

Neural Network with Memory

Hung-yi Lee

Memory is important

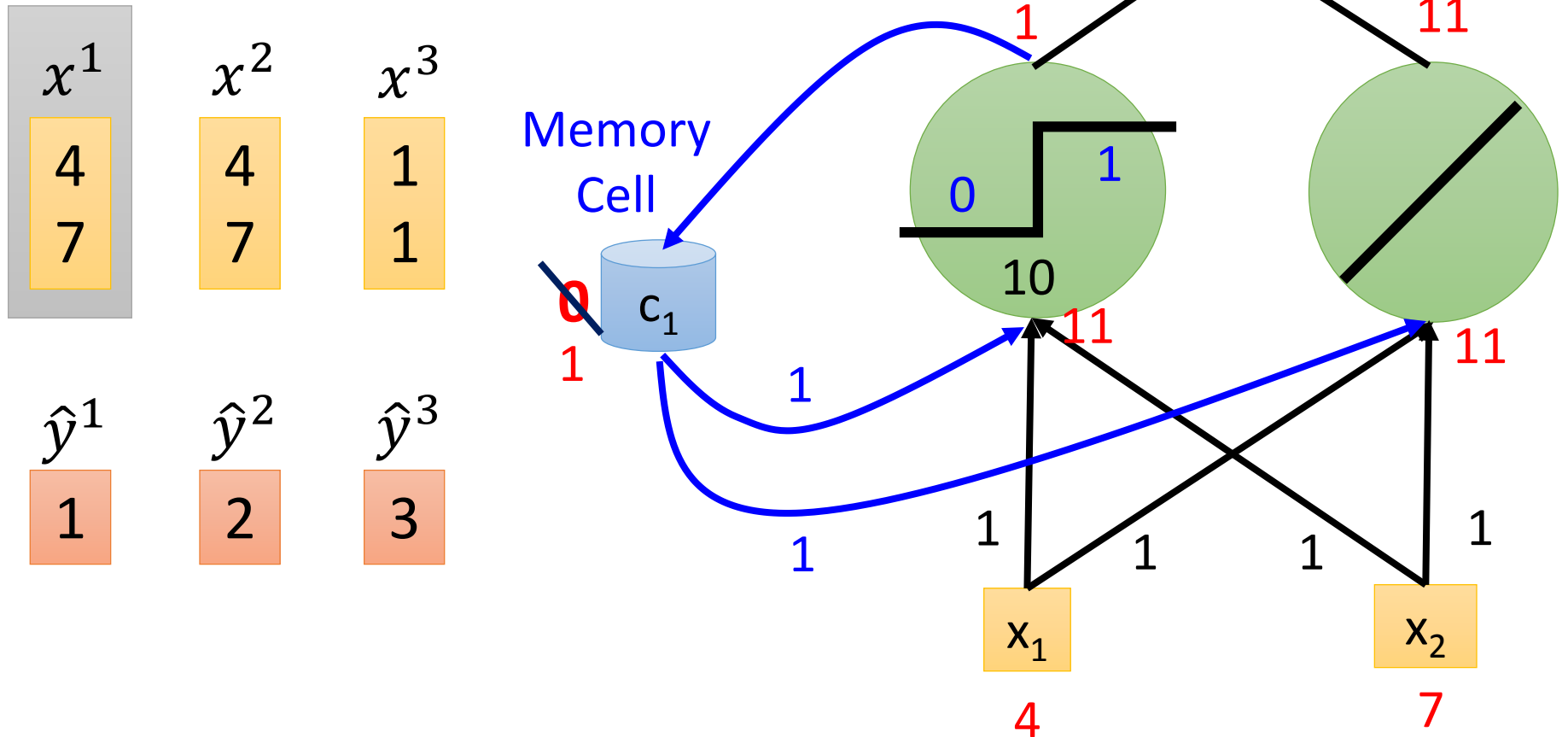
	x^1	x^2	x^3
Input: 2 dimensions	4 7	4 7	1 1
Output: 1 dimension	\hat{y}^1 1	\hat{y}^2 2	\hat{y}^3 3

$$\begin{array}{r} \\ \\ + \\ \hline 3 \end{array}$$

Network needs memory to achieve this

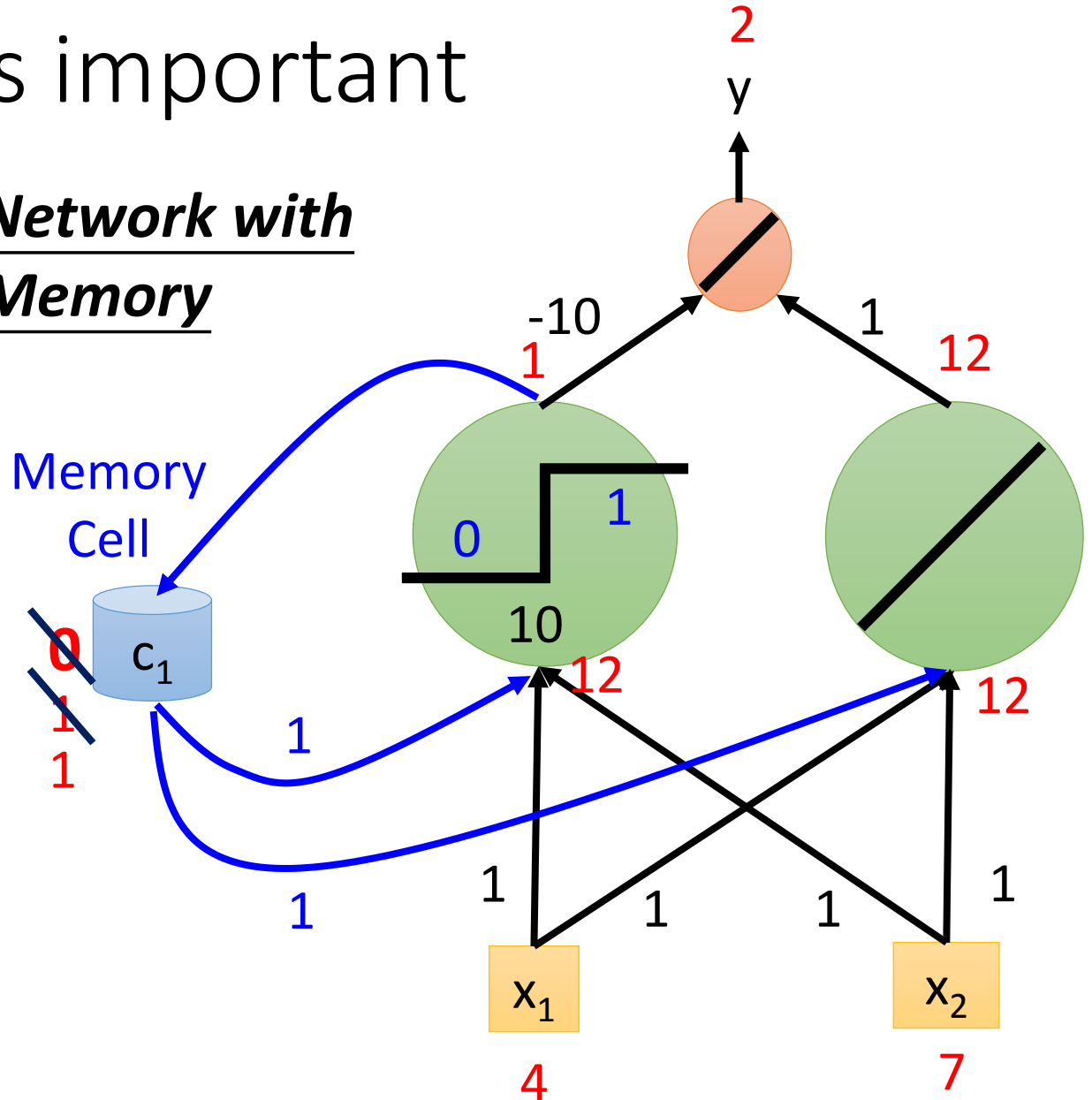
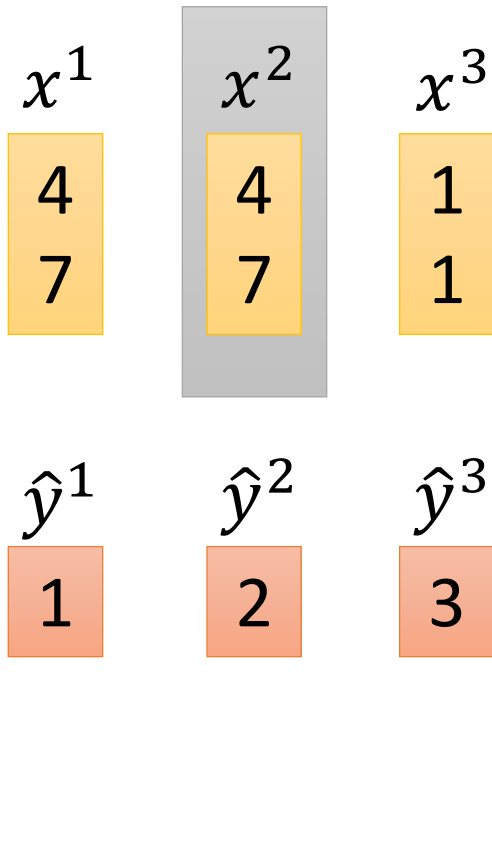
Memory is important

Network with Memory



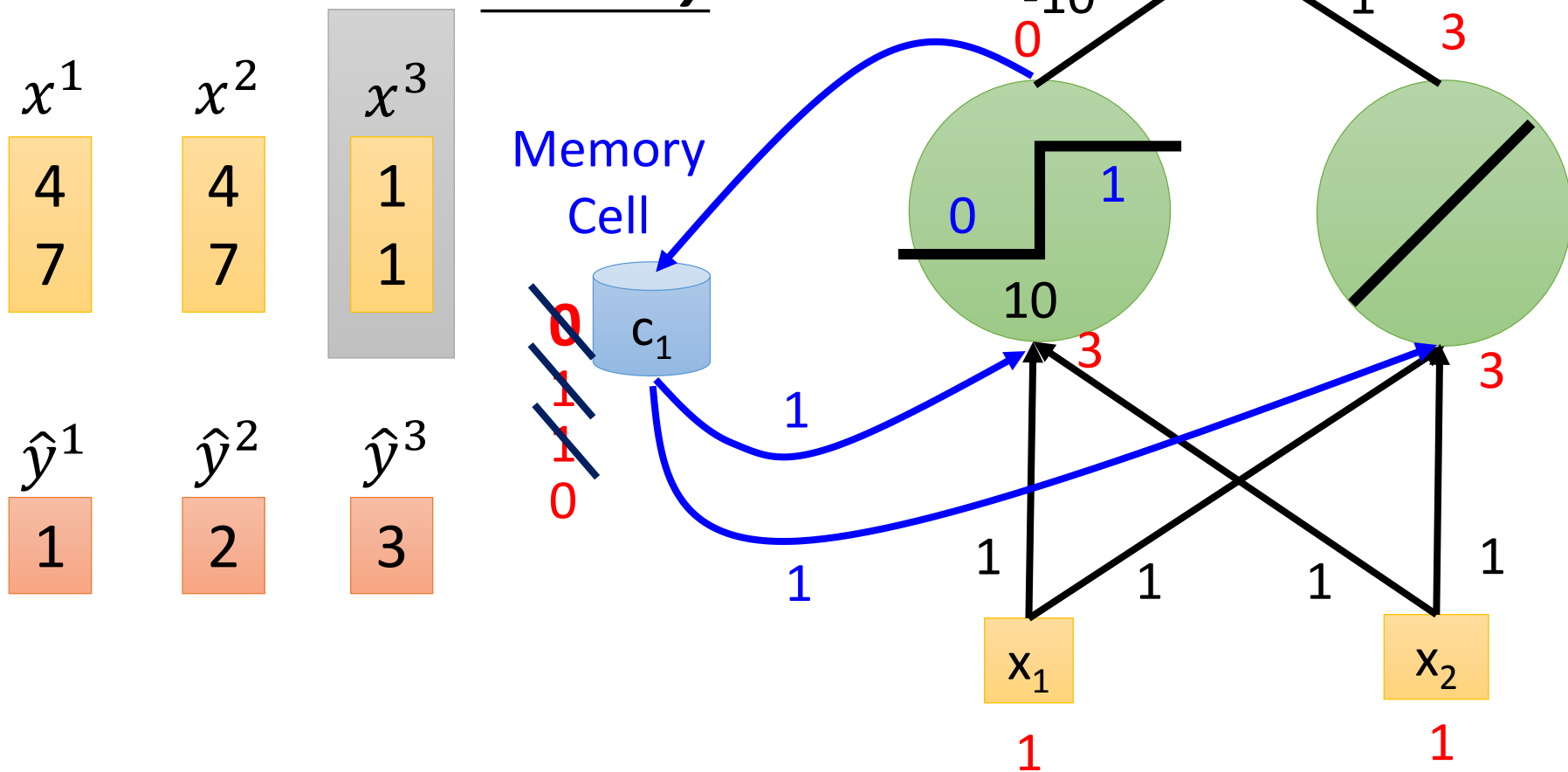
Memory is important

Network with Memory



Memory is important

Network with Memory



Outline

Vanilla Recurrent Neural Network (RNN)



Variants of RNN



Long Short-term Memory (LSTM)

Outline

Vanilla Recurrent Neural Network (RNN)

