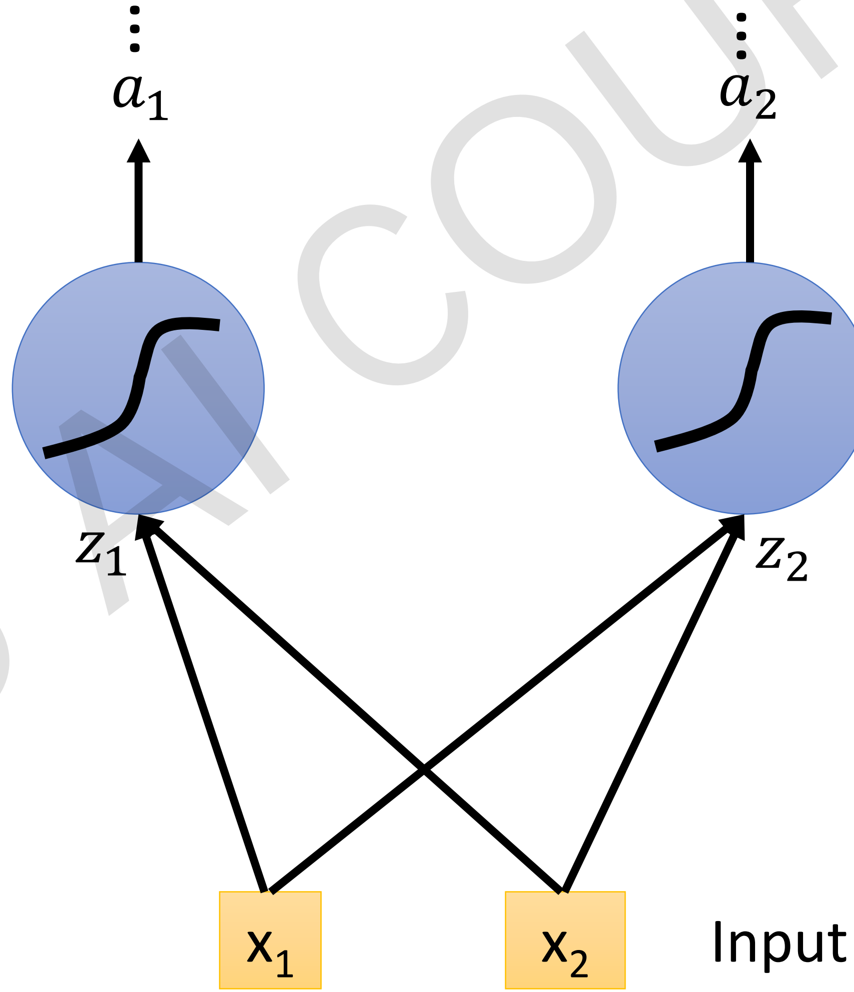


y
≈0
Output Gate
≈0
0
Forget Gate
0
≈0
≈0
0
Input Gate
3
Block
3 0
-1 0
0 100
1 -10
3 0
-1 100
0 0
1 -10
0 3
100 -1
0 0
10 1
1 0 0 0
3 -1 0 1
y 0 0 0 7 0
x1 3 4 2 1 3
x2 1 1 0 0 -1
x3 0 0 0 1 0
SYSU AI COURSE

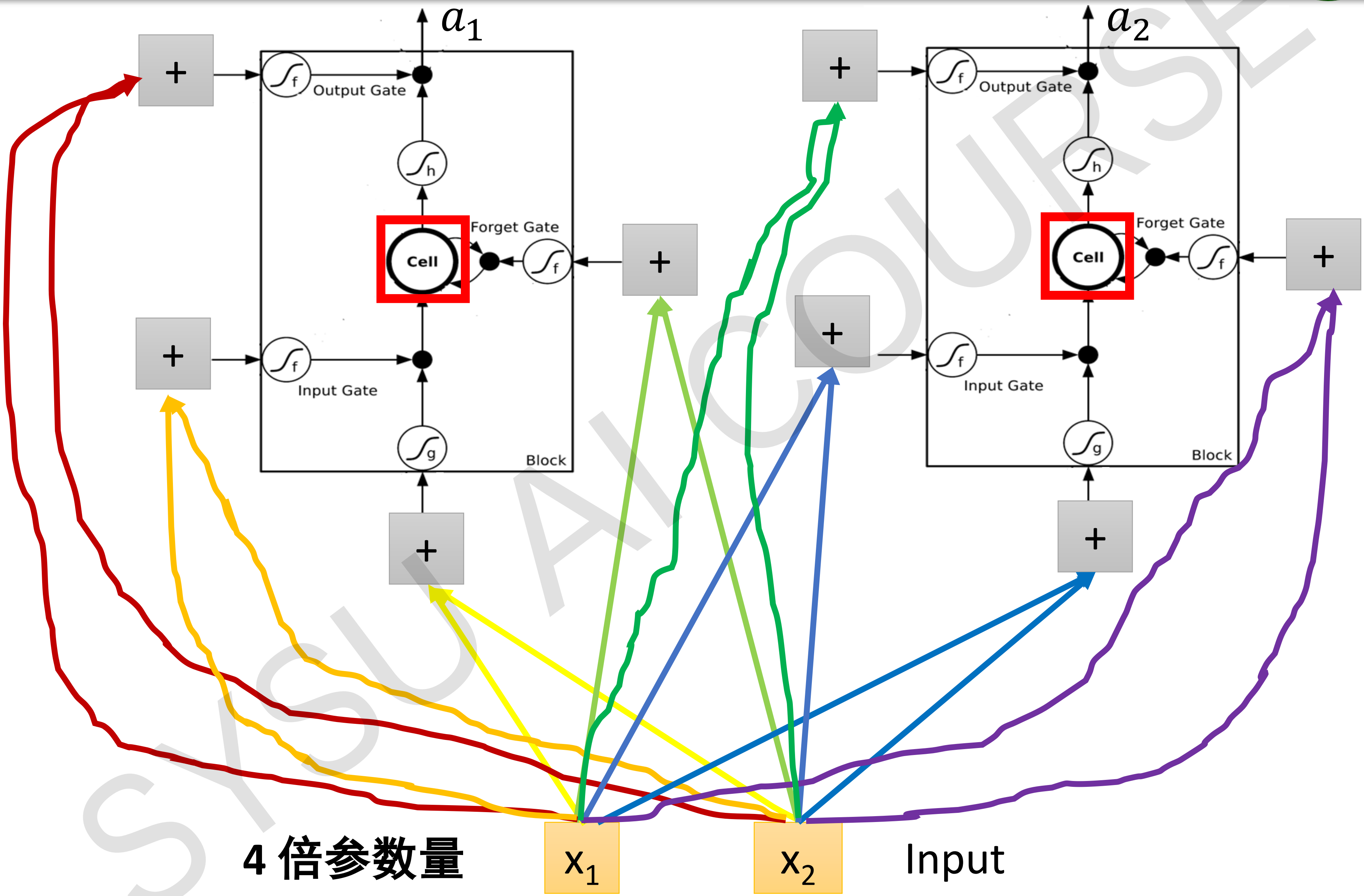
循环神经网络 (RNN) —— LSTM

网络结构:

