Deep Learning
Neural Network with Memory (1)

Hung-yi Lee

Memory is important

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

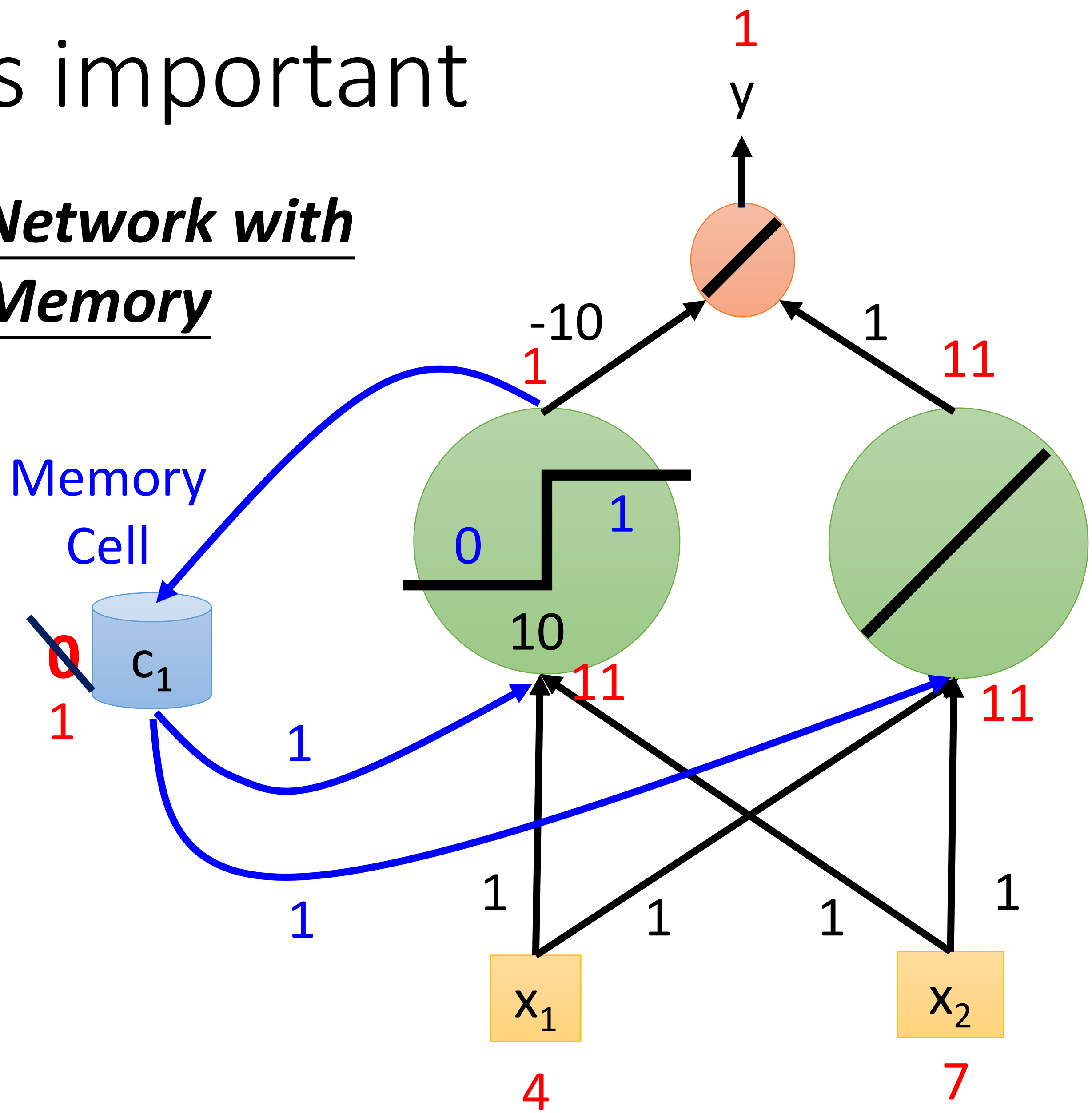
$$\begin{array}{cccc} & \boxed{1} & \boxed{1} & \\ & 1 & 4 & 4 \\ + & 1 & 7 & 7 \\ \hline & 3 & 2 & 1 \end{array}$$

Network needs memory
to achieve this

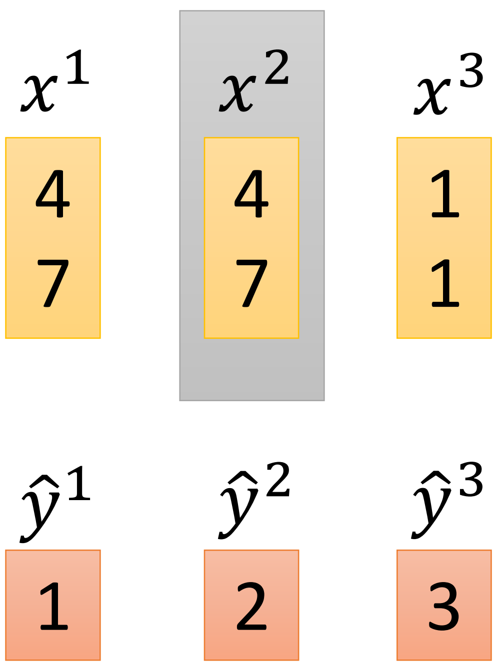
Memory is important

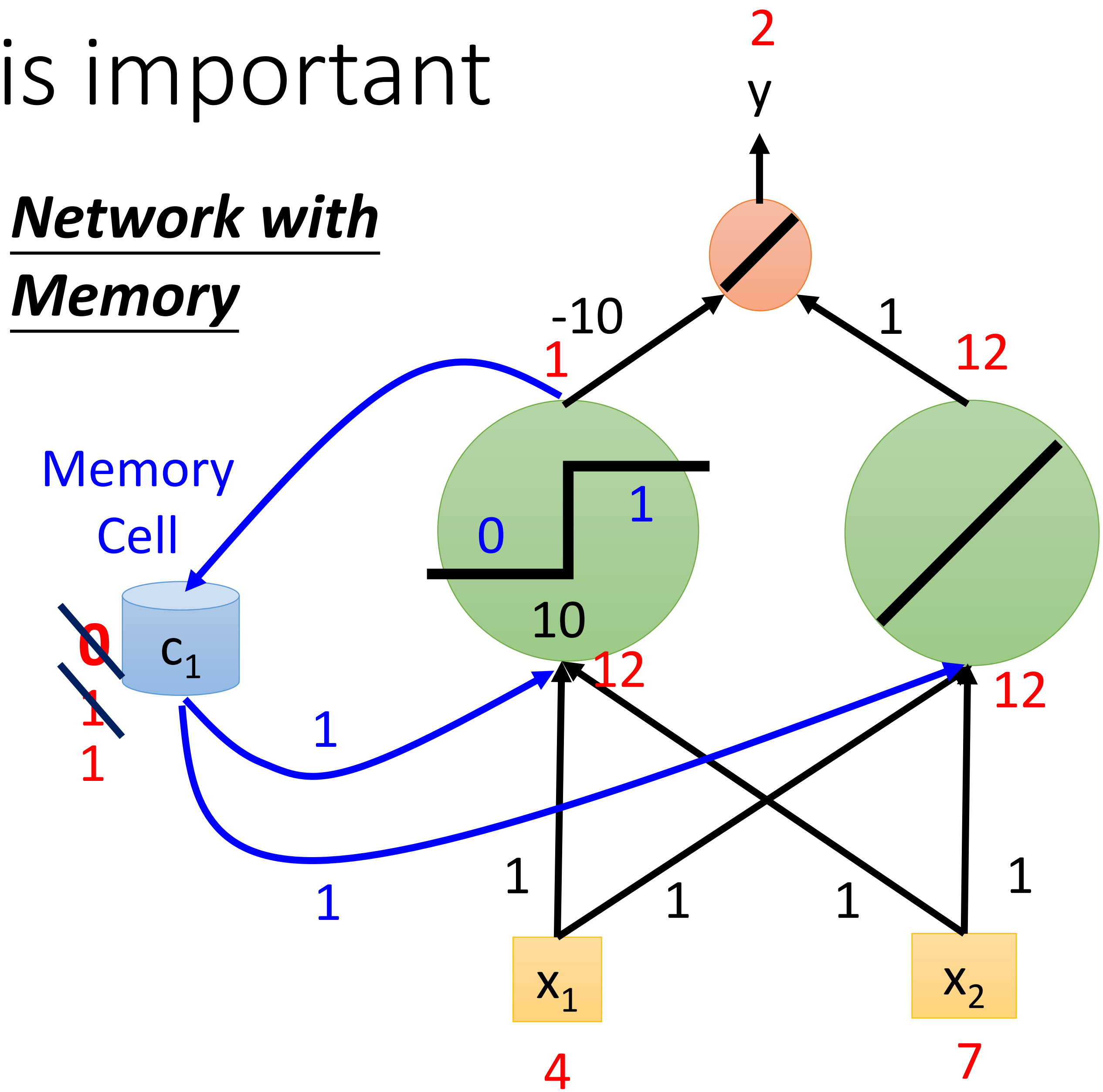
	x^1	x^2	x^3
	4	4	1
	7	7	1
	$\hat{y}^1$	$\hat{y}^2$	$\hat{y}^3$
	1	2	3

