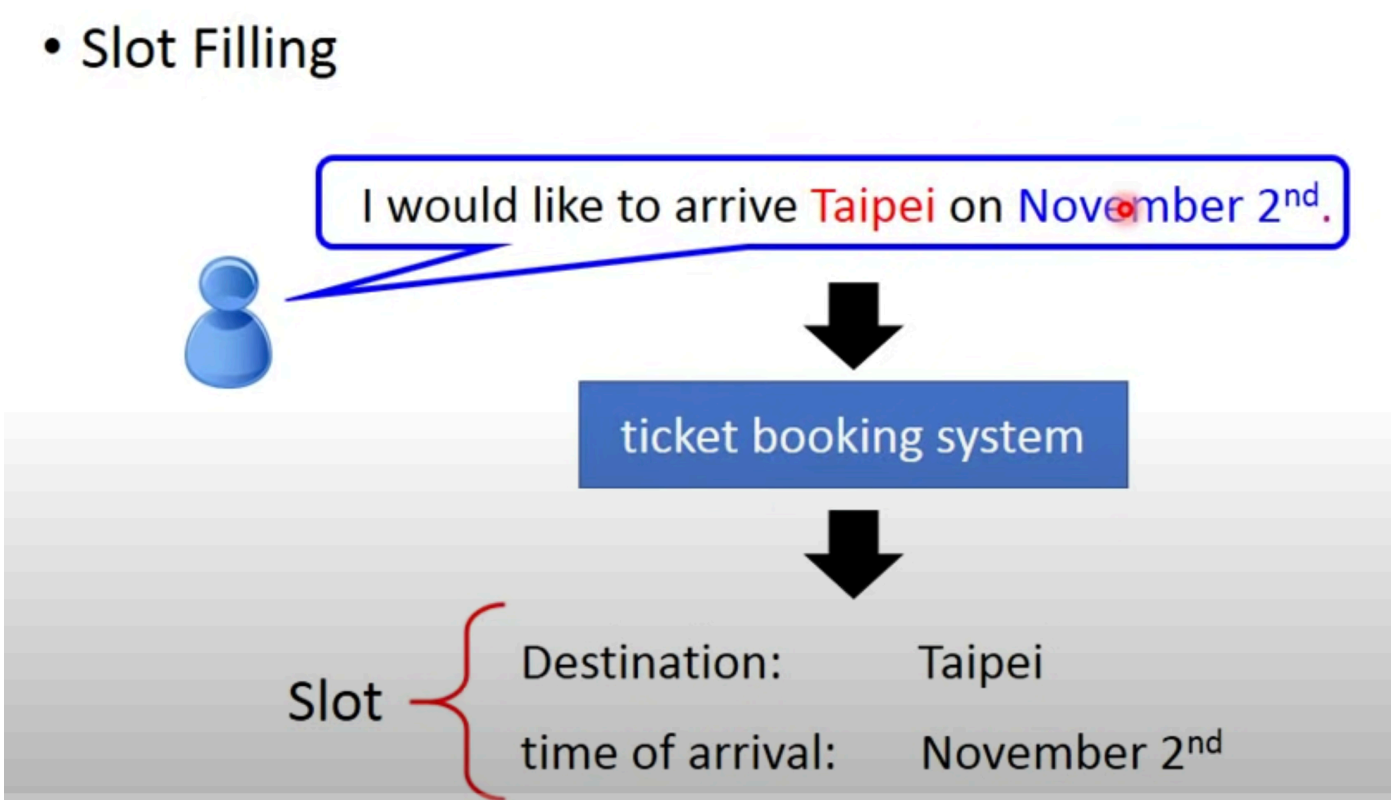
RNN

Slot Filling

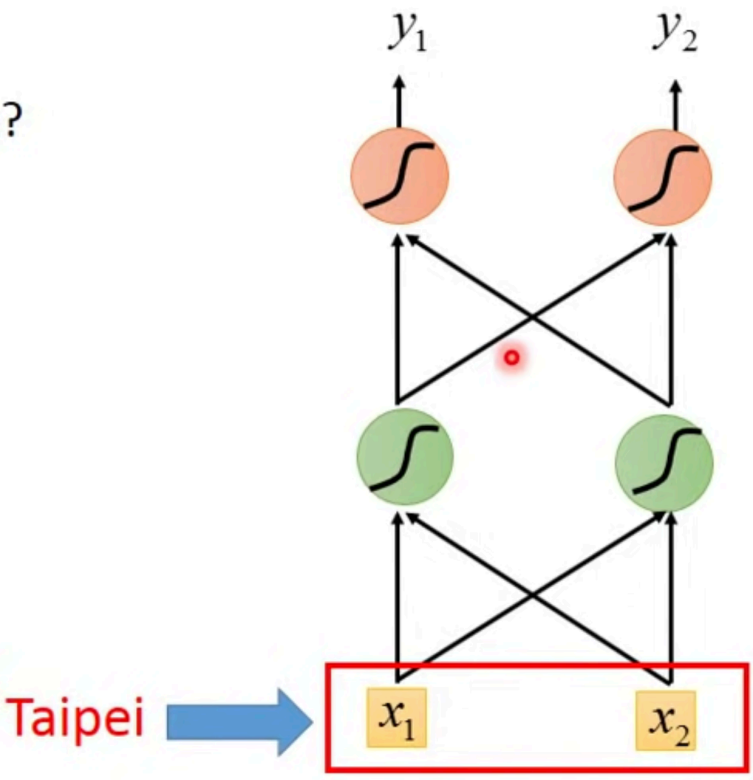


Trivial Sol.

Example Application

Solving slot filling by Feedforward network?

Input: a word



输出一个关于词汇的distribution

1-of-N encoding

How to represent each word as a vector?

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

The vector is lexicon size.

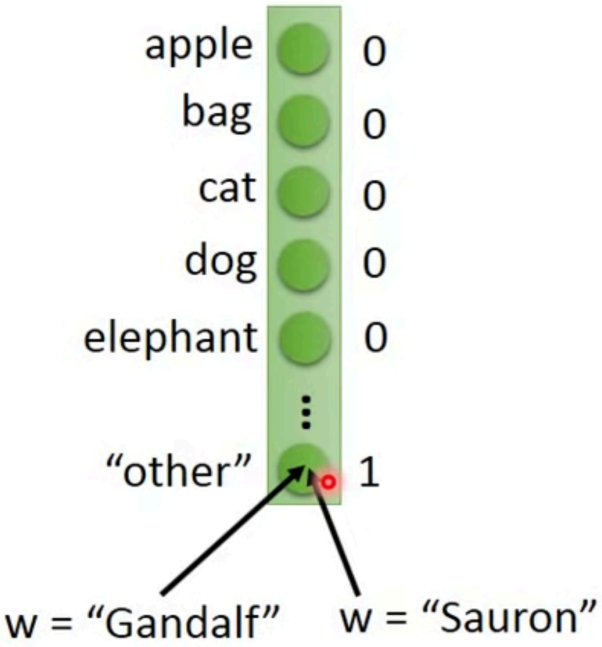
Each dimension corresponds to a word in the lexicon

The dimension for the word is 1, and others are 0

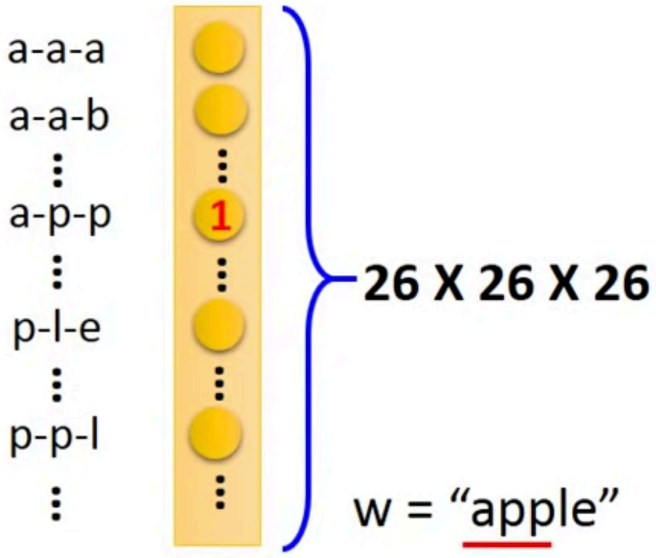
apple = [1 0 0 0 0]
bag = [0 1 0 0 0]
cat = [0 0 1 0 0]
dog = [0 0 0 1 0]
elephant = [0 0 0 0 1]

Beyond 1-of-N encoding

Dimension for "Other"