- ➢用 LSTM 替换原始神经元

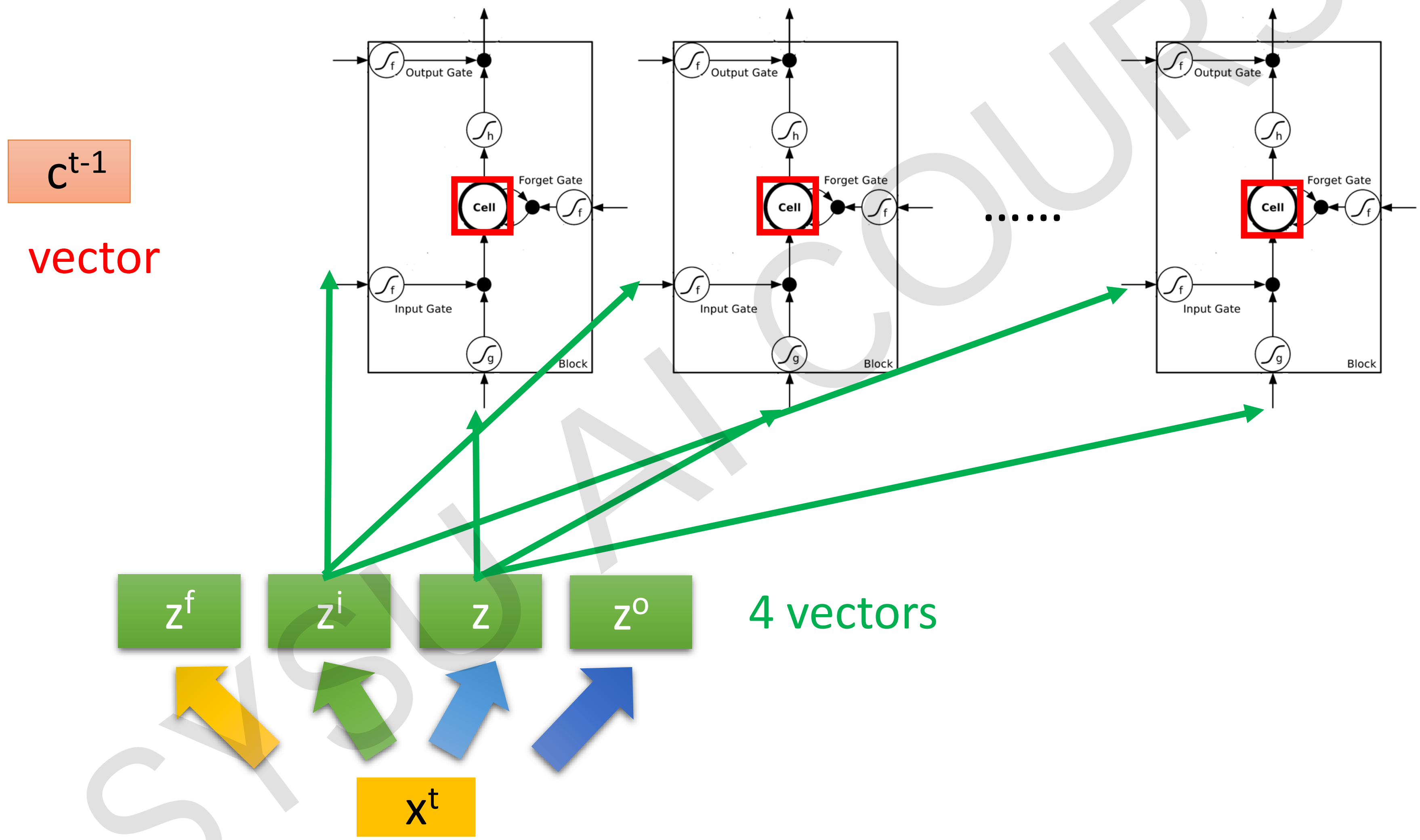


循环神经网络 (RNN) —— LSTM



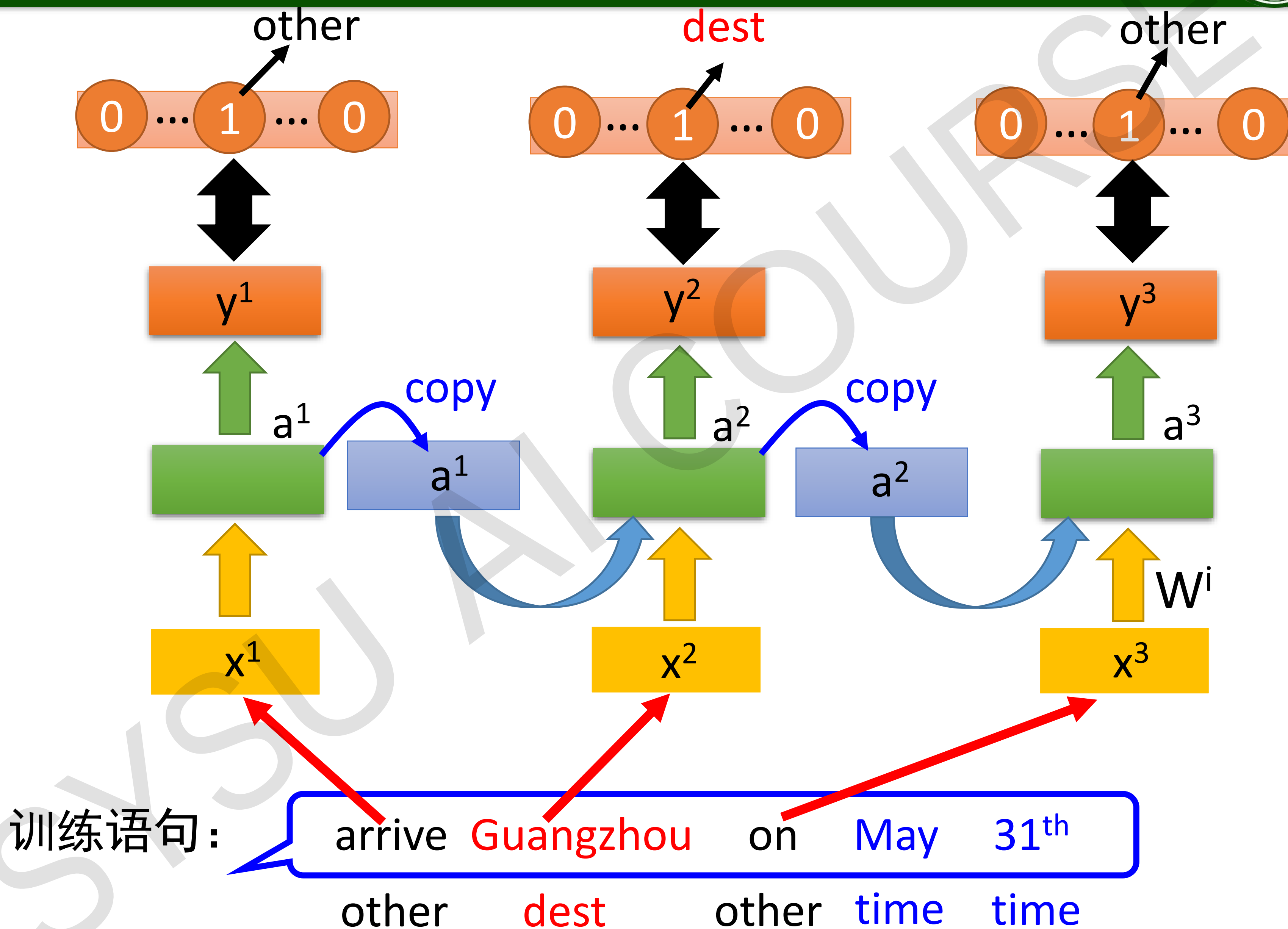