Network with Memory

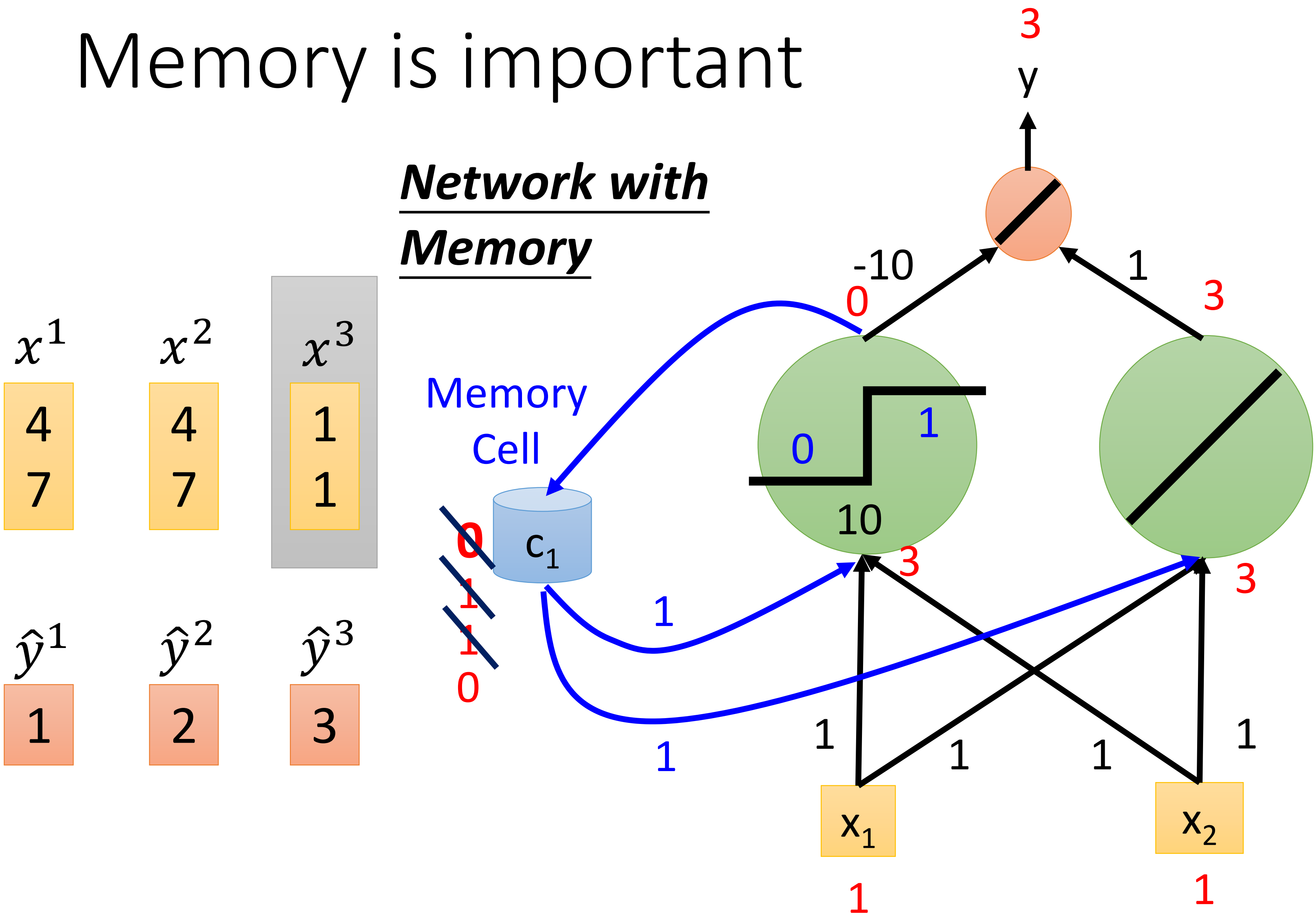


Memory is important





Memory is important

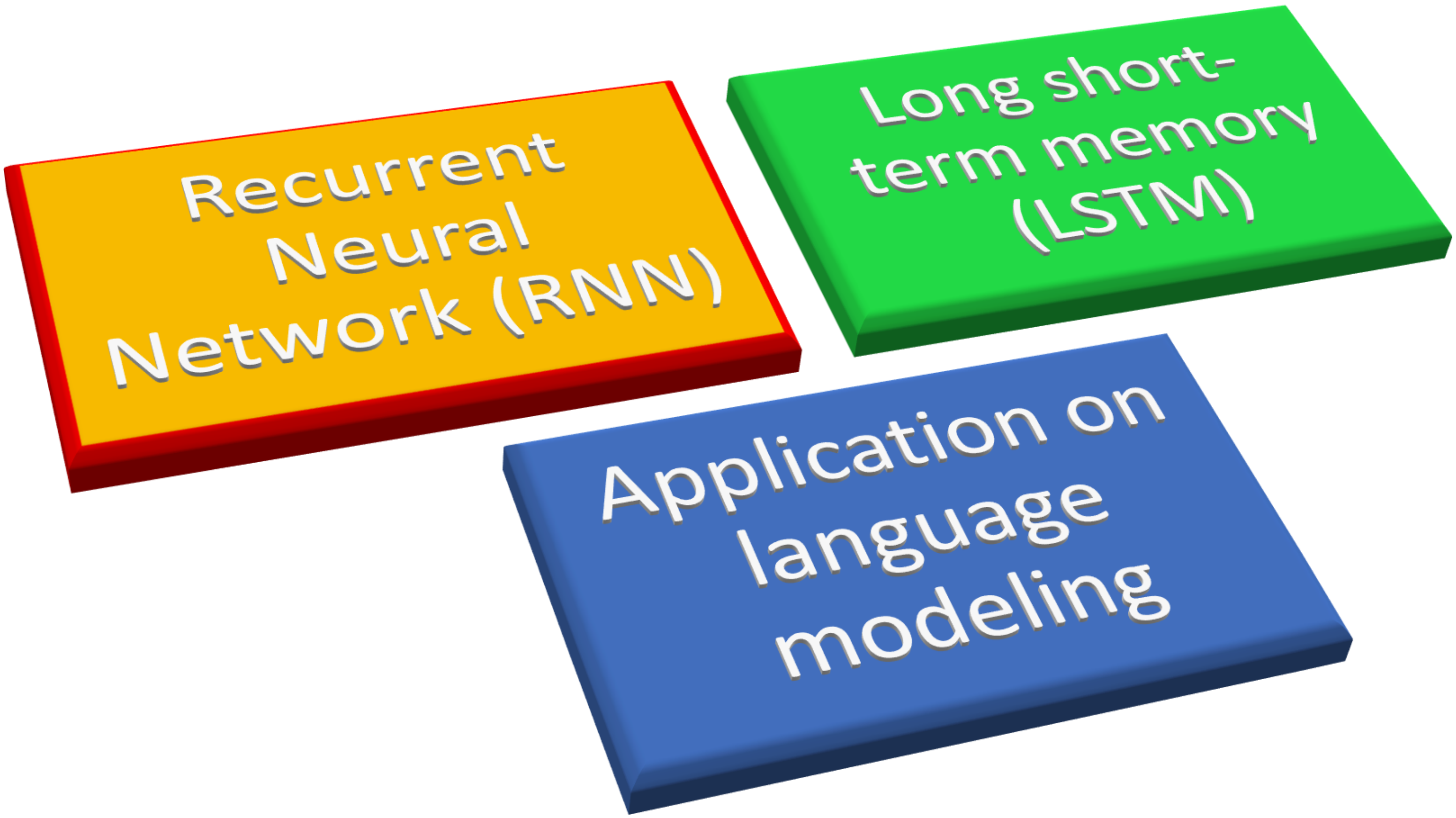


Outline

Today we will simply assume we already know the parameters in the network.
Training will be discussed in the next week.