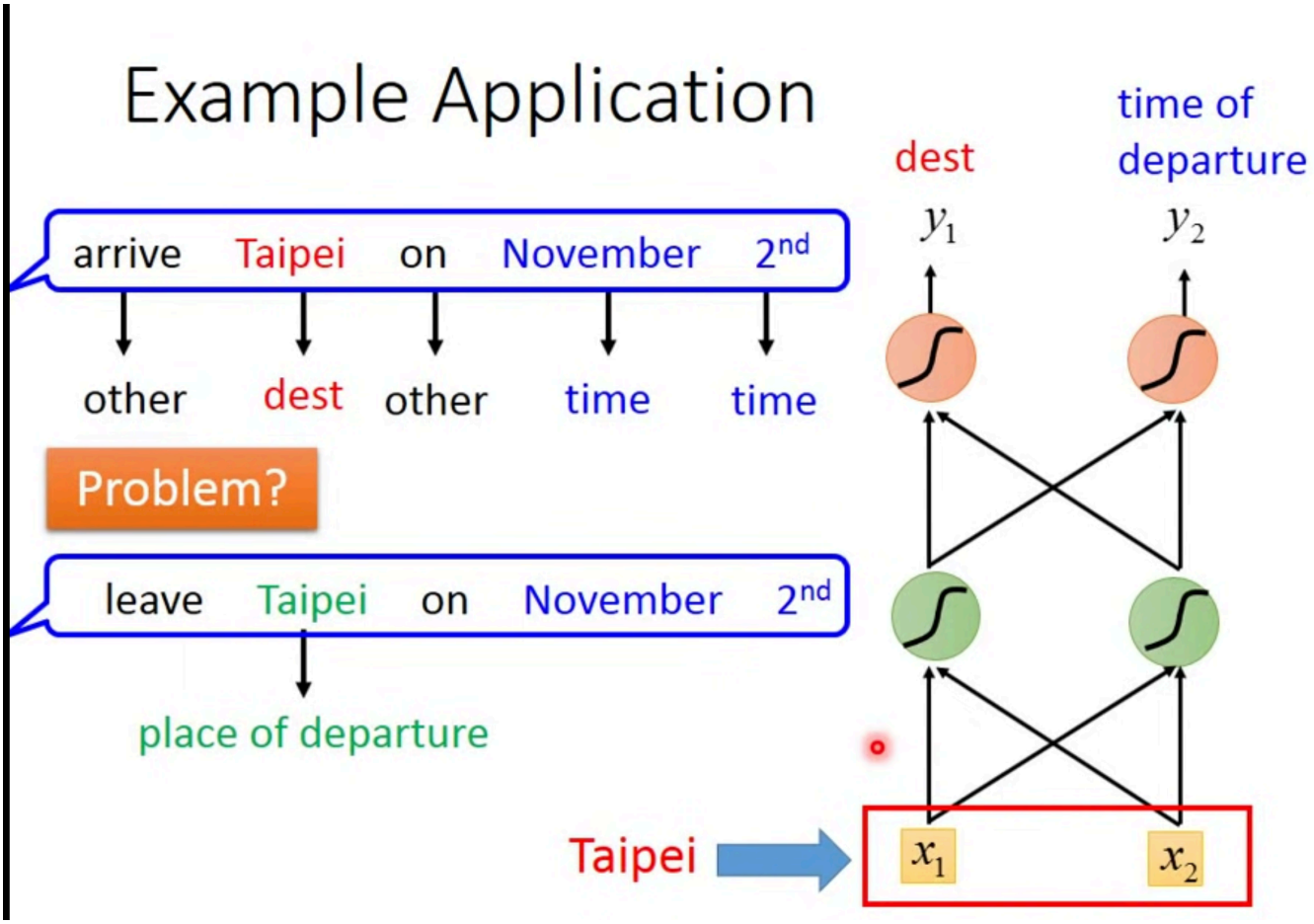


Word hashing



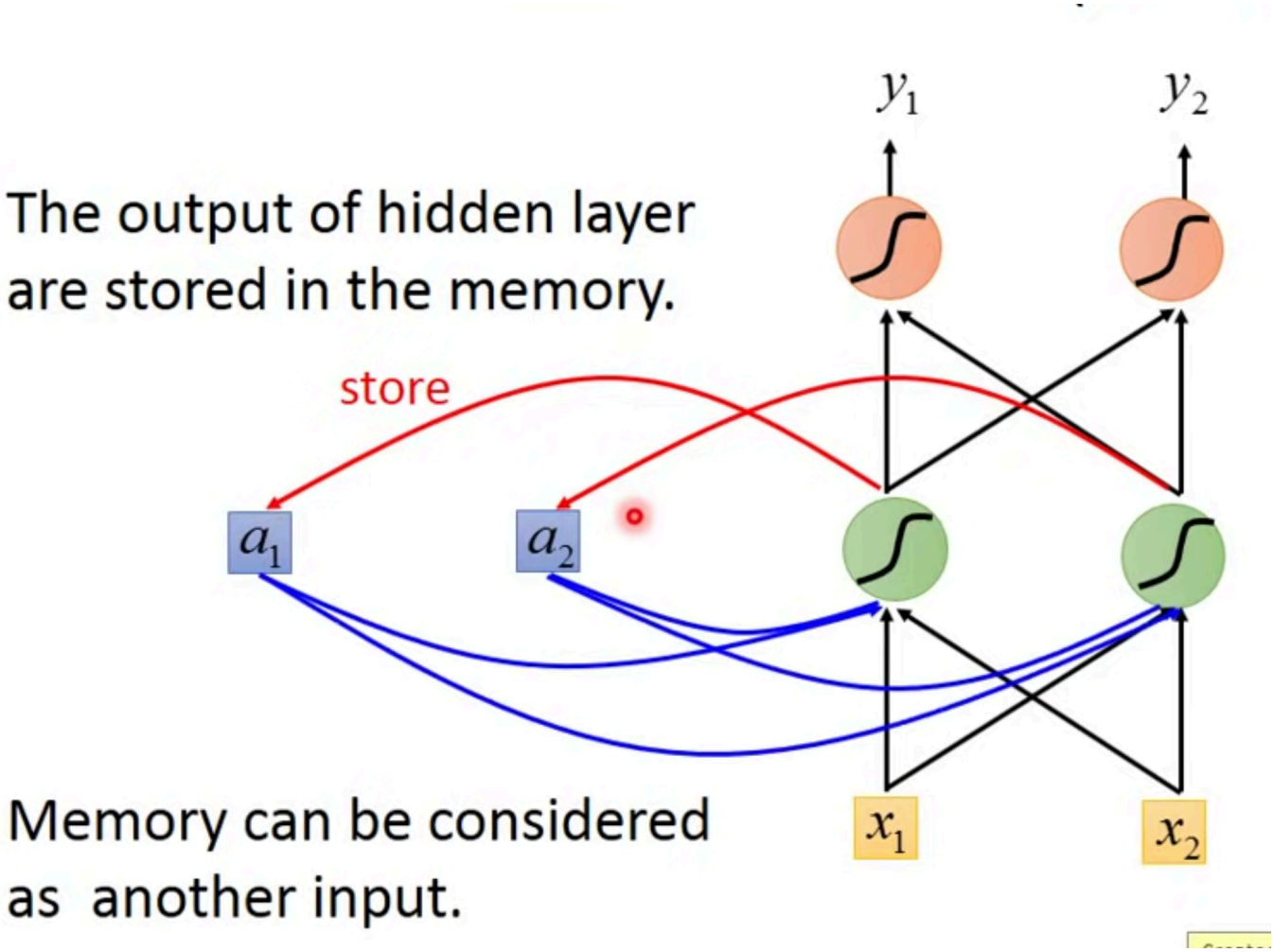
独热编码

Example Application



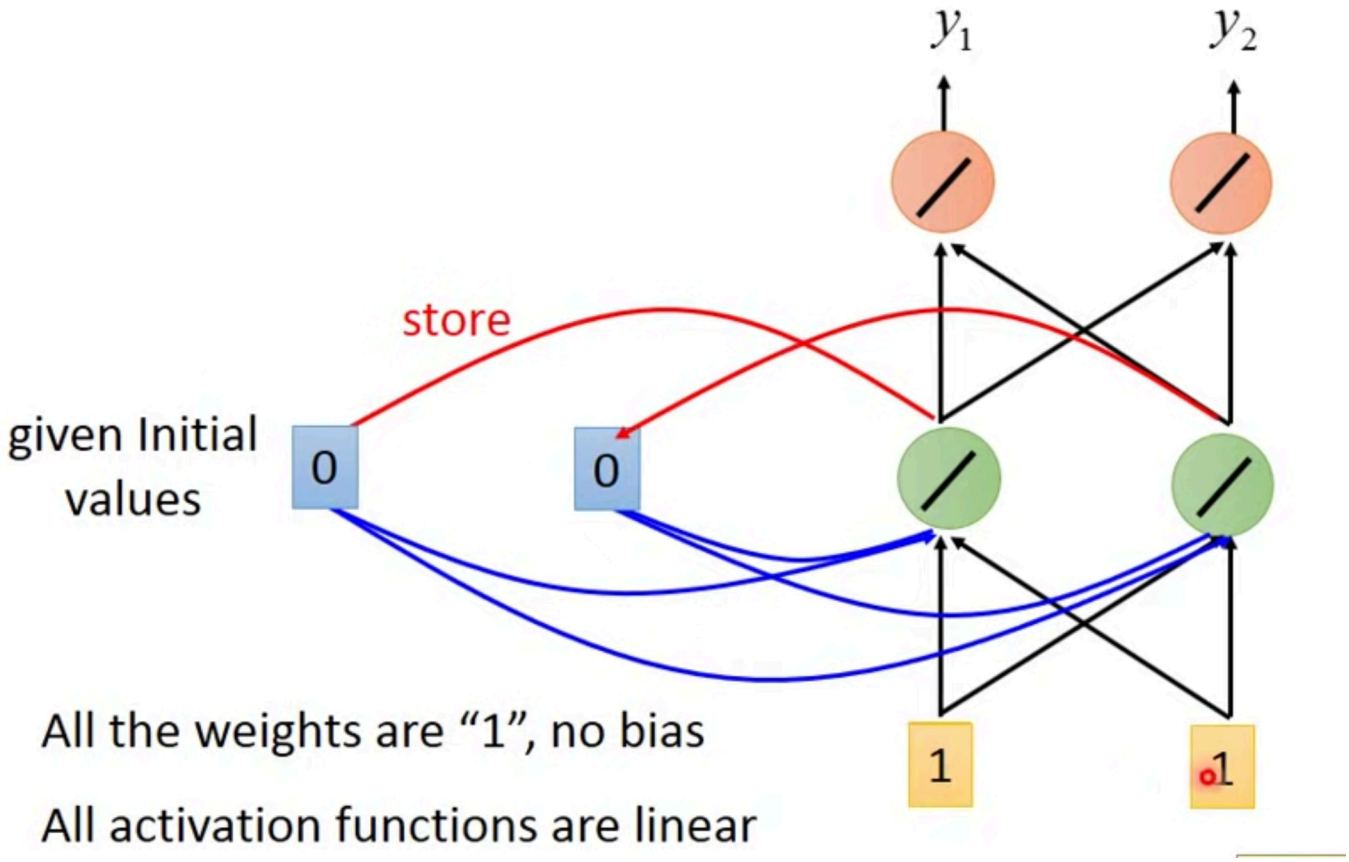
fc神经网络输出有固定性

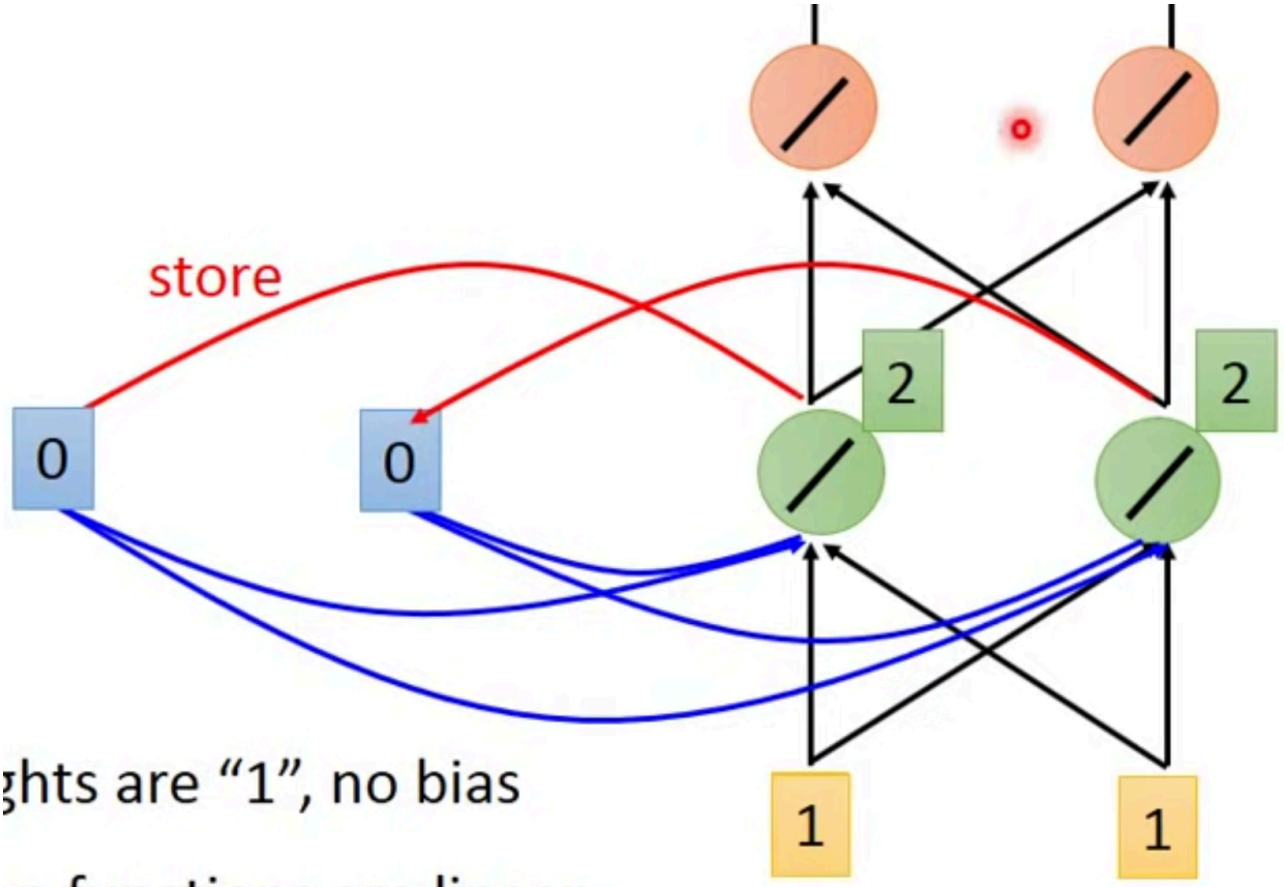
加入状态 类似FSM -> RNN



Input sequence: $\begin{bmatrix}1\\1\end{bmatrix}$ $\begin{bmatrix}1\\1\end{bmatrix}$ $\begin{bmatrix}2\\2\end{bmatrix}$ ……