4 倍参数量

循环神经网络 (RNN) —— LSTM



循环神经网络 (RNN)

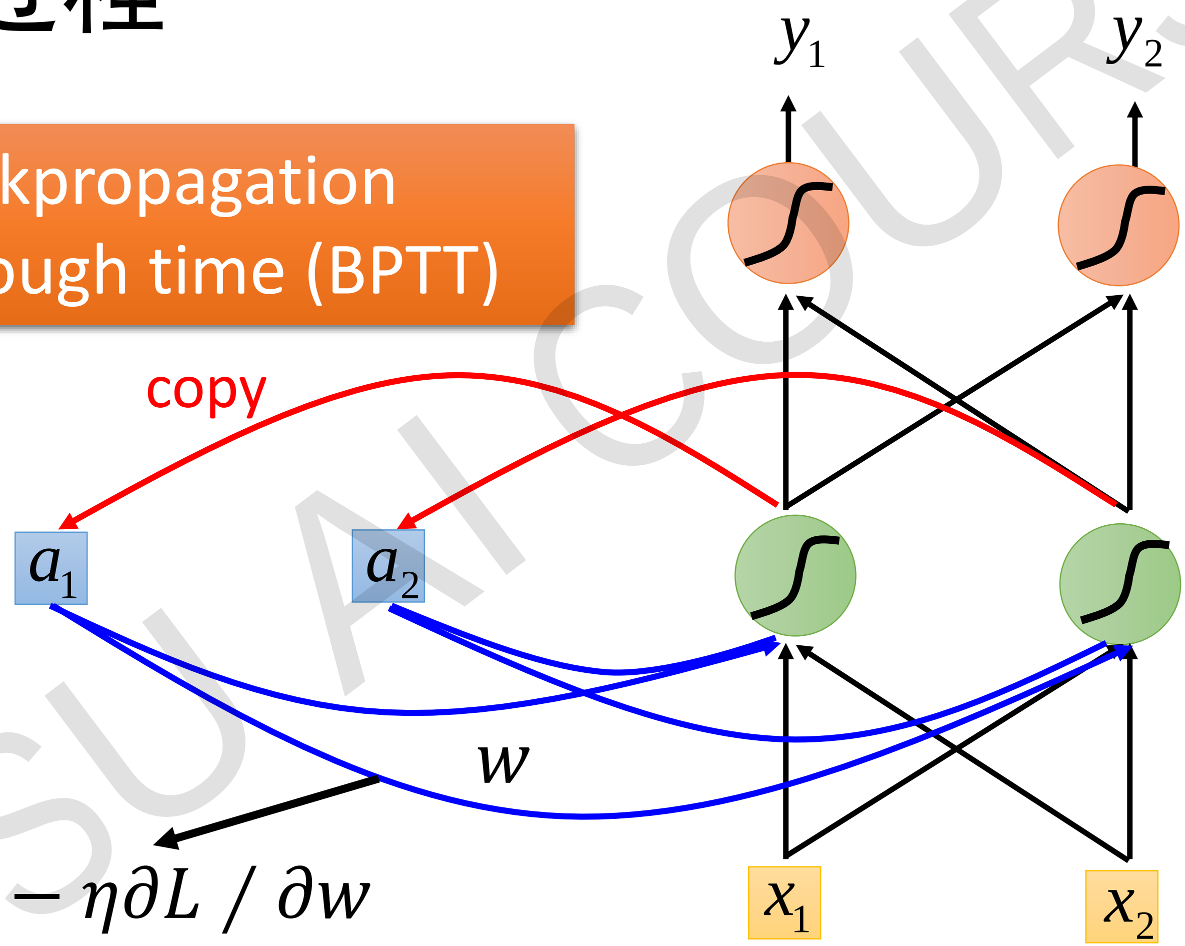
学习目标



循环神经网络 (RNN)