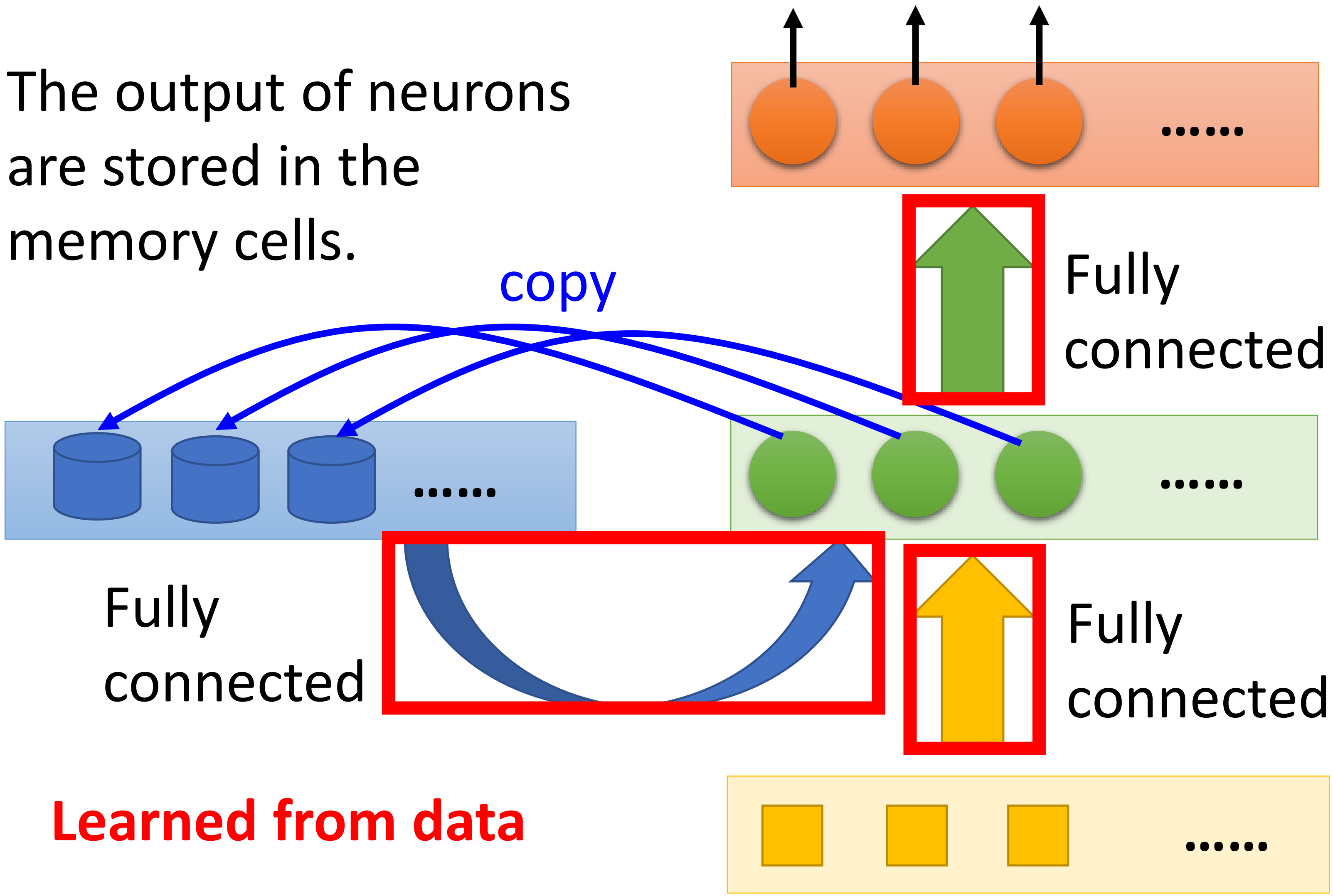
- Recurrent Neural Network (RNN)
- Long short-term memory (LSTM)
- Application on language modeling

Outline


Recurrent Neural Network (RNN)
Long short-term memory (LSTM)
Application on language modeling

Recurrent Neural Network (RNN)

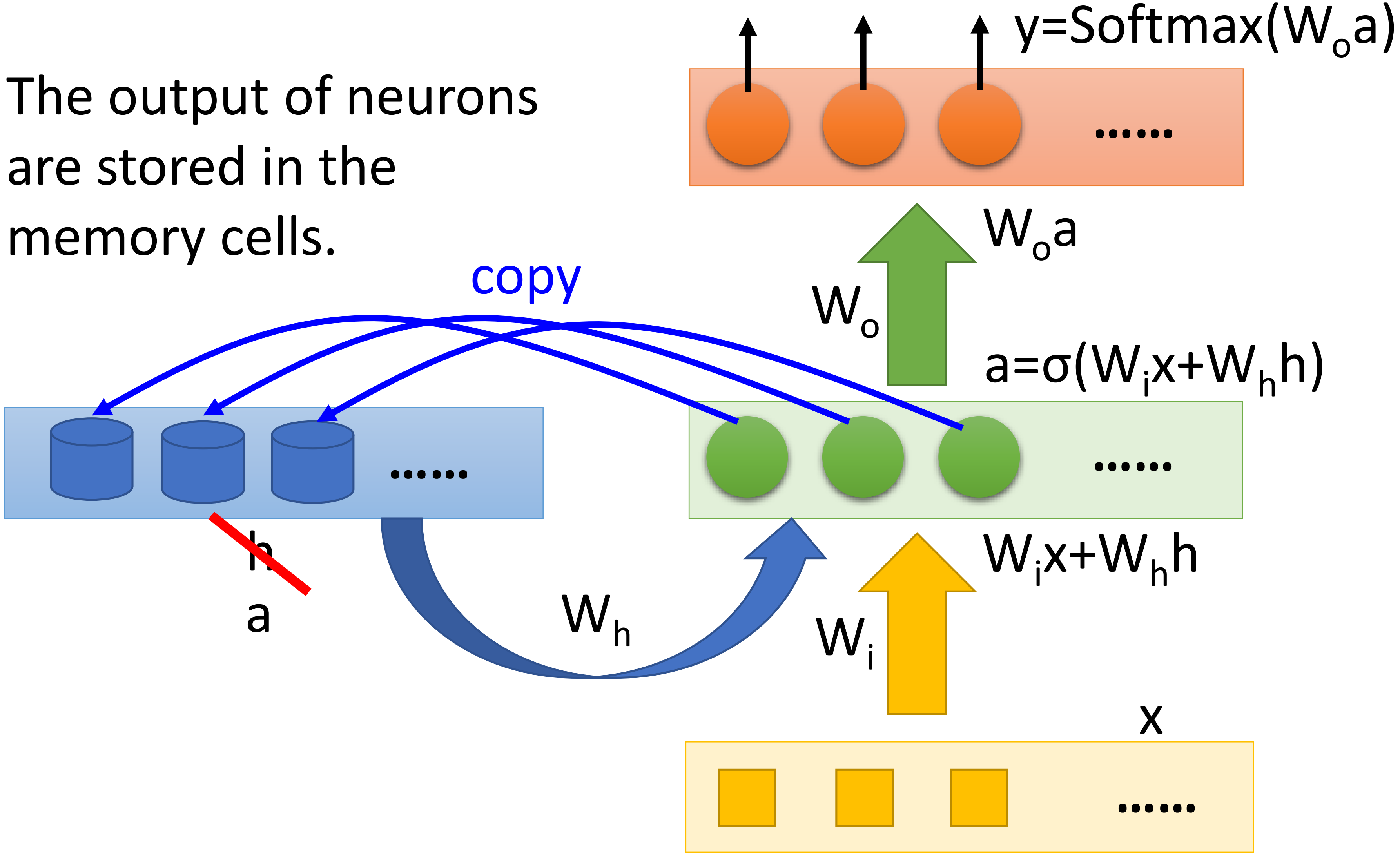
The output of neurons are stored in the memory cells.



