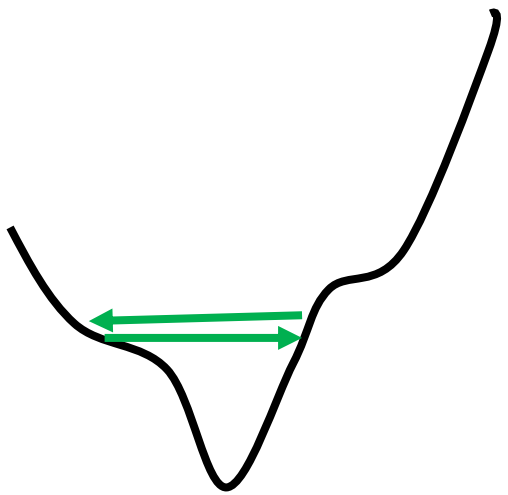
机器学习

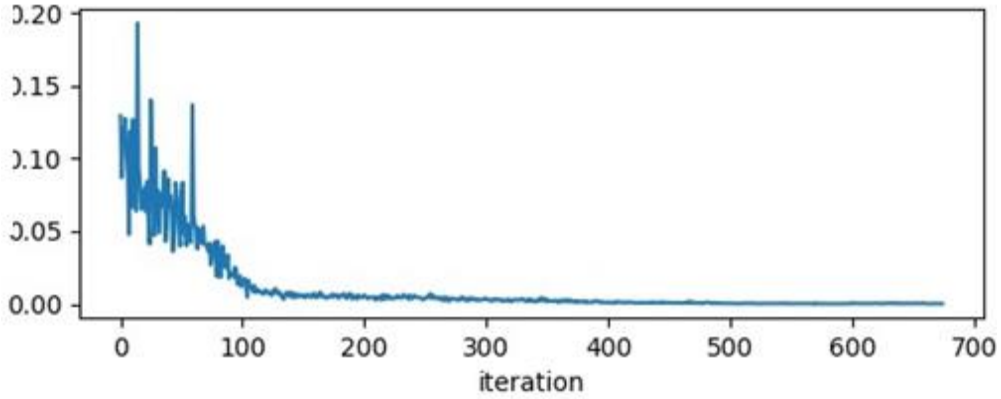
苏州大学计算机科学与技术学院
自然语言处理实验室
主讲：周夏冰
邮箱：zhouxiabing@suda.edu.cn

Learning rate

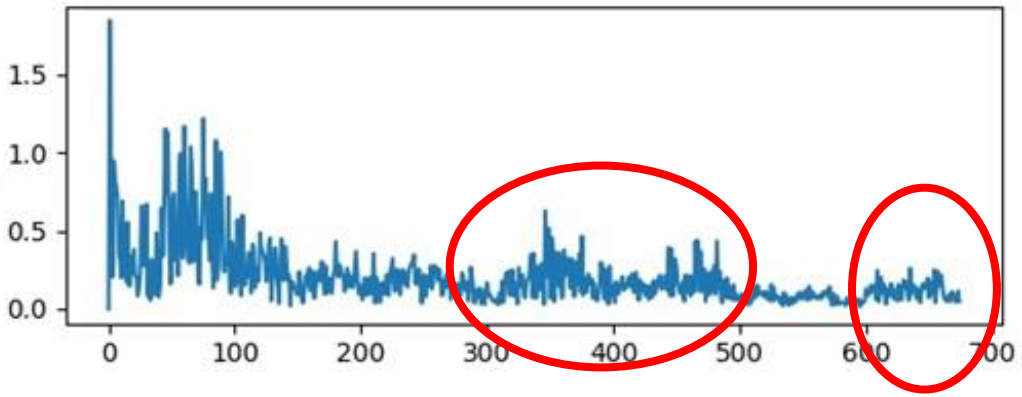
Training stuck ≠ Small Gradient

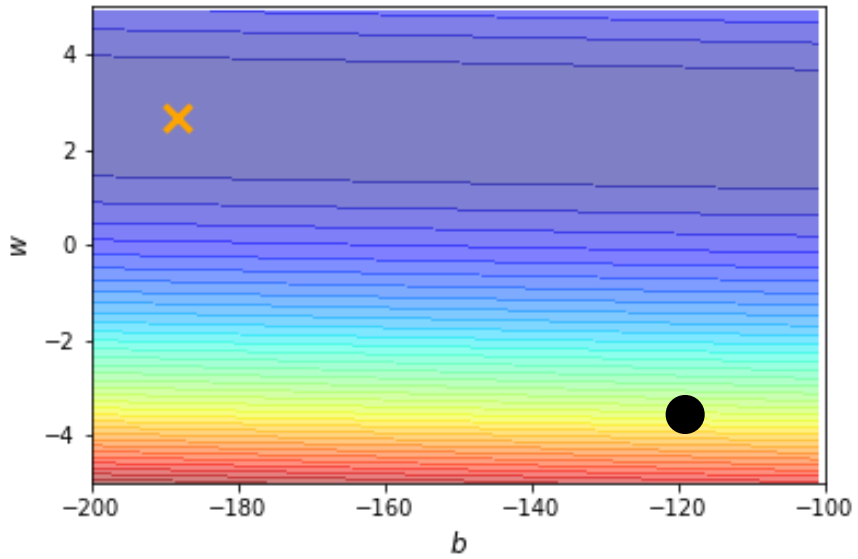


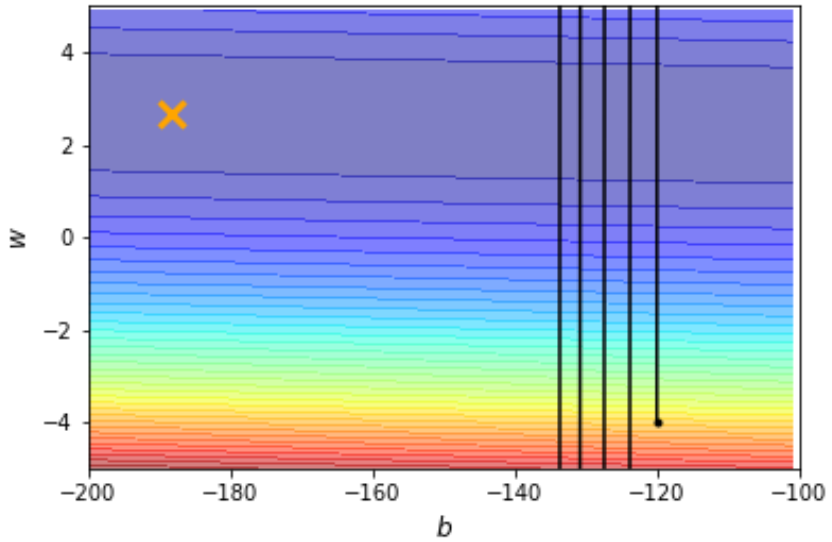
loss



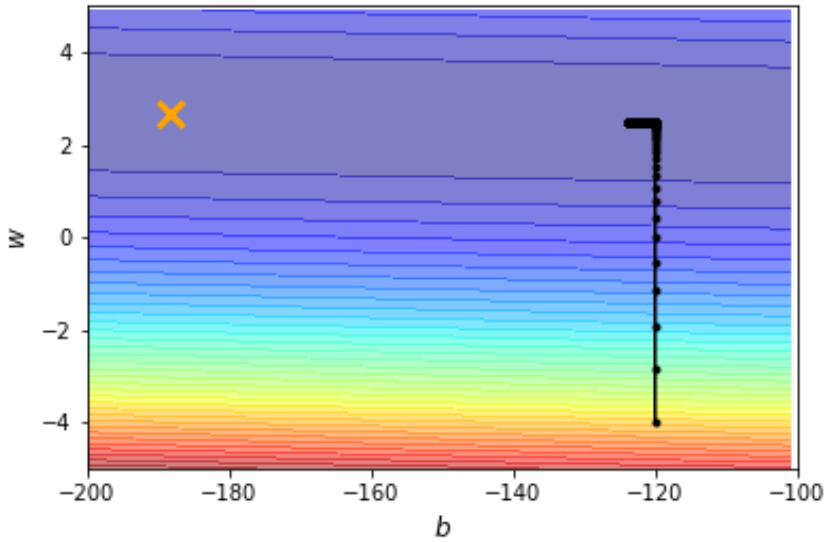
norm of gradient





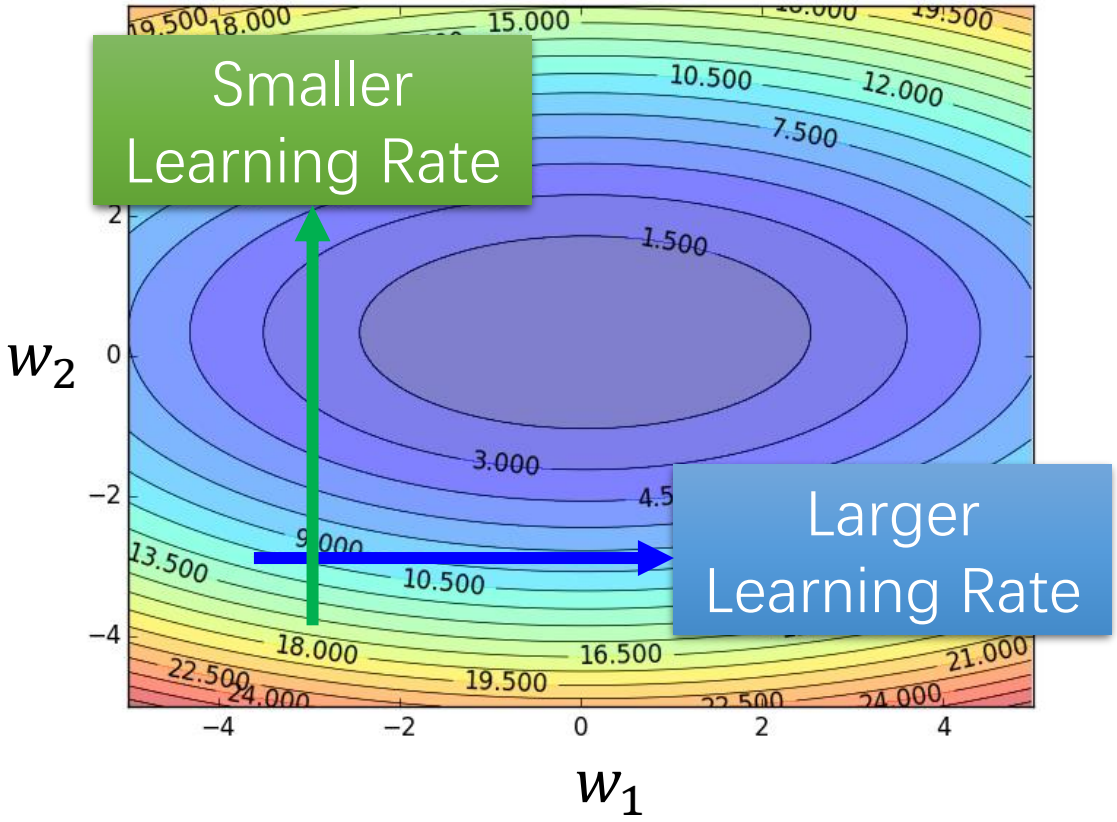


$\eta = 10^{-2}$



$\eta = 10^{-7}$

Learning rate



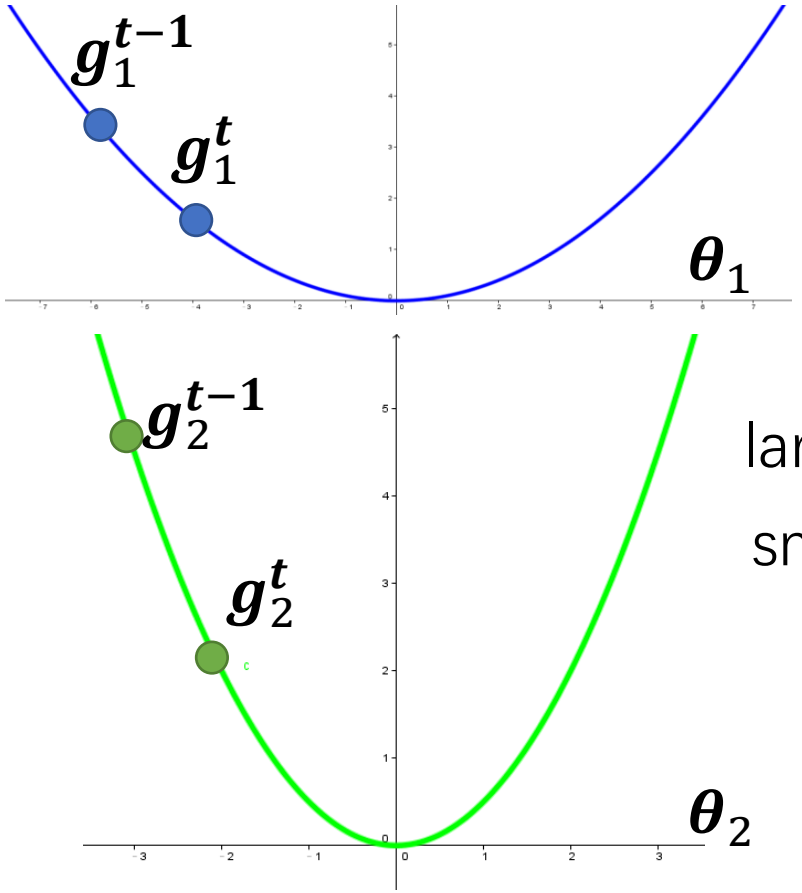
$$\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^t - \eta \boldsymbol{g}_i^t$$

$$\boldsymbol{g}_i^t = \frac{\partial L}{\partial \boldsymbol{\theta}_i}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^t}$$

$$\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^t - \frac{\eta}{\sigma_i^t} \boldsymbol{g}_i^t$$