学习过程

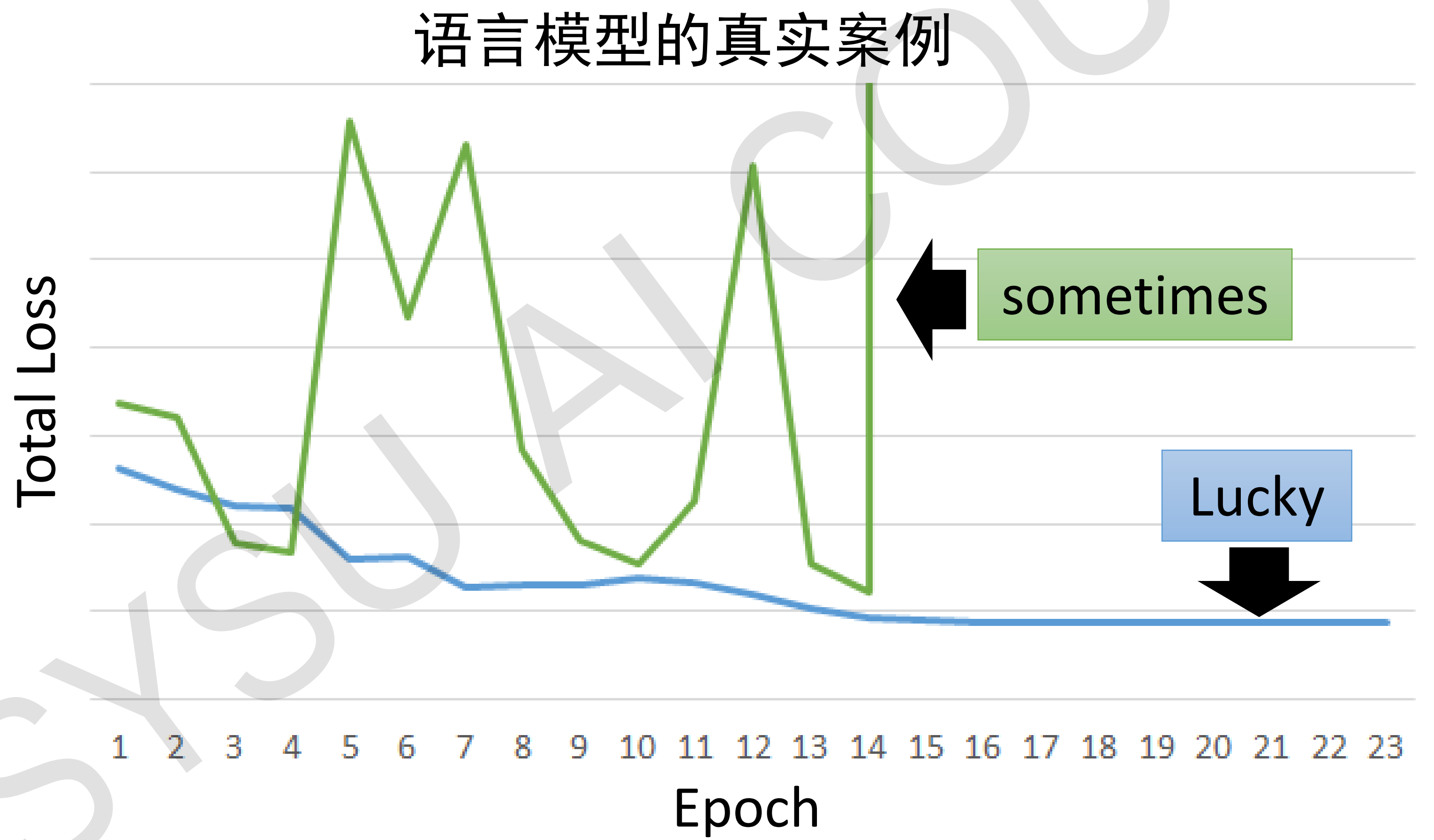
Backpropagation through time (BPTT)



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

循环神经网络 (RNN)