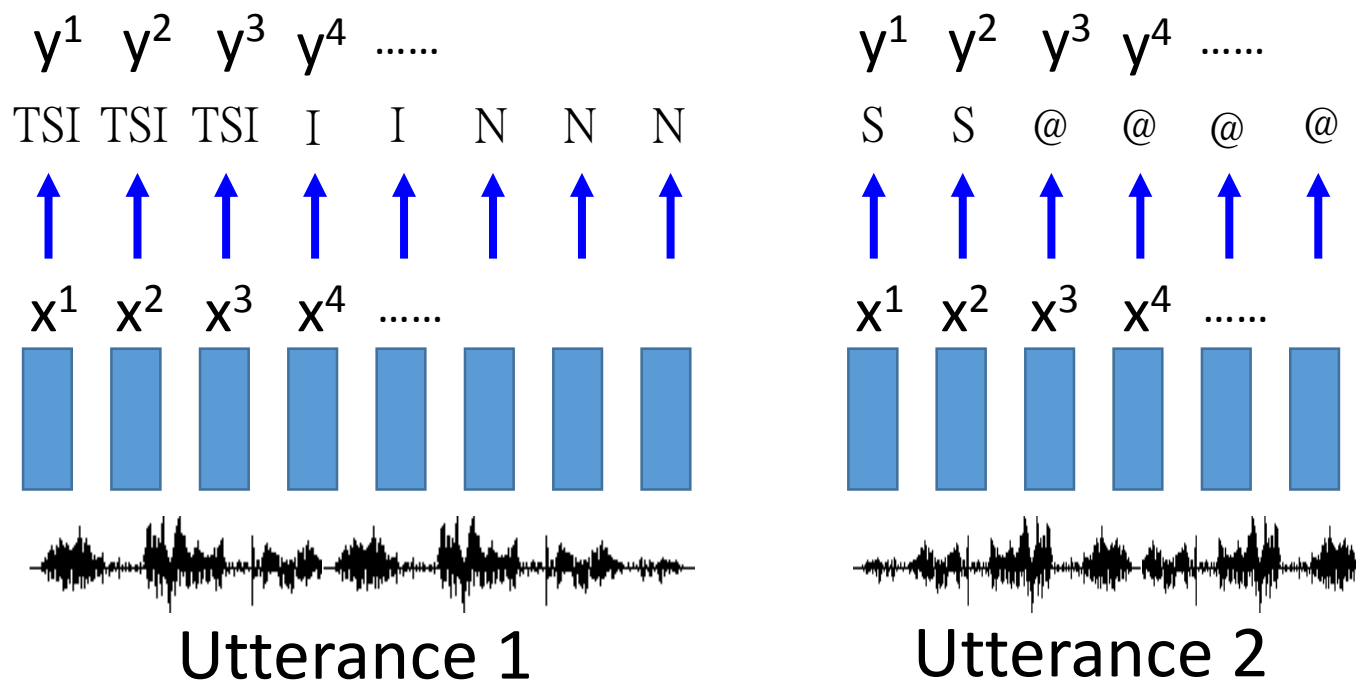
```
graph TD; A[Vanilla Recurrent Neural Network (RNN)] --> B[Variants of RNN]; B --> C[Long Short-term Memory (LSTM)];
```

Variants of RNN

Long Short-term Memory (LSTM)

Application

- (Simplified) Speech Recognition

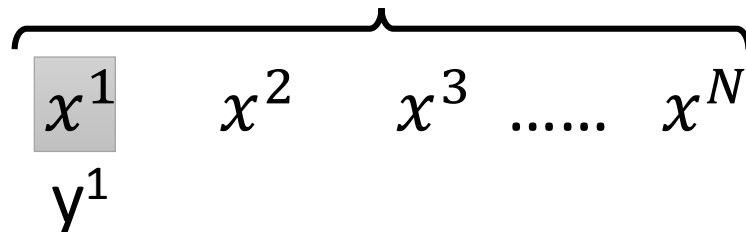


We use DNN. All the frames are considered independently.

RNN

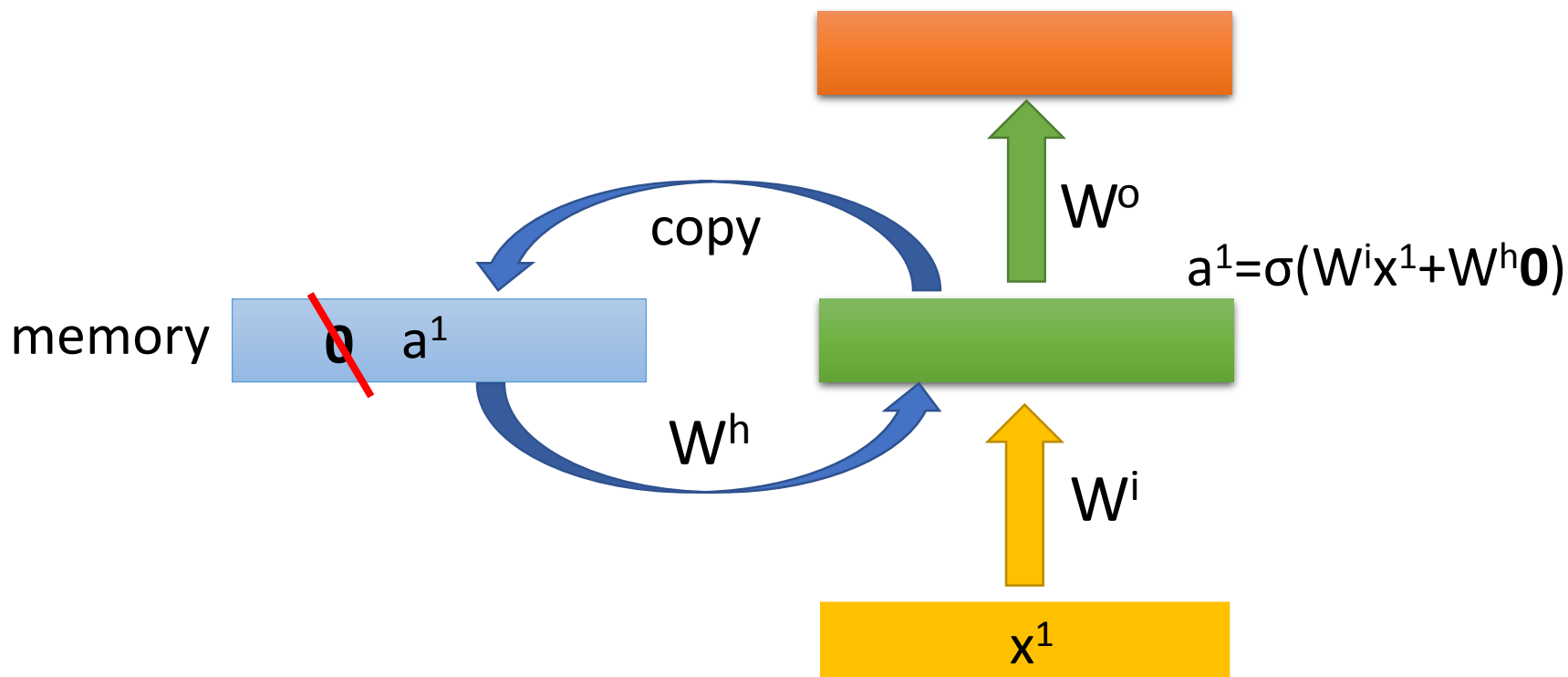
The order cannot change.

RNN input:



Input of RNN is one utterance

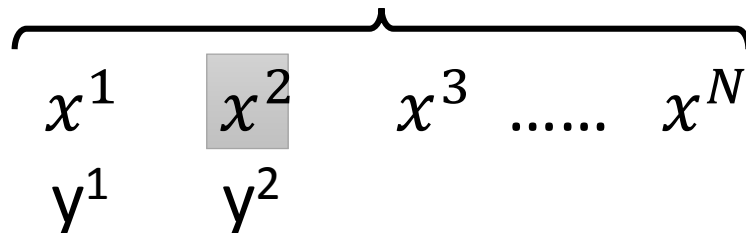
$$y^1 = \text{softmax}(W^o a^1)$$



RNN

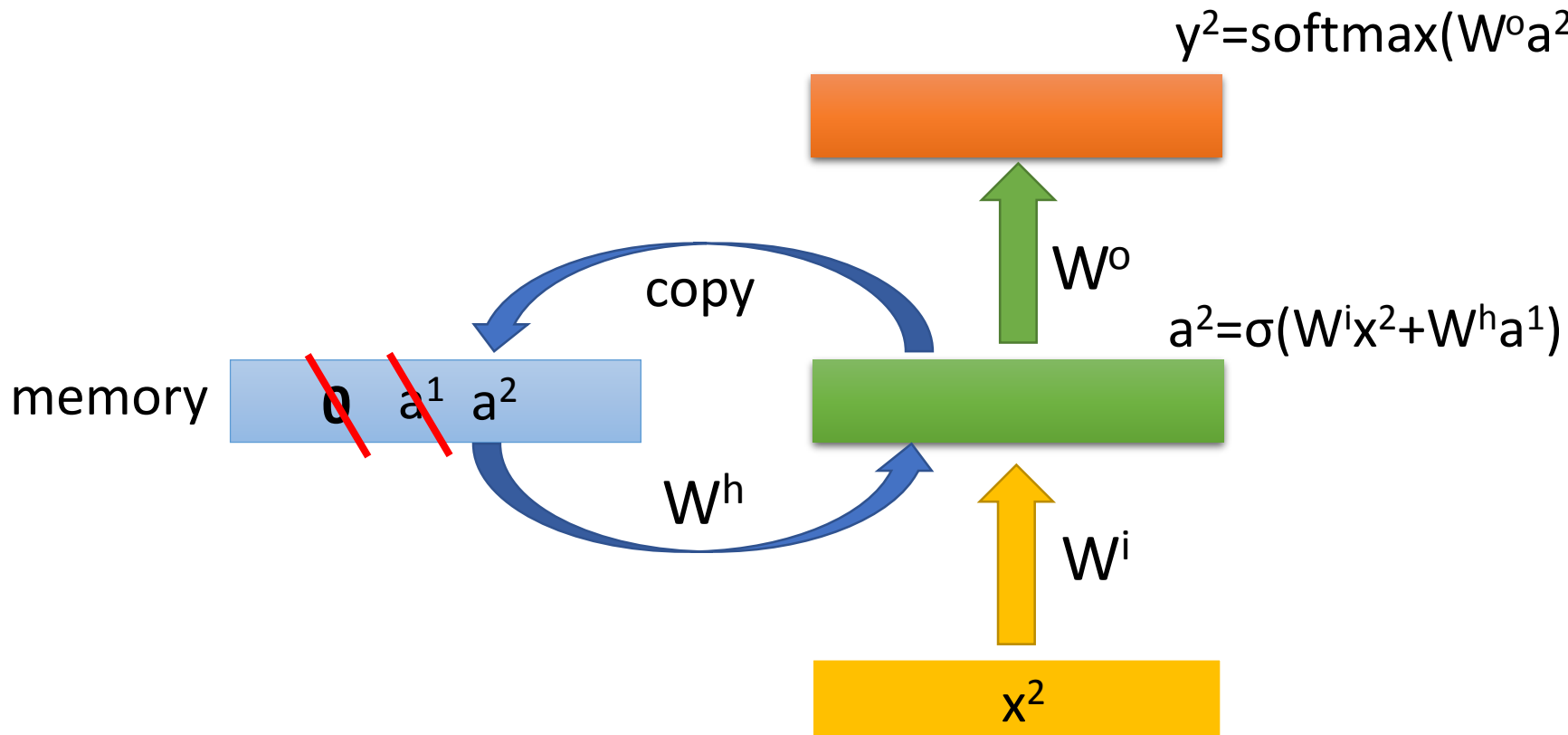
The order cannot change.

RNN input:



Input of RNN is one utterance

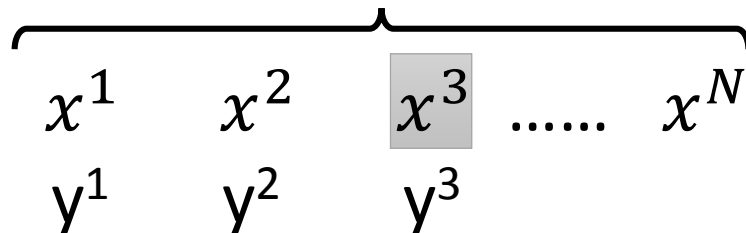
$$y^2 = \text{softmax}(W^o a^2)$$



RNN

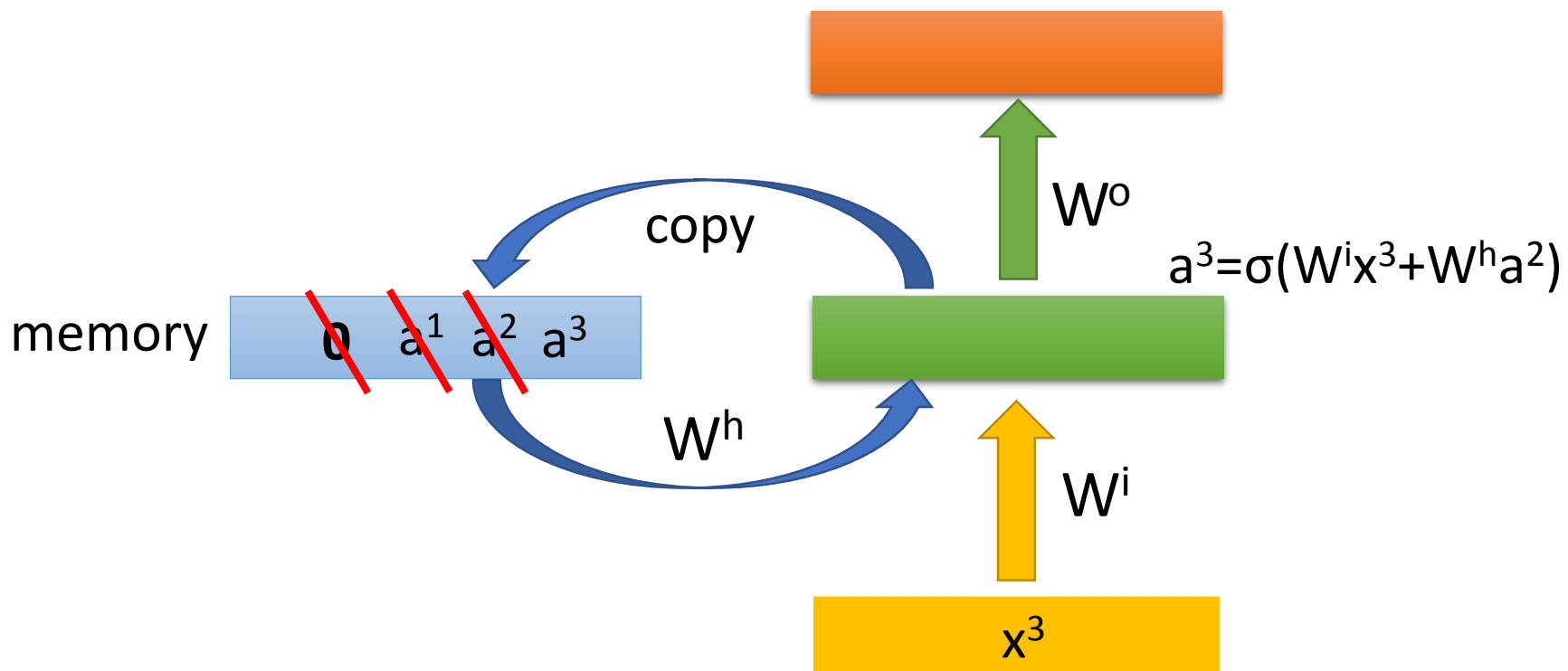
The order cannot change.