Learning rate

- $\boldsymbol{\theta_i^{t+1} \leftarrow \theta_i^t - \frac{\eta}{\sigma_i^t} g_i^t}$
  - $\sigma_i^t = \sqrt{\frac{1}{t+1}\sum_{i=0}^{t}\left(g_i^t\right)^2}$
    - **Adagrad**



smaller σ_1^t

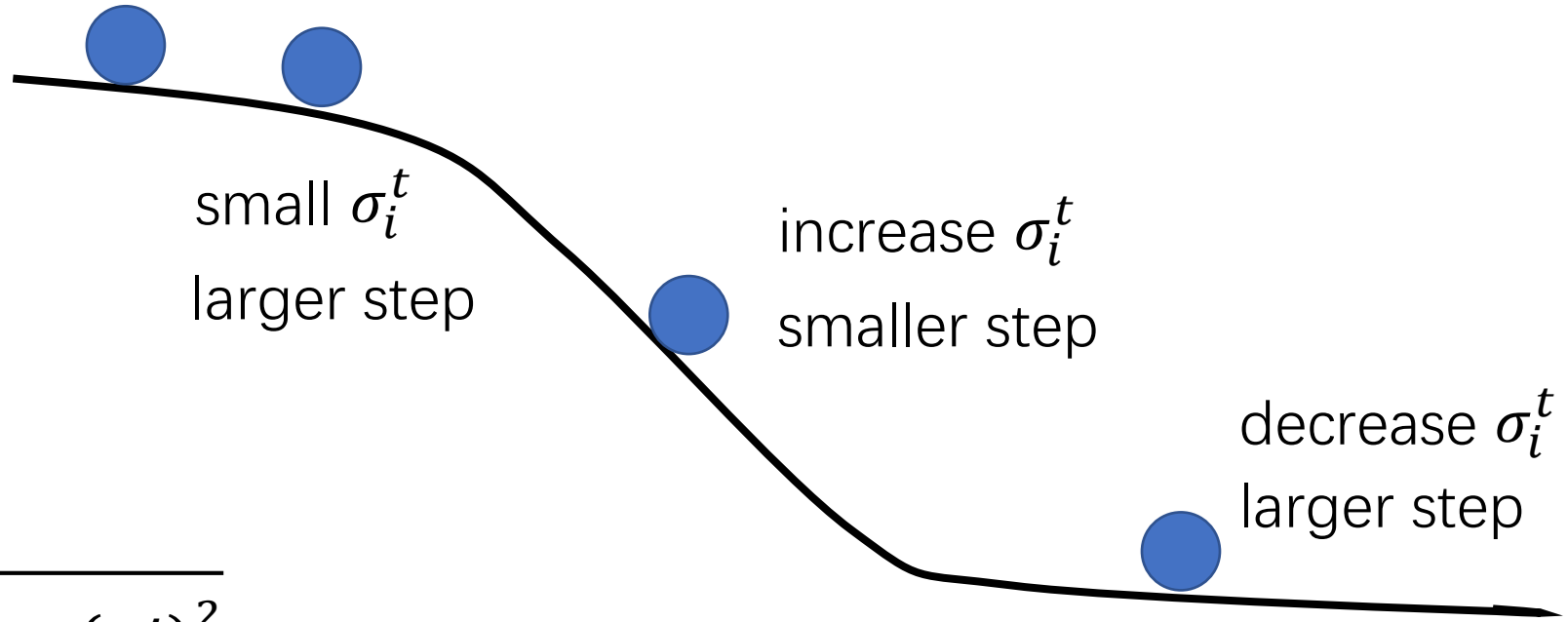
larger step

larger σ_2^t

smaller step

Learning rate

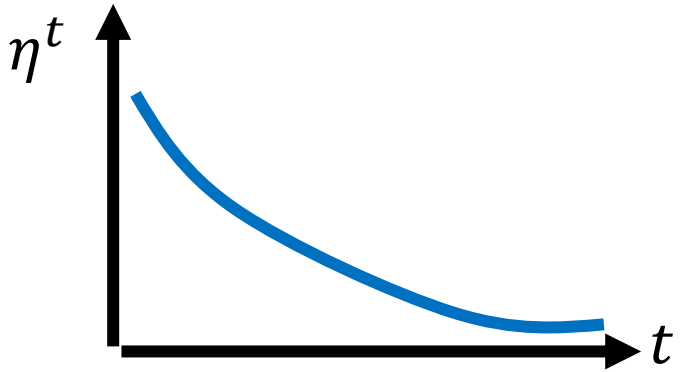
- $\boldsymbol{\theta_i^{t+1} \leftarrow \theta_i^t - \frac{\eta}{\sigma_i^t} g_i^t}$
  - $\sigma_i^t = \sqrt{\frac{1}{t+1}\sum_{i=0}^{t}\left(g_i^t\right)^2}$
    - **Adagrad**
  - $\sigma_i^t = \sqrt{\alpha\left(\sigma_i^{t-1}\right)^2 + (1-\alpha)\left(g_i^t\right)^2}$
    - **RMSProp**



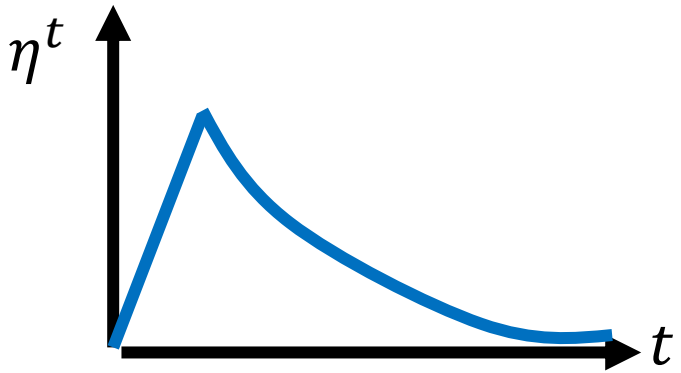
Adam:结合了动量梯度下降和RMSProp两种算法的优点

Learning rate

$$\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^t - \frac{\eta}{\sigma_i^t}\boldsymbol{g}_i^t$$



Learning Rate Decay



Warm Up

Summary of Optimization

(Vanilla) Gradient Descent

$$\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^t - \eta \boldsymbol{g}_i^t$$

Various Improvements

$$\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^t - \frac{\eta^t}{\sigma_i^t} \boldsymbol{m}_i^t$$