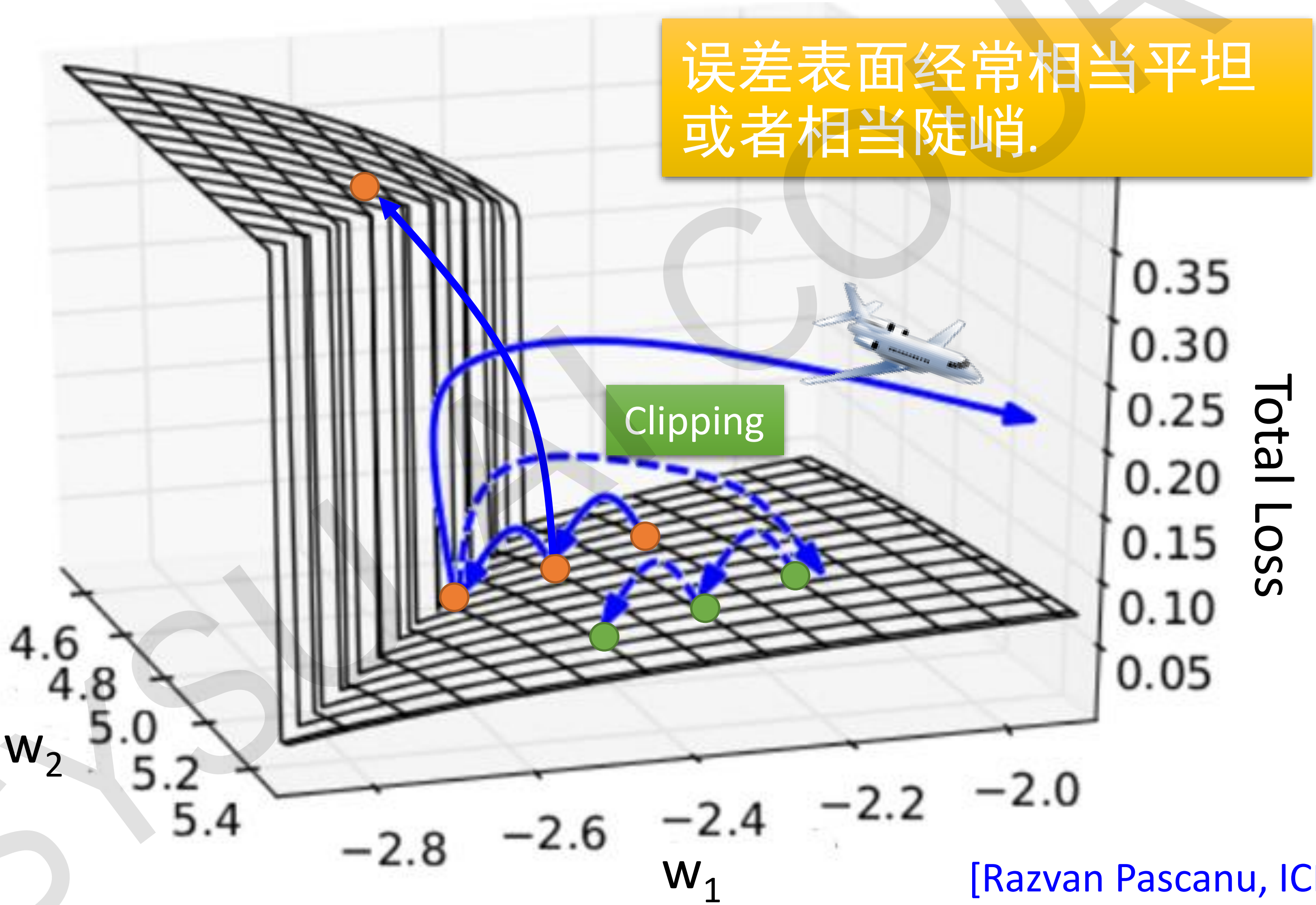
但是 … 基于 RNN 的网络并不总是容易学习的



循环神经网络 (RNN)

误差表面相当粗糙

误差表面经常相当平坦或者相当陡峭.



[Razvan Pascanu, ICML'13]

循环神经网络 (RNN)

Why?

$w = 1 \Rightarrow y^{1000} = 1$

$w = 1.01 \Rightarrow y^{1000} \approx 20000$

Large $\partial L / \partial w$ → Small Learning rate?

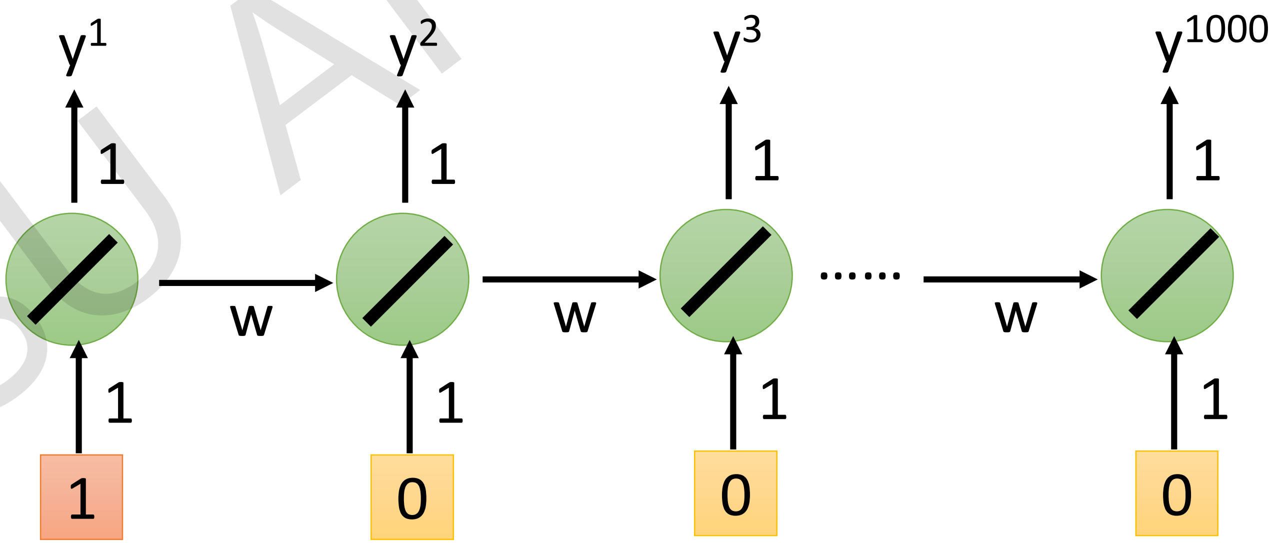
$w = 0.99 \Rightarrow y^{1000} \approx 0$

$w = 0.01 \Rightarrow y^{1000} \approx 0$

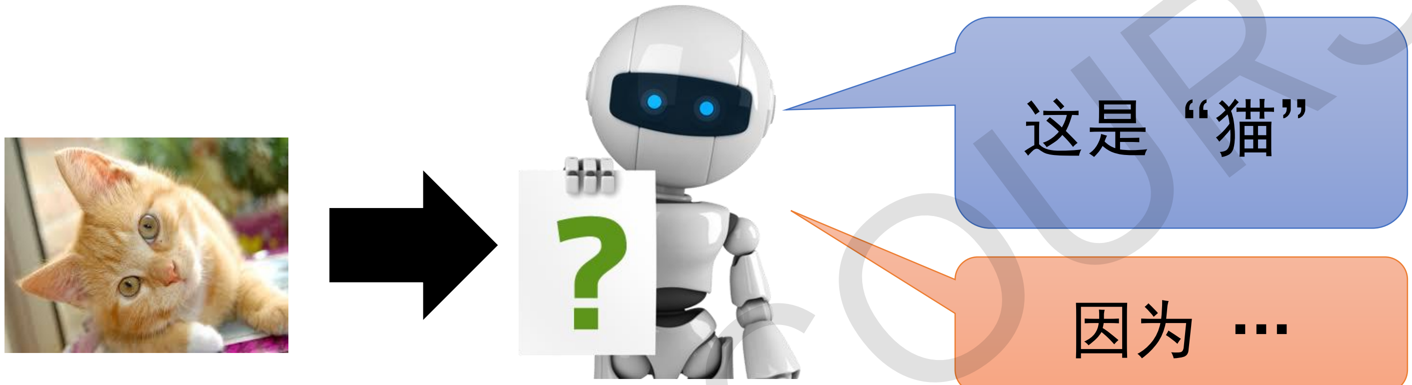
small $\partial L / \partial w$ → Large Learning rate?