RNN input:



Input of RNN is one utterance

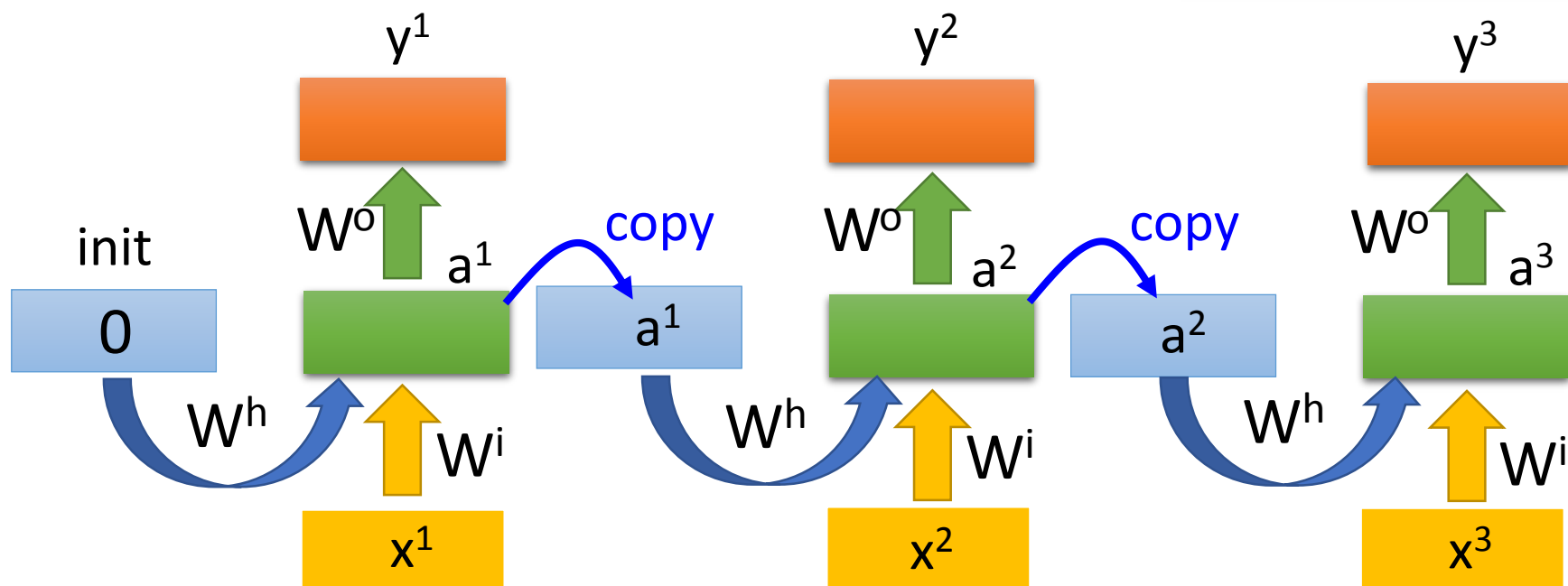
$$y^3 = \text{softmax}(W^o a^3)$$



RNN

Input data: x^1 x^2 x^3 x^N

Input of RNN is one utterance



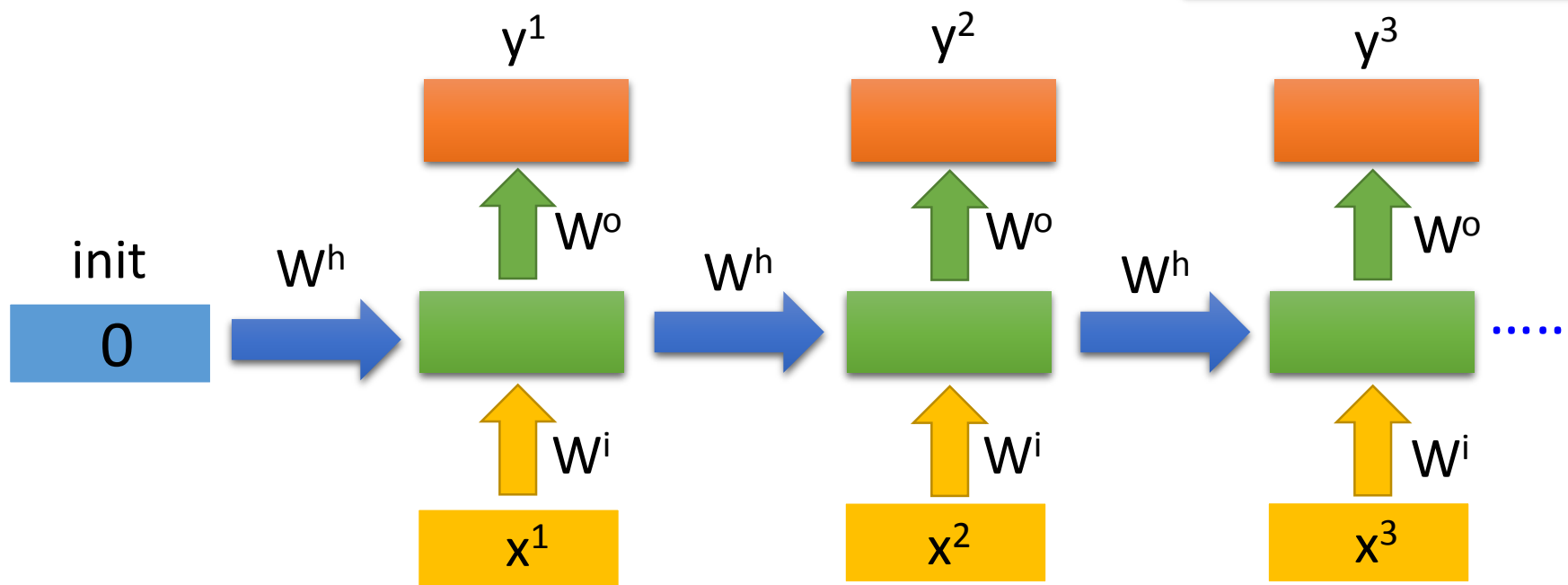
The same network is used again and again.

Output y^i depends on x^1, x^2, \dots, x^i

RNN

Input data: x^1 x^2 x^3 x^N

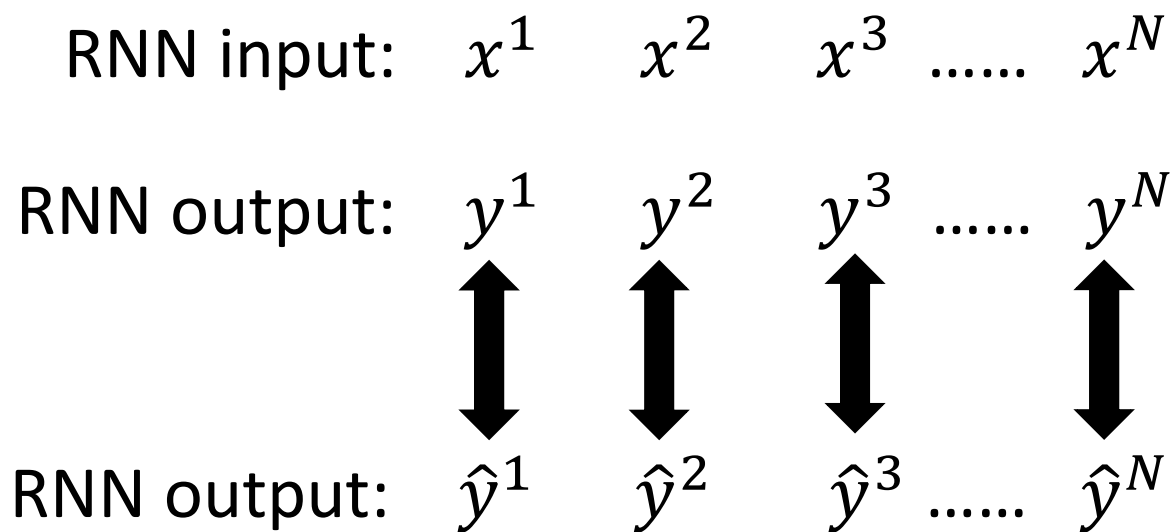
Input of RNN is one utterance



The same network is used again and again.

Output y^i depends on x^1, x^2, \dots, x^i

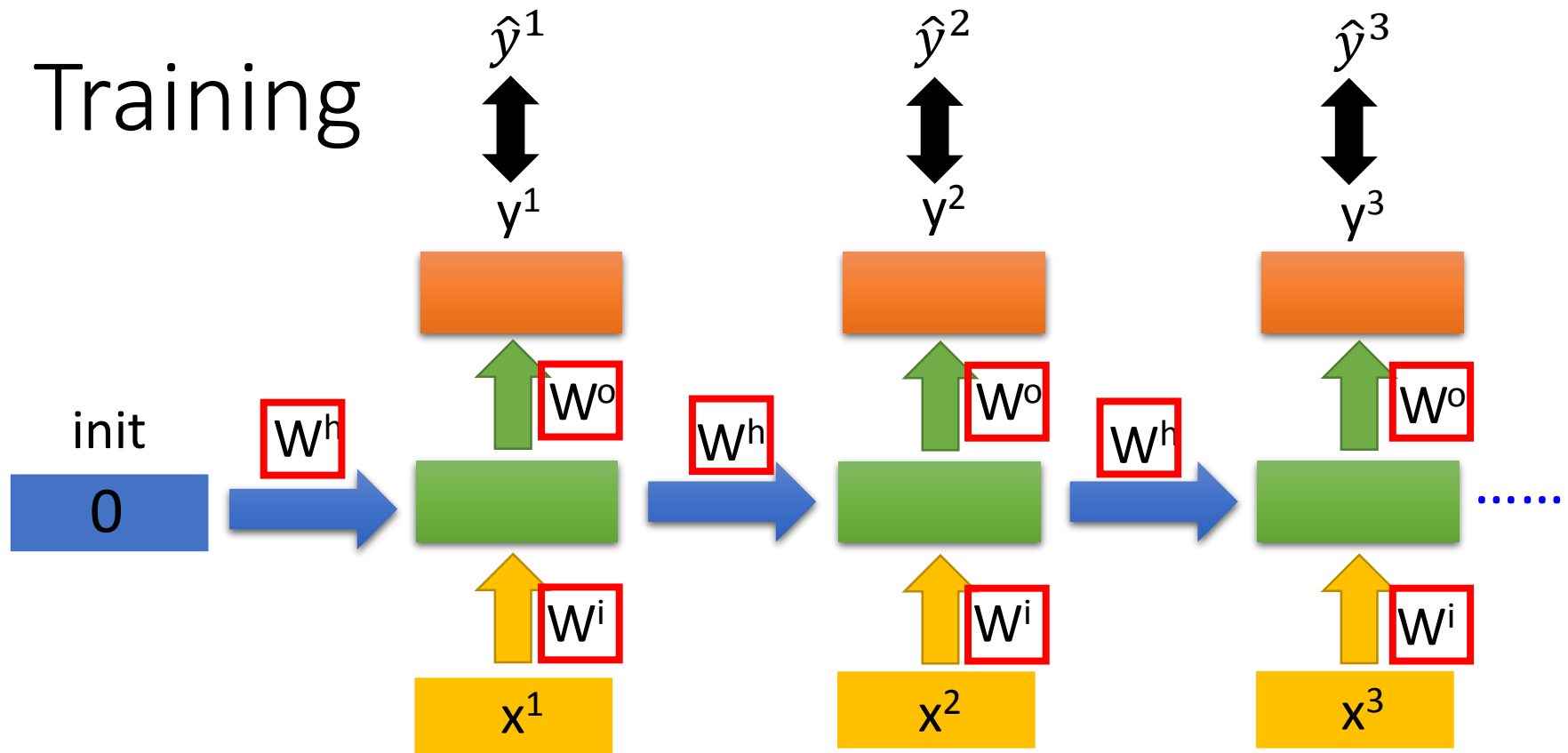
Cost



$$C = \frac{1}{2} \sum_{n=1}^N \|y^n - \hat{y}^n\|^2$$

$$C = \frac{1}{2} \sum_{n=1}^N -\log y_{r^n}^n$$

Training



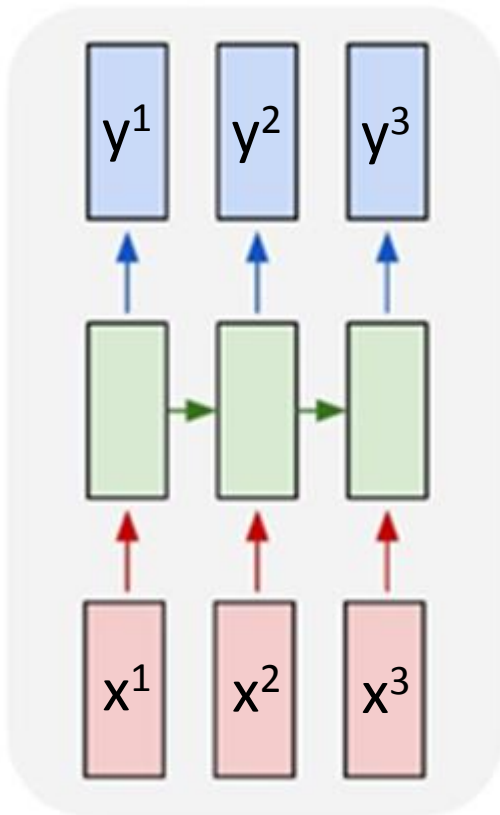
w is an element in W^h , W^i or $W^o \Rightarrow w \leftarrow w - \eta \partial C / \partial w$

➡ Backpropagation through time (BPTT)

RNN Training is very difficult in practice.