Learning rate scheduling

Momentum: weighted sum of the previous gradients

确定方向

root mean square of the gradients

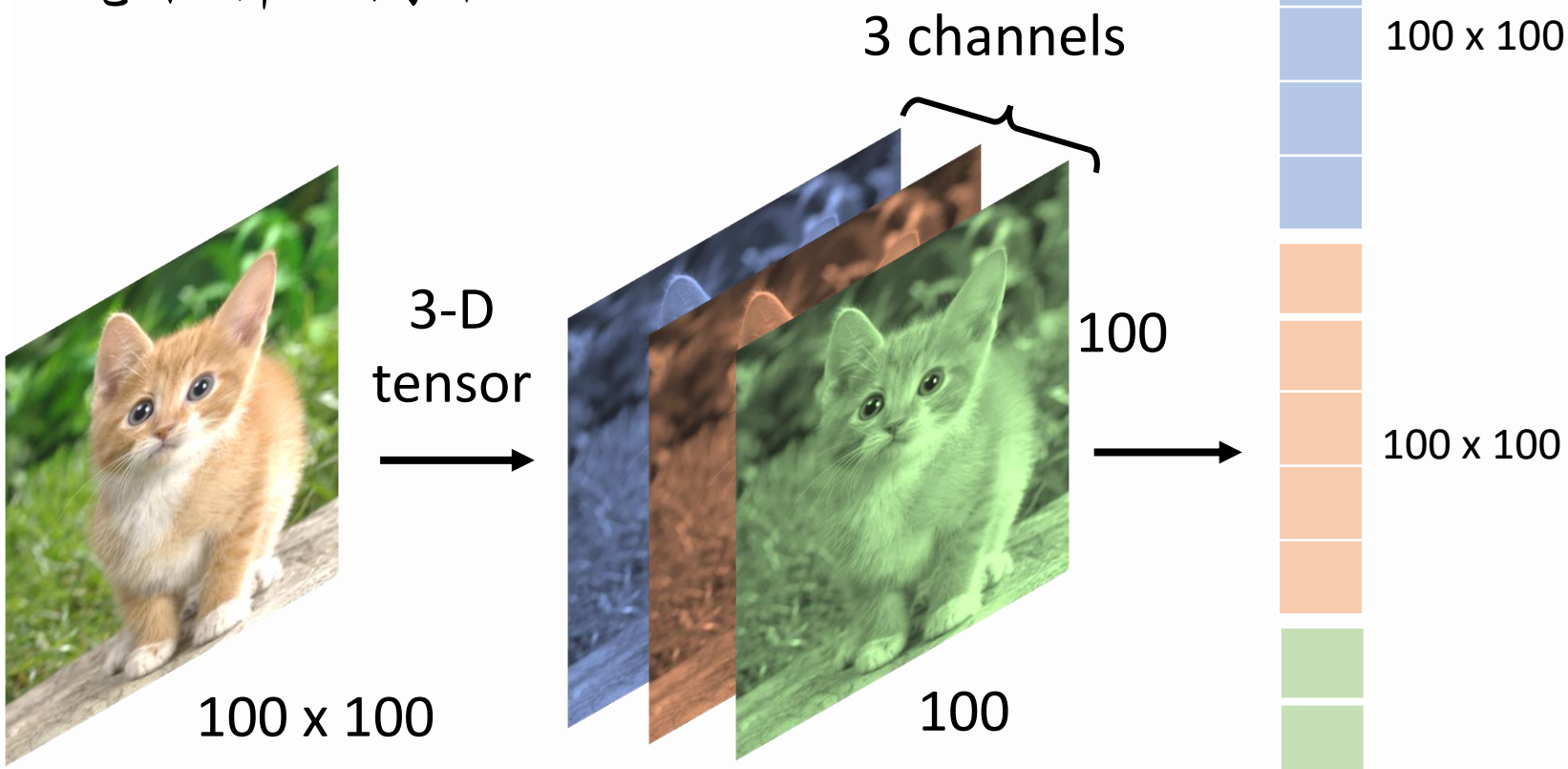
确定大小

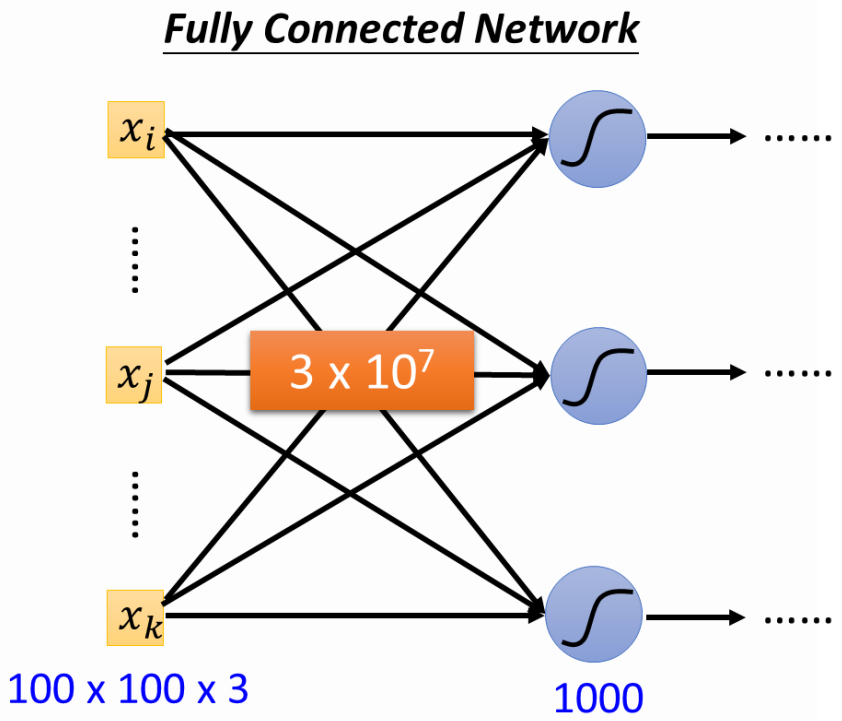
02

卷积神经网络
CNN

CNN

- 卷积神经网络

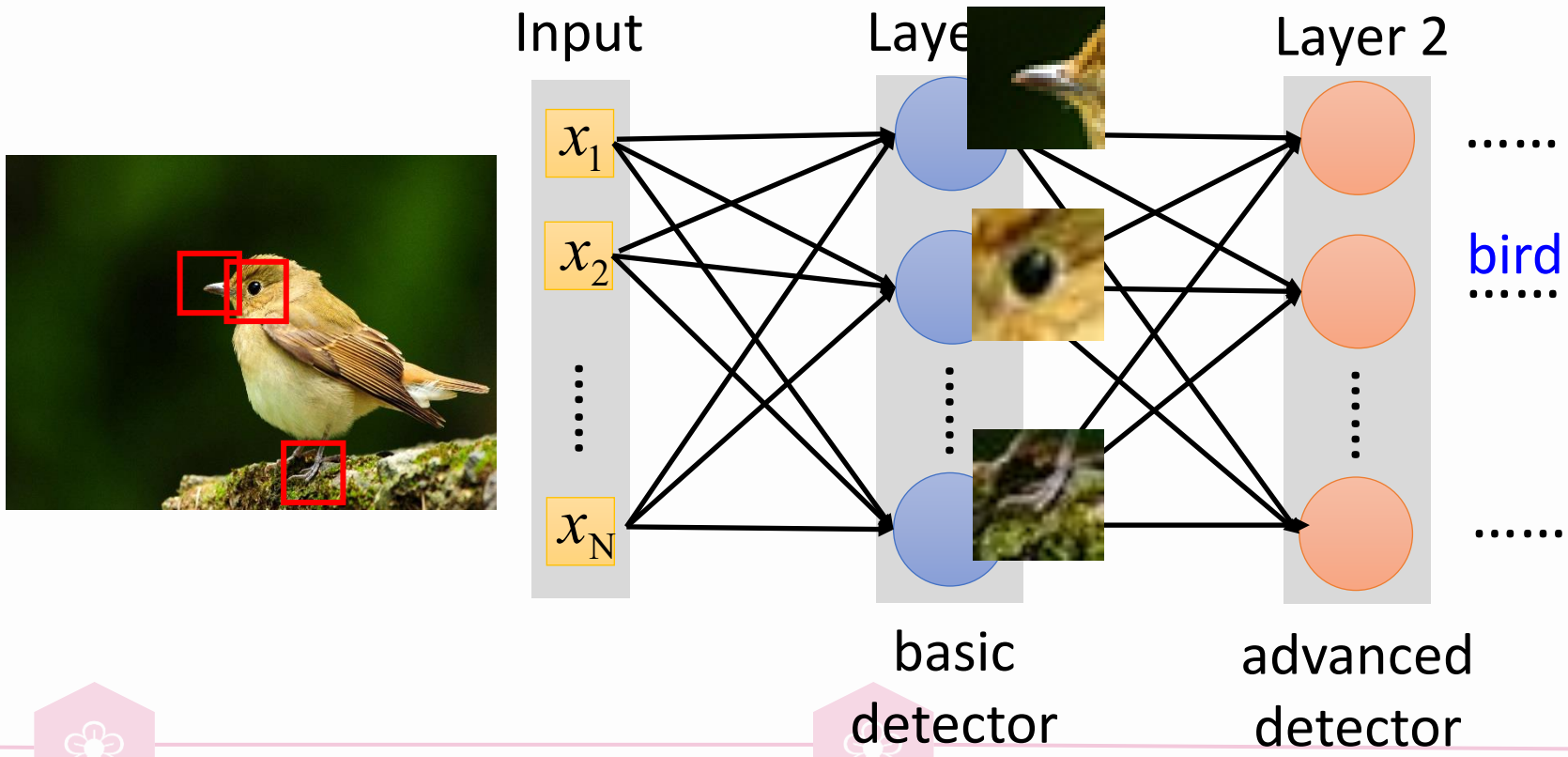

3 channels
3-D
tensor
100
100 x 100
100
100 x 100
100 x 100
100 x 100


Fully Connected Network
x_i
x_j
x_k
3 x 10^7
100 x 100 x 3
1000

CNN

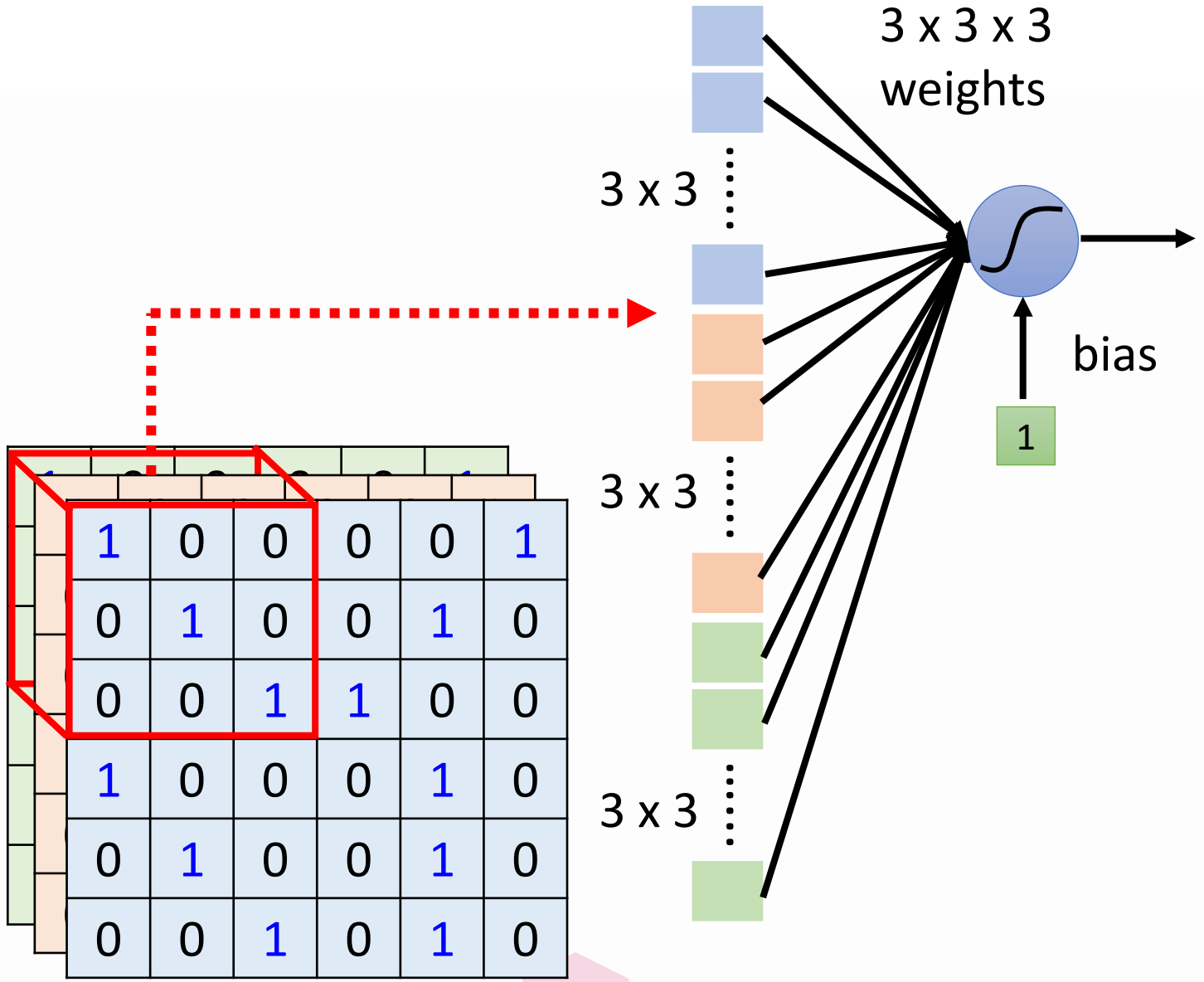
- 关键的模式只占整个图片的很小的区域

对于神经元，它不需要观测整个图像去识别pattern

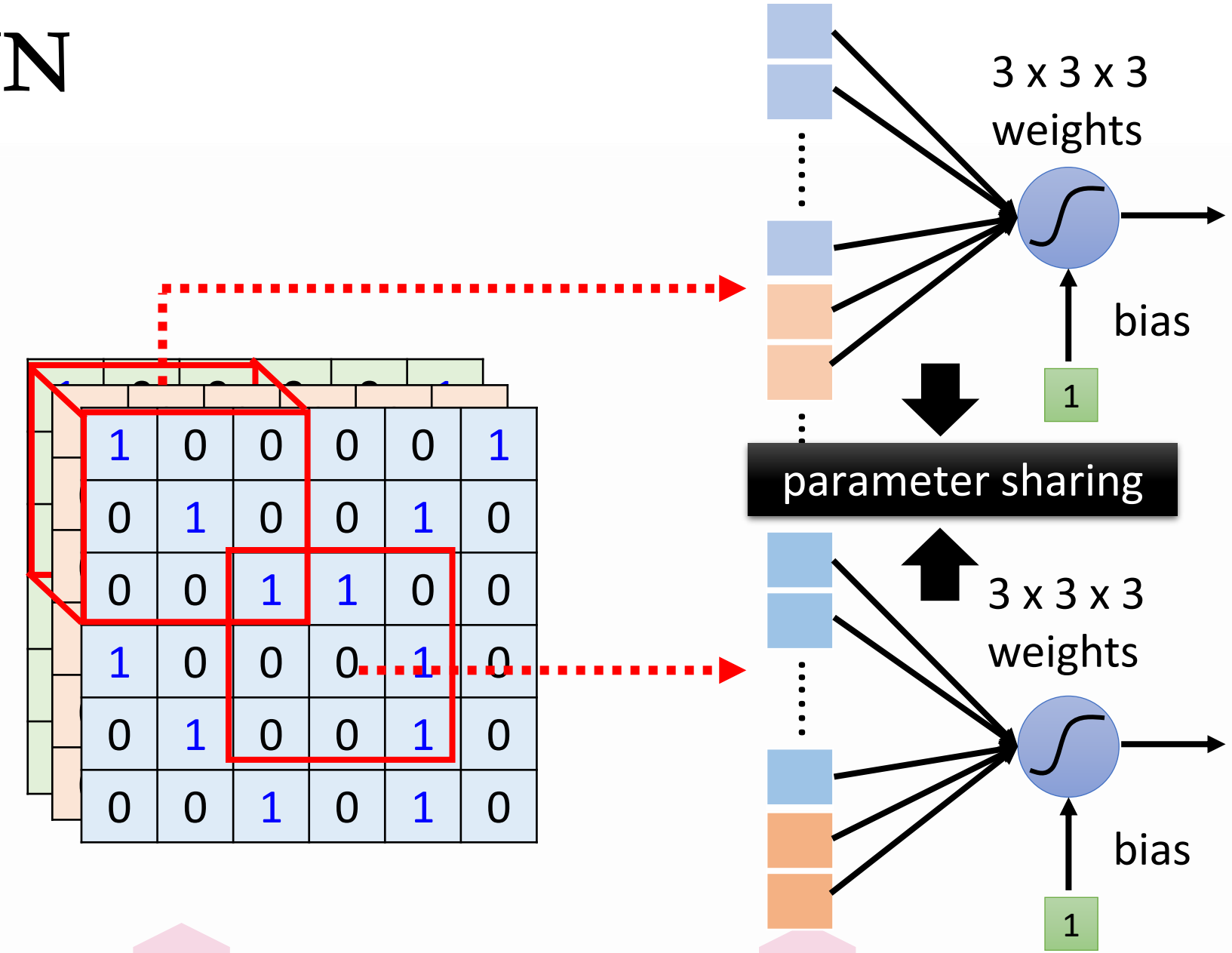


CNN

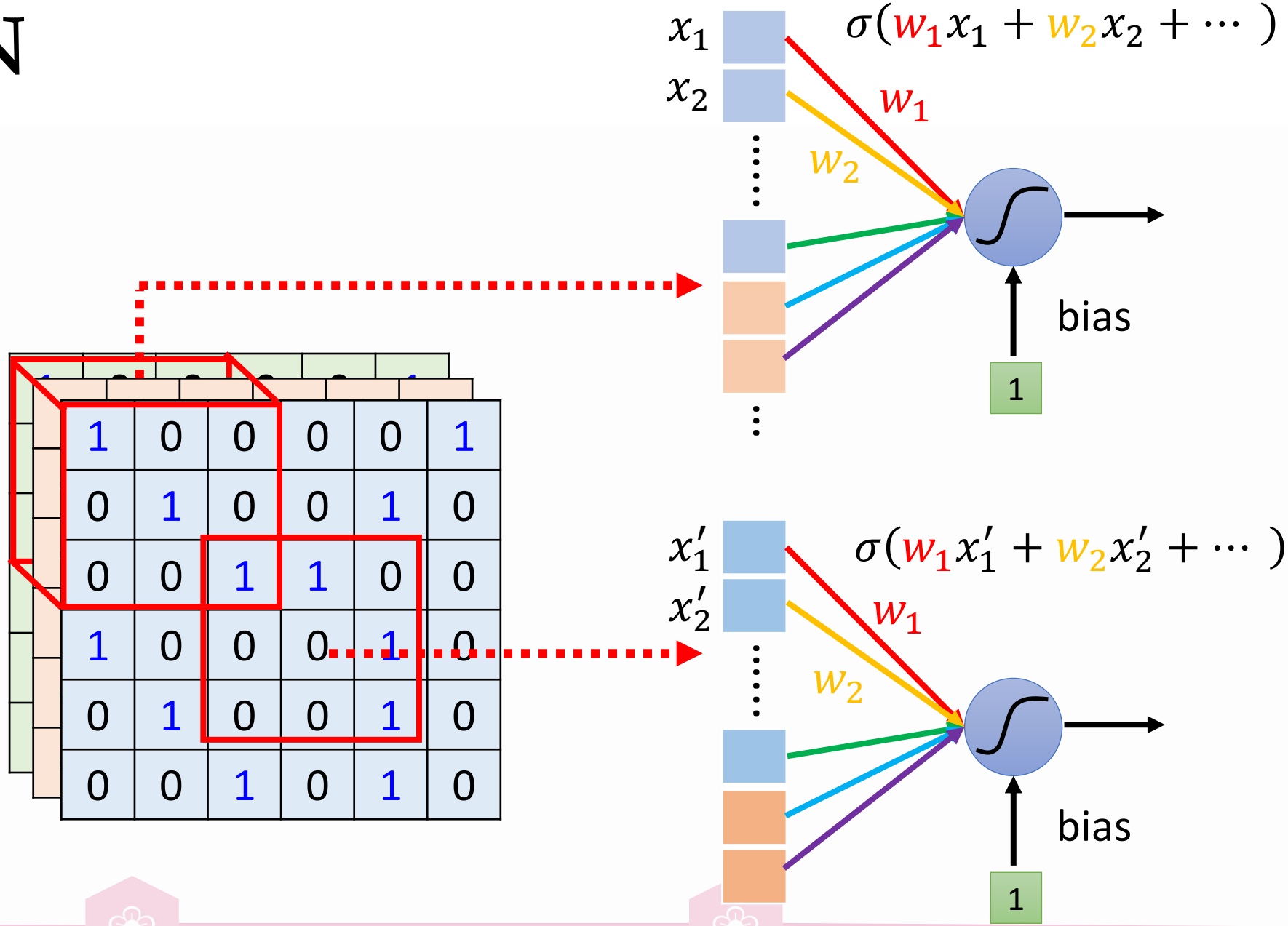
感受野
Receptive field



CNN


3 x 3 x 3
weights
bias
1
parameter sharing
3 x 3 x 3
weights
bias
1

CNN


x_1
x_2
$\sigma(w_1x_1 + w_2x_2 + \cdots)$
w_1
w_2
bias
1
x_1'
x_2'
$\sigma(w_1x_1' + w_2x_2' + \cdots)$
w_1
w_2
bias
1