Learned from data

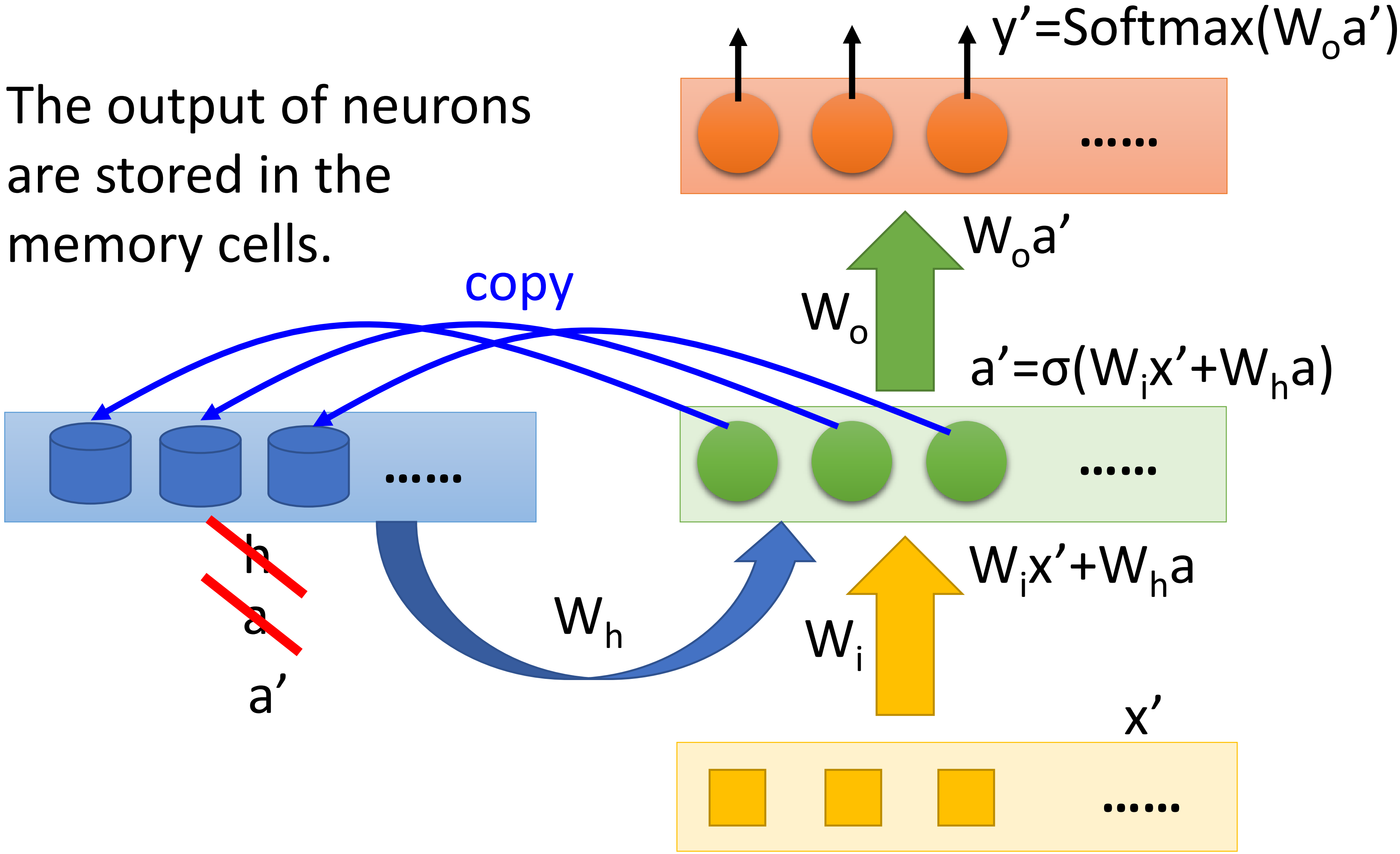
Recurrent Neural Network (RNN)

The output of neurons are stored in the memory cells.



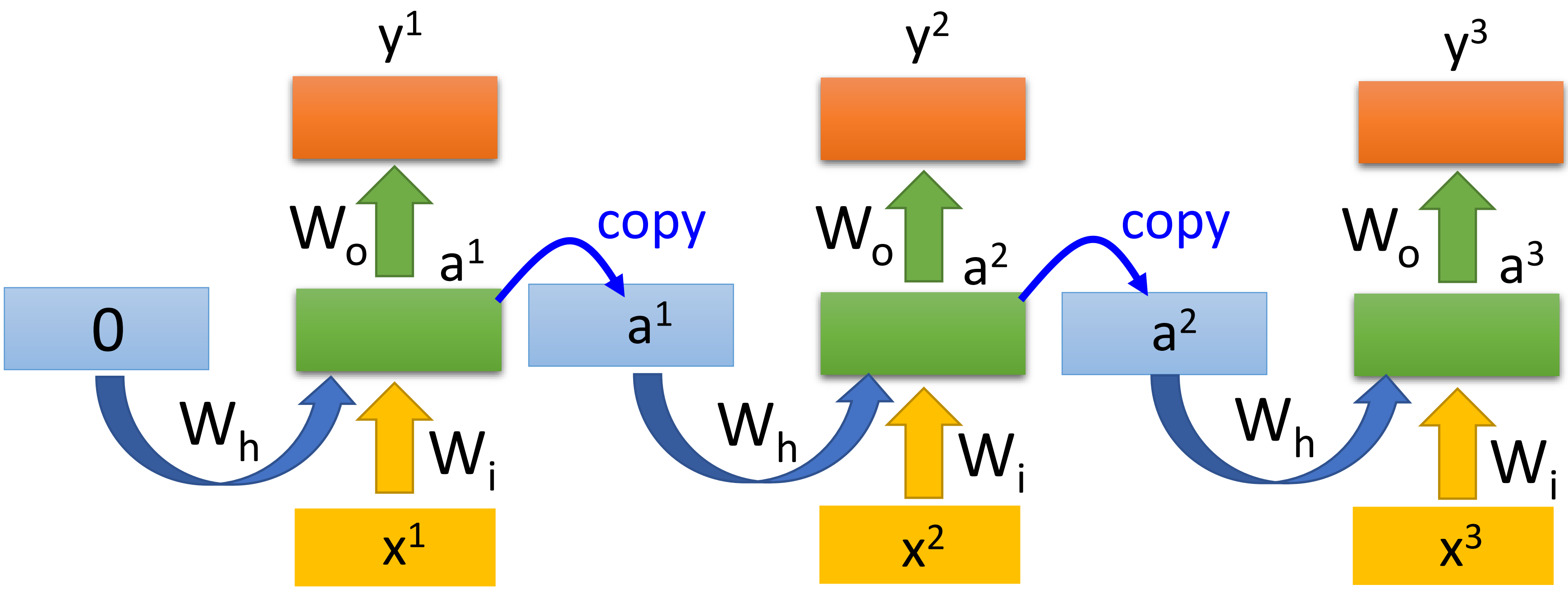
Recurrent Neural Network (RNN)

The output of neurons are stored in the memory cells.



Recurrent Neural Network (RNN)

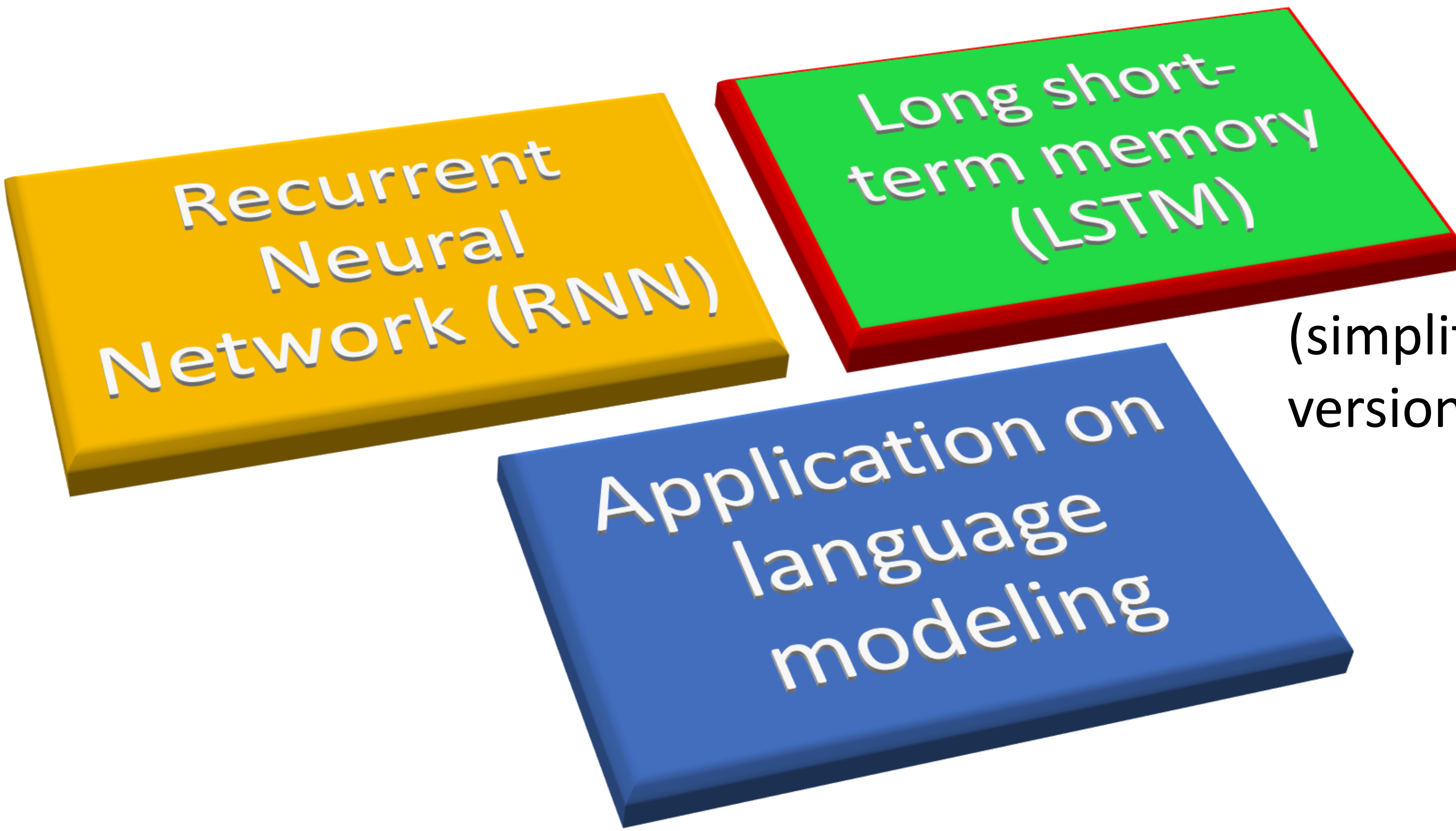
Input data: x^1 x^2 x^3 ...... (x^i are vectors)



The same network is used again and again.