More Applications

- Input and output are vector sequences with *the same length*



y^1 y^2 y^3 y^4
PN V D N
↑ ↑ ↑ ↑
 x^1 x^2 x^3 x^4
John saw the saw.

POS Tagging

More Applications

- Name entity recognition
 - Identifying names of people, places, organizations, etc. from a sentence
 - **Harry Potter** is a student of **Hogwarts** and lived on **Privet Drive**.
 - **people, organizations, places**, not a name entity
- Information extraction
 - Extract pieces of information relevant to a specific application, e.g. flight booking
 - I would like to leave **Boston** on **November 2nd** and arrive in **Taipei** before **2 p.m.**
 - **place of departure, destination, time of departure, time of arrival**, other

Outline

Vanilla Recurrent Neural Network (RNN)



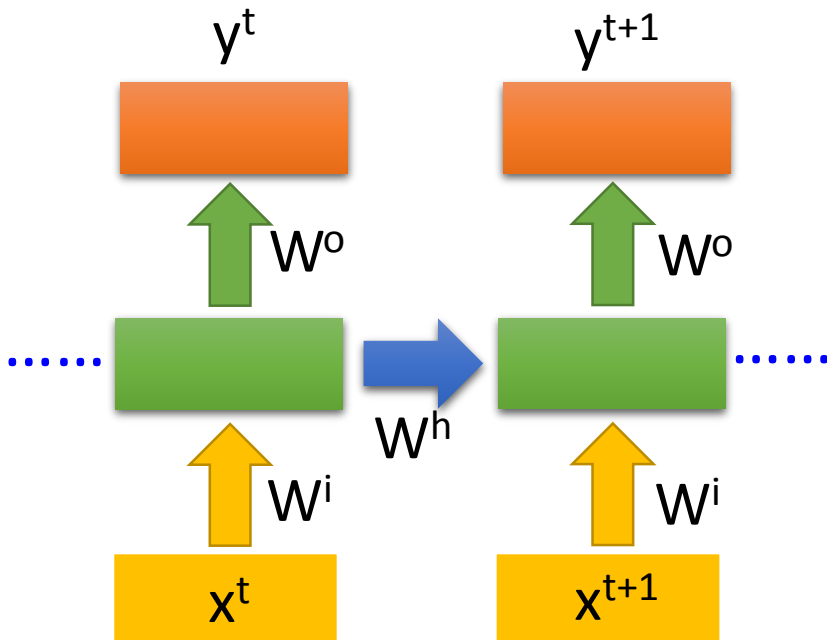
Variants of RNN



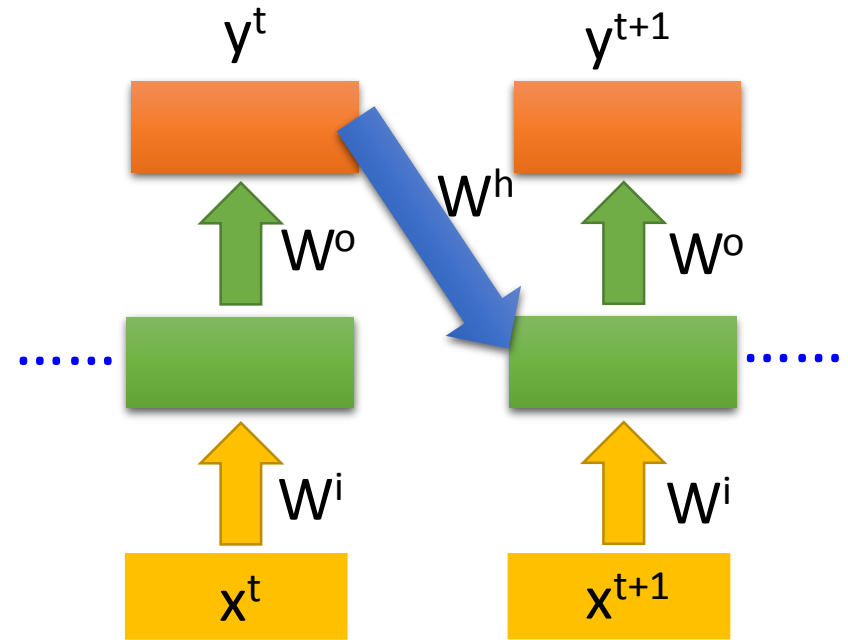
Long Short-term Memory (LSTM)

Elman Network & Jordan Network

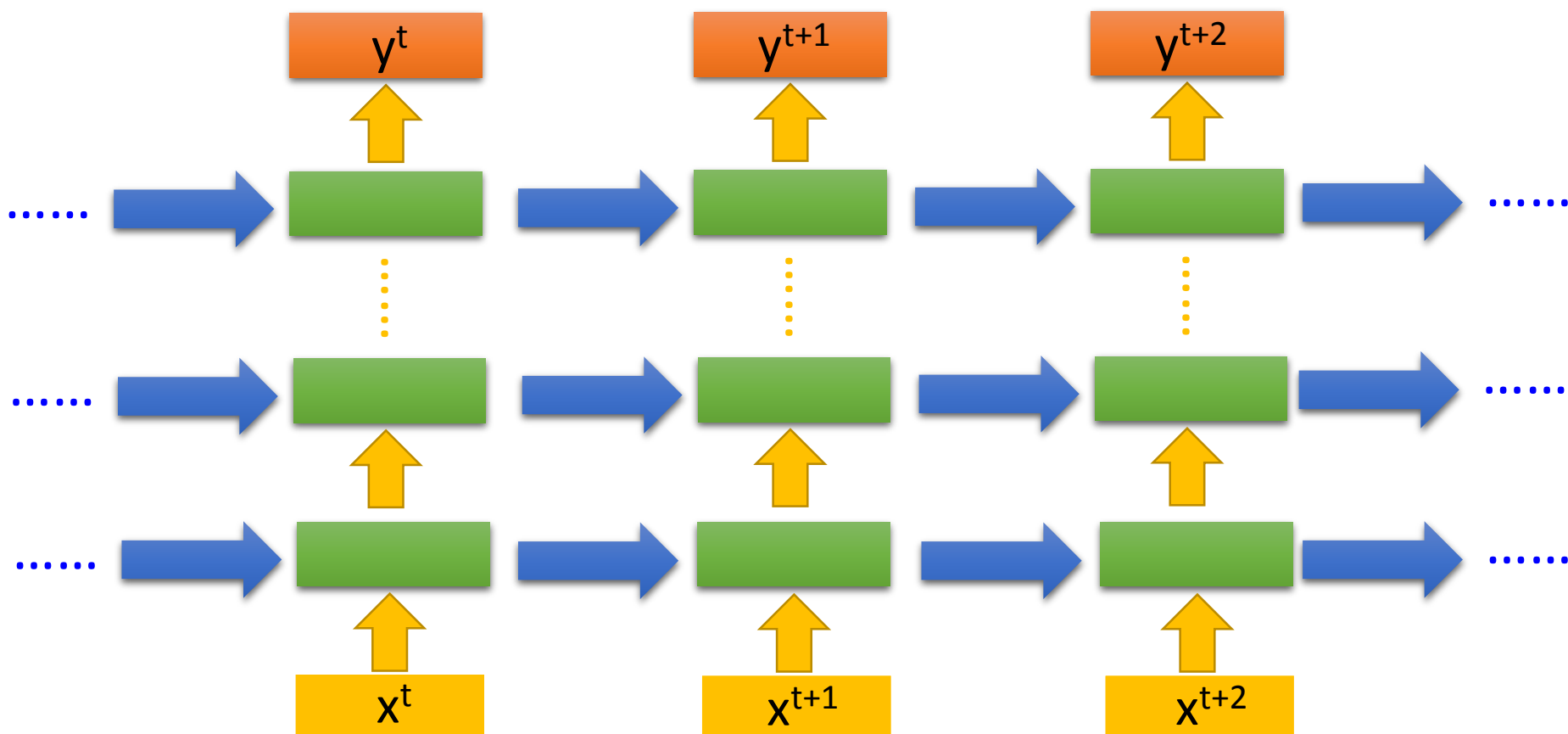
Elman Network



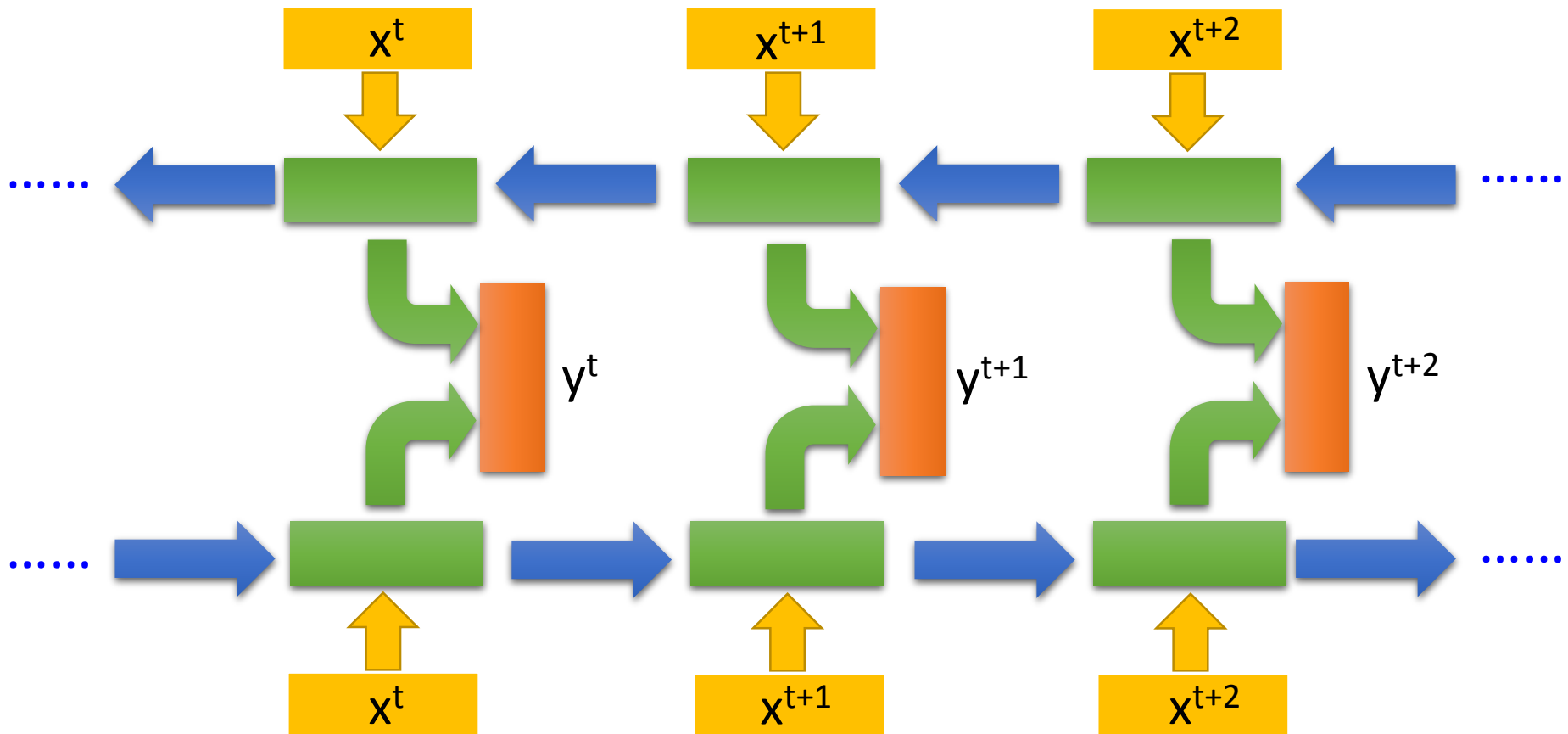
Jordan Network



Deep RNN



Bidirectional RNN



Many to one

- Input is a vector sequence, but output is only one vector

Sentiment Analysis

看了這部電影覺得很高興

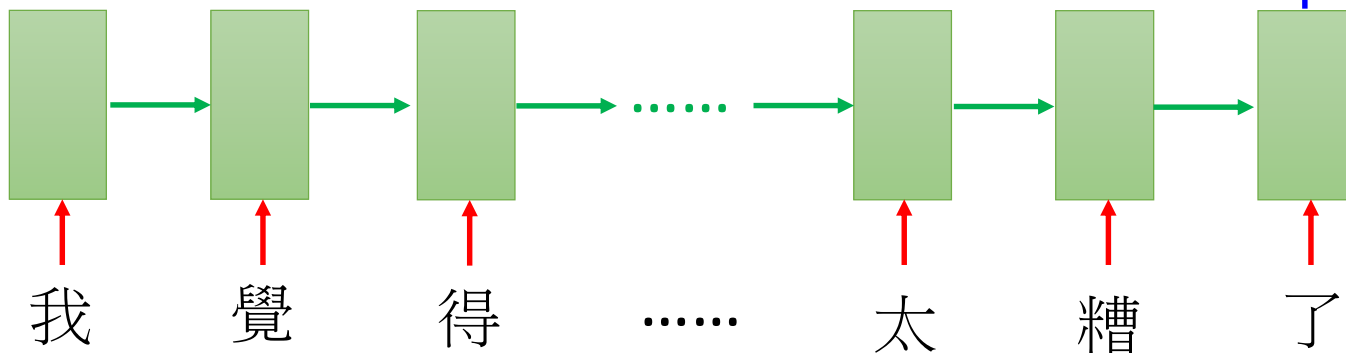
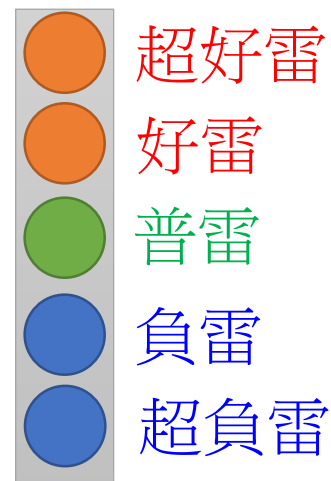
Positive (正雷)

這部電影太糟了

Negative (負雷)