CNN

- 一定的压缩不影响识别

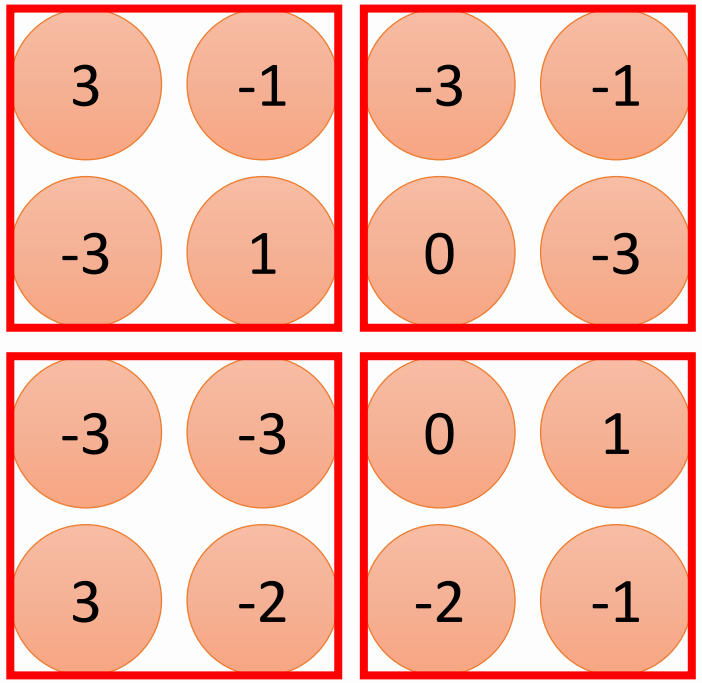
bird

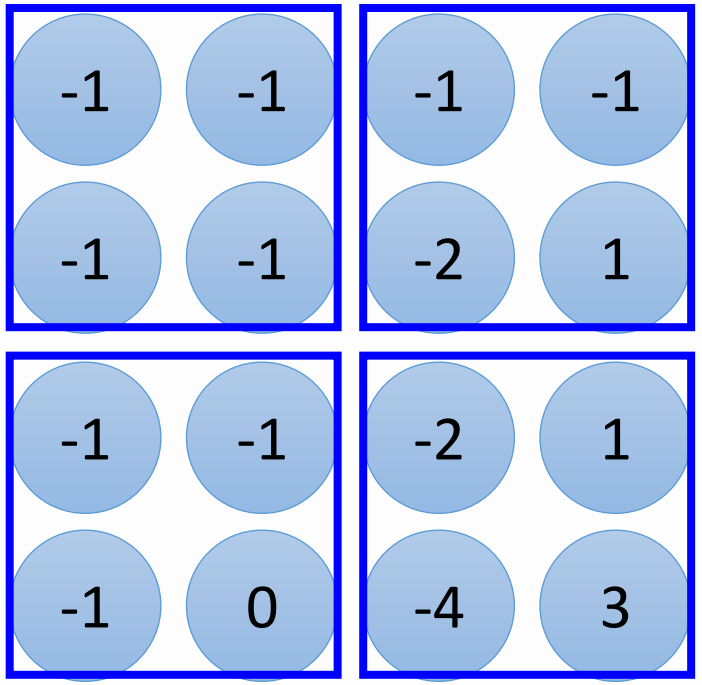
subsampling

bird

CNN

- 池化层
  - 最大池化
  - 平均池化





CNN

Property 1

- Some patterns are much smaller than the whole image

Property 2

- The same patterns appear in different regions.

Property 3

- Subsampling the pixels will not change the object

Convolution

Max Pooling

Convolution

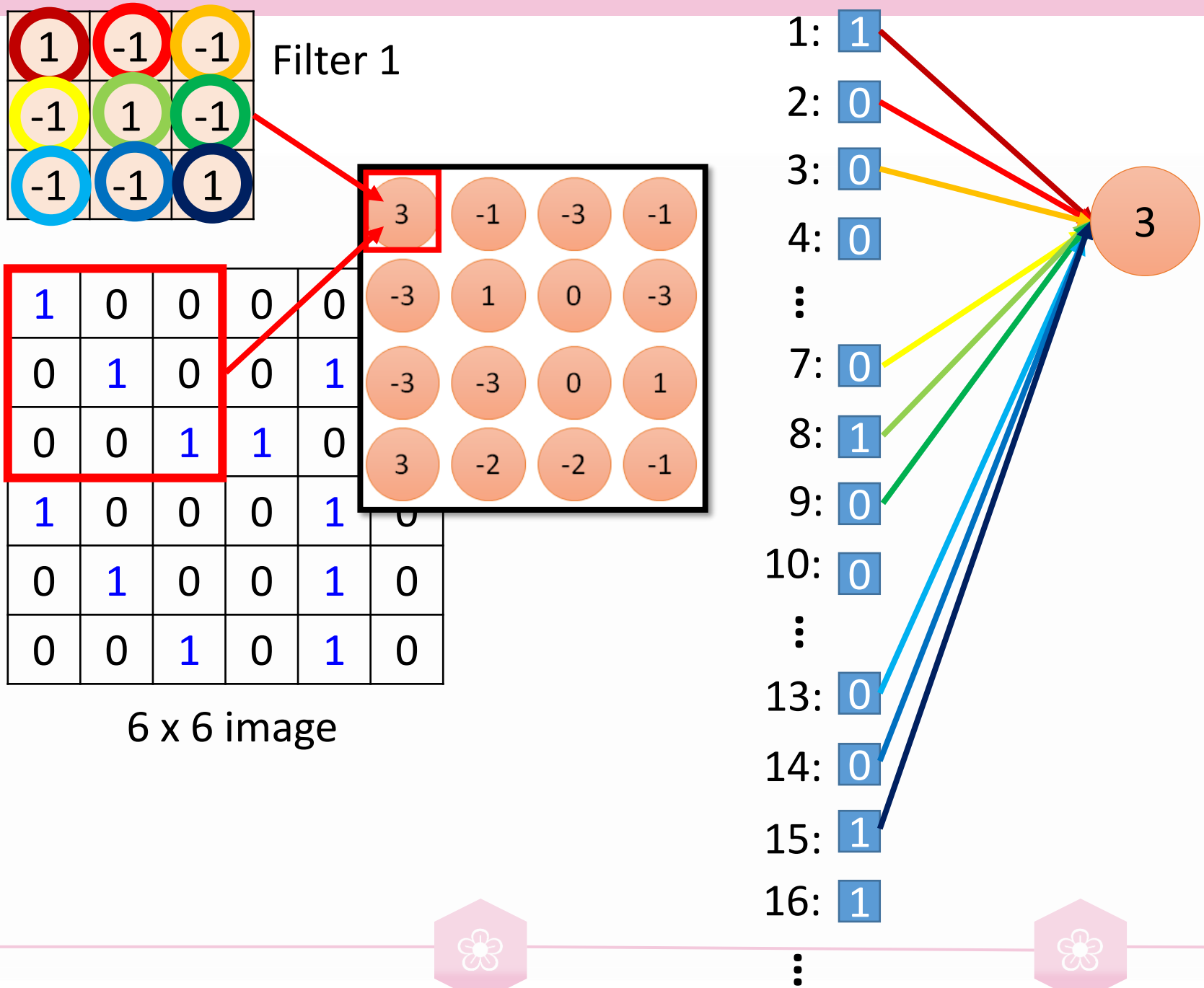
Max Pooling

Can repeat many times

Flatten

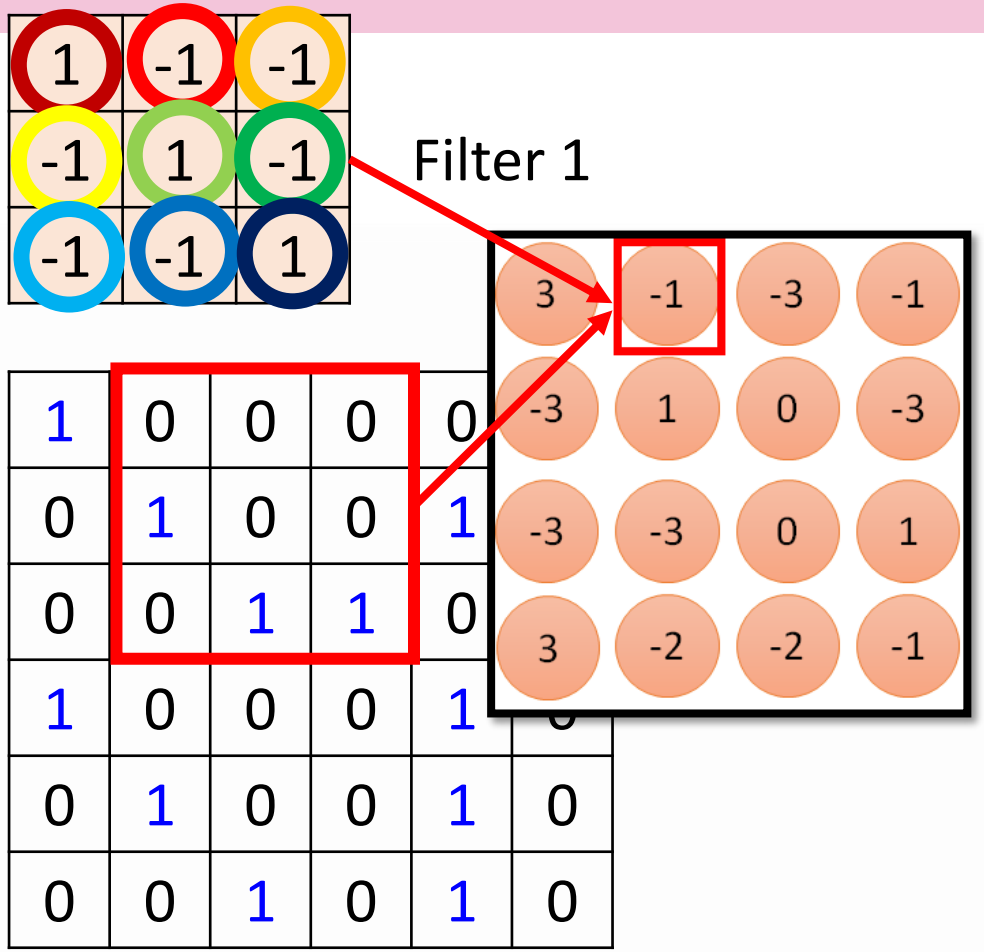
CNN

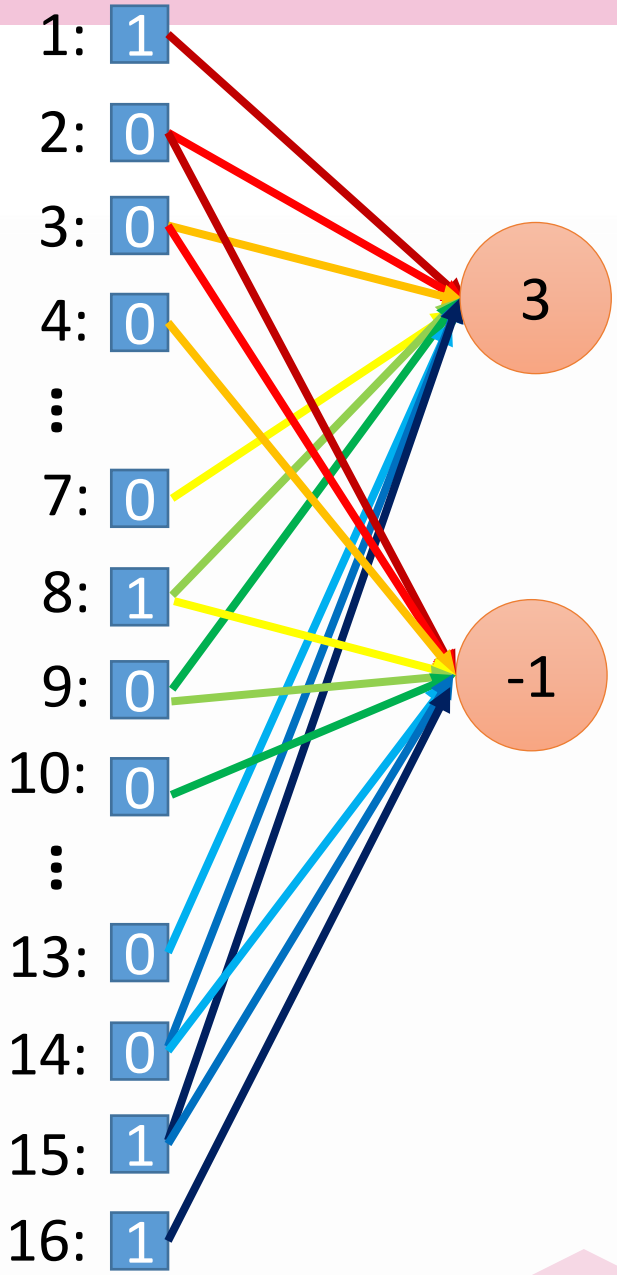
- 卷积层


1 -1 -1
-1 1 -1
-1 -1 1
Filter 1
3 -1 -3 -1
-3 1 0 -3
-3 -3 0 1
3 -2 -2 -1
1 0 0 0 0
0 1 0 0 1
0 0 1 1 0
1 0 0 0 1
0 1 0 0 1 0
0 0 1 0 1 0
6 x 6 image
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