$=w^{999}$

Toy Example



深度学习的可解释性



分类器

Local Explanation

为什么你认为 ***这张图片*** 是猫？

Global Explanation

你认为“猫”应该是什么样子？

深度学习的可解释性

为什么我们需要可解释的机器学习？

深度学习的可解释性

为什么我们需要可解释的机器学习？

用机器来协助判断简历

具体能力？还是性别？

用机器来协助判断犯人是否可以假释

具体证据？还是肤色？

金融相关的决策常常依法需要提供理由

为什麽拒绝了某個人的贷款？

模型诊断：到底机器学到了什么

不能只看正确率？想想神马汉斯的故事

深度学习的可解释性

我们可以根据可解释性来改进机器学习模型

我知道答案为什么错，所以我可以修正它

With explainable ML

https://www.explainxkcd.com/wiki/index.php/1838:_Machine_Learning

深度学习的可解释性

机器学习可解释性的目标 ≠ 完全清楚机器学习模型的工作机理

- 人脑也是一个黑盒模型!
- 人类不相信深度学习因为它是黑盒，但他们却相信其他人类的决策!

机器学习可解释性的目标是(个人观点)

- Make people (your customers, your boss, yourself) comfortable.
- Make machine controllable.

让人觉得“舒坦”

让机器变得“可控”

可解释性 v.s. 功能强大

- 某些模型本质上是可解释的
  - 例如，线性模型(可以从权重知道特征的重要性)
  - 但是，这种模型功能并不强大.
- 深度神经网络很难解释
  - 深度神经网络是黑盒

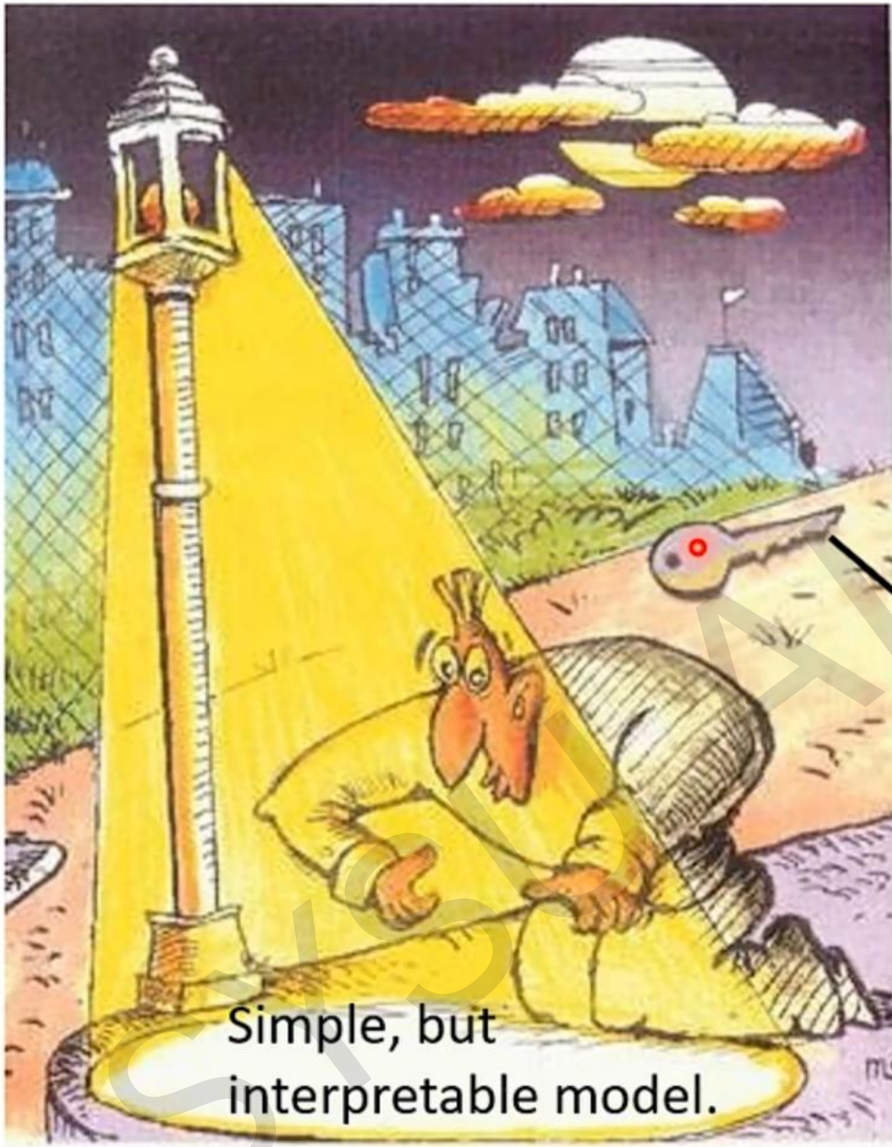
因为深度网络是黑盒，
所以我们不用它。

= 削足适履 ☹

  - 但它功能远强大于线性模型 …

Let’s make deep network interpretable.

深度学习的可解释性

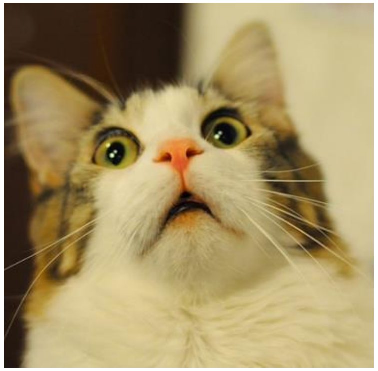


Powerful Model

应该扩大“照明”范围，让强大的模型（钥匙）处于可解释（光照）范围内，而不是放弃使用照明范围外的物体。

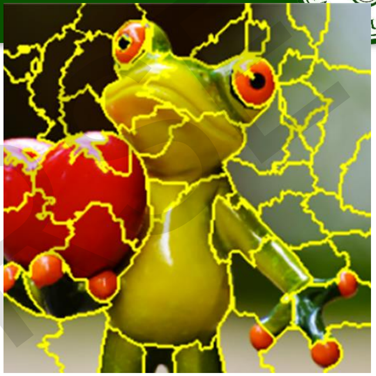
Local Explanation: Explain the Decision

Questions: Why do you think this image is a cat?