Example





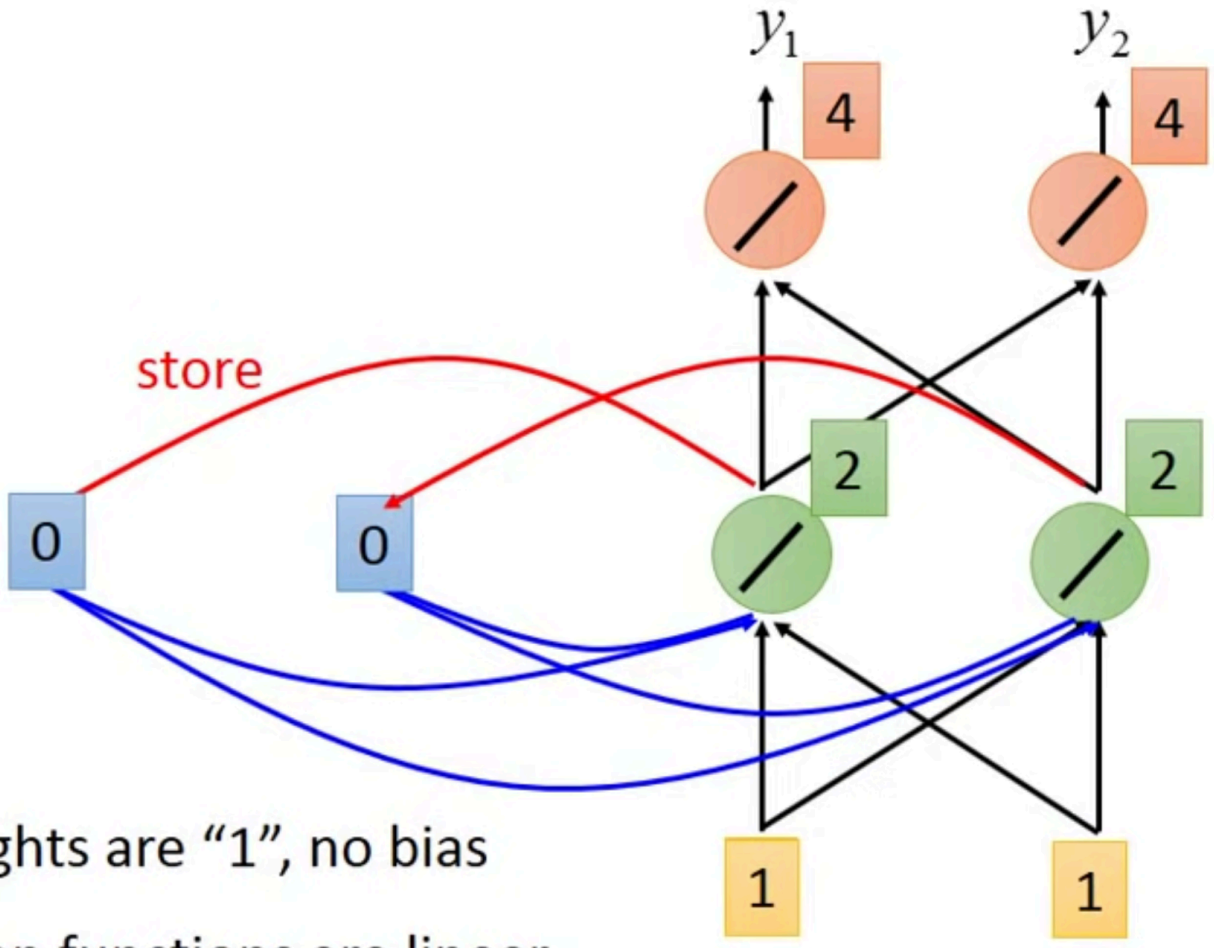
ghts are “1”, no bias

on functions are linear

ple

Input sequence: $\begin{bmatrix}1\\1\end{bmatrix}$ $\begin{bmatrix}1\\1\end{bmatrix}$ $\begin{bmatrix}2\\2\end{bmatrix}$ ……

output sequence: $\begin{bmatrix}4\\4\end{bmatrix}$



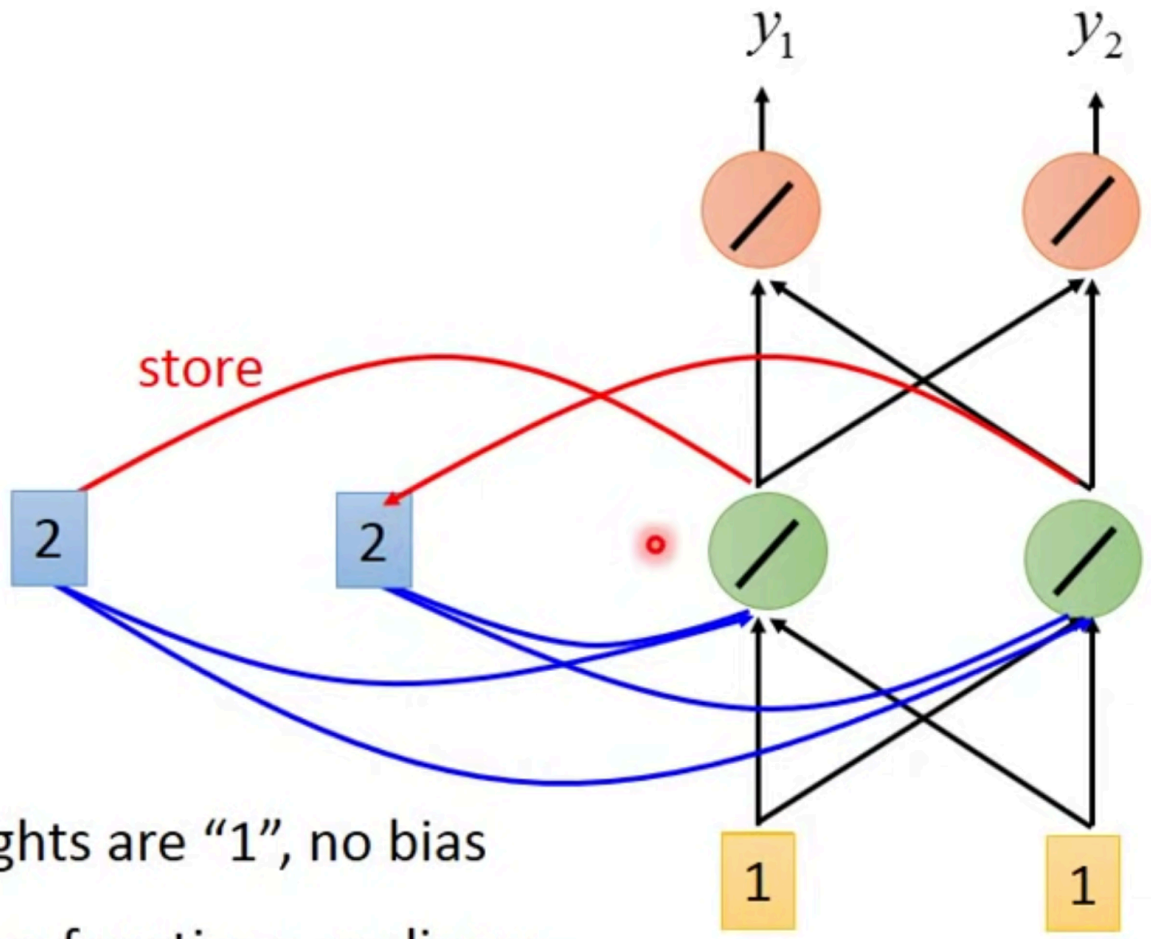
ghts are “1”, no bias

on functions are linear

Input sequence: $\begin{bmatrix}1\\1\end{bmatrix}$ $\begin{bmatrix}1\\1\end{bmatrix}$ $\begin{bmatrix}2\\2\end{bmatrix}$ ……

ıple

output sequence: $\begin{bmatrix}4\\4\end{bmatrix}$



eights are “1”, no bias

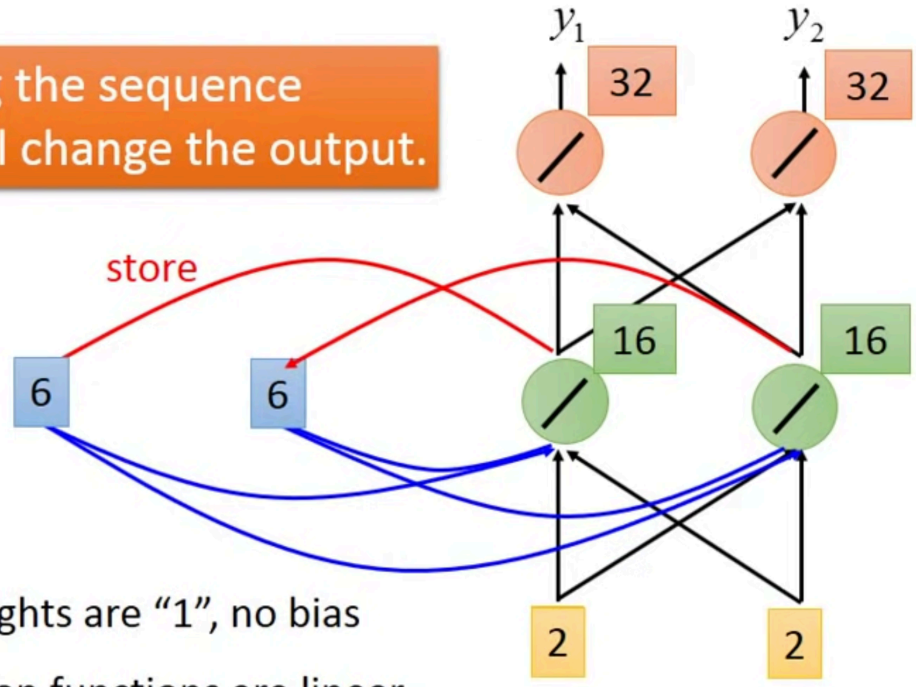
tion functions are linear

Example

Input sequence: $\begin{bmatrix}1\\1\end{bmatrix}$ $\begin{bmatrix}1\\1\end{bmatrix}$ $\begin{bmatrix}2\\2\end{bmatrix}$ ……

output sequence: $\begin{bmatrix}4\\4\end{bmatrix}\begin{bmatrix}12\\12\end{bmatrix}\begin{bmatrix}32\\32\end{bmatrix}$

Changing the sequence order will change the output.