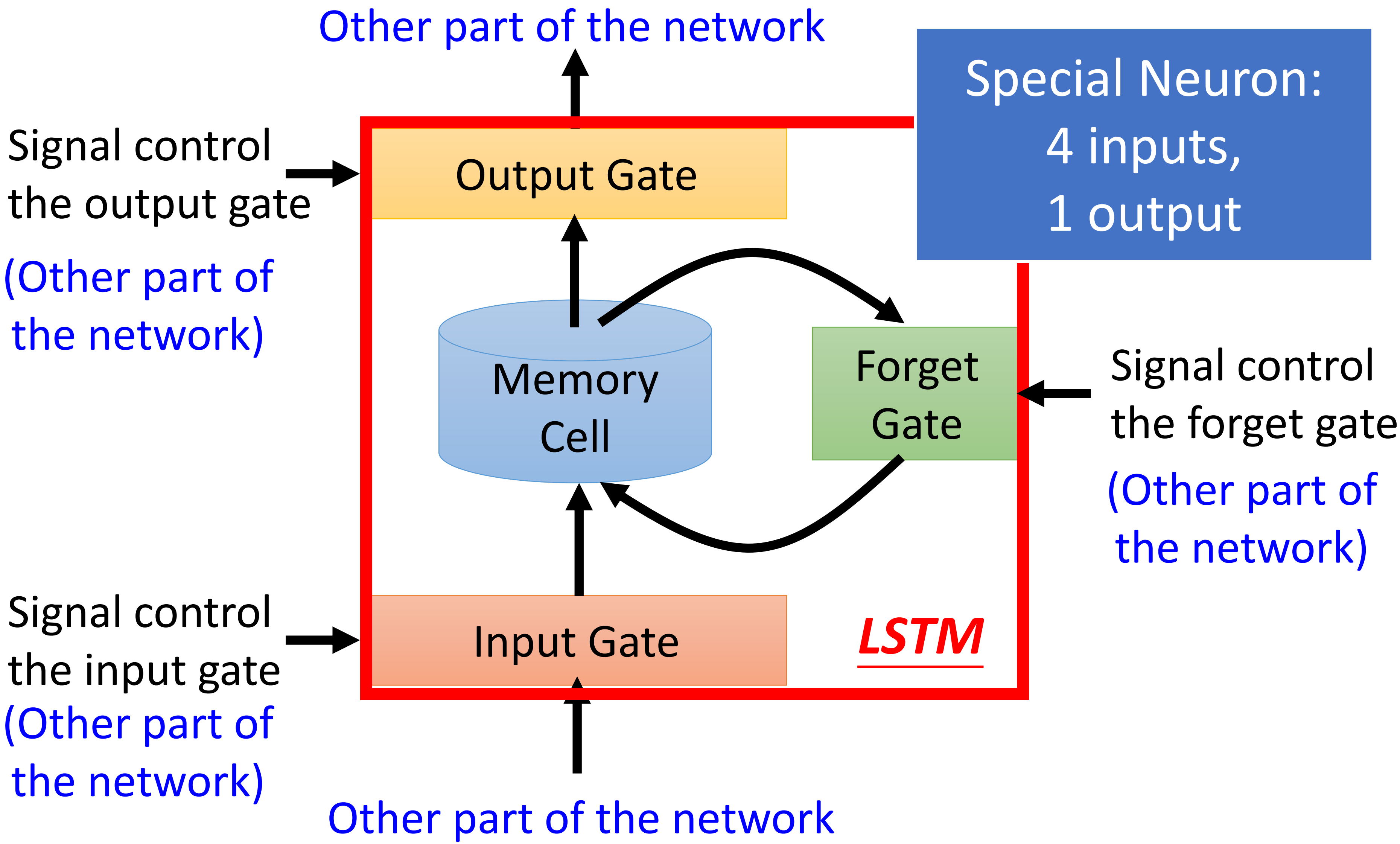
Output y^i depends on x^1, x^2, ... x^i

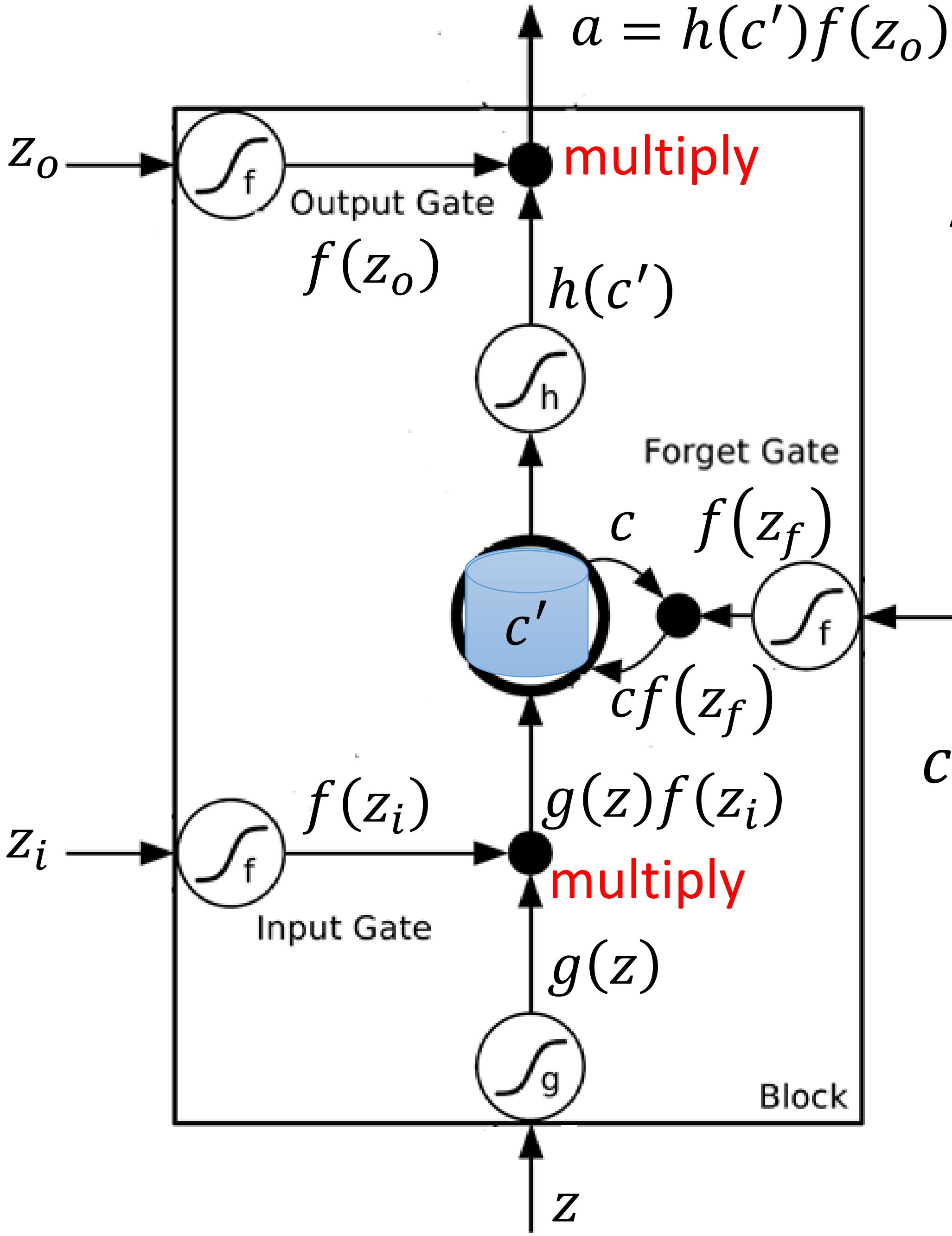
Outline



(simplified version)

Long Short-term Memory (LSTM)