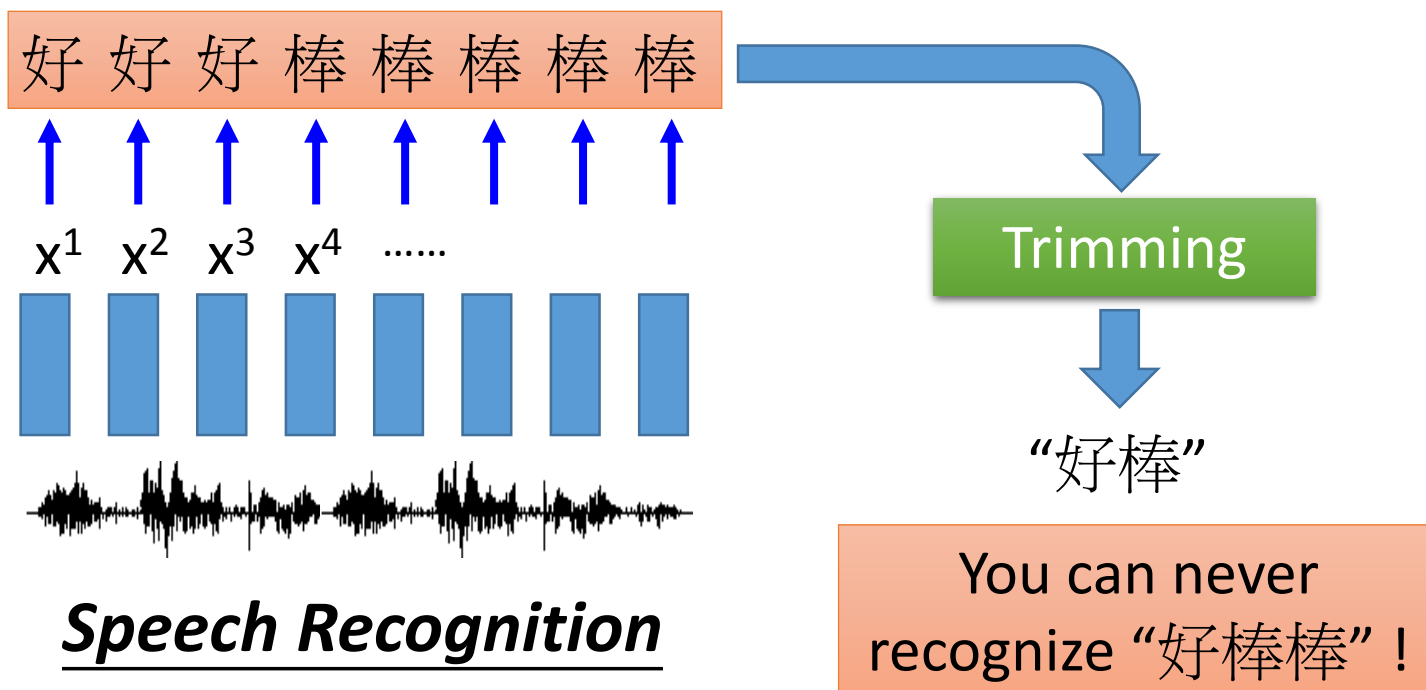
這部電影很棒

Positive (正雷)




Many to Many (Output is shorter)

- Both input and output are vector sequences, **but the output is shorter.**



Many to Many (Output is shorter)

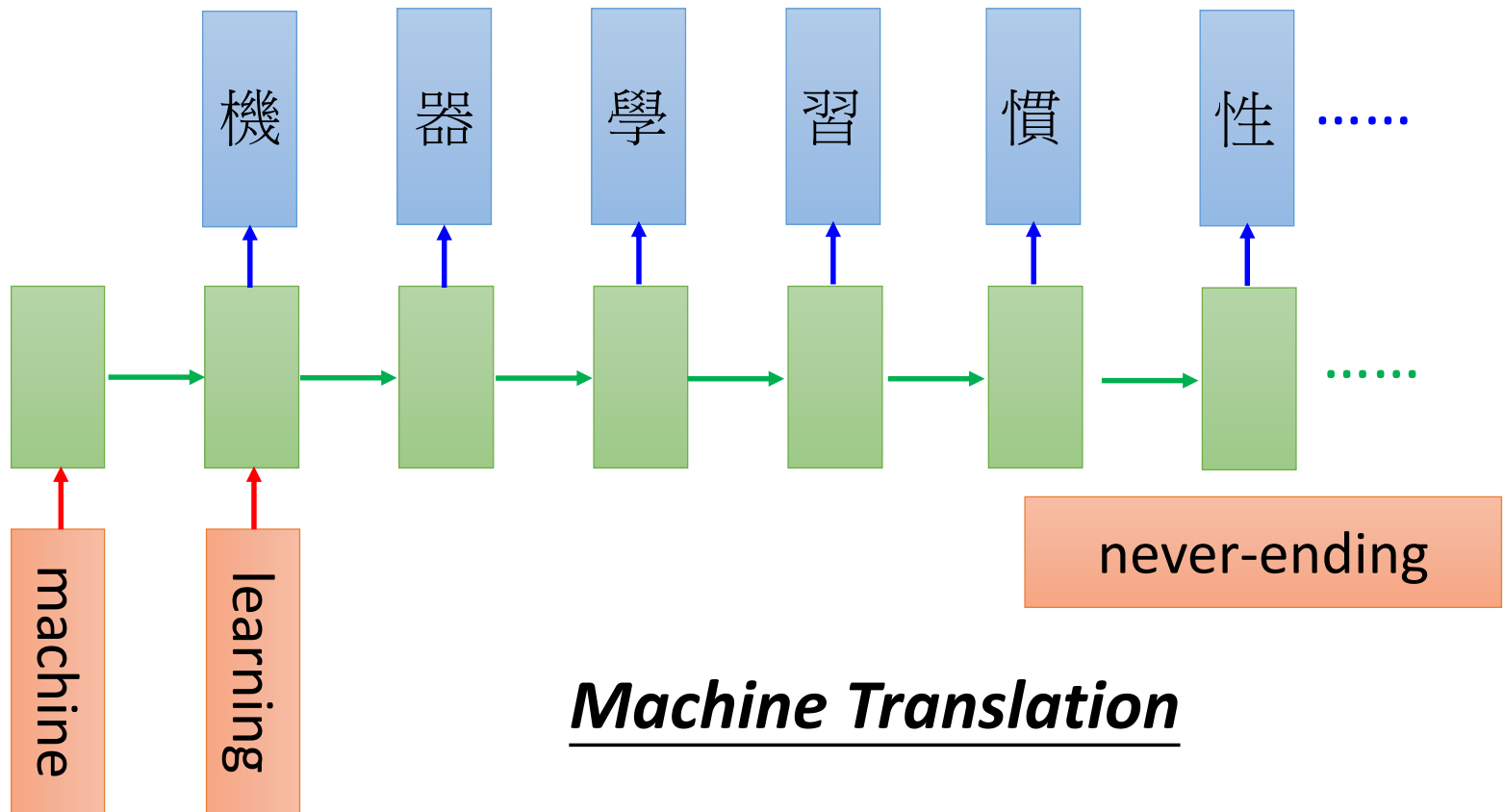
- Both input and output are vector sequences, **but the output is shorter.**
- Connectionist Temporal Classification (CTC)
 - Add an extra symbol “ ϕ ” (同上)

好 ϕ ϕ 棒 ϕ ϕ ϕ ϕ  “好棒”

好 ϕ ϕ 棒 ϕ 棒 ϕ ϕ  “好棒棒”

Many to Many (No Limitation)

- Both input and output are vector sequences with different lengths. → Sequence to sequence learning



Many to Many (No Limitation)

- 推文接龍

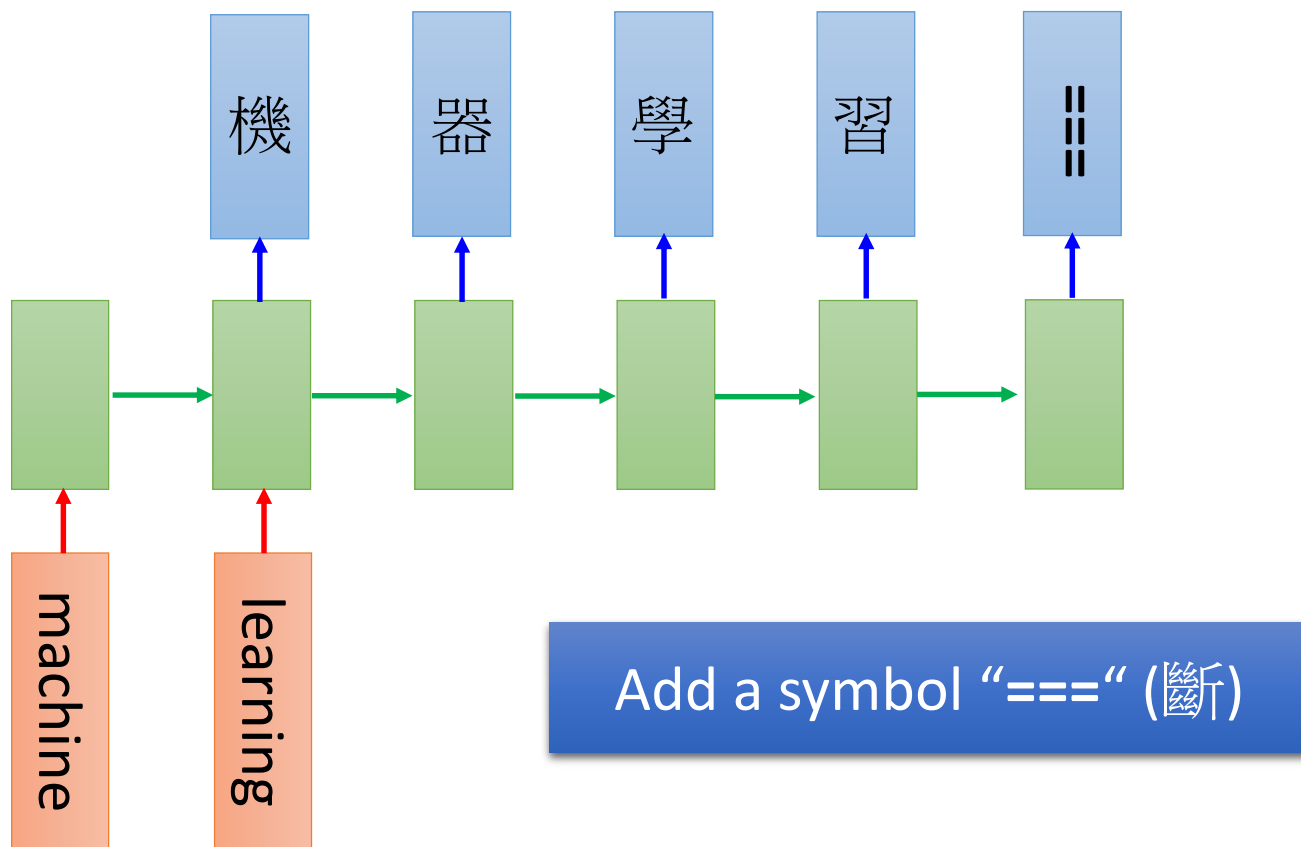
- Ref: <http://pttpedia.pixnet.net/blog/post/168133002-%E6%8E%A5%E9%BE%8D%E6%8E%A8%E6%96%87>

推xxx: ptt萬歲
推dd: 歲平安
噓dddf: 全
推zzzzzzzzzzzz: 家就是你家
 ⋮

推tlkagk: =====斷=====

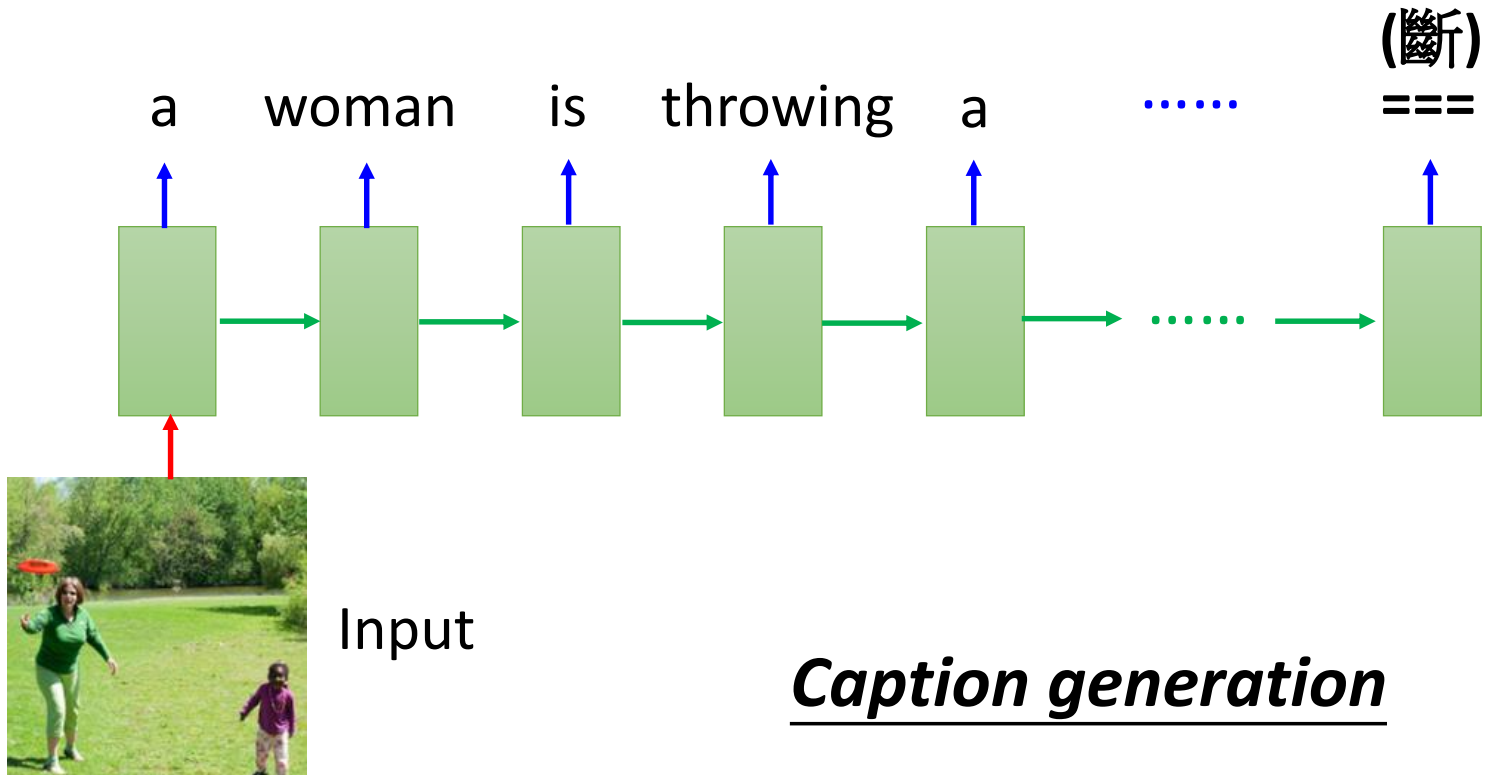
Many to Many (No Limitation)