All the weights are “1”, no bias

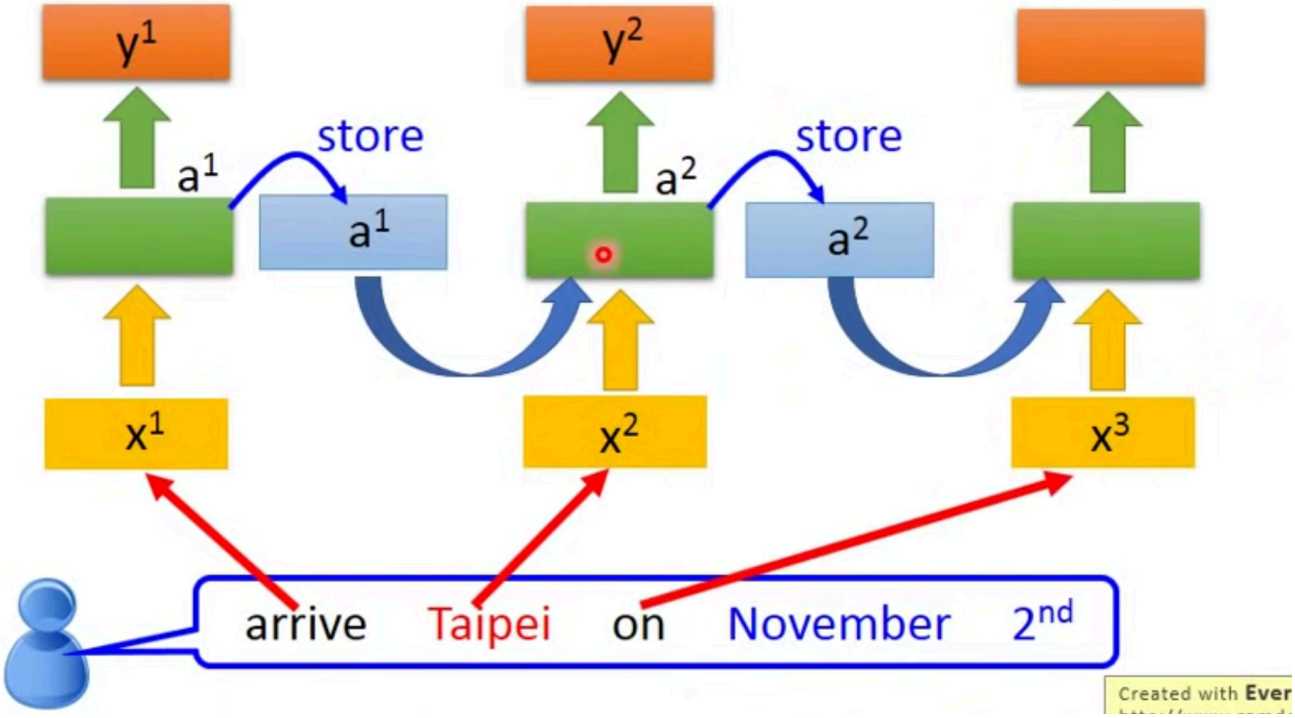
All activation functions are linear

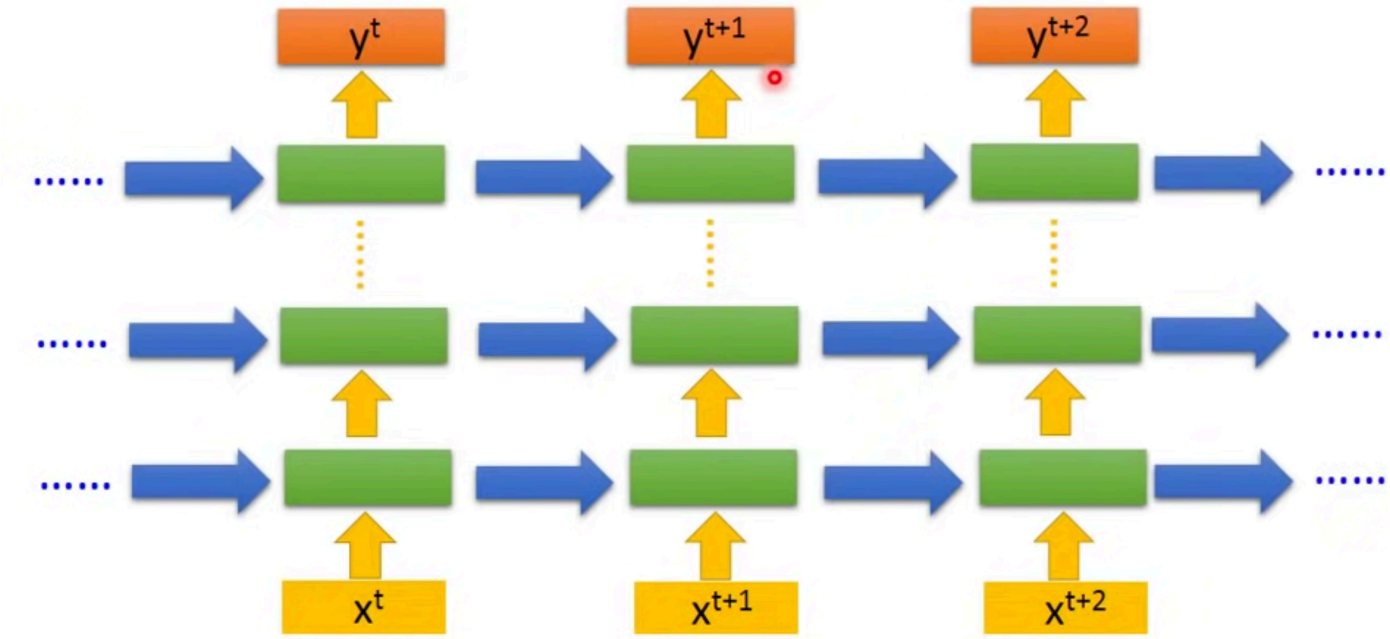
RNN

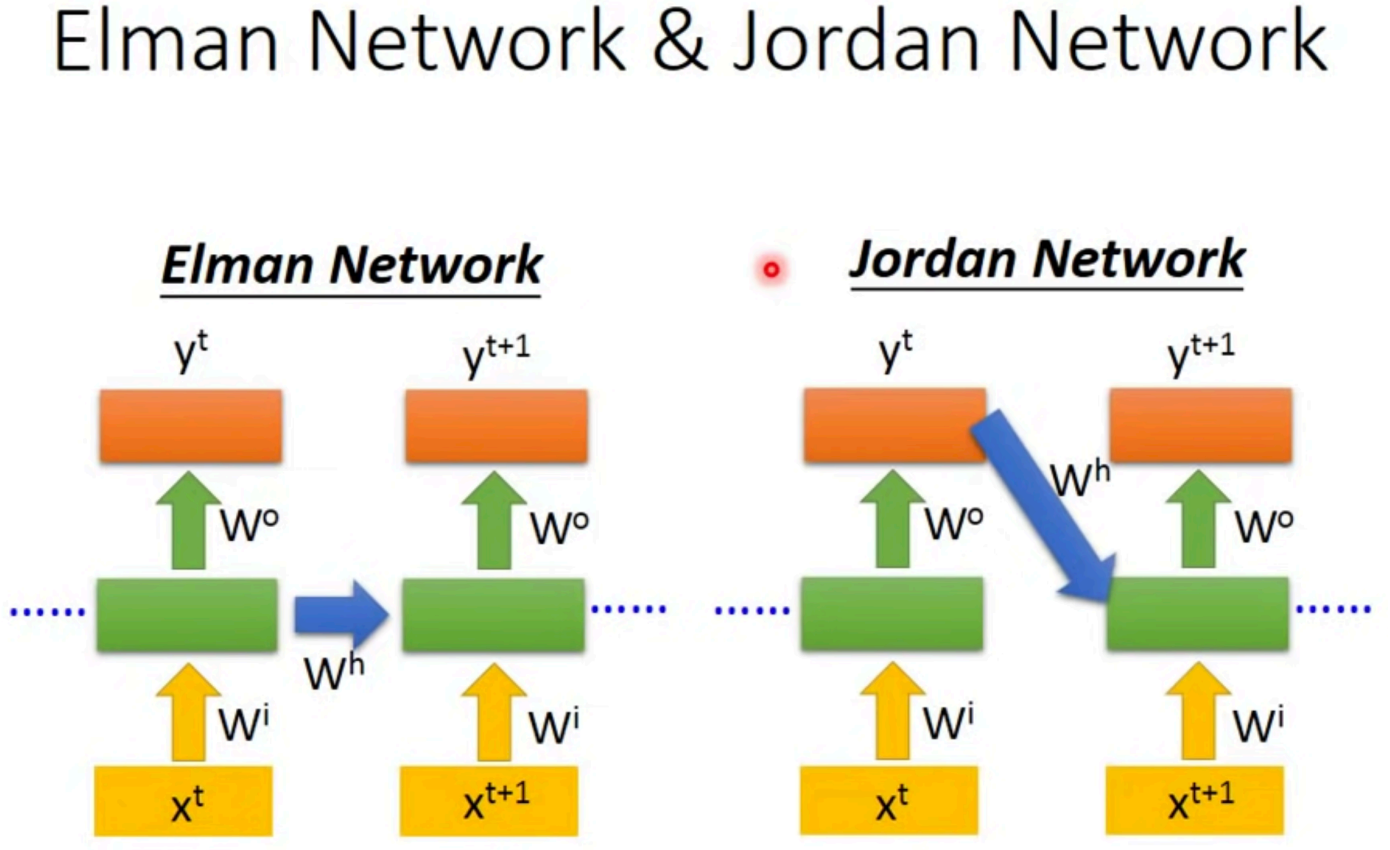
The same network is used again and again.