Activation function f is usually a sigmoid function

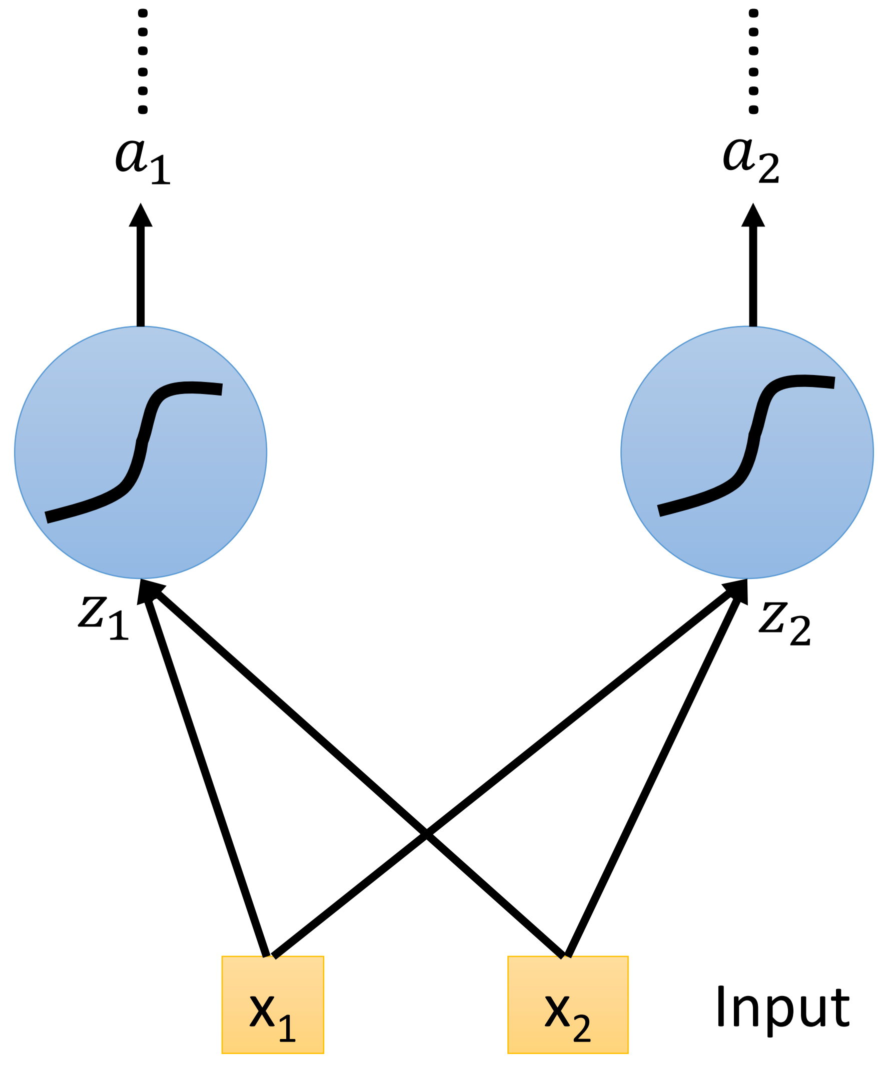
Between 0 and 1

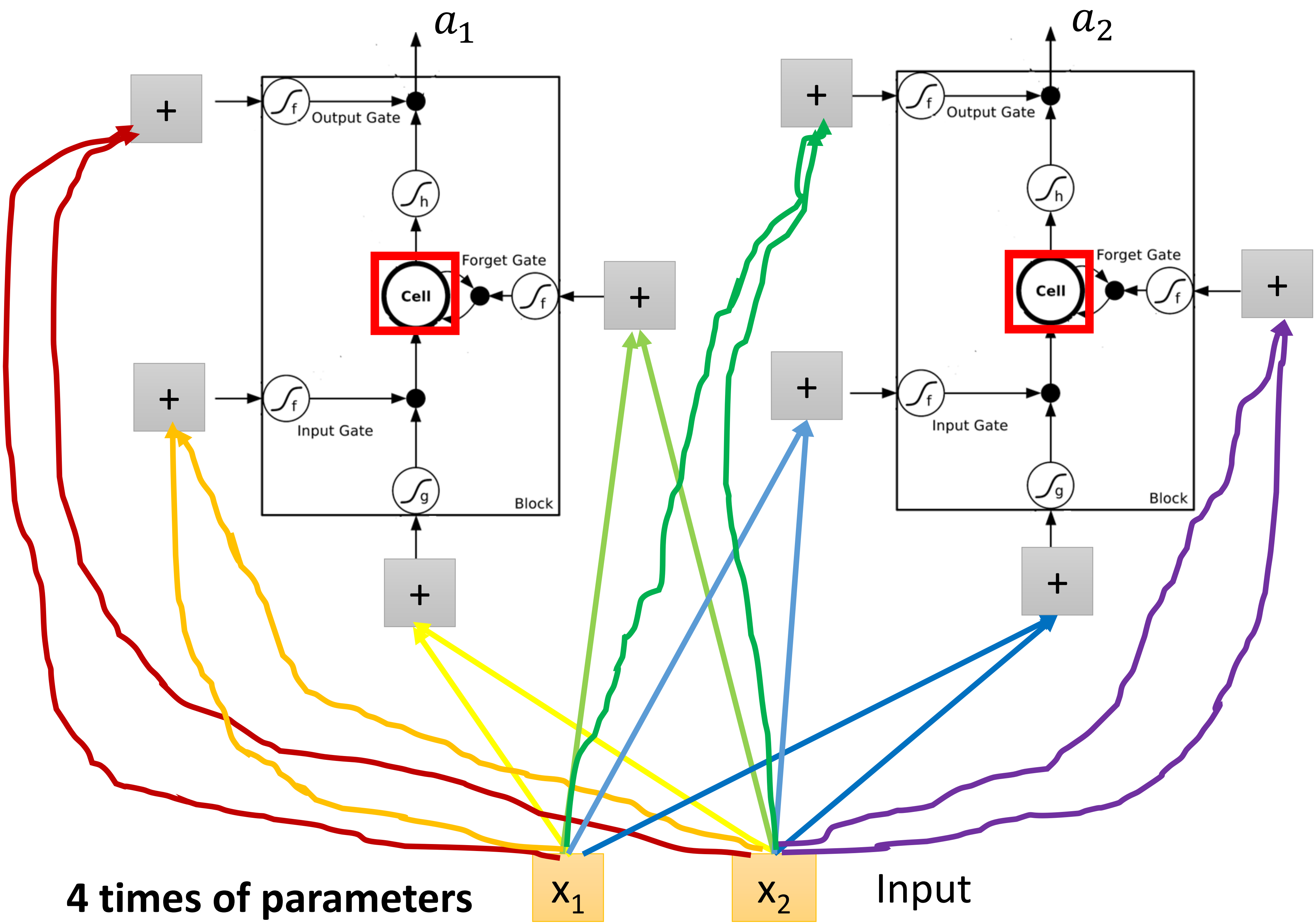
Mimic open and close gate

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

Original Network:

➢ Simply replace the neurons with LSTM





a_1
a_2
Output Gate
Forget Gate
Cell
Input Gate
Block
Output Gate
Forget Gate
Cell
Input Gate
Block
4 times of parameters
x_1
x_2
Input