- Both input and output are vector sequences with different lengths. → Sequence to sequence learning



One to Many

- Input is one vector, but output is a vector sequence



Outline

Vanilla Recurrent Neural Network (RNN)

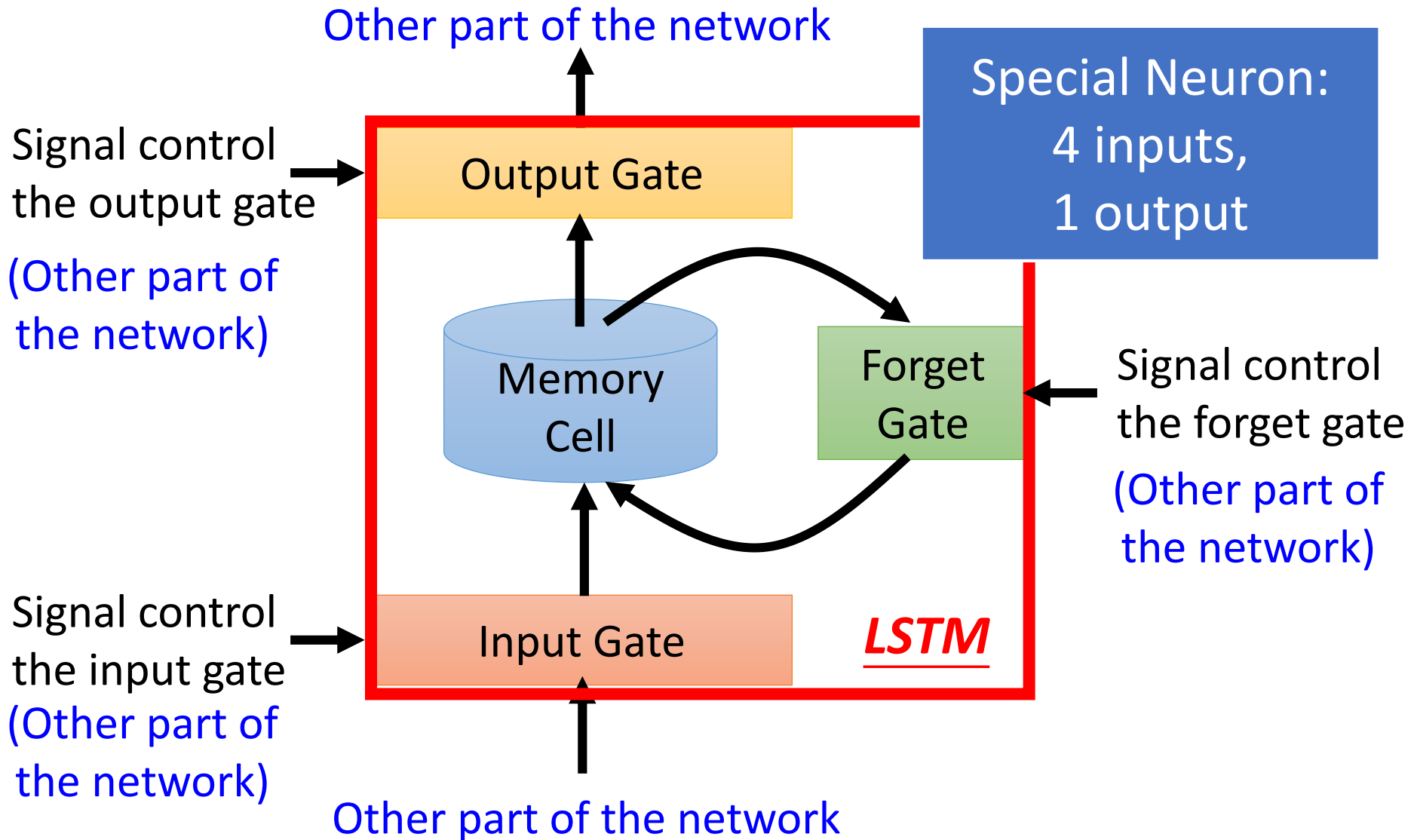


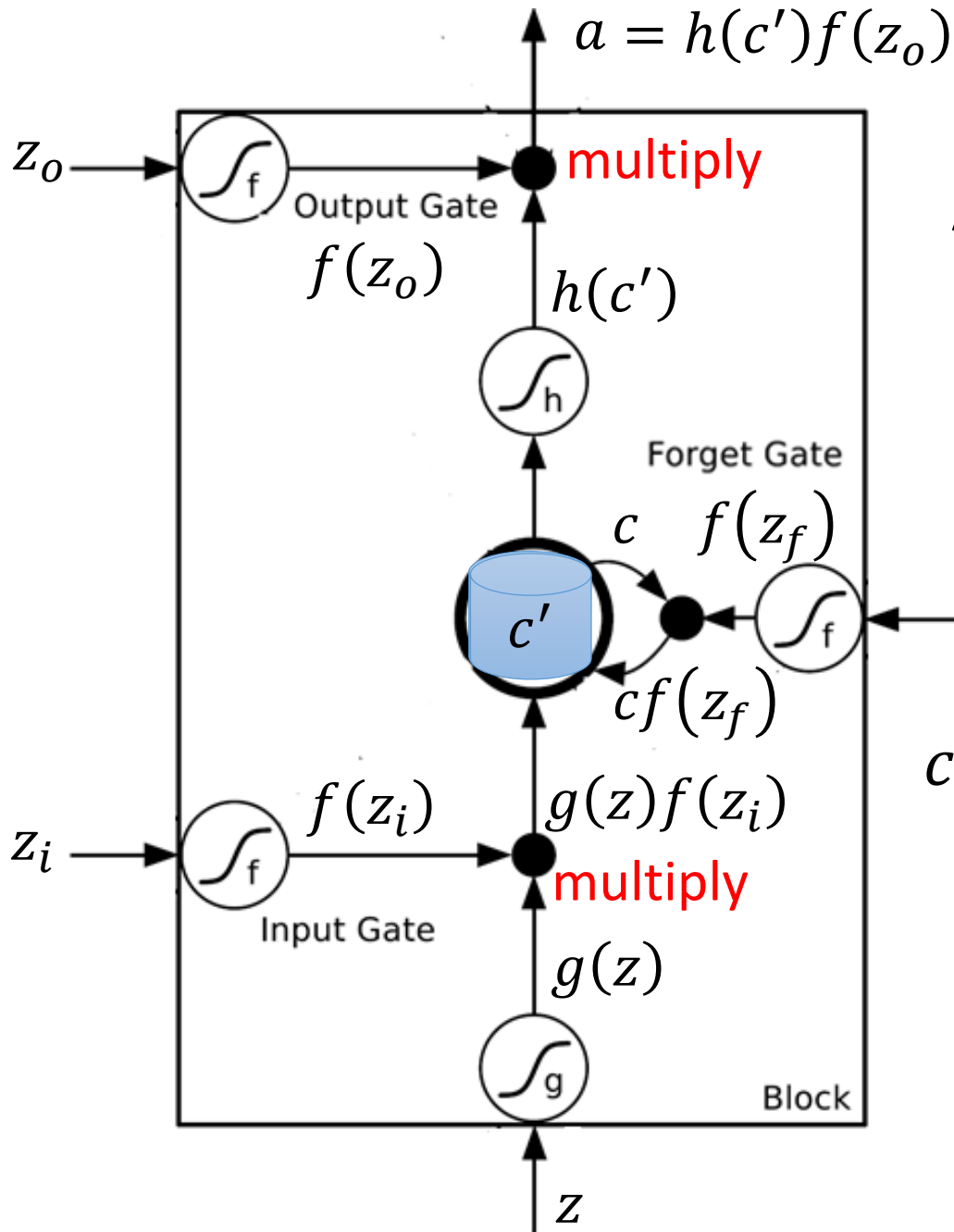
Variants of RNN



Long Short-term Memory (LSTM)

Long Short-term Memory (LSTM)





Activation function f is usually a sigmoid function

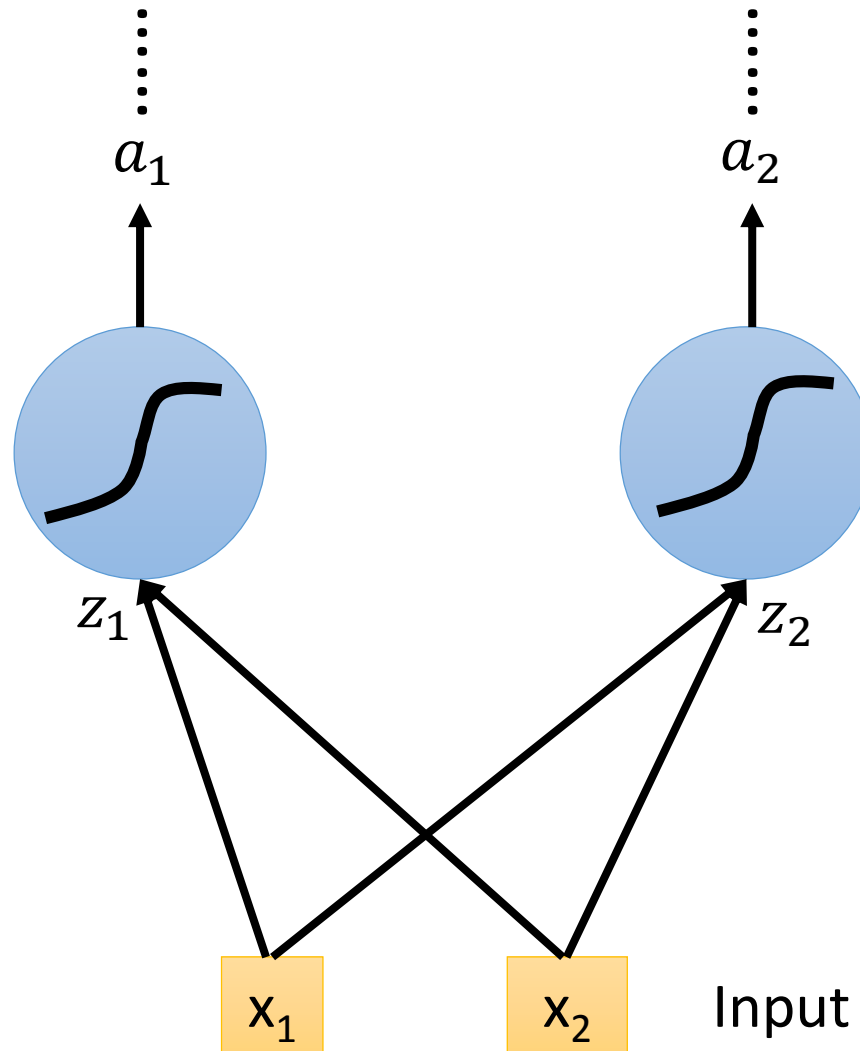
Between 0 and 1

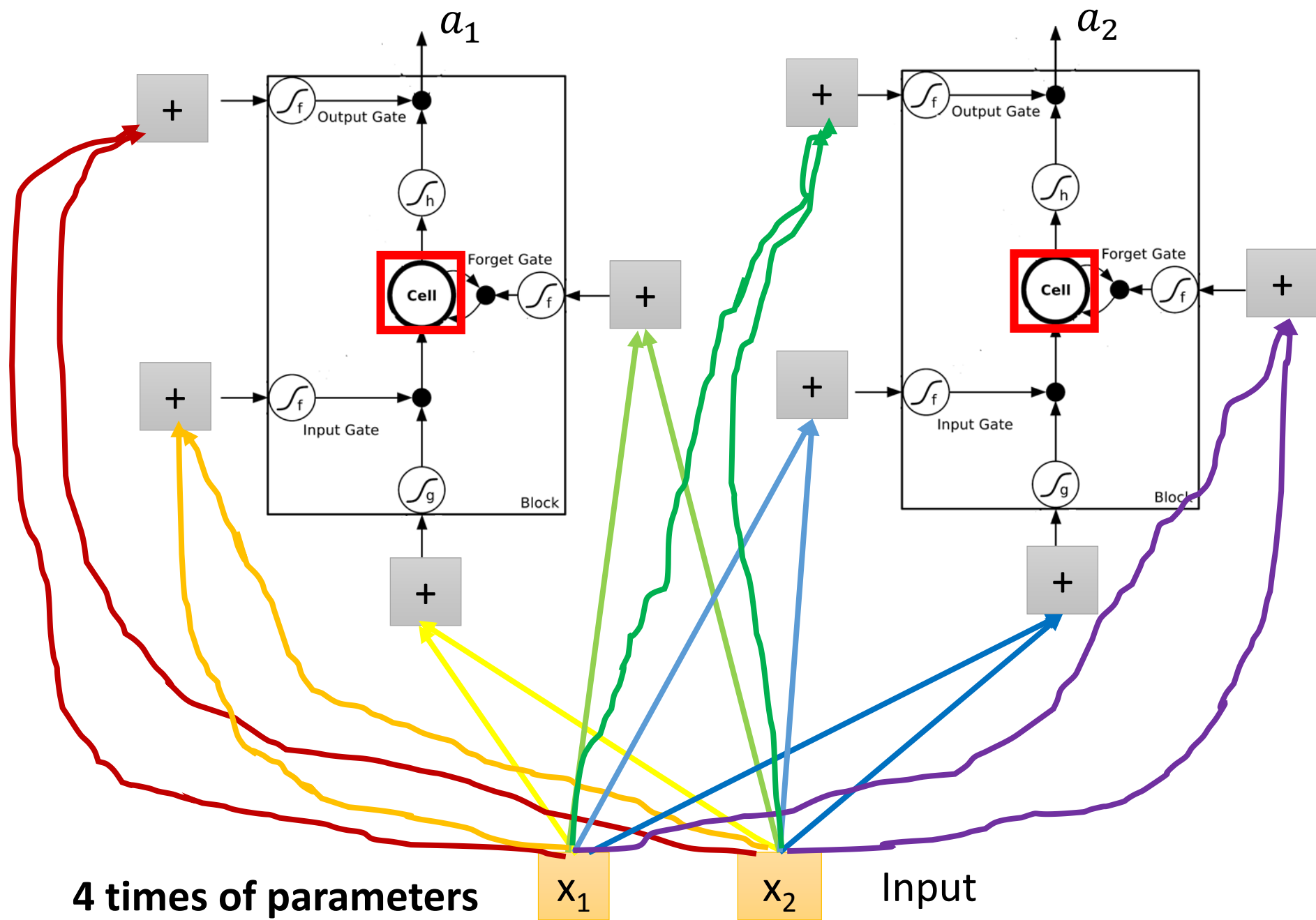
Mimic open and close gate

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

Original Network:

- Simply replace the neurons with LSTM





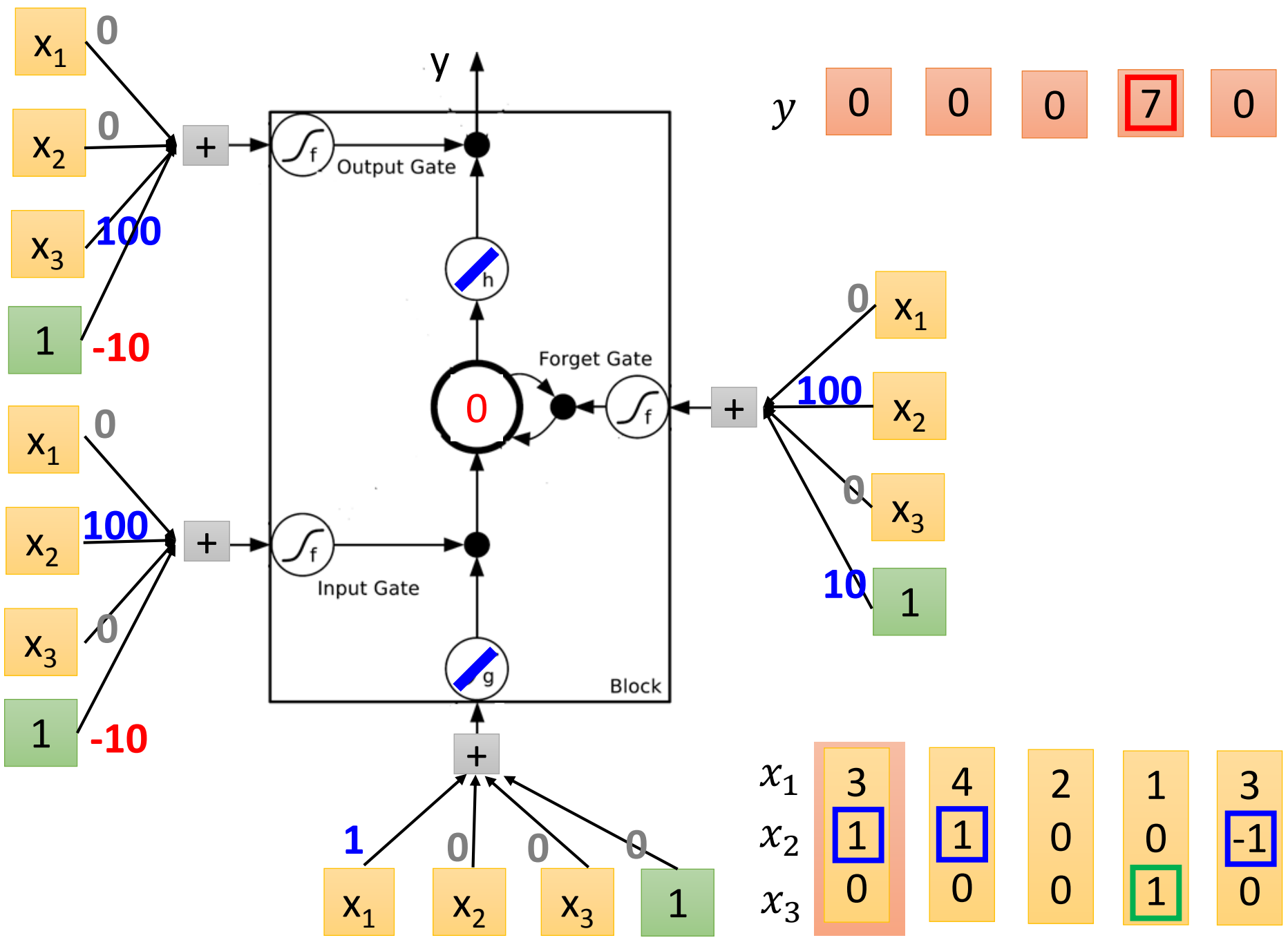
LSTM - Example

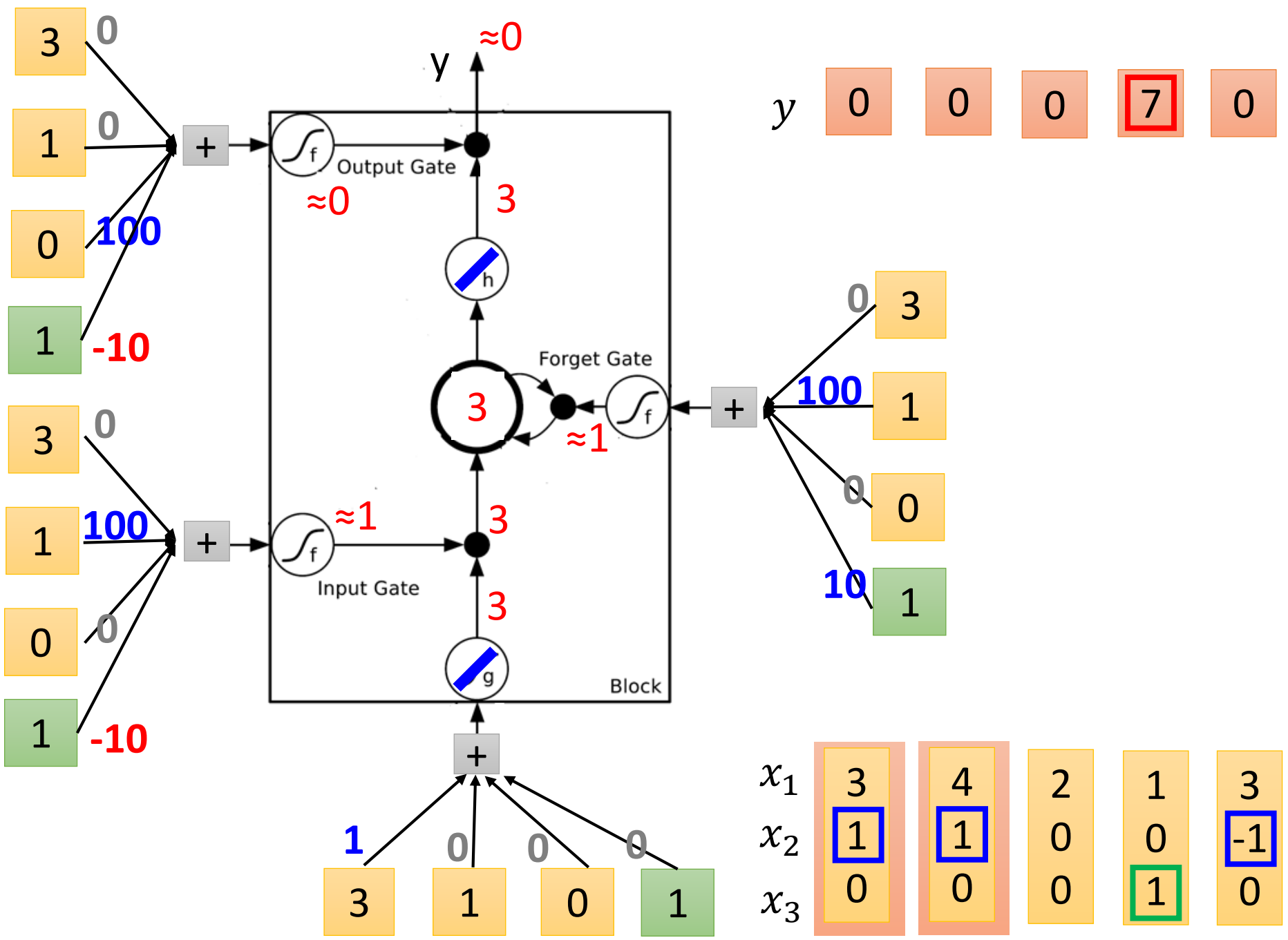
	0	0	3	3	7	7	7	0	6
x_1	1	3	2	4	2	1	3	6	1
x_2	0	1	0	1	0	0	-1	1	0
x_3	0	0	0	0	0	1	0	0	1
y	0	0	0	0	0	7	0	0	6

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.