Probability of "arrive" in each slot

Probability of "Taipei" in each slot

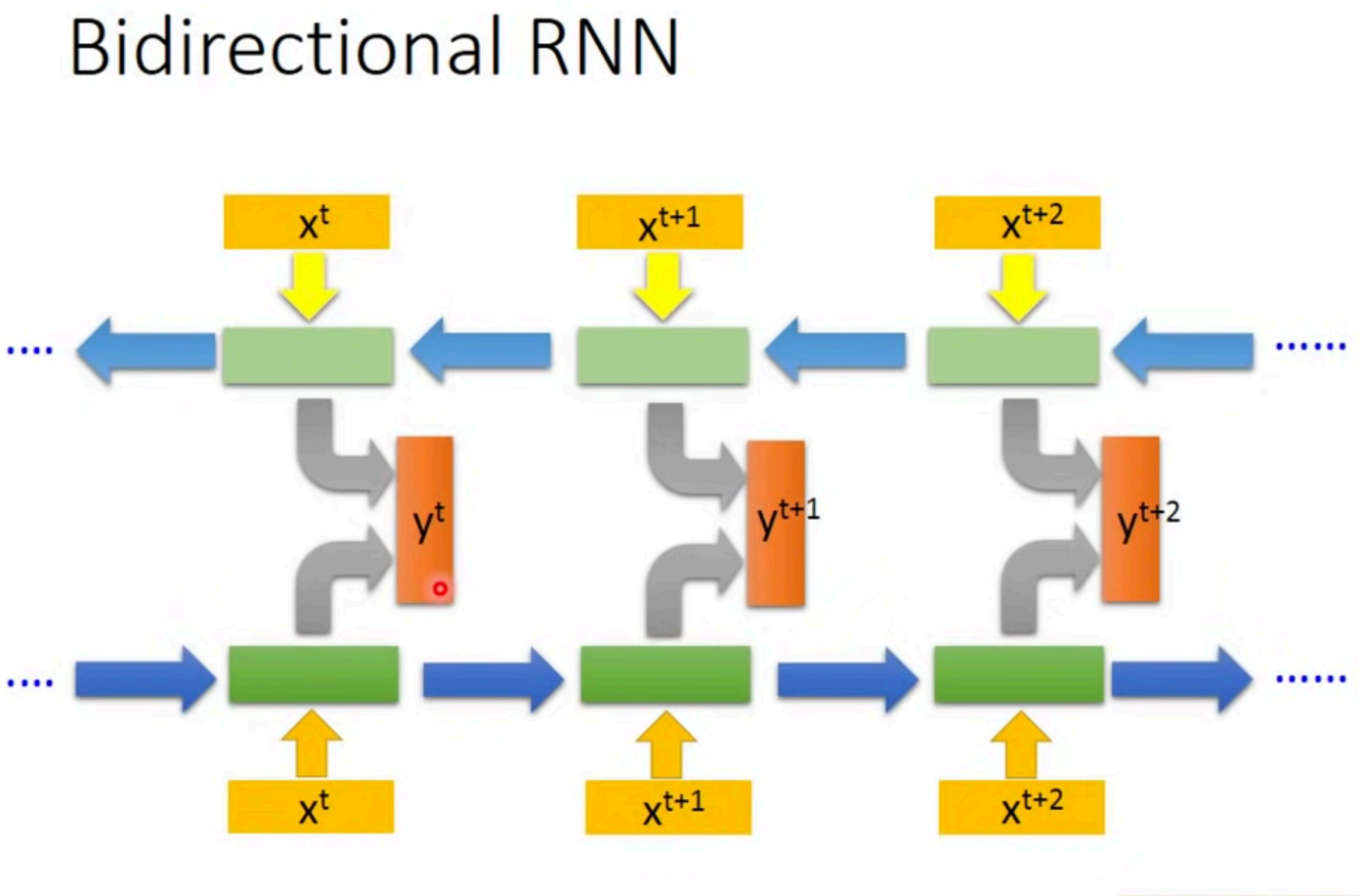






Elman : 存hidden layer

Jordan : 存output 结果好点

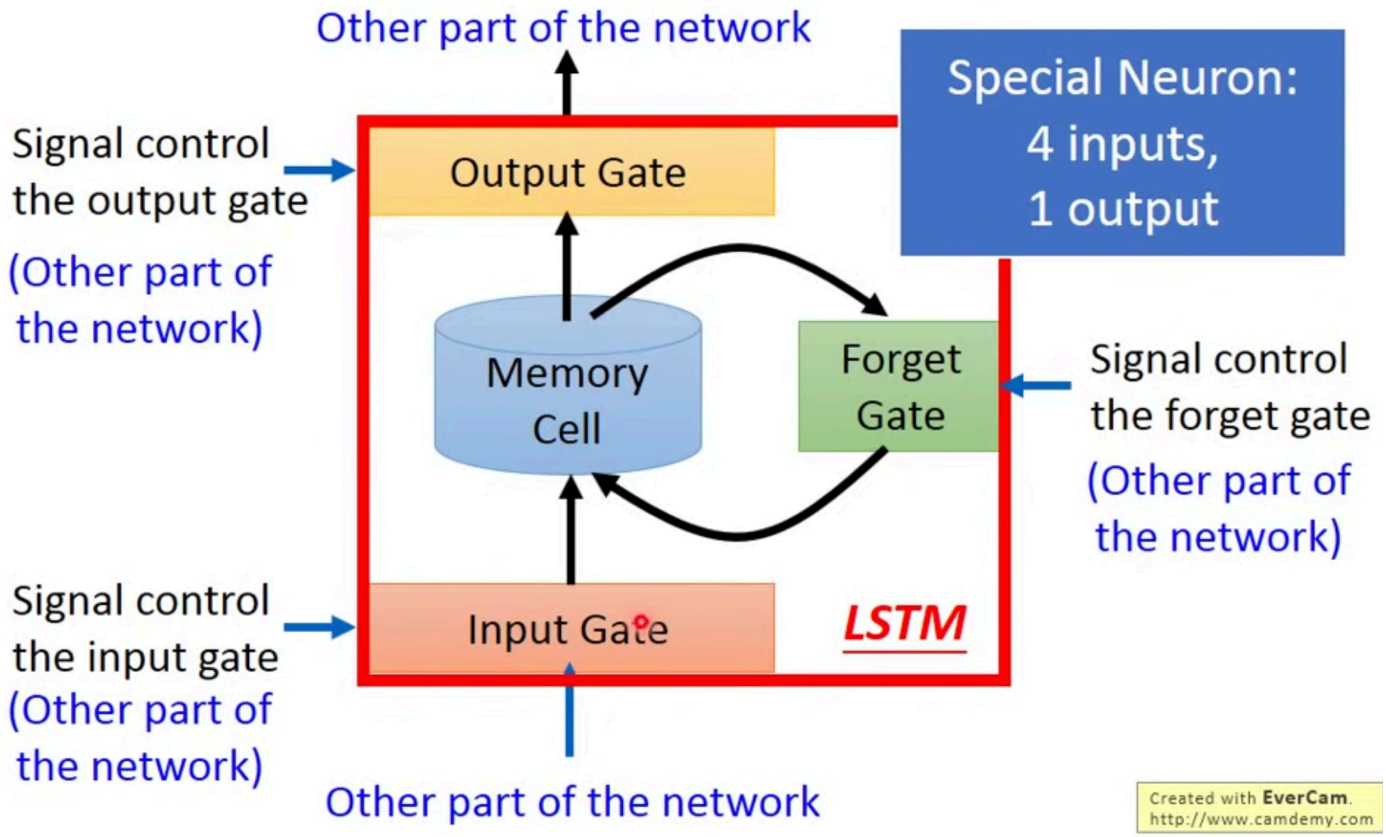


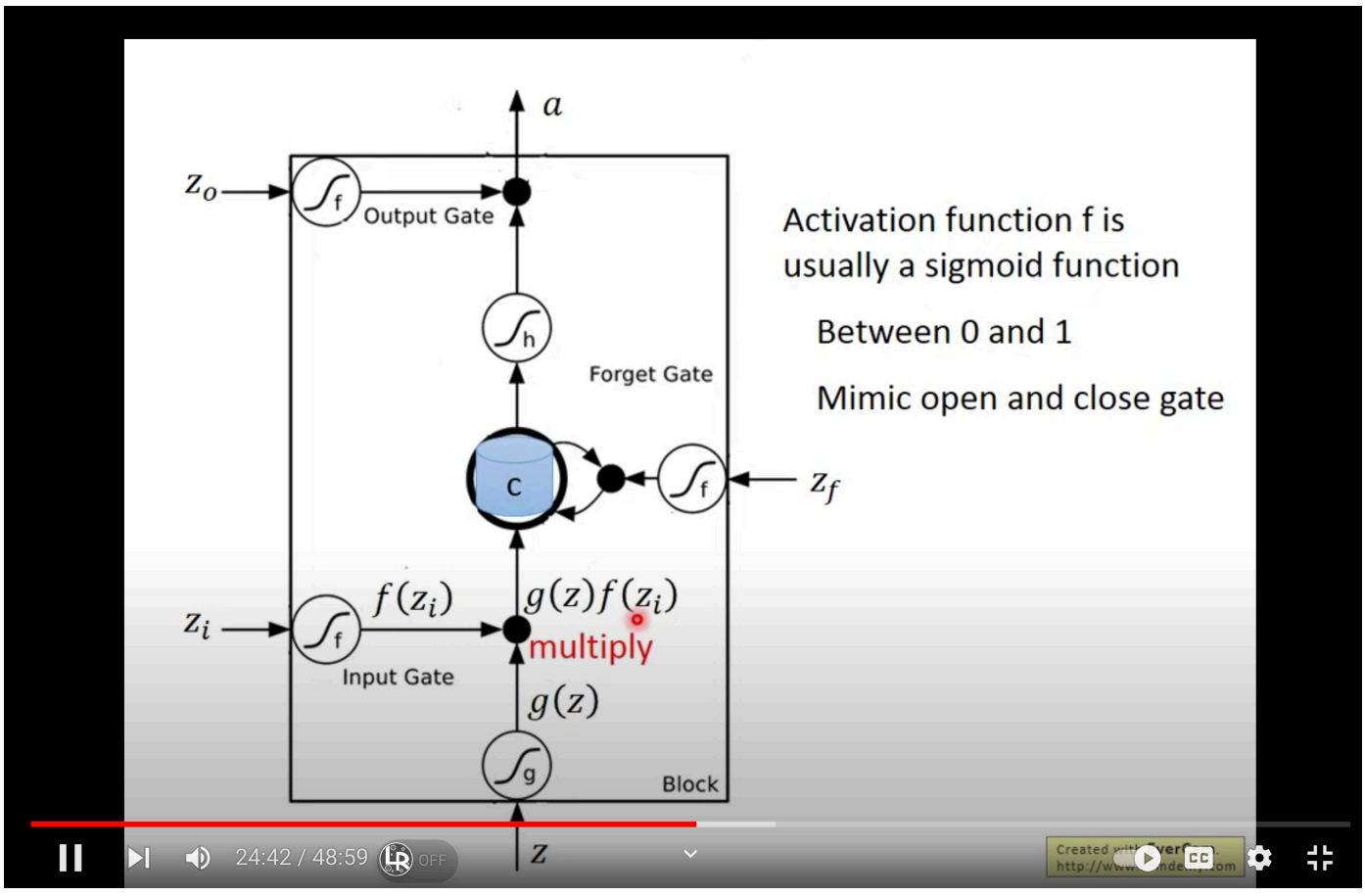
训练两个不同方向的RNN 同时产生输出

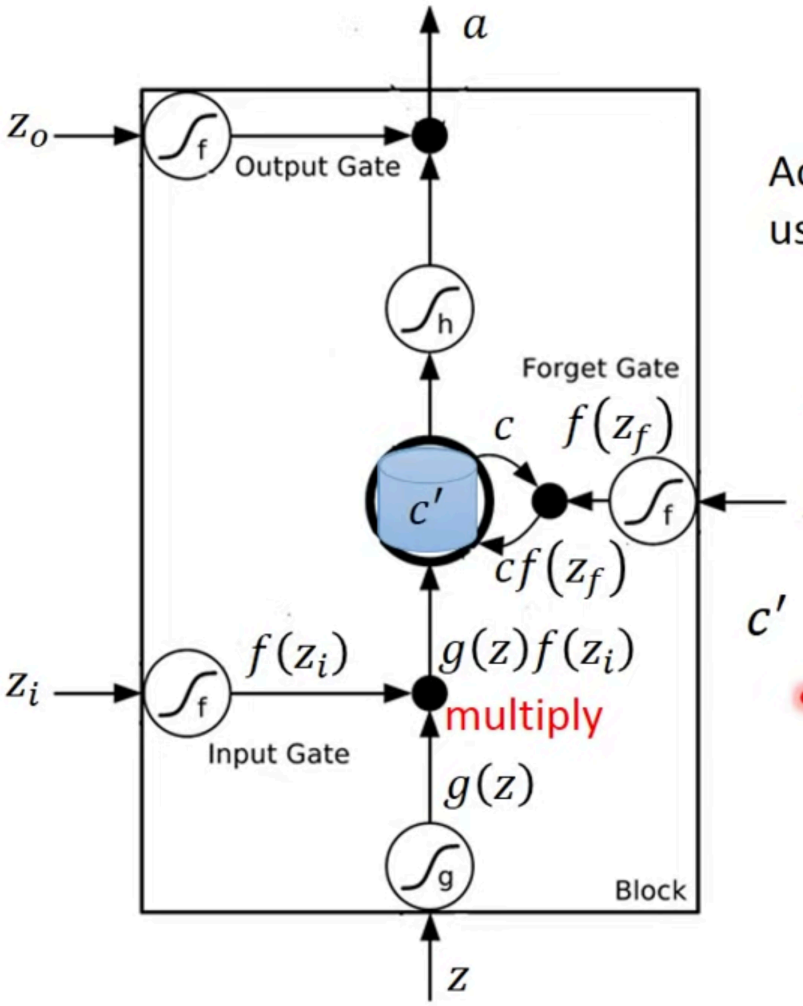
LSTM

记忆较长的(相较于RNN) short-term

Long Short-term Memory (LSTM)





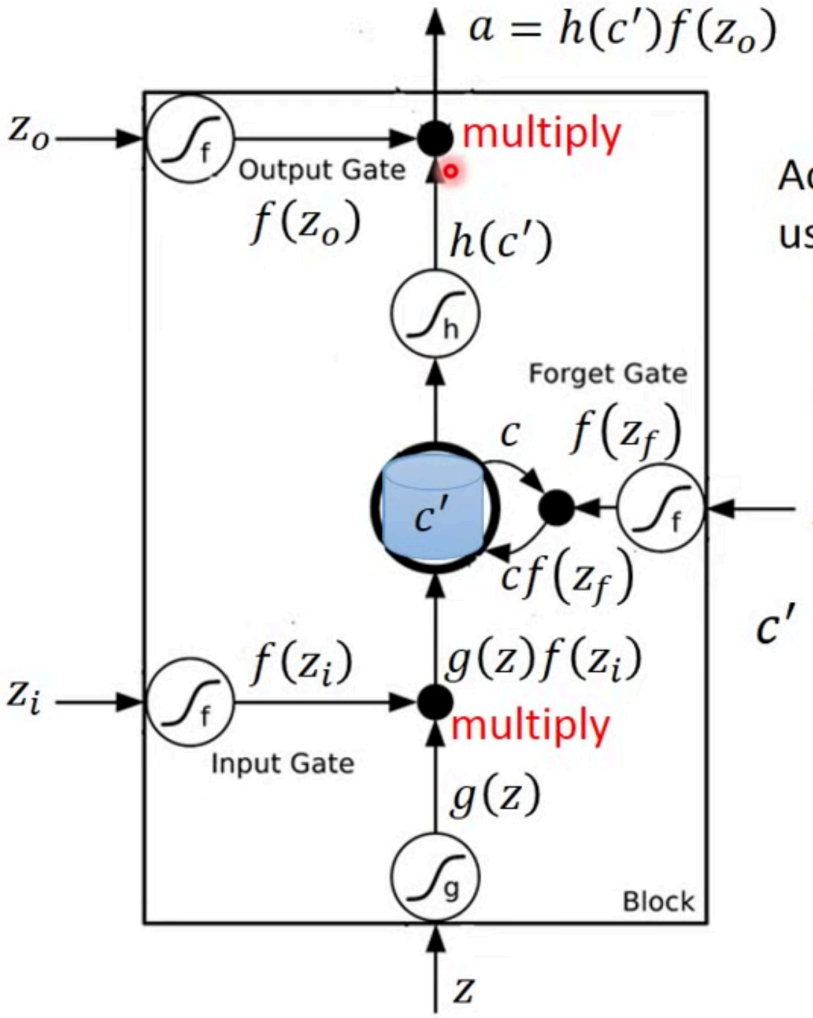


Activation function f is usually a sigmoid function

Between 0 and 1

Mimic open and close gate

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



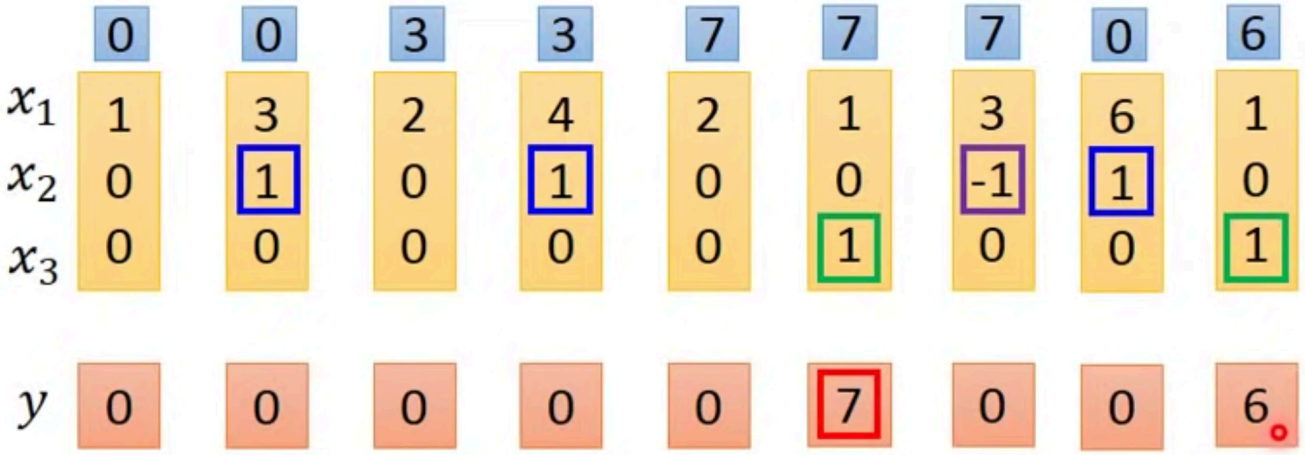
Activation function f is usually a sigmoid function