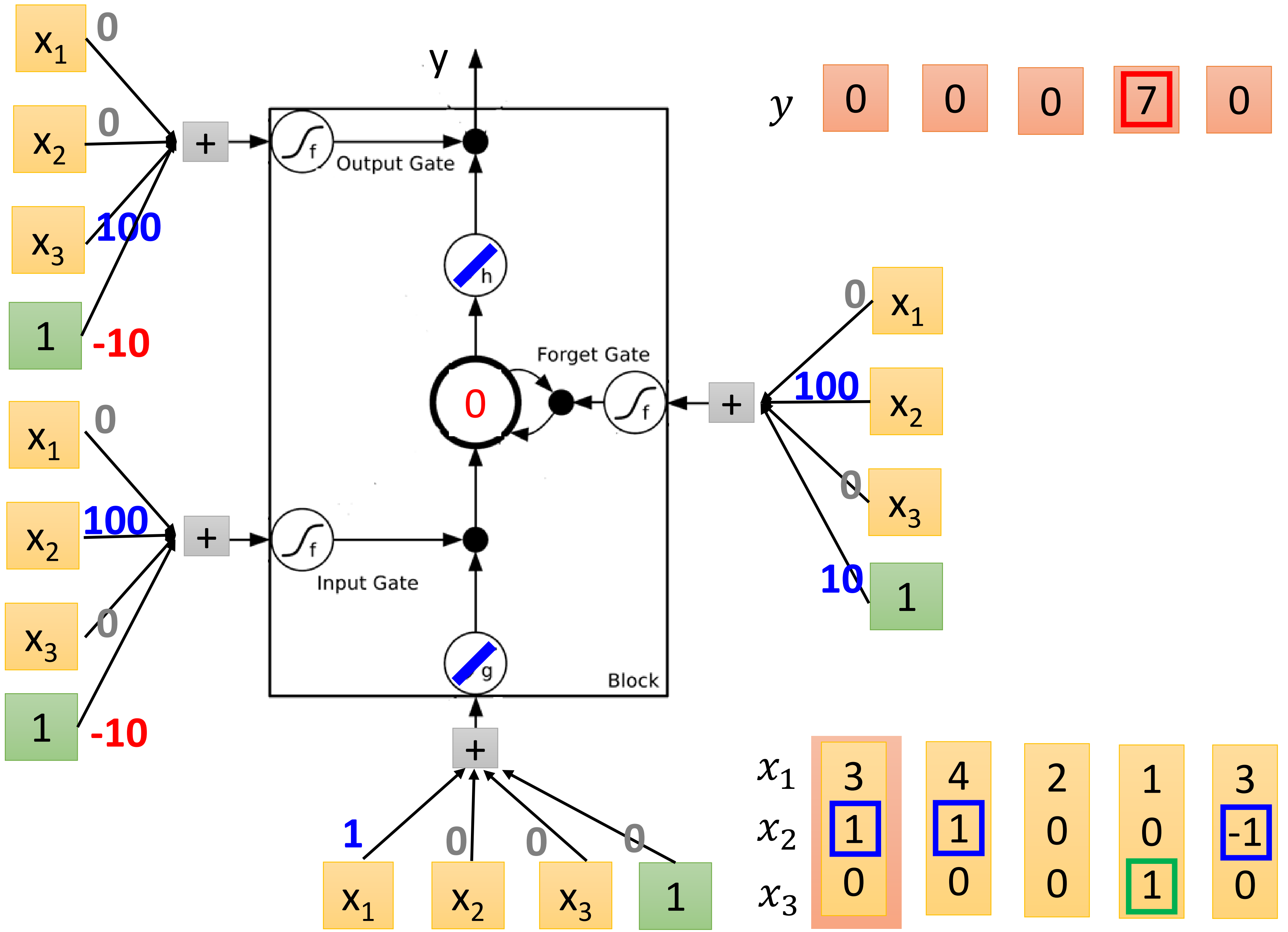
LSTM - Example

	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

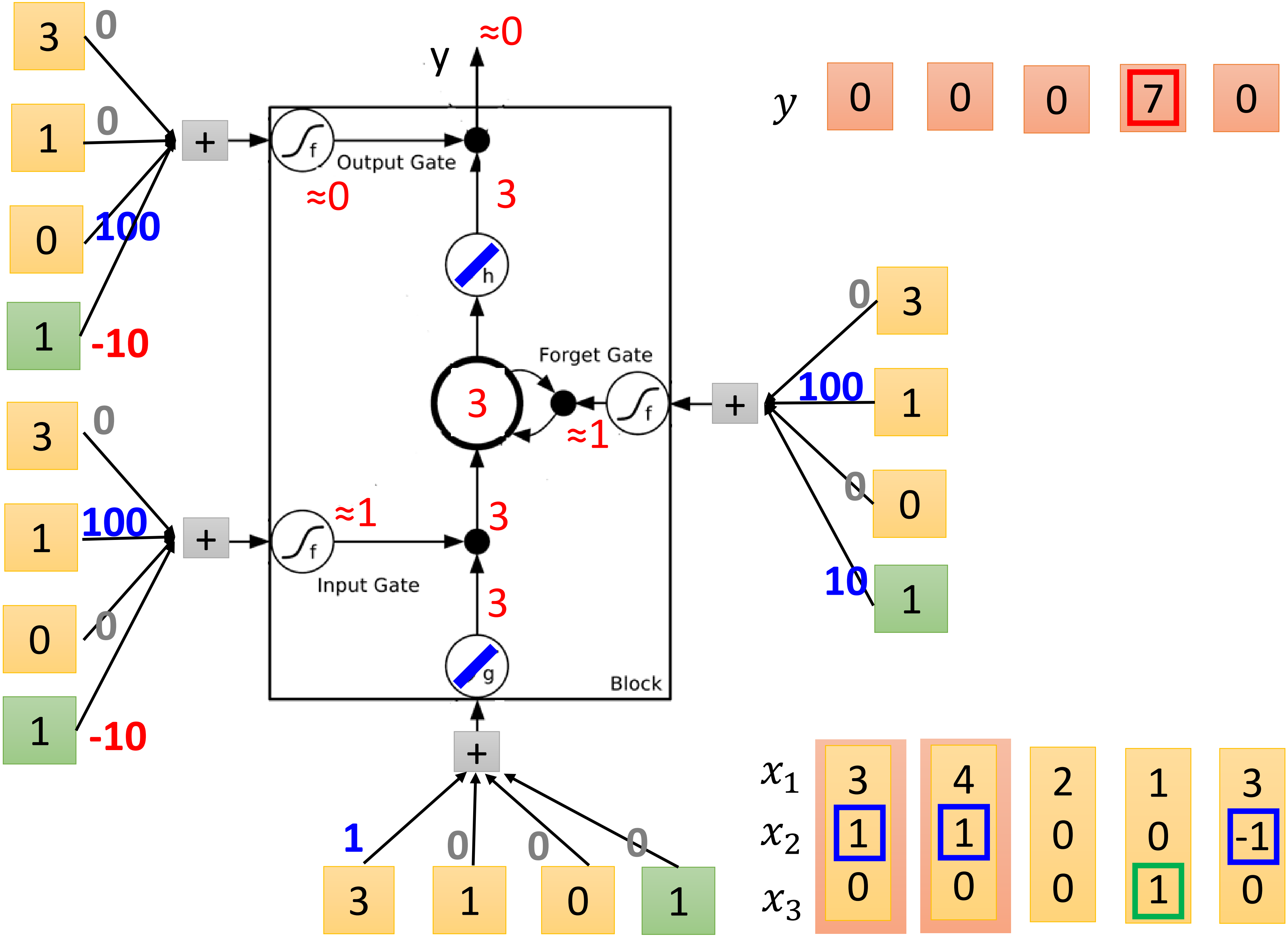
When x_2 = 1, add the numbers of x_1 into the memory

When x_2 = -1, reset the memory

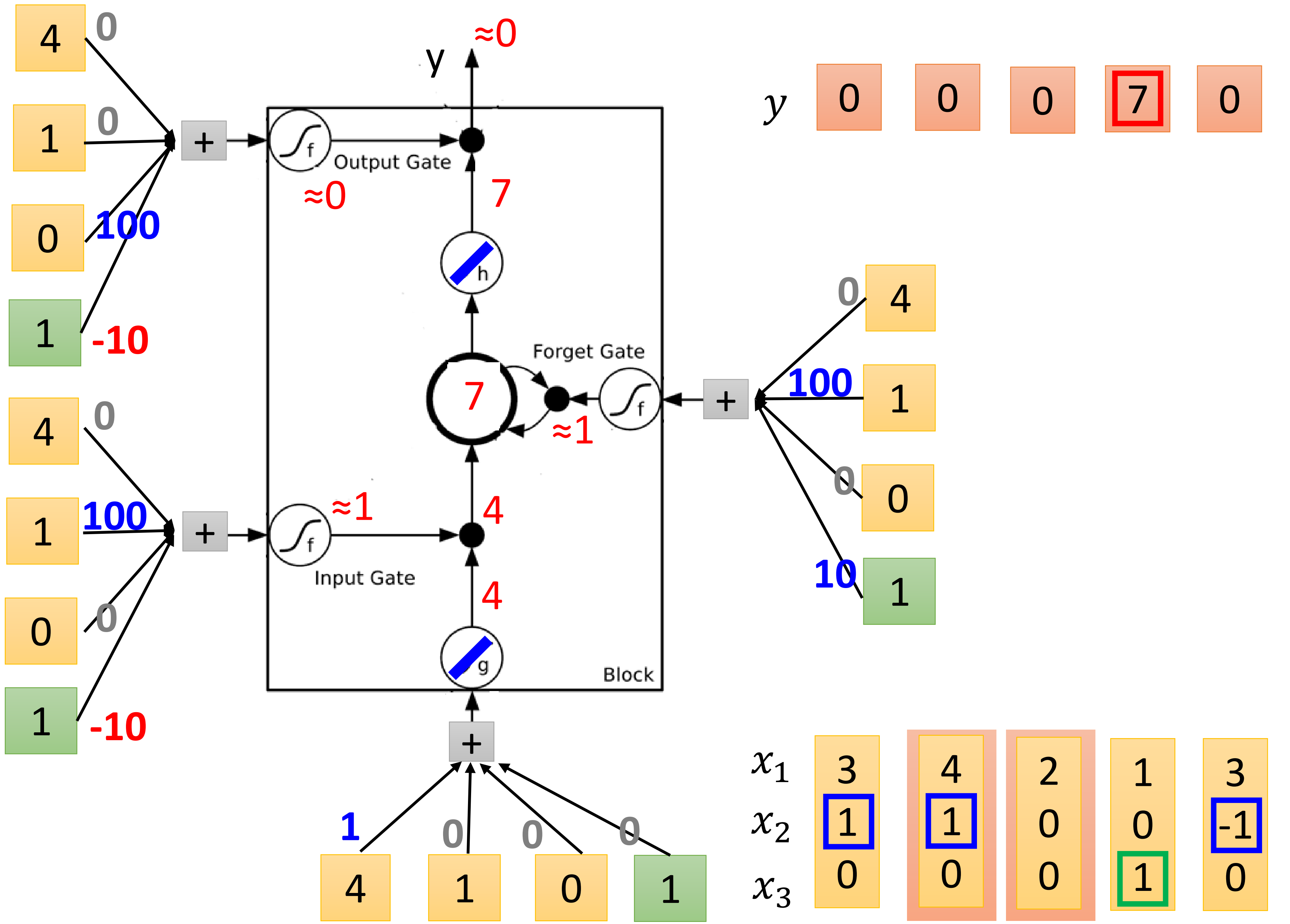
When x_3 = 1, output the number in the memory.