Basic Idea

Image: pixel, segment, etc.
Text: a word



Object x ➡ Components: $\{x_1, \cdots, x_n, \cdots, x_N\}$

我们想知道每个部分对于决策的重要程度

Idea: 移除或改变该部分的值，观察决策结果是否变化

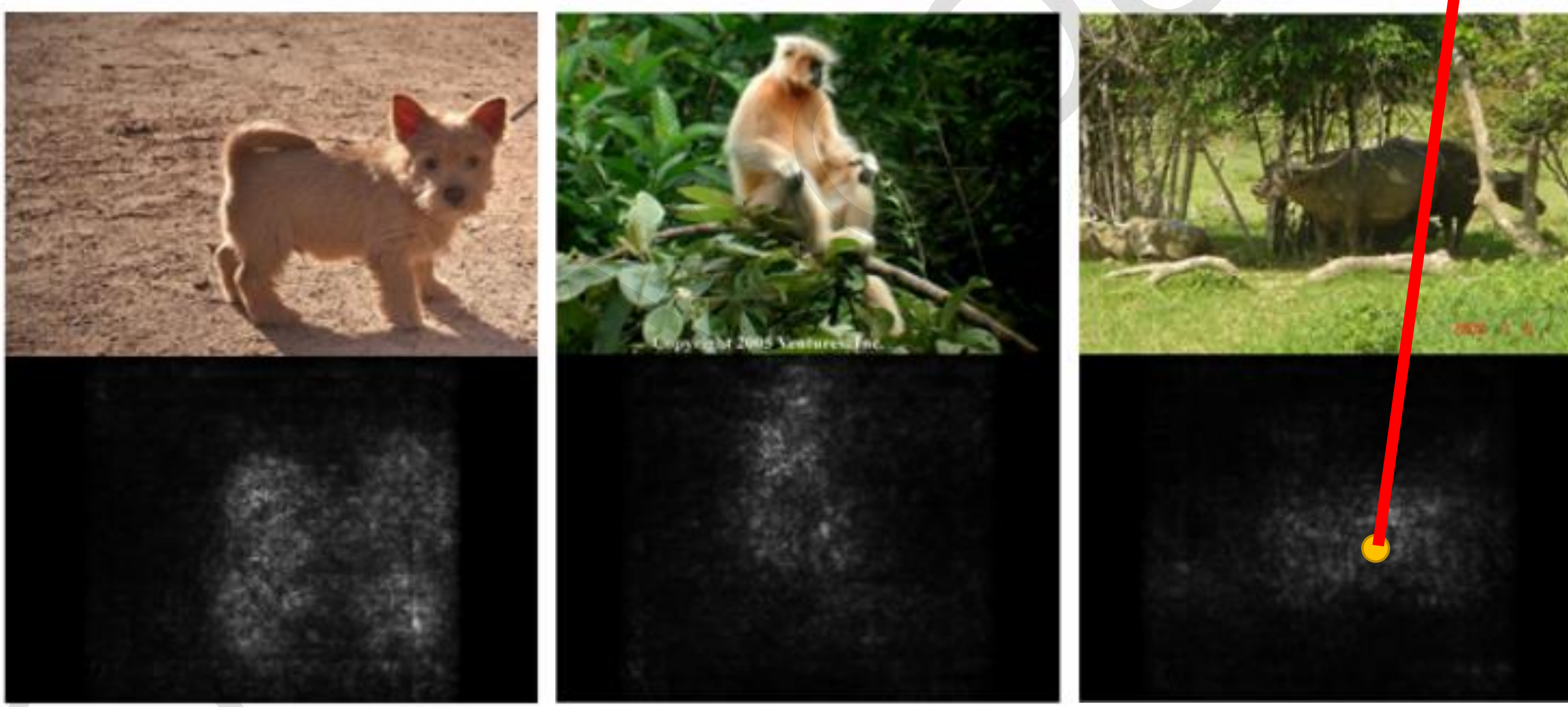
决策变化较大 ➡ 该部分很重要

深度学习的可解释性

$$\{x_1, \cdots, x_n, \cdots, x_N\} \longrightarrow \{x_1, \cdots, x_n + \Delta x, \cdots, x_N\}$$

$$y_k \longrightarrow y_k + \Delta y \qquad \left|\frac{\Delta y}{\Delta x}\right| \longrightarrow \left|\frac{\partial y_k}{\partial x_n}\right|$$