3 0

1 0

0 100

1 -10

3 0

1 100

0 0

1 -10

1 0 0 0

3 1 0 1

y 0 0 0 7 0

0 3

100 1

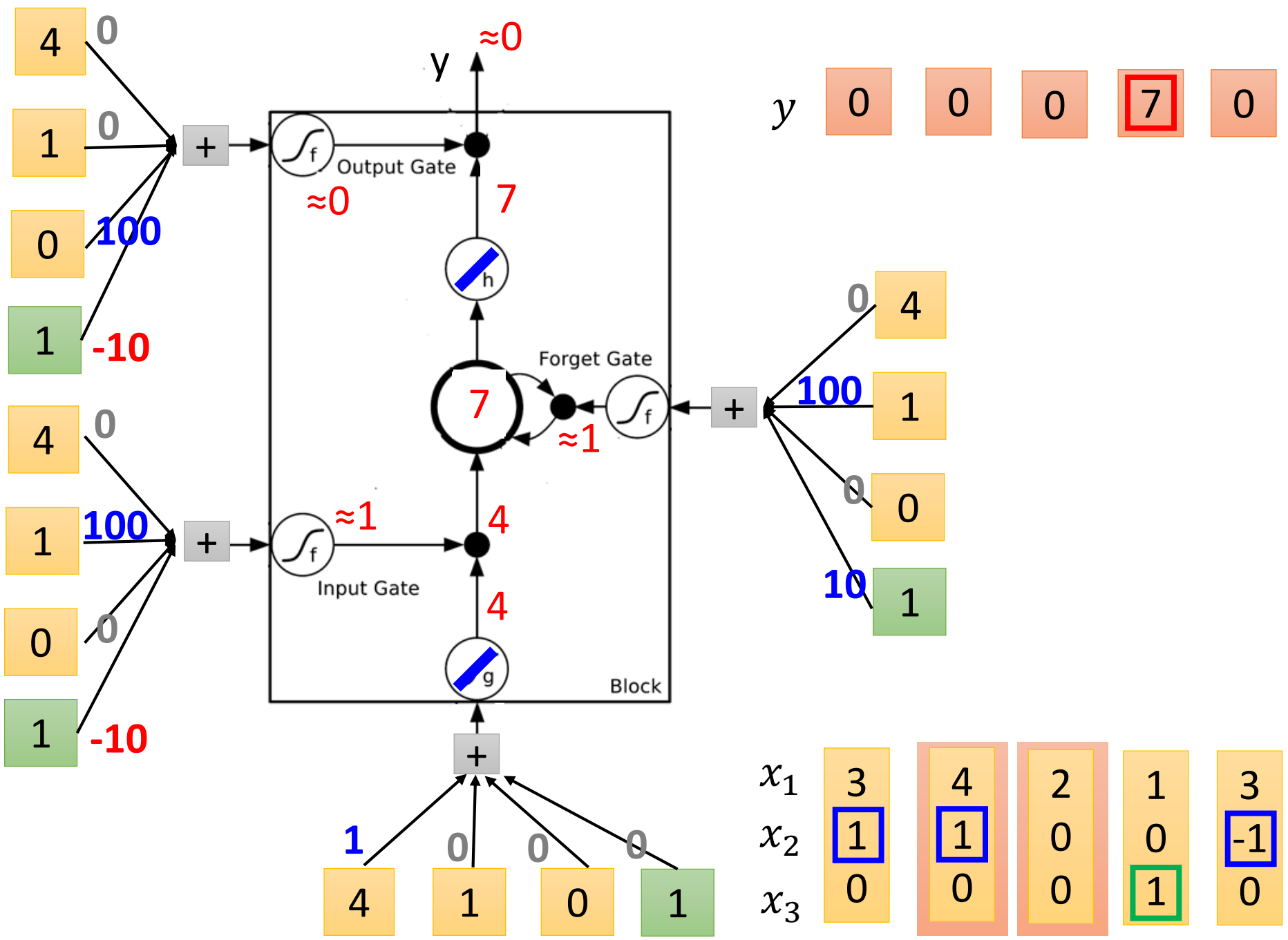
0 0

10 1

x_1 3 4 2 1 3

x_2 1 1 0 0 -1

x_3 0 0 0 1 0



4 0

1 0

0 100

1 -10

4 0

1 100

0 0

1 -10

1 0 0 0

4 1 0 1

y 0 0 0 7 0

0 4

100 1

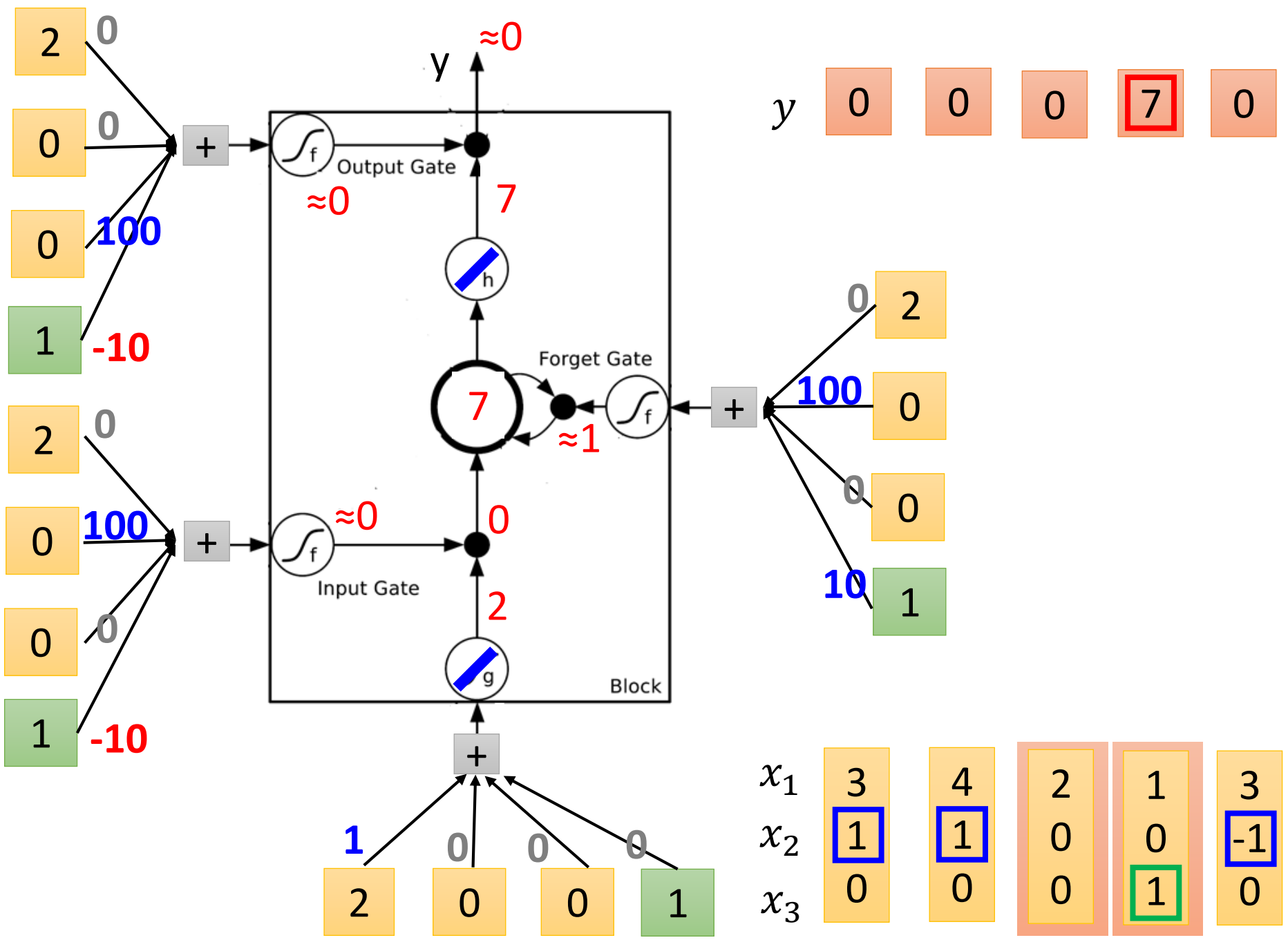
0 0

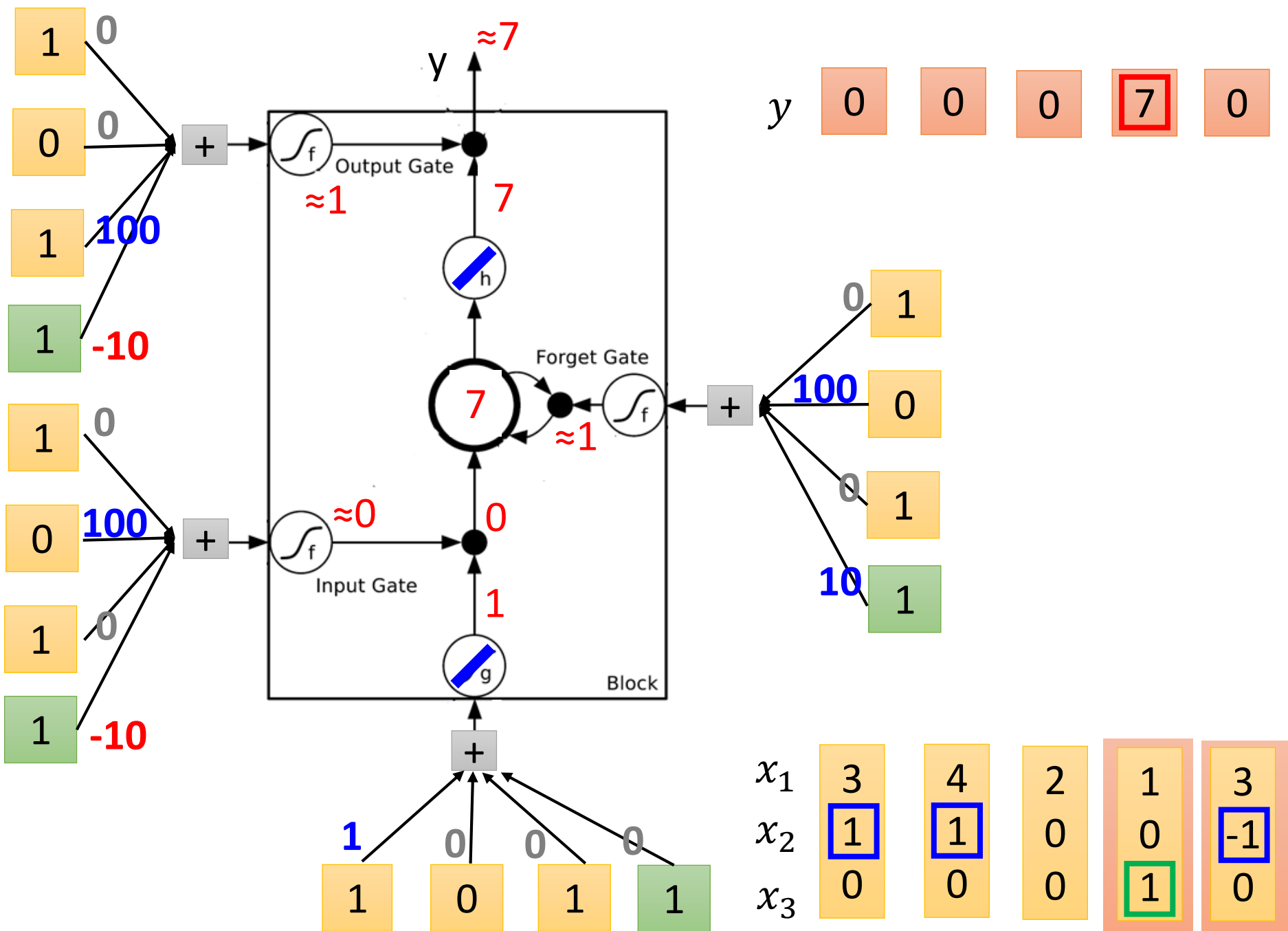
10 1

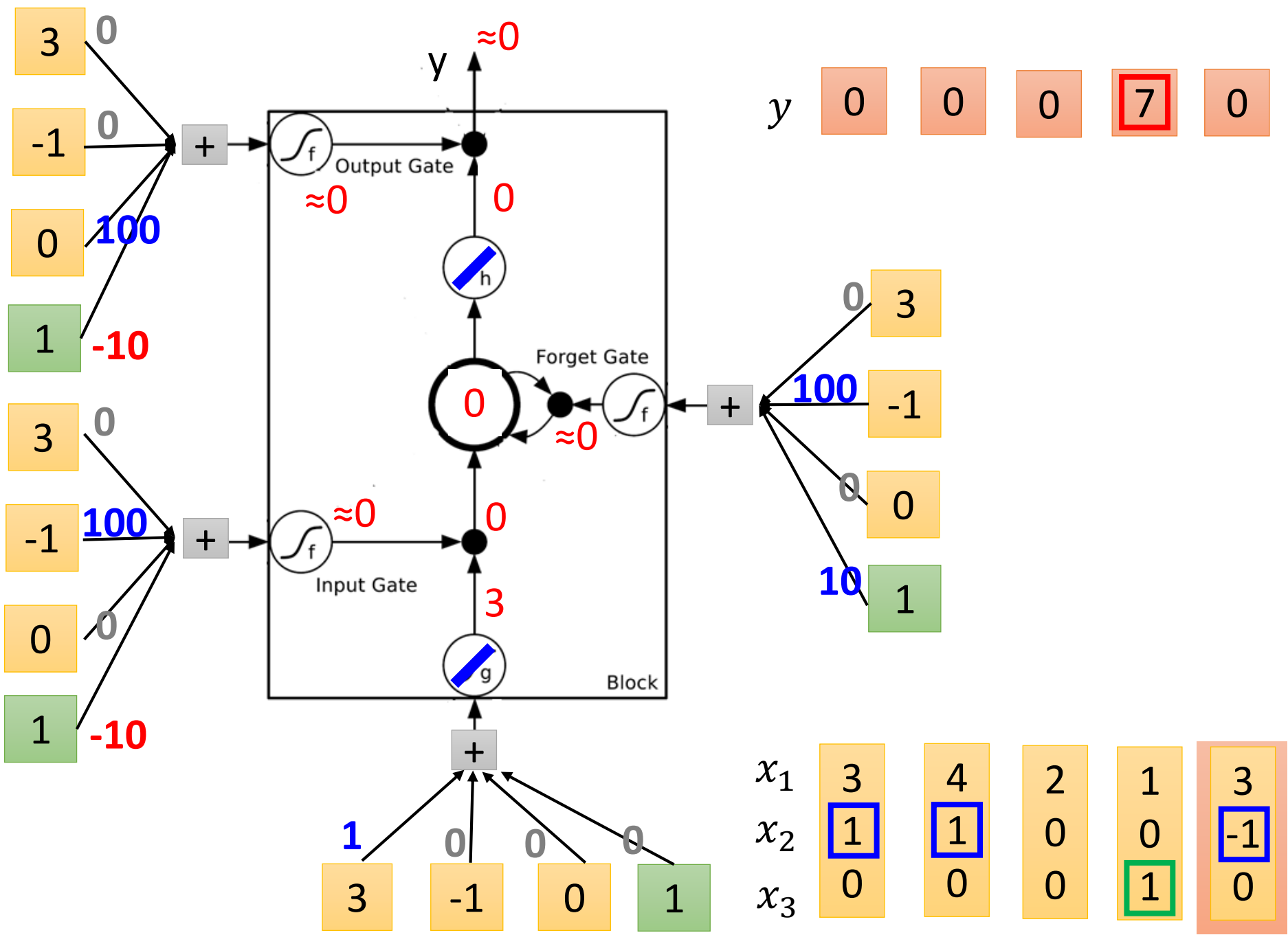
x_1 3 4 2 1 3

x_2 1 1 0 0 -1

x_3 0 0 0 1 0

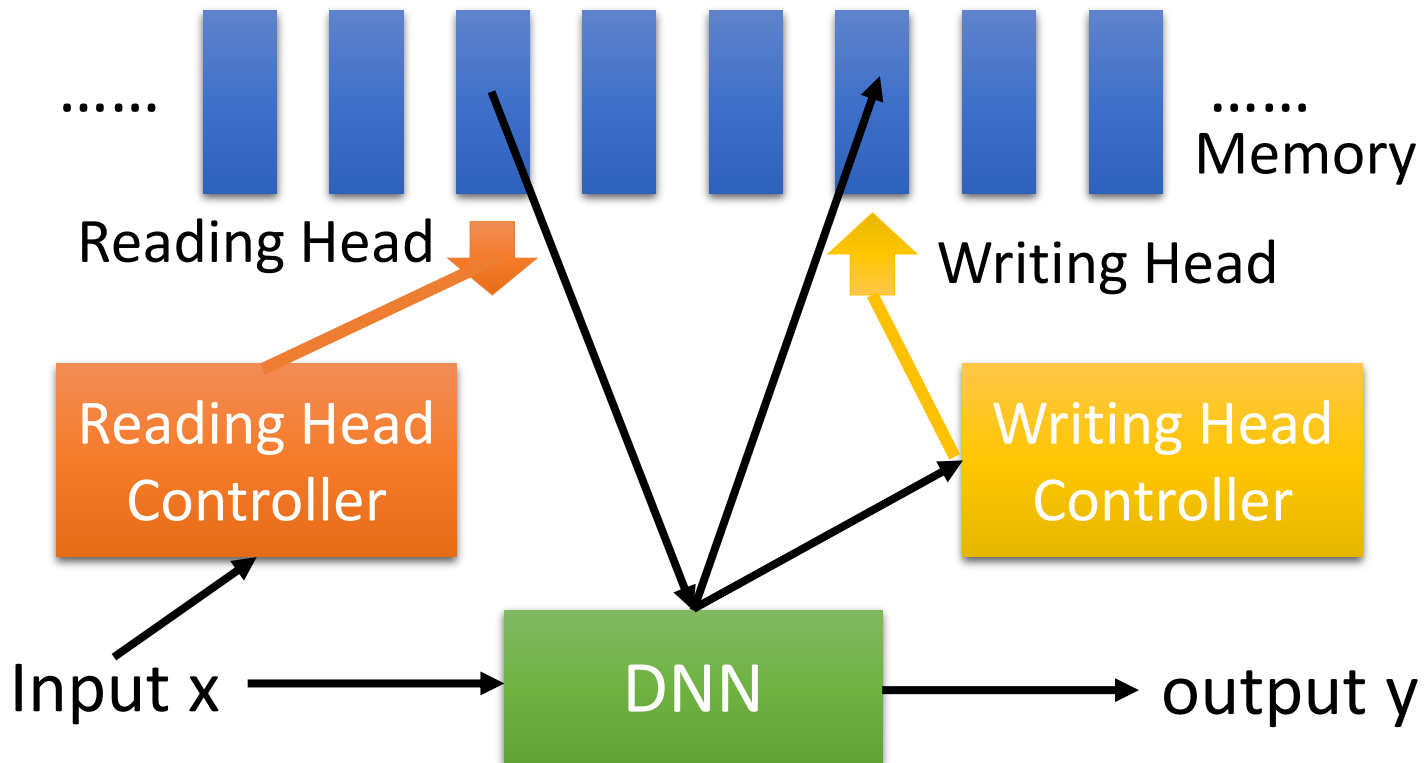






What is the next wave?

- Attention-based Model



Recommended Reading List

- The Unreasonable Effectiveness of Recurrent Neural Networks
 - <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
- Understanding LSTM Networks
 - <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Acknowledgement

- 感謝 葉軒銘 同學於上課時發現投影片上的錯誤