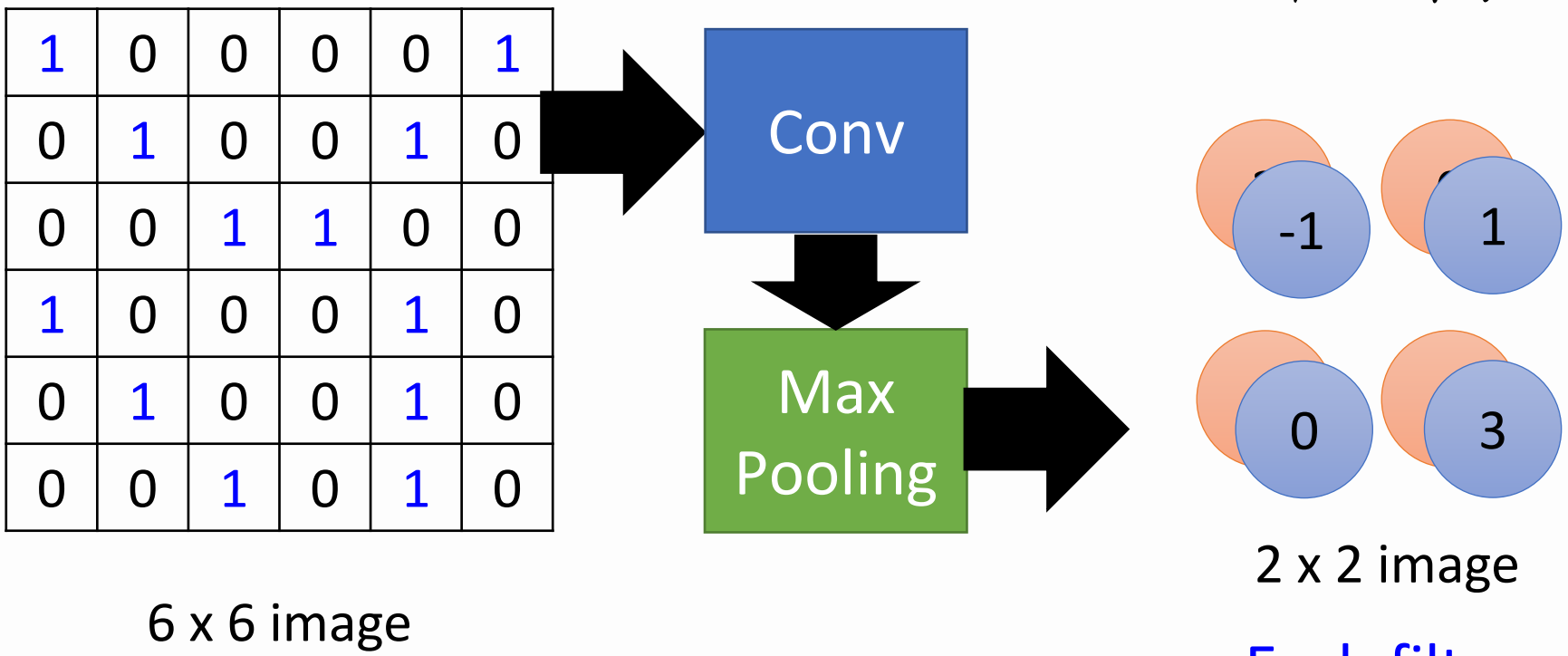
CNN

- 卷积层


1 -1 -1
-1 1 -1
-1 -1 1
Filter 1
3 -1 -3 -1
-3 1 0 -3
-3 -3 0 1
3 -2 -2 -1
1 0 0 0 0
0 1 0 0 1
0 0 1 1 0
1 0 0 0 1
0 1 0 0 1 0
0 0 1 0 1 0
6 x 6 image

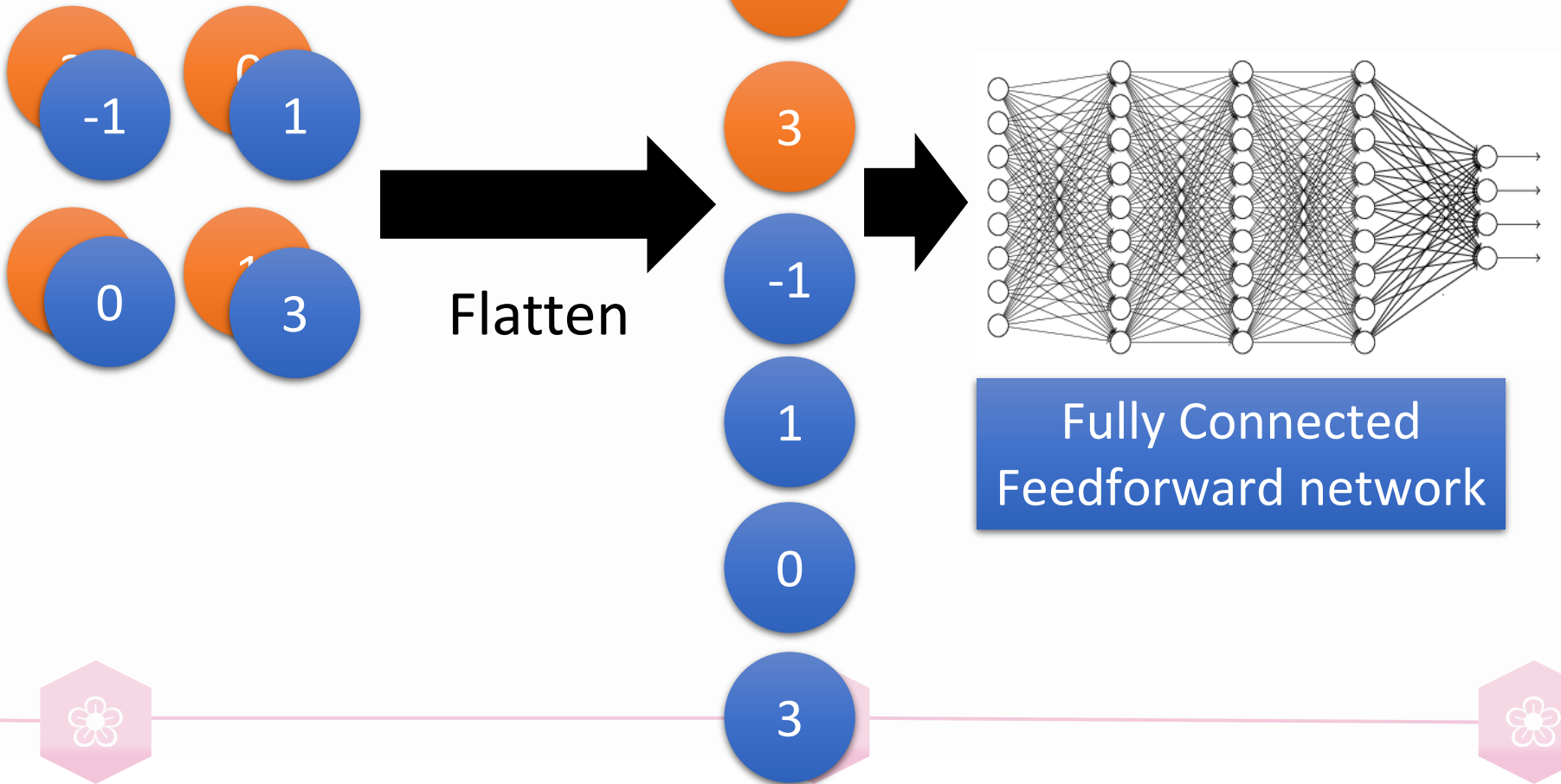

1: 1
2: 0
3: 0
4: 0
7: 0
8: 1
9: 0
10: 0
13: 0
14: 0
15: 1
16: 1
3
-1

CNN

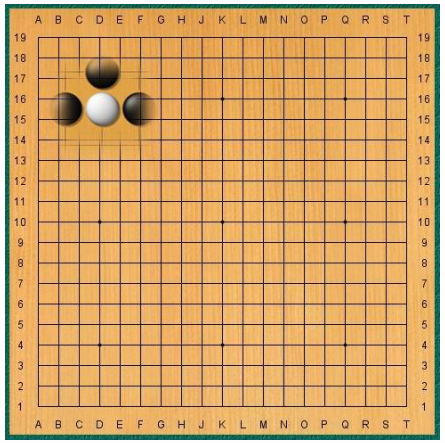

新的图像
1 0 0 0 0 1
0 1 0 0 1 0
0 0 1 1 0 0
1 0 0 0 1 0
0 1 0 0 1 0
0 0 1 0 1 0
Conv
Max Pooling
-1
1
0
3
6 x 6 image
2 x 2 image
Each filter is a channel

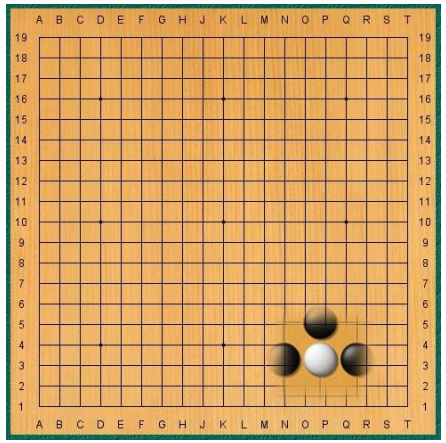
CNN

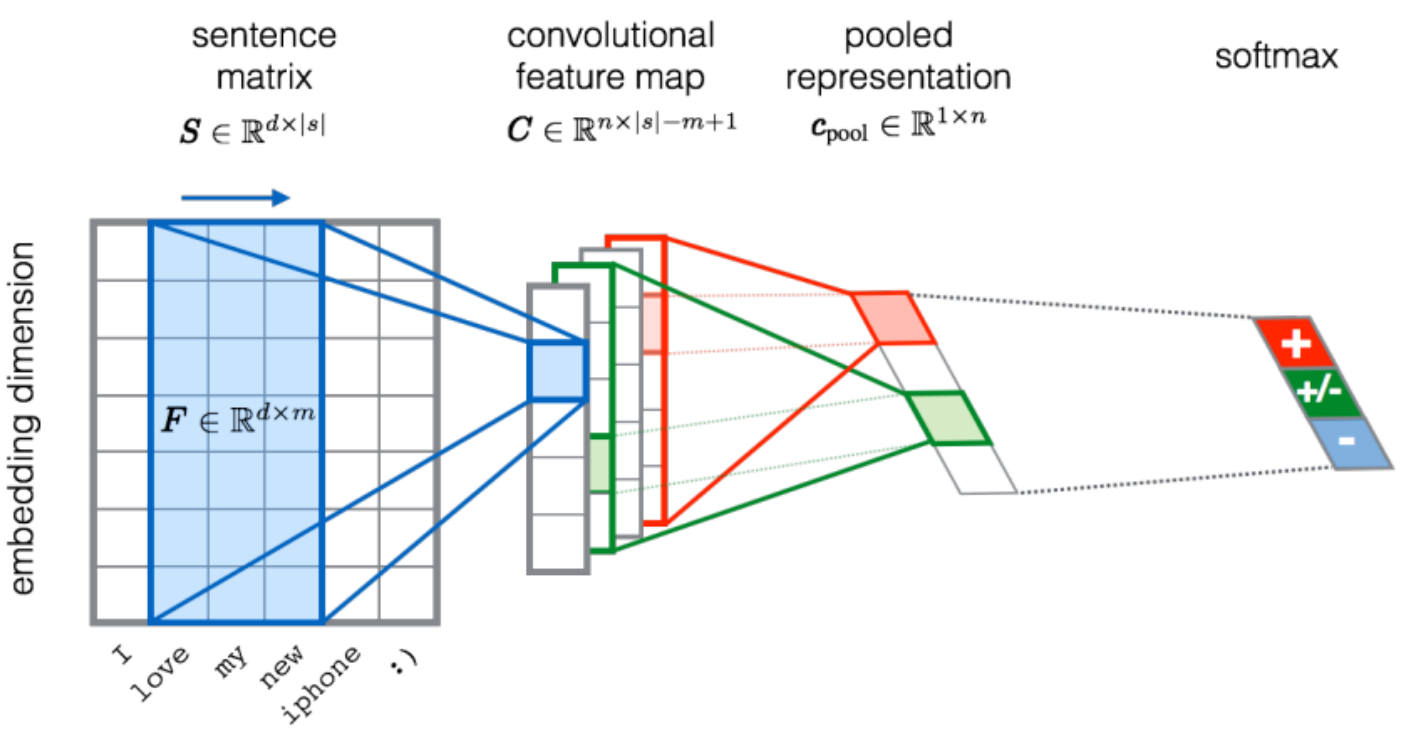
- faltten


-1
1
0
3
Flatten
3
0
1
3
-1
1
0
3
Fully Connected
Feedforward network

CNN





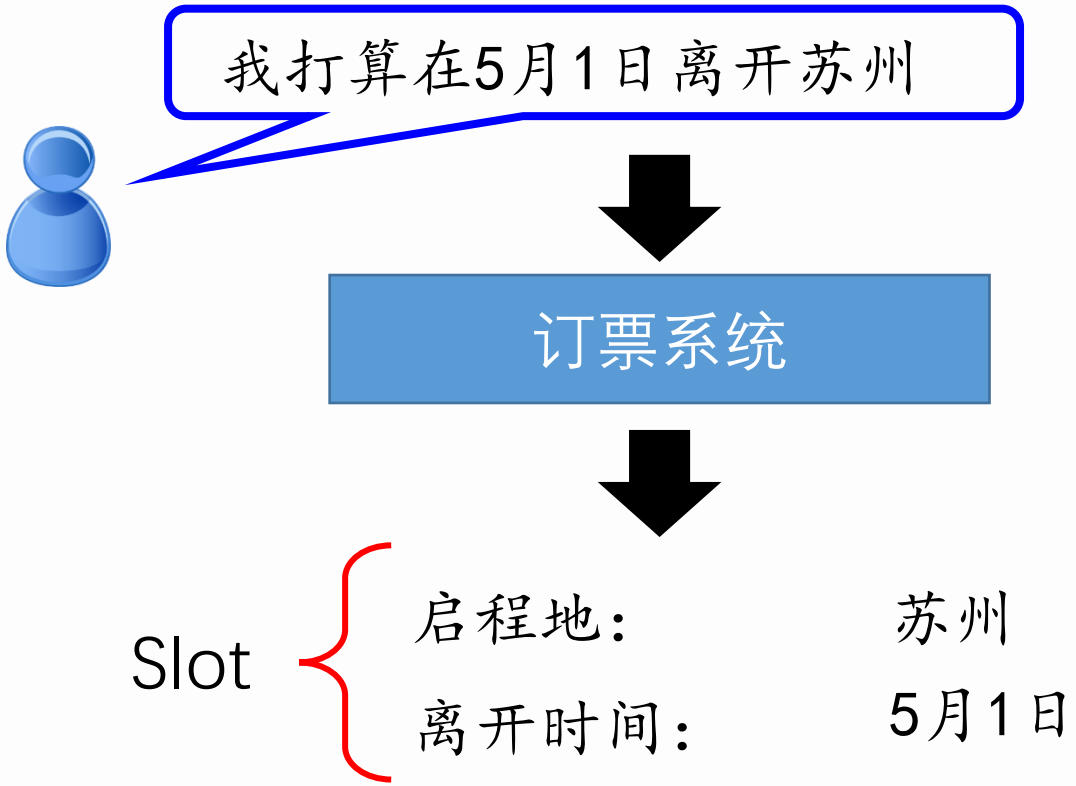

sentence matrix
$S \in \mathbb{R}^{d \times |s|}$
convolutional feature map
$C \in \mathbb{R}^{n \times |s|-m+1}$
pooled representation
$c_{\text{pool}} \in \mathbb{R}^{1 \times n}$
softmax
embedding dimension
$F \in \mathbb{R}^{d \times m}$
I love my new iphone :)
+
+/-
-