y_k: the prob of the predicted class of the model



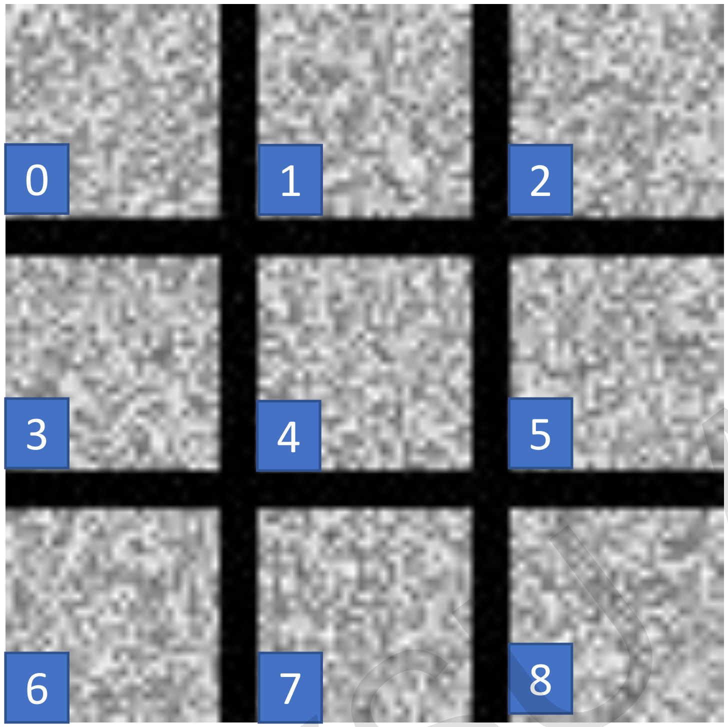
特征图

Global Explanation: Explain the whole Model

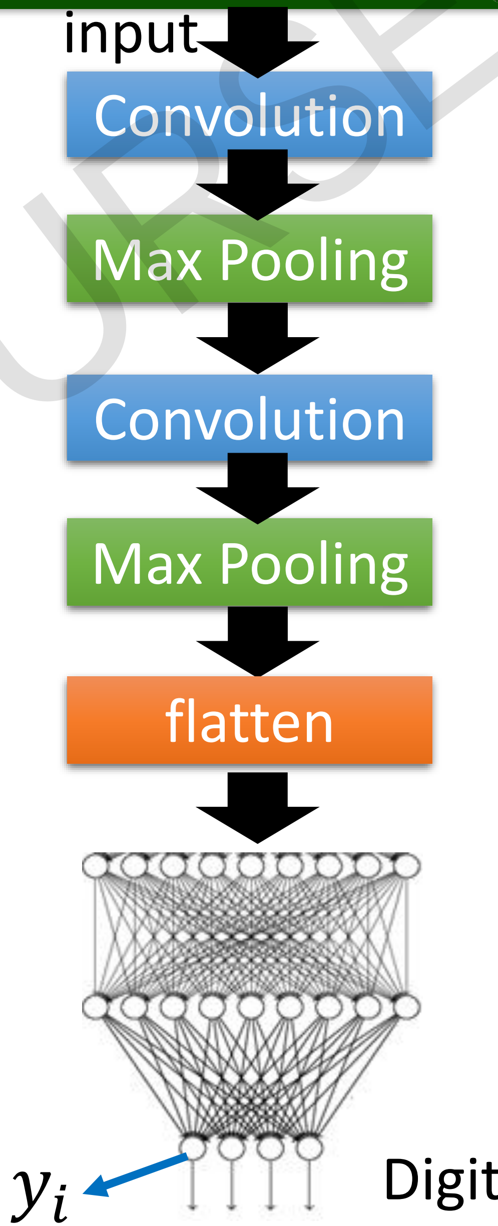
Question: What do you think a “cat” looks like?

深度学习的可解释性

$$x^* = arg \max_x y_i$$



我们找出让卷积为最容易辨识的各个类别的输入
（给定输出类别，卷积层最希望看到的输入）



Deep Neural Networks are Easily Fooled
https://www.youtube.com/watch?v=M2IebCN9Ht4

深度学习的可解释性

Deep Neural Networks are Easily Fooled

High Prediction Scores for Unrecognizable Images

来自生成器 (Generator) 的限制

Training a generator

Training Examples

(by GAN, VAE, etc.)

low-dim vector z → Image Generator G → Image x

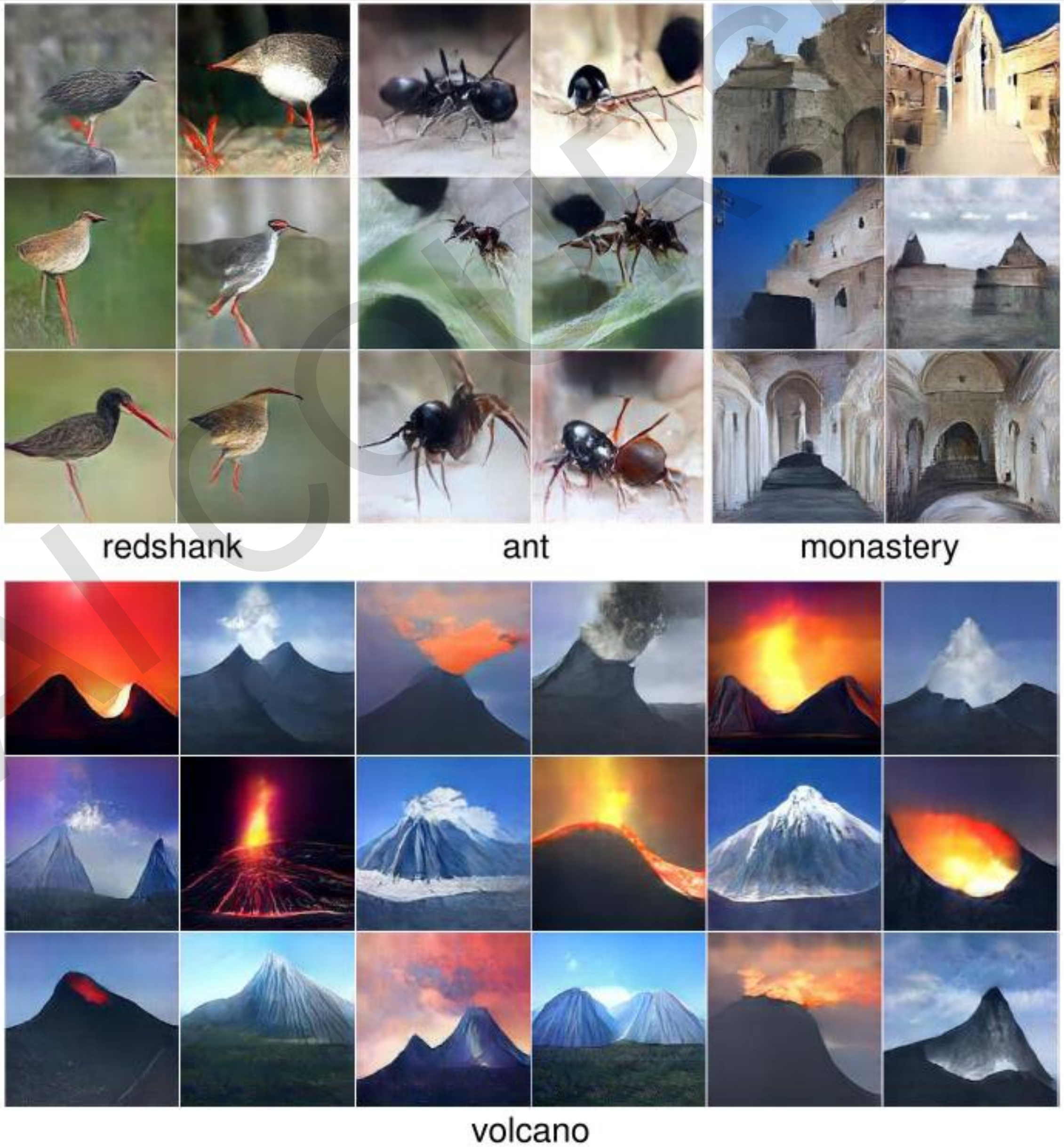
$x = G(z)$

$$x^* = arg\max_x y_i \Rightarrow z^* = arg\max_z y_i$$

$$x^* = G(z^*)$$

z → Image Generator → Image x → Image Classifier → y

深度学习的可解释性



https://arxiv.org/abs/1612.00005

Endless to explore ...

Thanks

Thanks the support from Prof. Yu, Prof. Lee, for some refers to their slides.