Between 0 and 1

Mimic open and close gate

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

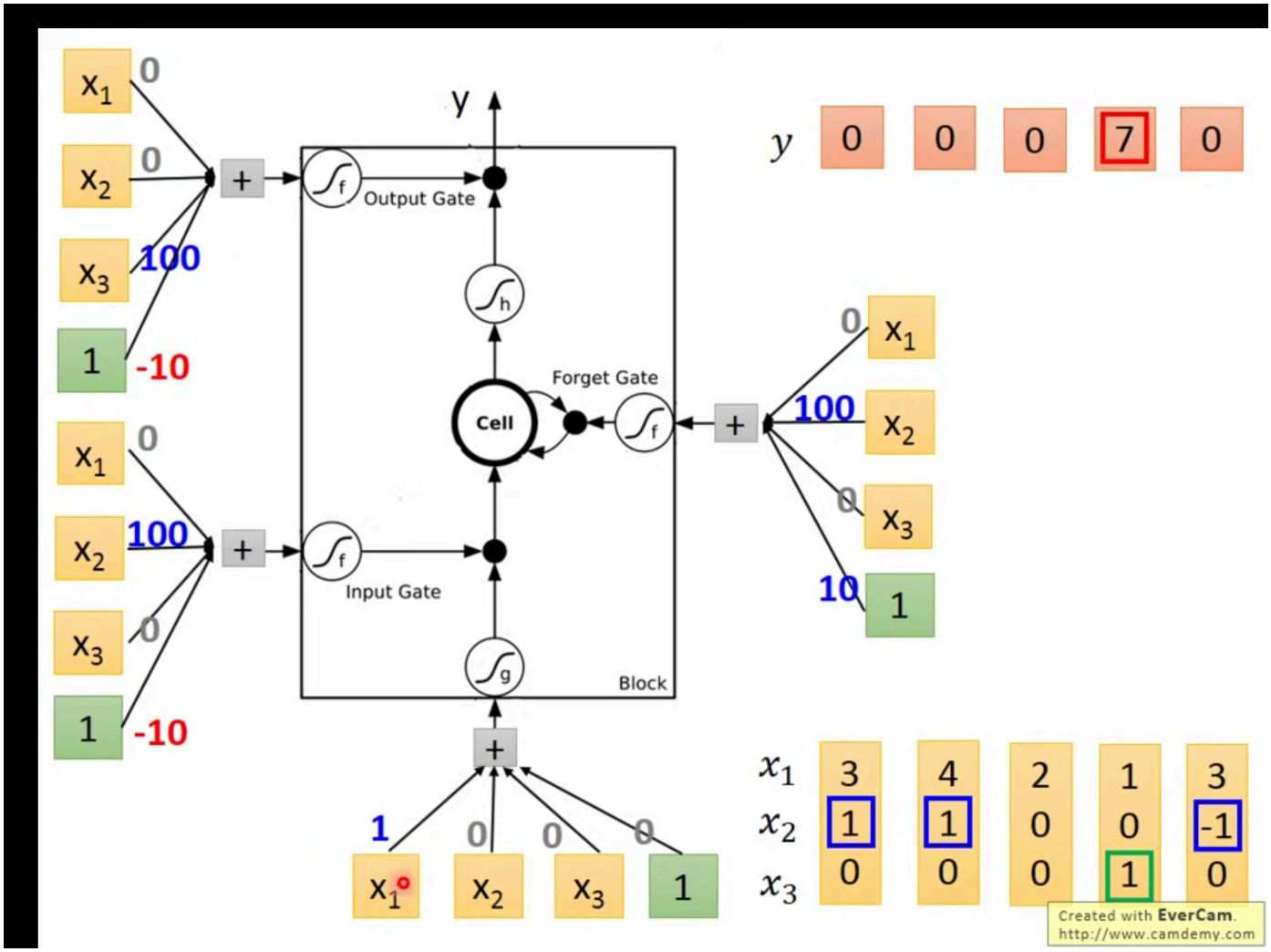
LSTM - Example

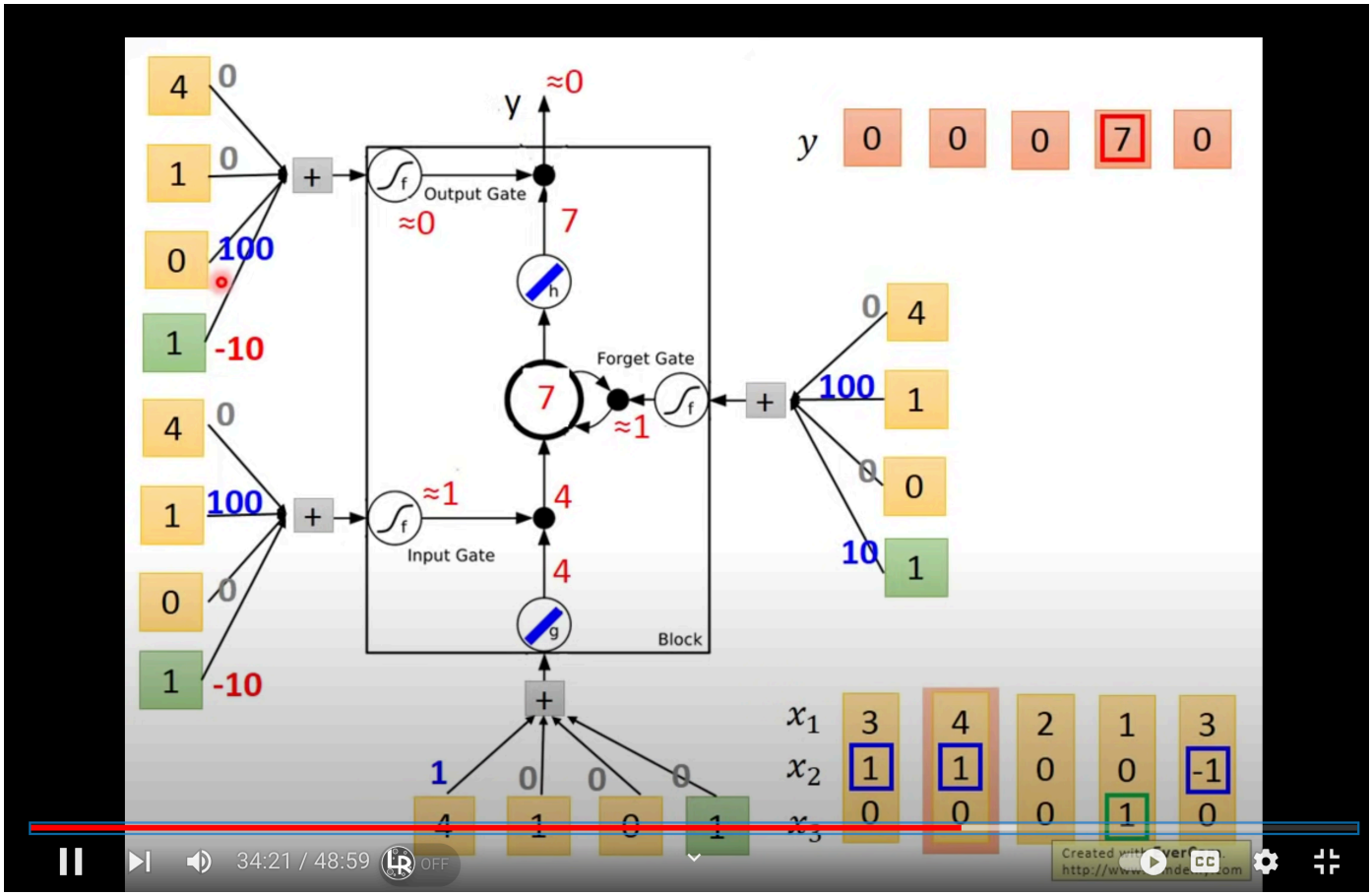


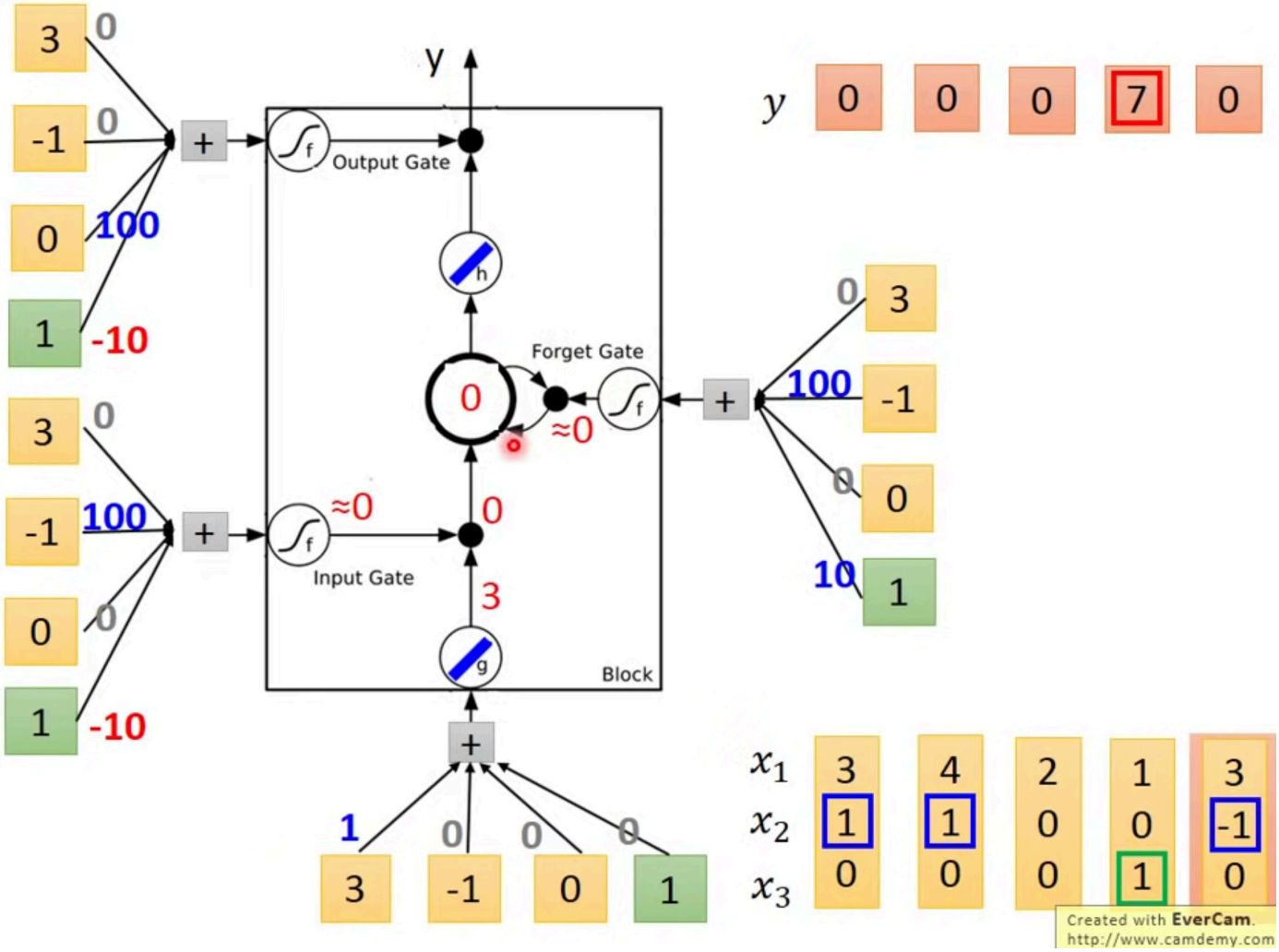
When x_2 = 1, add the numbers of x_1 into the memory

When x_2 = -1, reset the memory

When x_3 = 1, output the number in the memory.