03

循环神经网络
RNN&LSTM

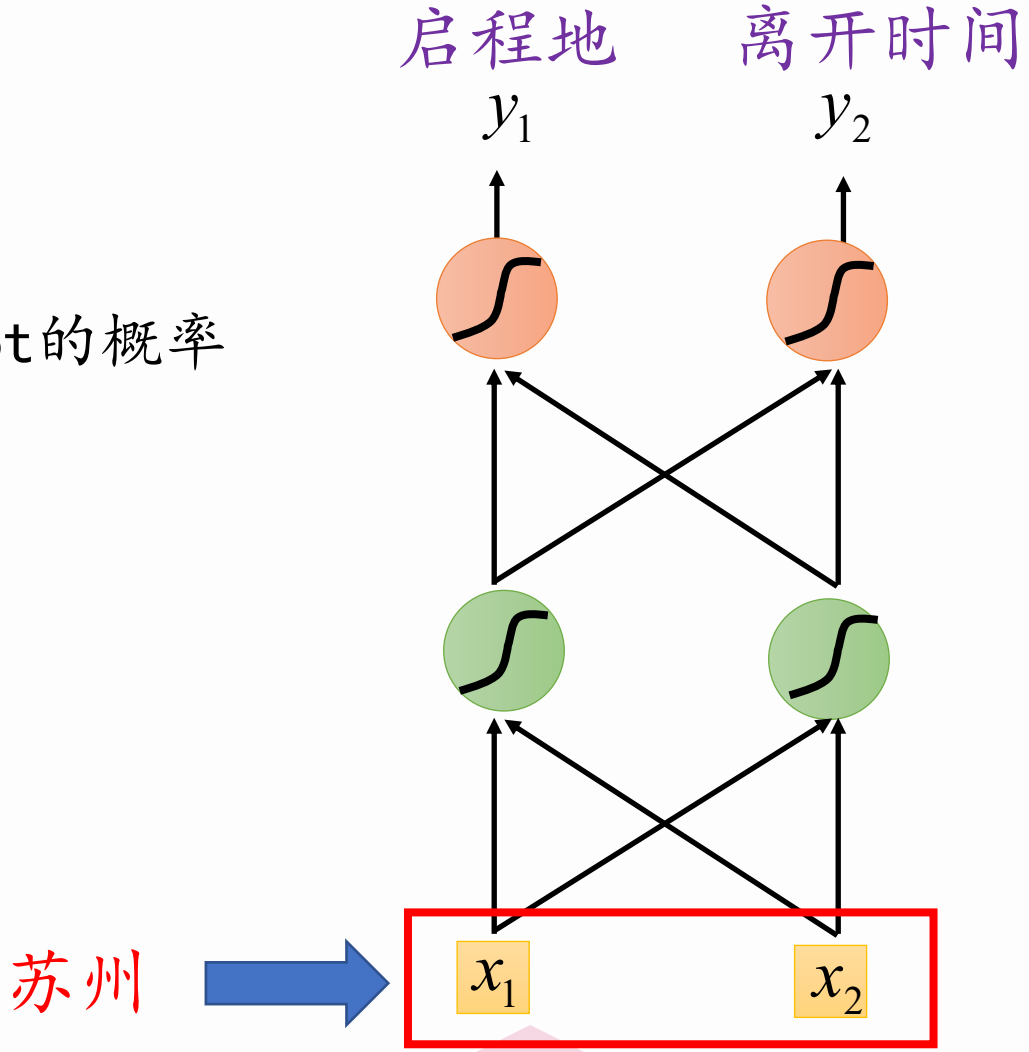
一个小例子

- Slot Filling

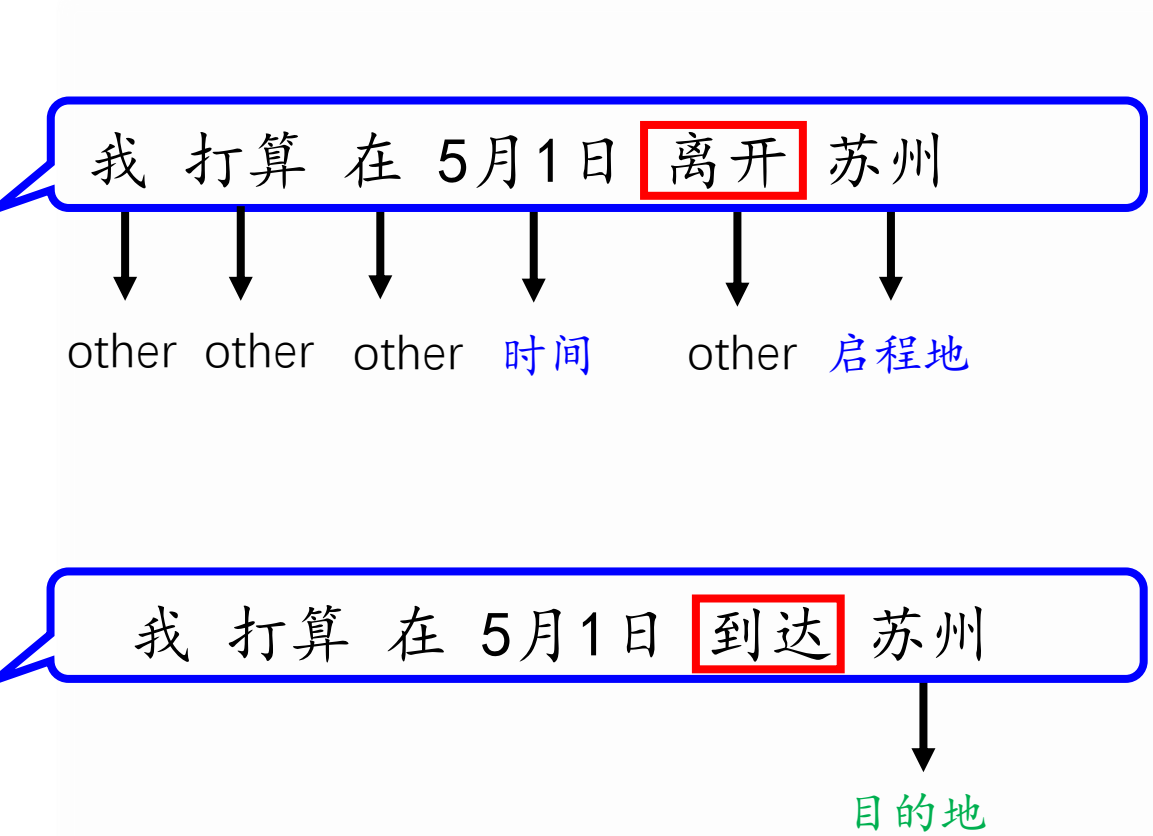


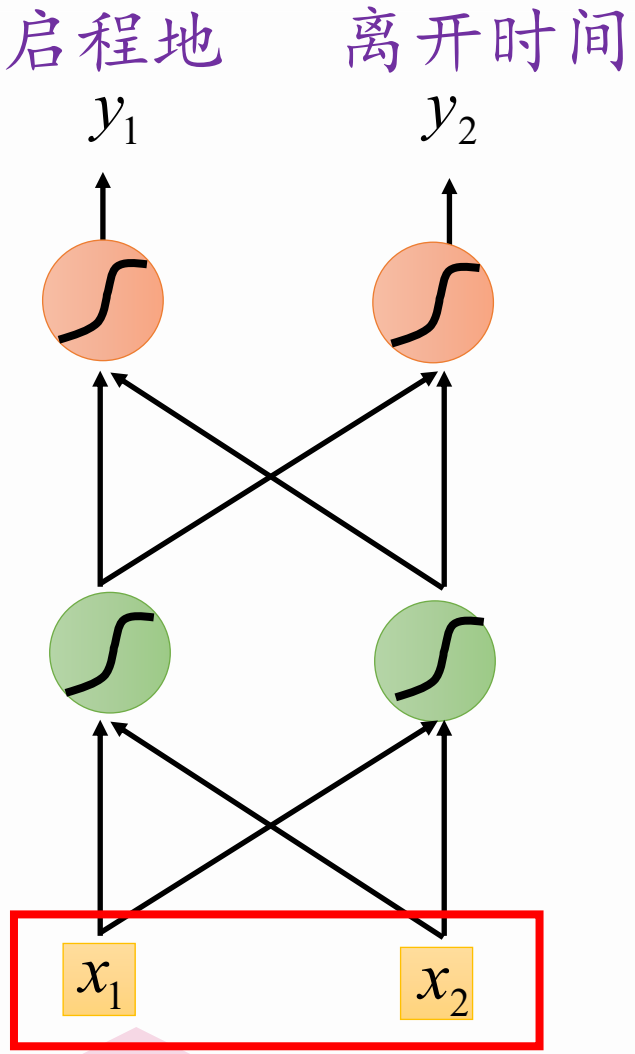
一个小例子

- 采用前馈神经网络
  - 输入词向量
  - 输出：隶属不同slot的概率



一个小例子



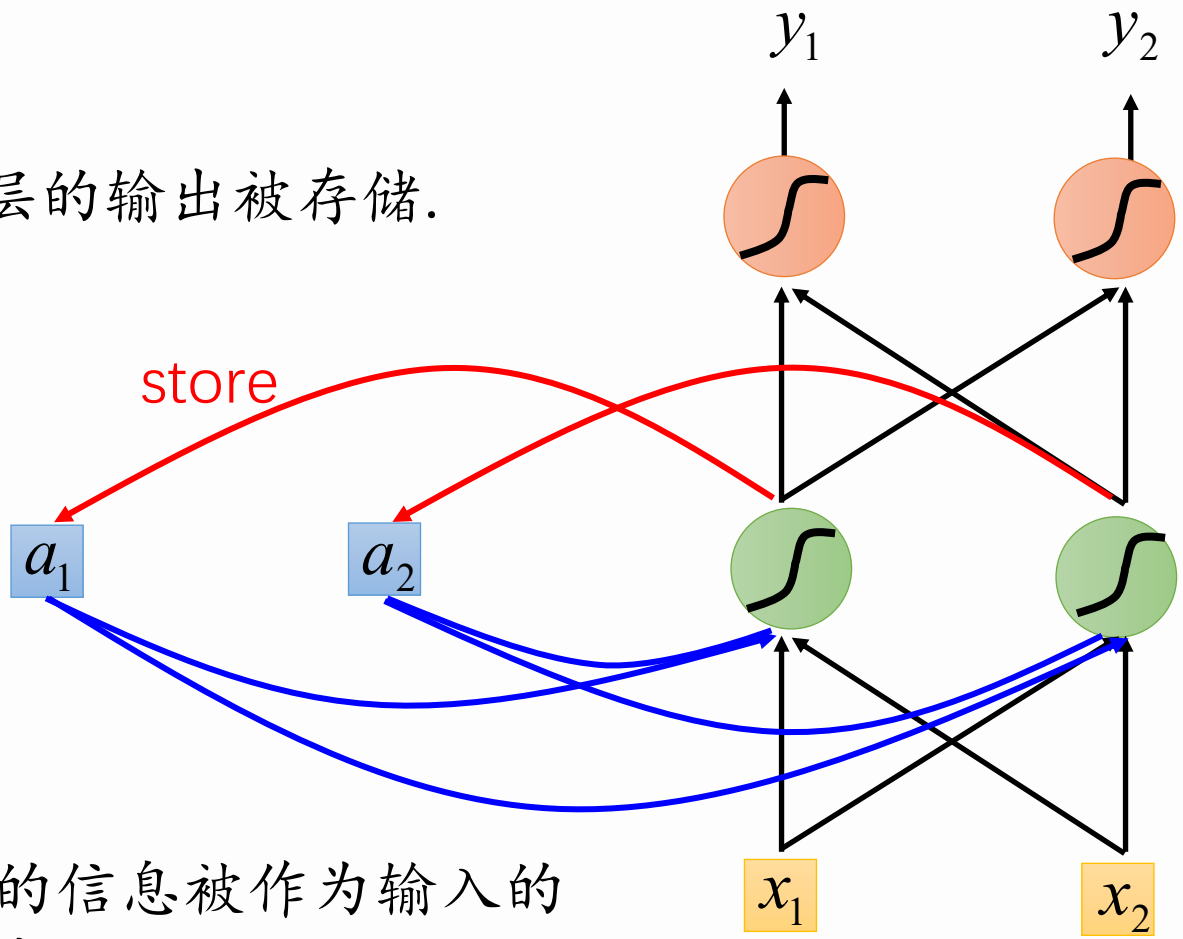


苏州

Neural network needs memory!

循环神经网络RNN

隐层的输出被存储.



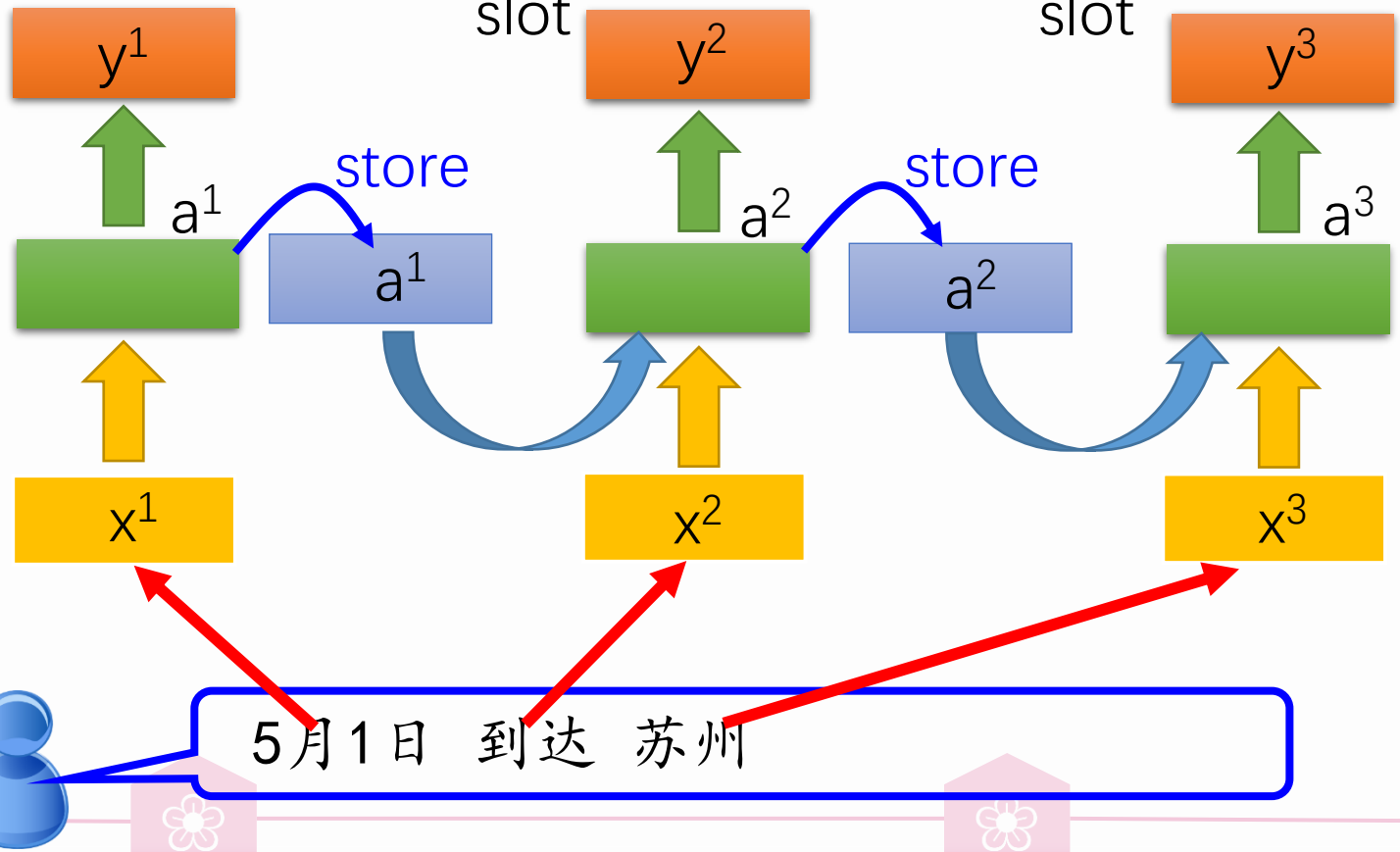
存储的信息被作为输入的一部分

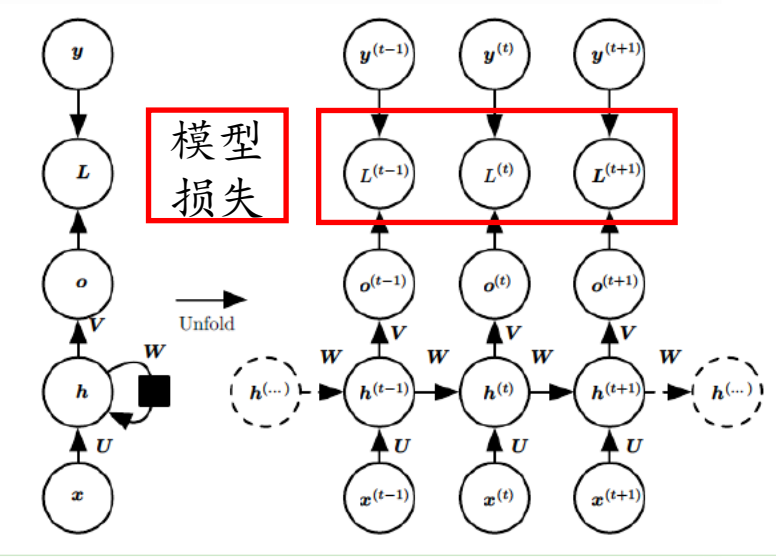
RNN

Probability of “time” in each slot

Probability of “Arrive” in each slot

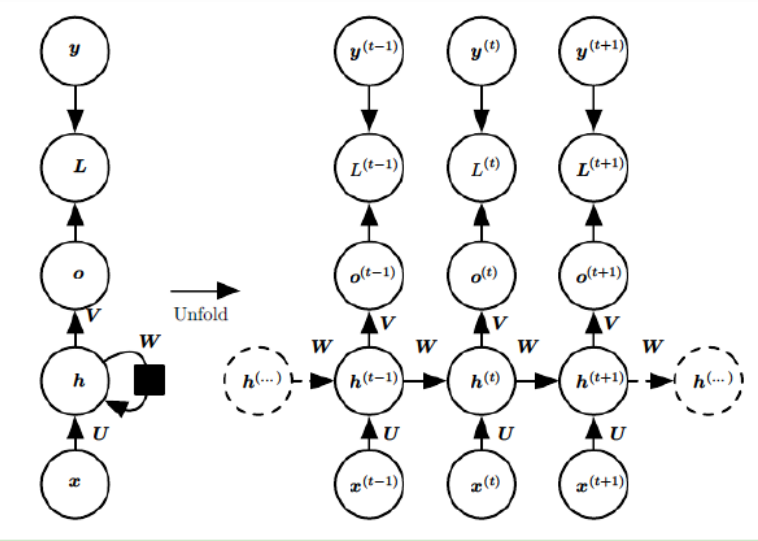
Probability of “Suzhou” in each slot



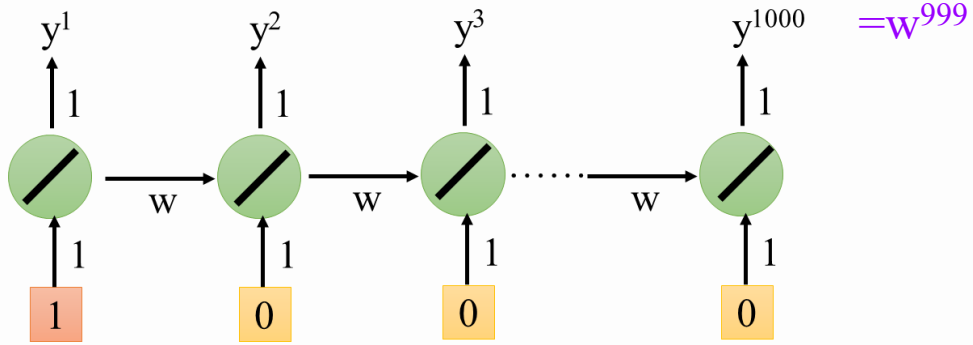


RNN

- 前向传播
  - 任一序列索引号t
  - $h^t = \sigma(z^t) = \sigma(\boldsymbol{U}x^t + \boldsymbol{W}h^{t-1})$
  - $\hat{y}^t = \sigma(o^t) = \sigma(\boldsymbol{V}h^t)$



$$h^t = \sigma(\boldsymbol{U}x^t + \boldsymbol{W}(\sigma(\boldsymbol{U}x^{t-1} + \boldsymbol{W}h^{t-2}))) $$



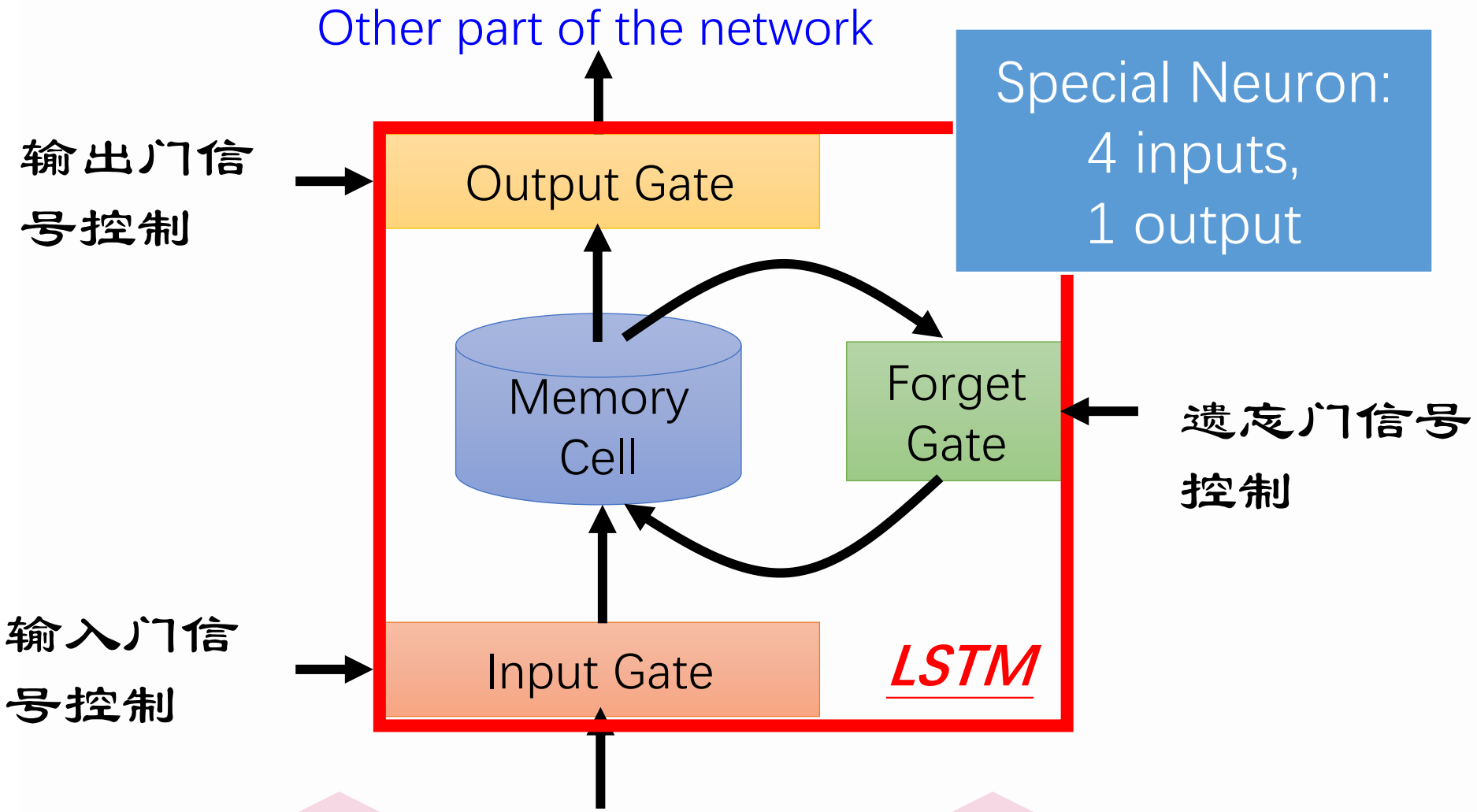
$w = 1 \Rightarrow y^{1000} = 1$
$w = 1.01 \Rightarrow y^{1000} \approx 20000$

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

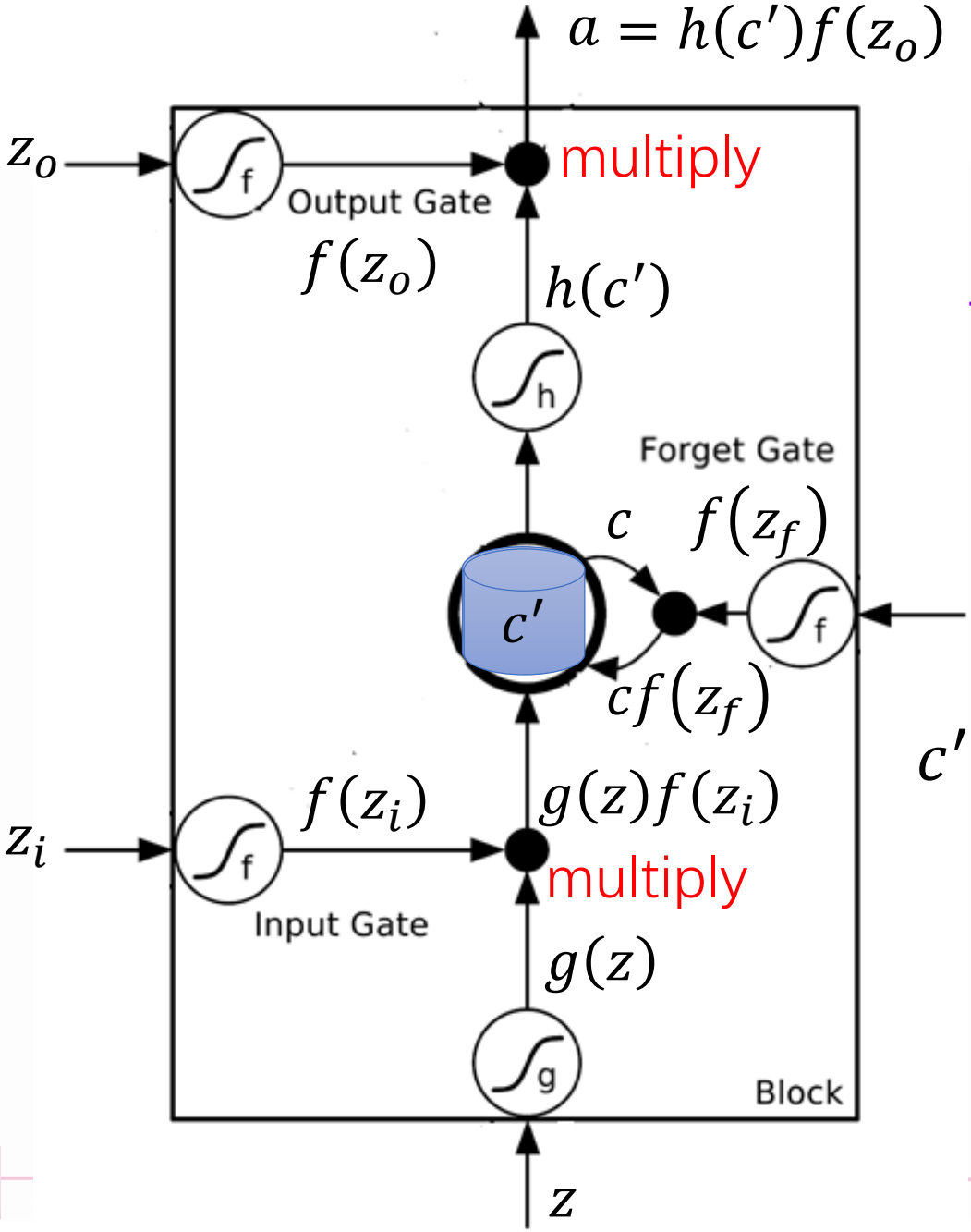
$w = 0.99 \Rightarrow y^{1000} \approx 0$
$w = 0.01 \Rightarrow y^{1000} \approx 0$