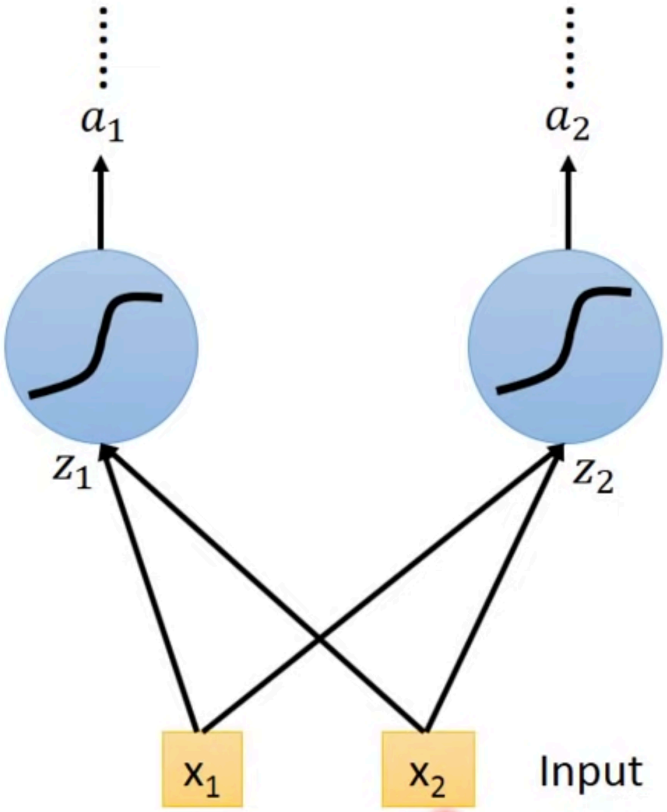

y
Output Gate
Forget Gate
Cell
Input Gate
Block
Created with EverCam.
http://www.camdemy.com

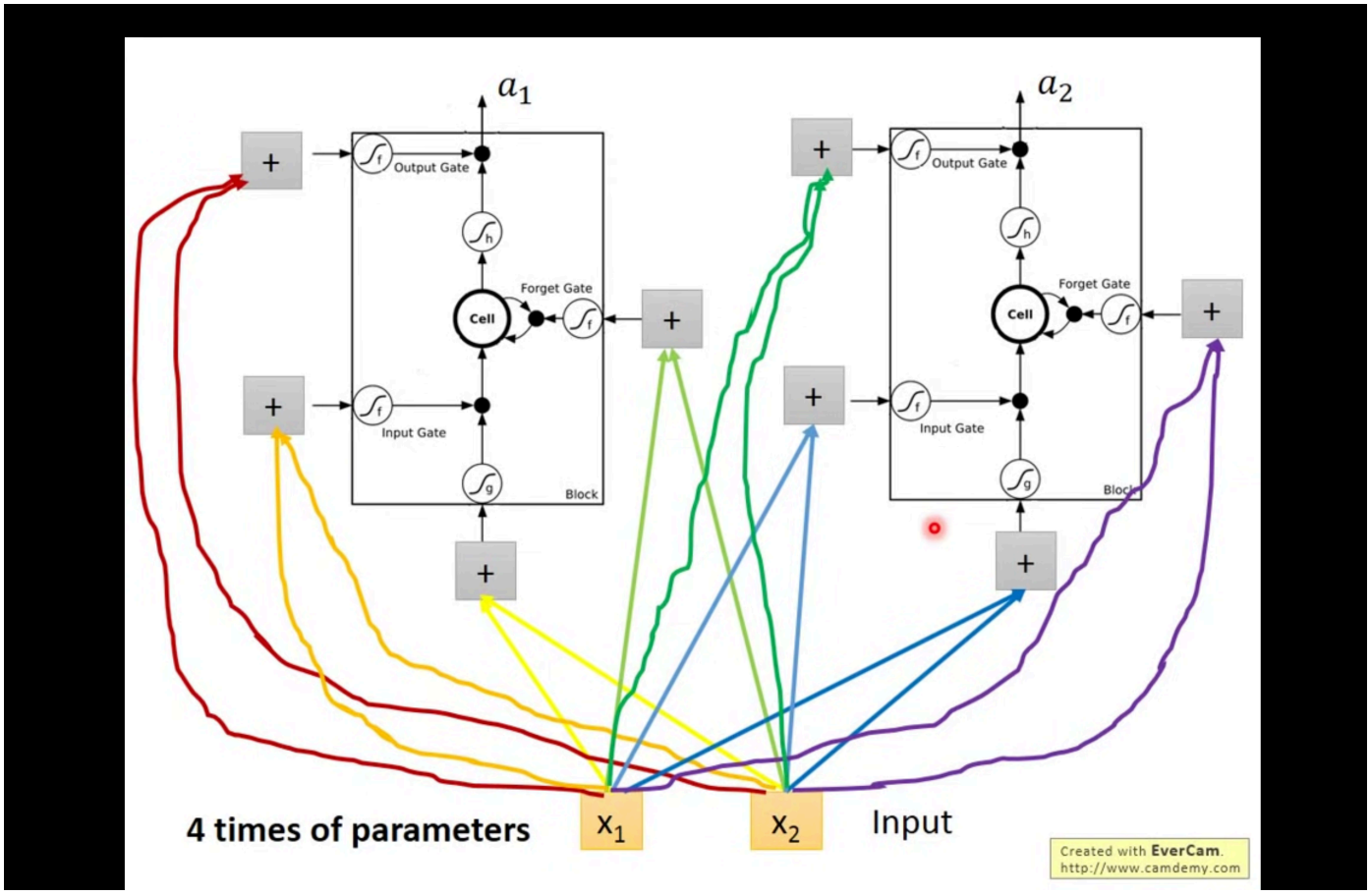

y
≈0
Output Gate
7
Forget Gate
≈1
Input Gate
4
Block
34:21 / 48:59
OFF


y
Output Gate
Forget Gate
Input Gate
Block
≈0
≈0
0
3
1
Created with EverCam.
http://www.camdemy.com

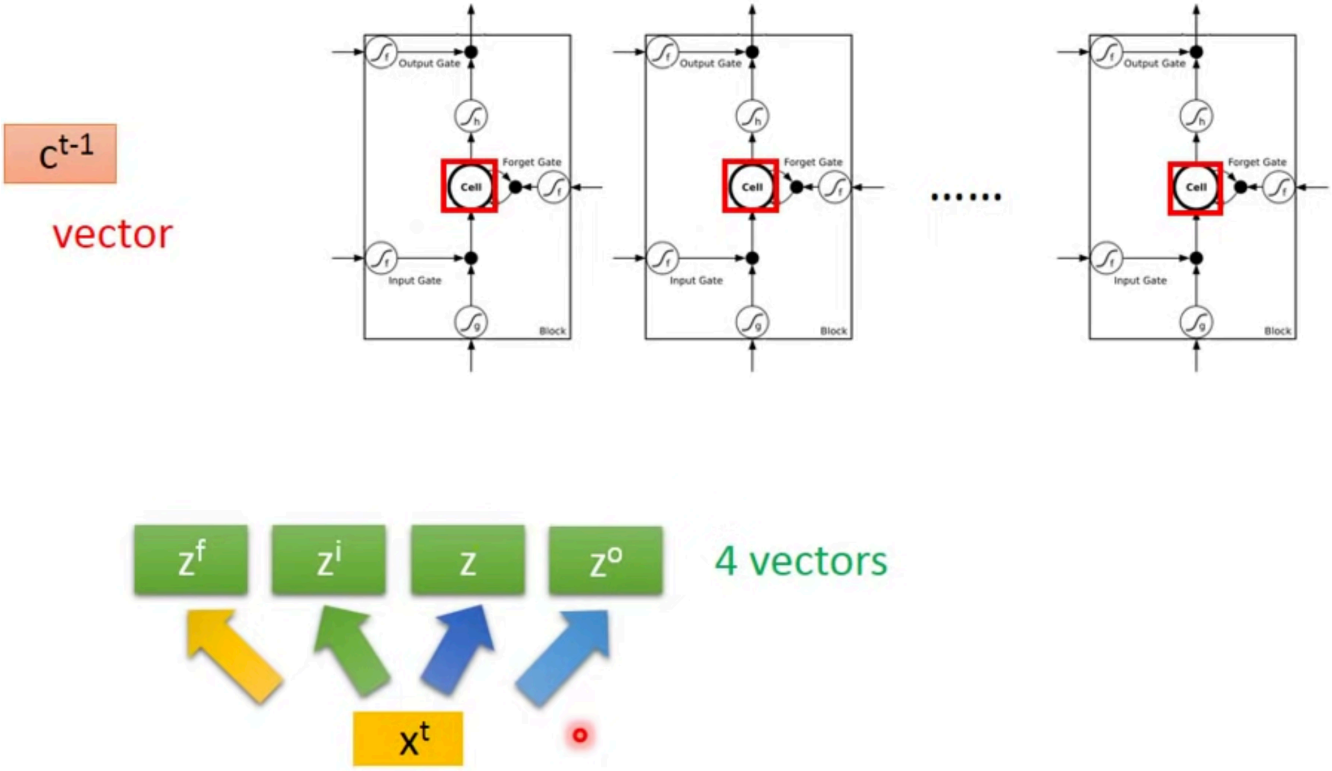
Original Network:

- Simply replace the neurons with LSTM

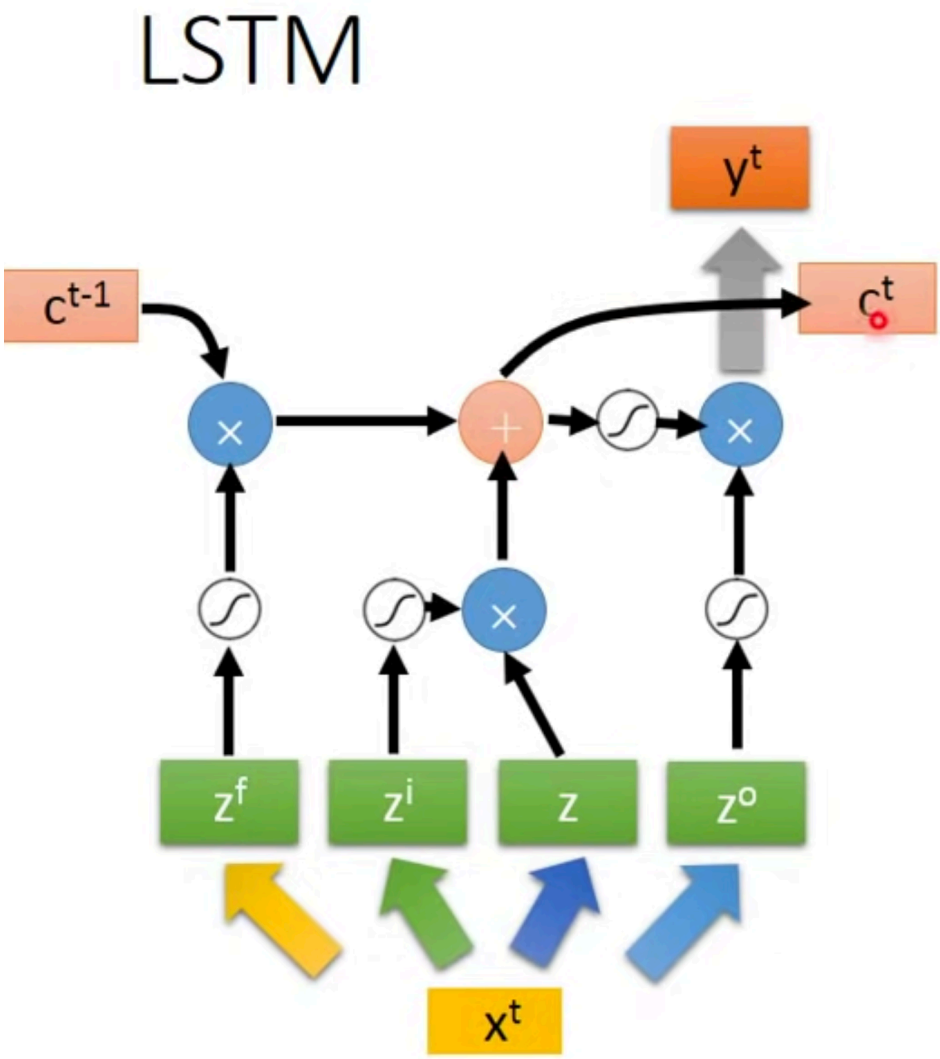




LSTM

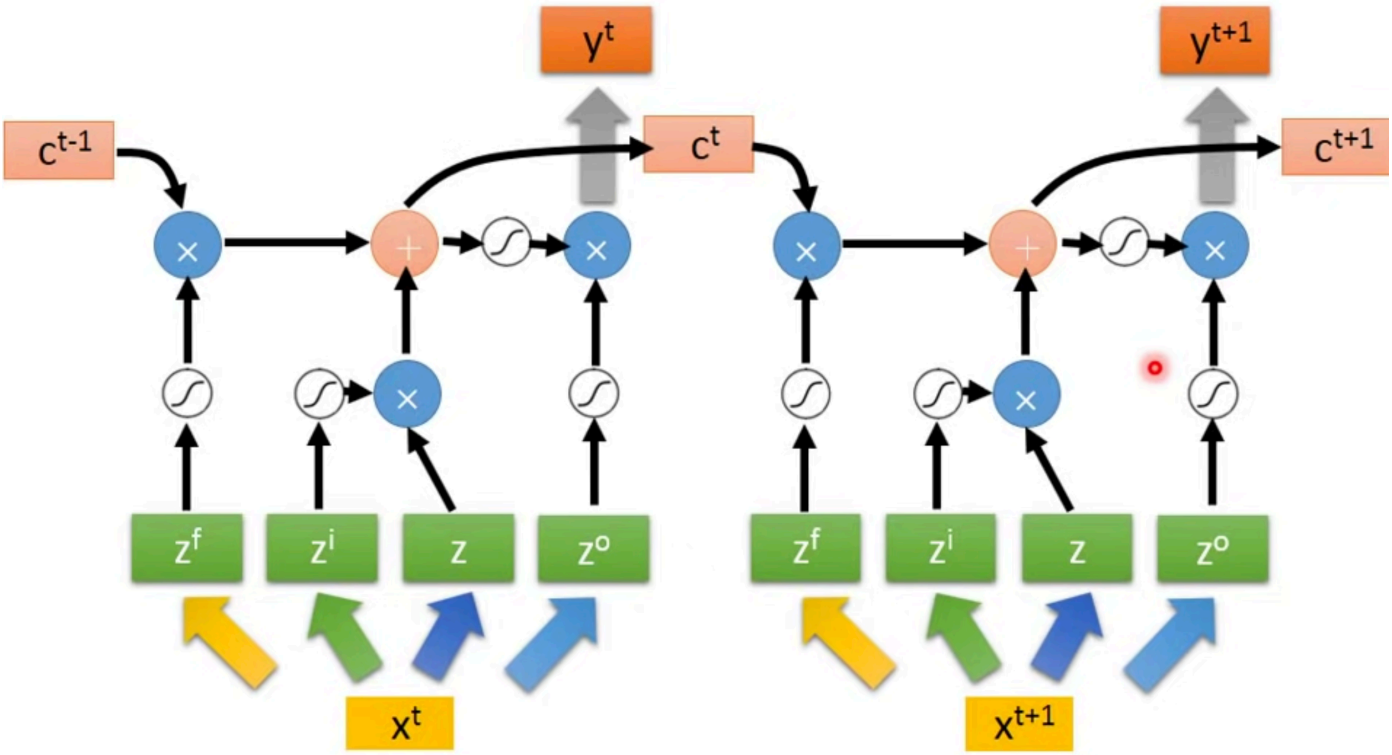


X^t分别乘以不同的权重矩阵,不同的分量控制不同的gate

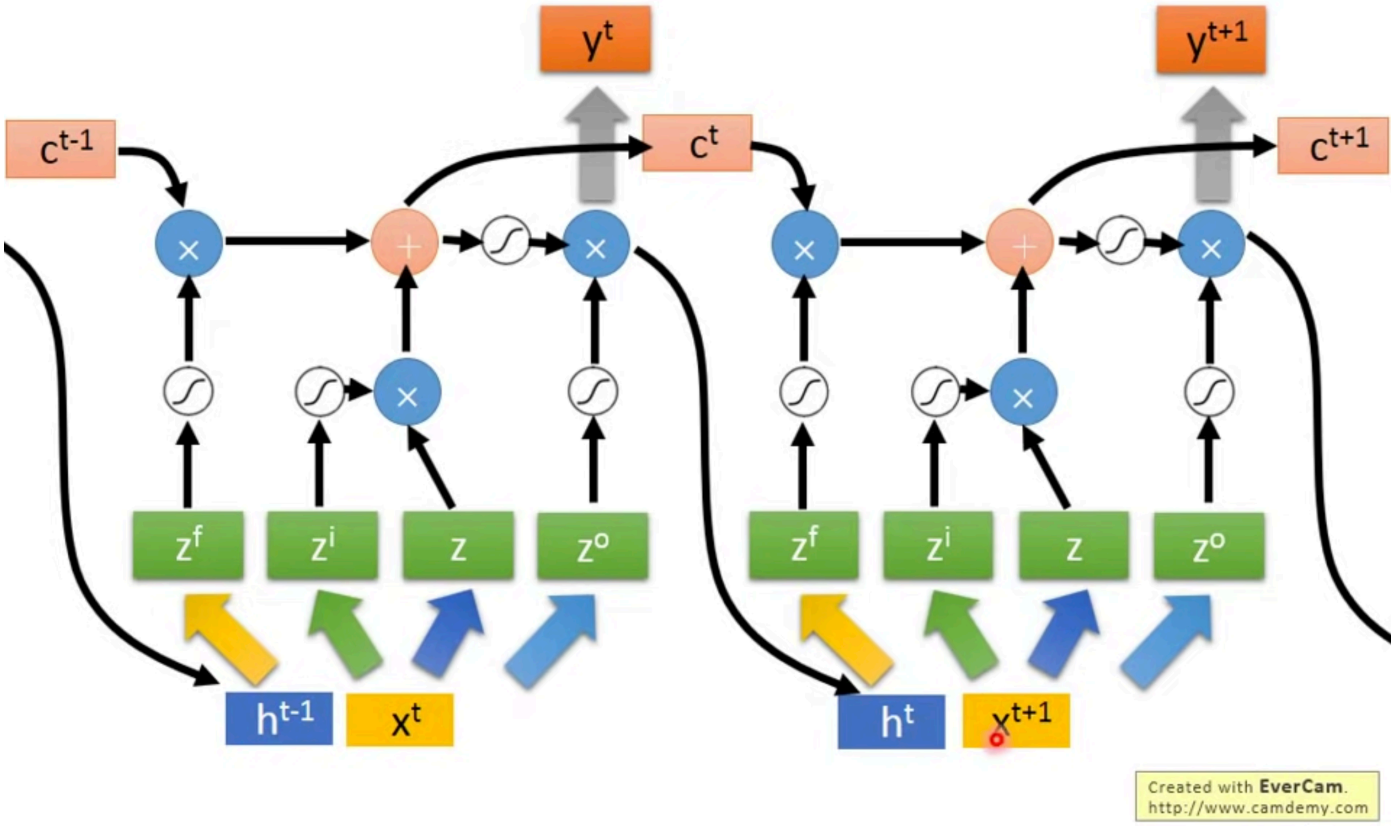


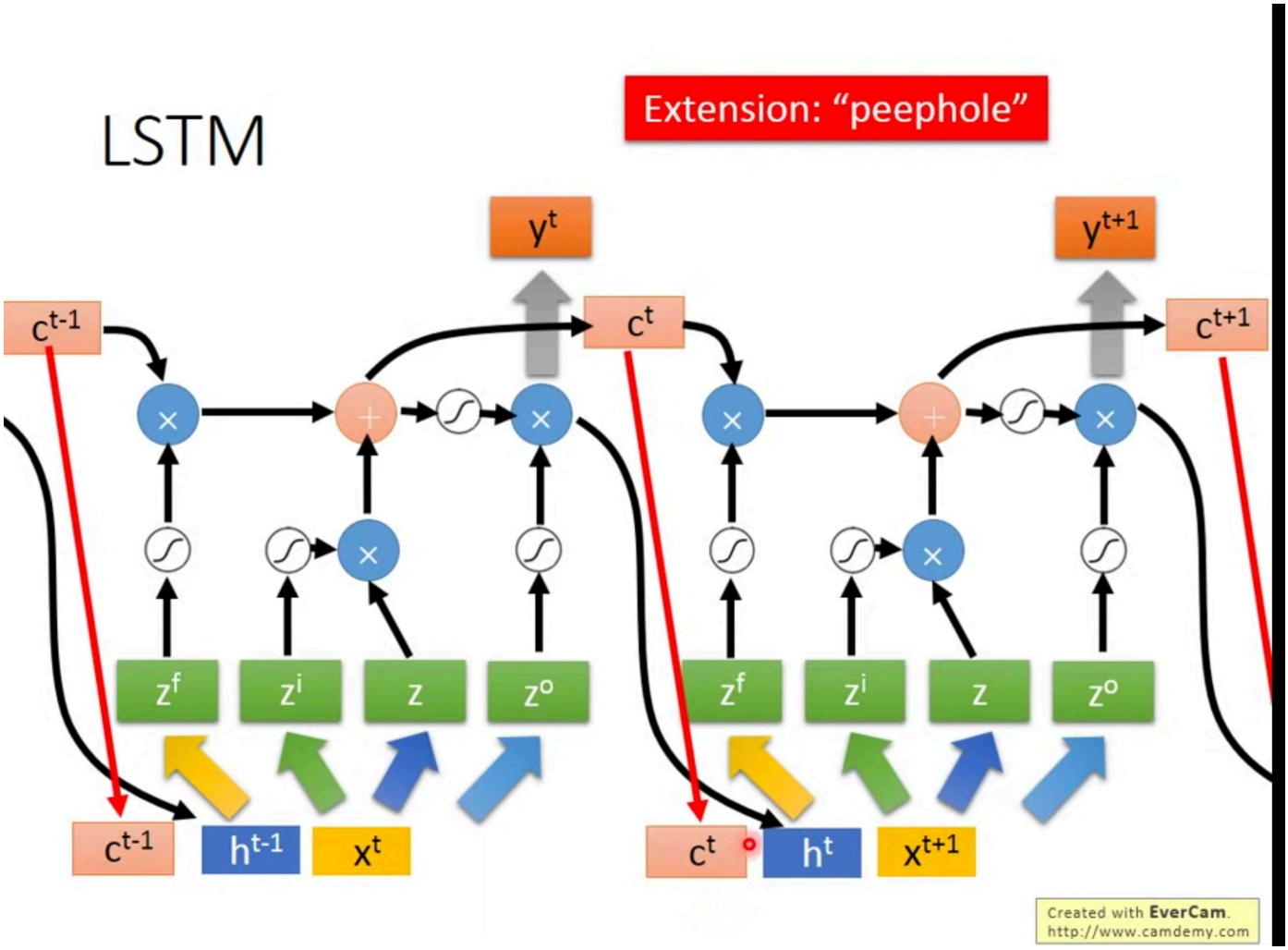
c^t代表存储的状态

LSTM



LSTM


y^t
y^t+1
c^t-1
c^t
c^t+1
z^f
z^i
z
z^o
h^t-1
x^t
h^t
x^t+1
Created with EverCam.
http://www.camdemy.com


LSTM
Extension: “peephole”
y^t
y^{t+1}
c^{t-1}
c^t
c^{t+1}
z^f
z^i
z
z^o
c^{t-1}
h^{t-1}
x^t
c^t
h^t
x^{t+1}
Created with EverCam.
http://www.camdemy.com