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

LSTM


Other part of the network
Special Neuron:
4 inputs,
1 output
输出门信号控制
Output Gate
Memory Cell
Forget Gate
遗忘门信号控制
输入门信号控制
Input Gate
LSTM
Other part of the network

LSTM

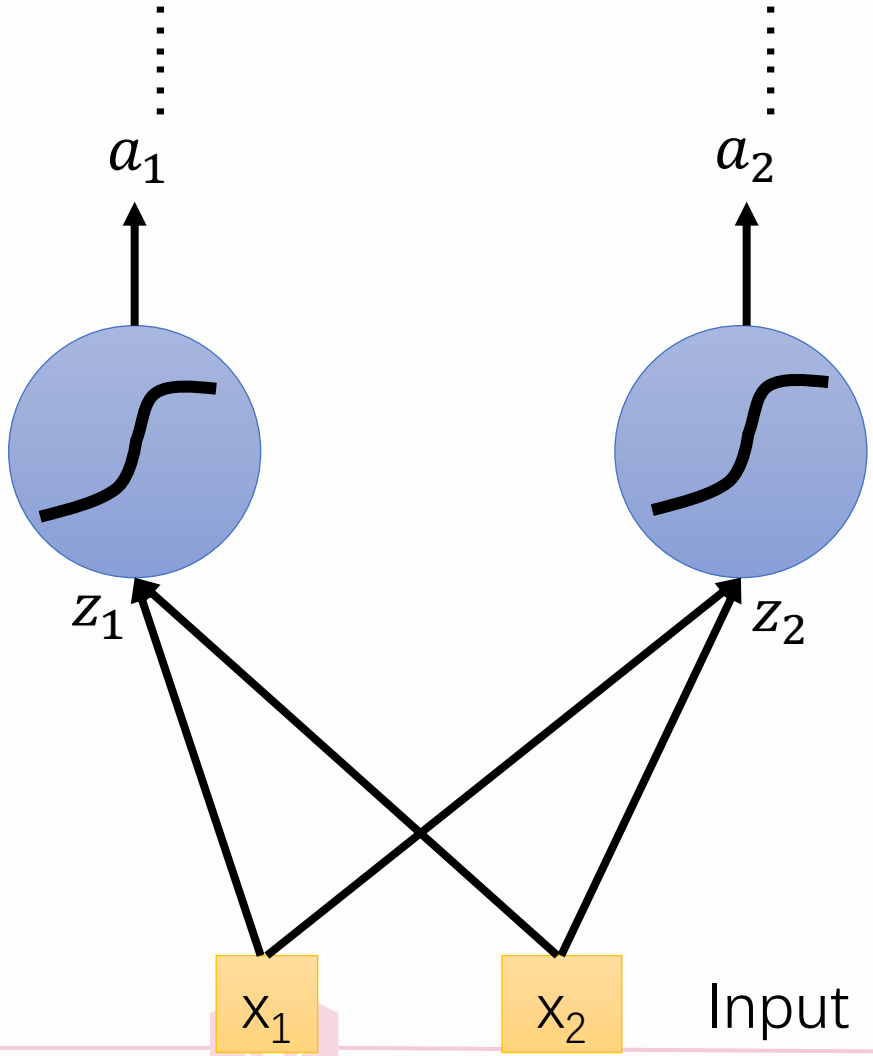


激活函数常用sigmoid function(0~1)来模拟门开关

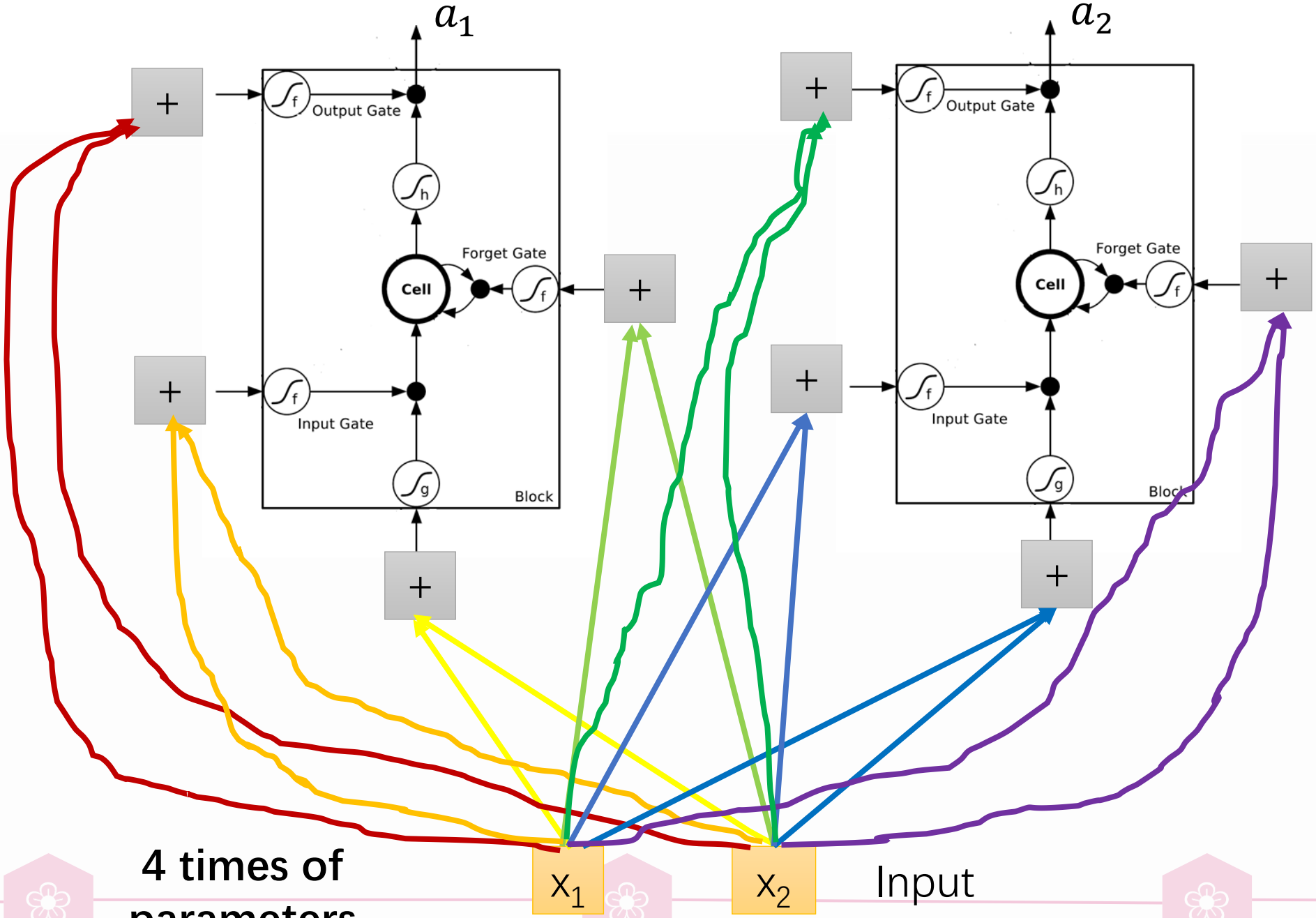
$$c' = g(z)f(z_i) + cf(z_f)$$

LSTM

- 常规形式



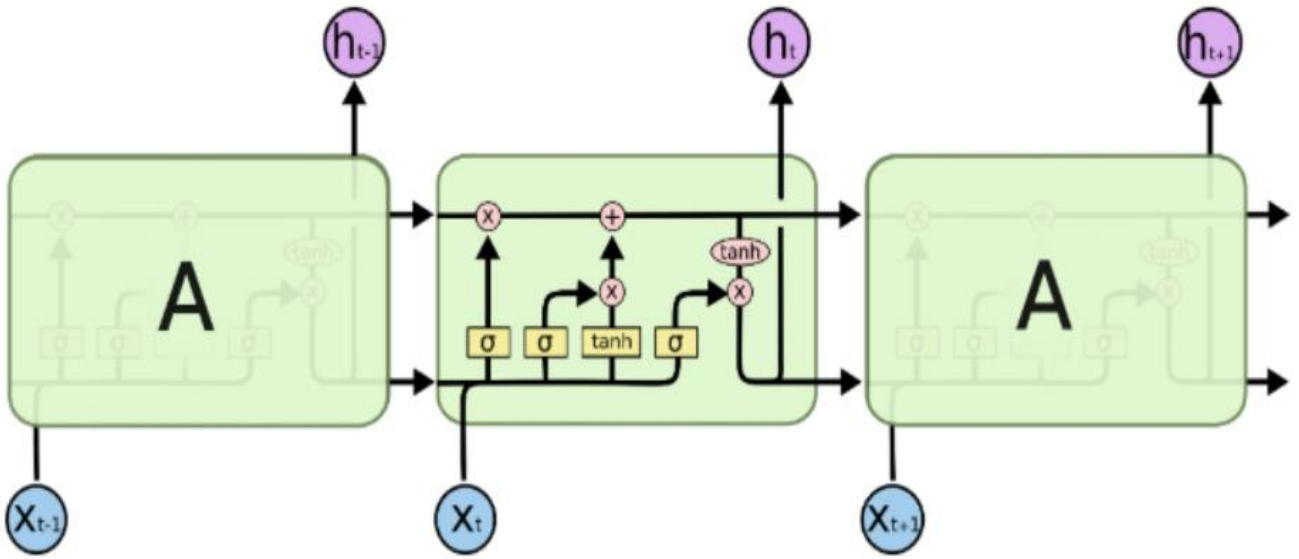
LSTM



4 times of parameters

LSTM

- 前向传播
  - 遗忘门 $f^t = \sigma\left(W_f h^{t-1} + U_f x^t + b_f\right)$
  - 输入部分：
    - 输入门 $i^t = \sigma(W_i h^{t-1} + U_i x^t + b_i)$
    - 输入 $a^t = tanh(W_a h^{t-1} + U_a x^t + b_a)$
  - 记忆存储部分
    - cell更新 $C^t = C^{t-1} \odot f^t + i^t \odot a^t$
  - 输出部分：
    - 输出门 $o^t = \sigma(W_o h^{t-1} + U_o x^t + b_o)$
    - 输出 $h^t = o^t \odot \tanh(C^t)$
  - $y^t = \sigma(V h^t + c)$

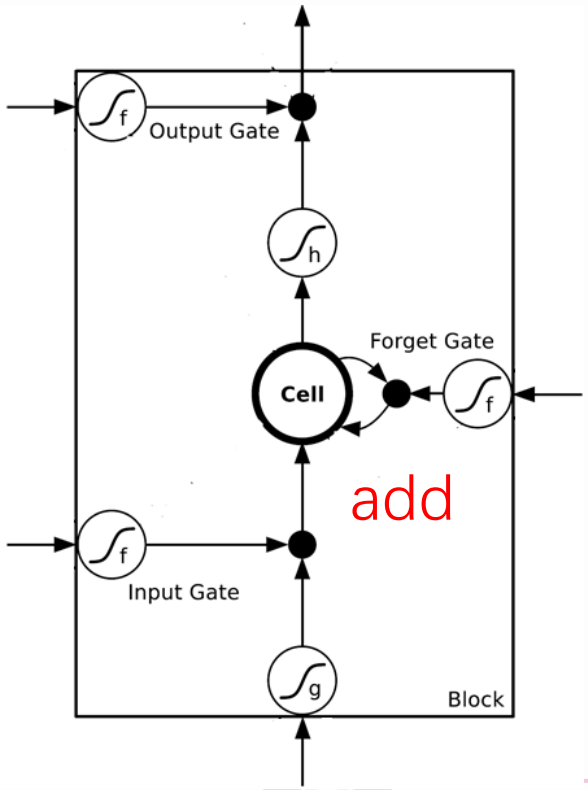


RNN

- $h^t = \sigma(z^t) = \sigma(\boldsymbol{U}x^t + \boldsymbol{W}h^{t-1})$
- $\frac{\partial L}{\partial W} = \sum_{t=1}^{\tau}\sum_{k=1}^{t}\frac{\partial L_t}{\partial \hat{y}^t}\frac{\partial \hat{y}^t}{\partial h^t}\prod_{j=k+1}^{t}\frac{\partial h^j}{\partial h^{j-1}}\frac{\partial h^k}{\partial W}$
- $\frac{\partial h^j}{\partial h^{j-1}} = \sigma' W$
  - 小于1，j和k距离过大，梯度消失
  - 大于1，j和k距离过大，梯度爆炸
  - 长程依赖问题

LSTM

- 记忆部分是相加
- 处理梯度消失，但不能处理梯度爆炸



$$C^t = C^{t-1} \odot f^t + i^t \odot a^t$$

$$\frac{\partial C_t}{\partial C_{t-1}} = \frac{\partial C_t}{\partial f^t}\frac{\partial f_t}{\partial h^{t-1}}\frac{\partial h_{t-1}}{\partial C_{t-1}} + \frac{\partial C_t}{\partial i^t}\frac{\partial i_t}{\partial h^{t-1}}\frac{\partial h_{t-1}}{\partial C_{t-1}} + \frac{\partial C_t}{\partial a^t}\frac{\partial a_t}{\partial h^{t-1}}\frac{\partial h_{t-1}}{\partial C_{t-1}} + f^t$$