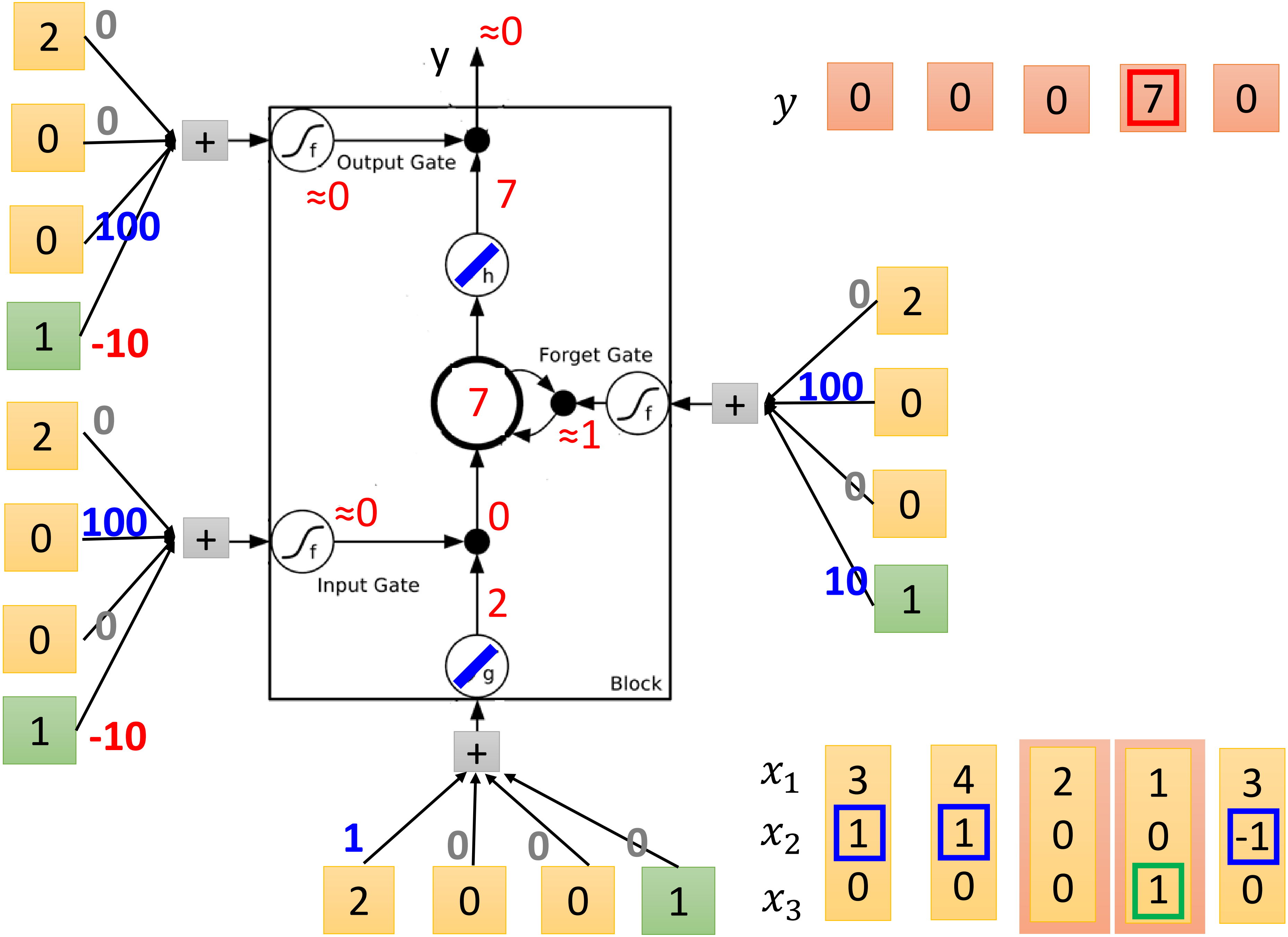
y
Output Gate
Forget Gate
Input Gate
Block
h
g
f
0
x1 x2 x3 1
0 0 100 -10
0 100 0 10
0 100 0 -10
1 0 0 0
y 0 0 0 7 0
x1 3 4 2 1 3
x2 1 1 0 0 -1
x3 0 0 0 1 0

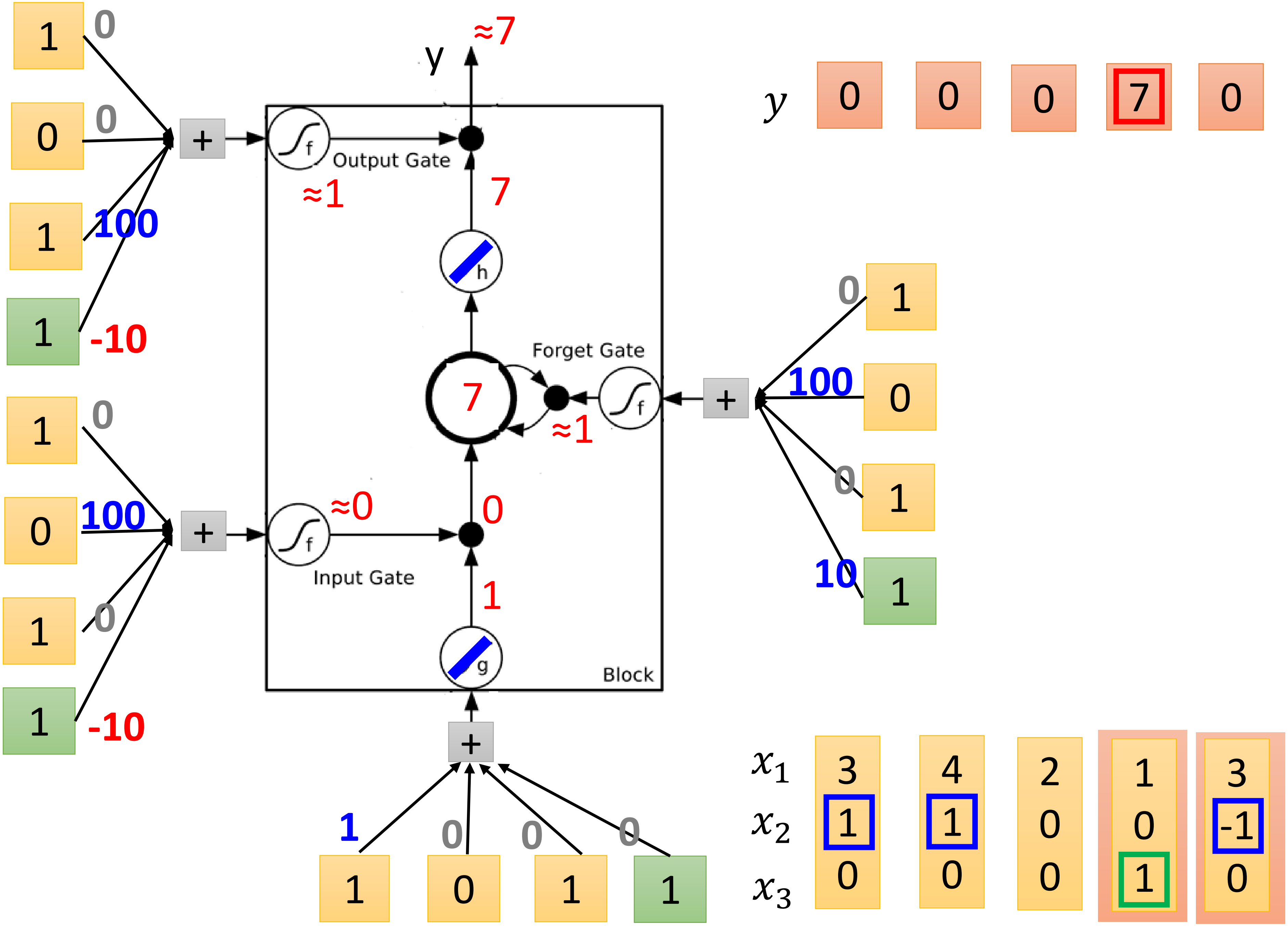


y
≈0
Output Gate
Forget Gate
Input Gate
Block
f
h
g
≈1
3
100
-10
10
0
1
y 0 0 0 7 0
x_1 3 4 2 1 3
x_2 1 1 0 0 -1
x_3 0 0 0 1 0

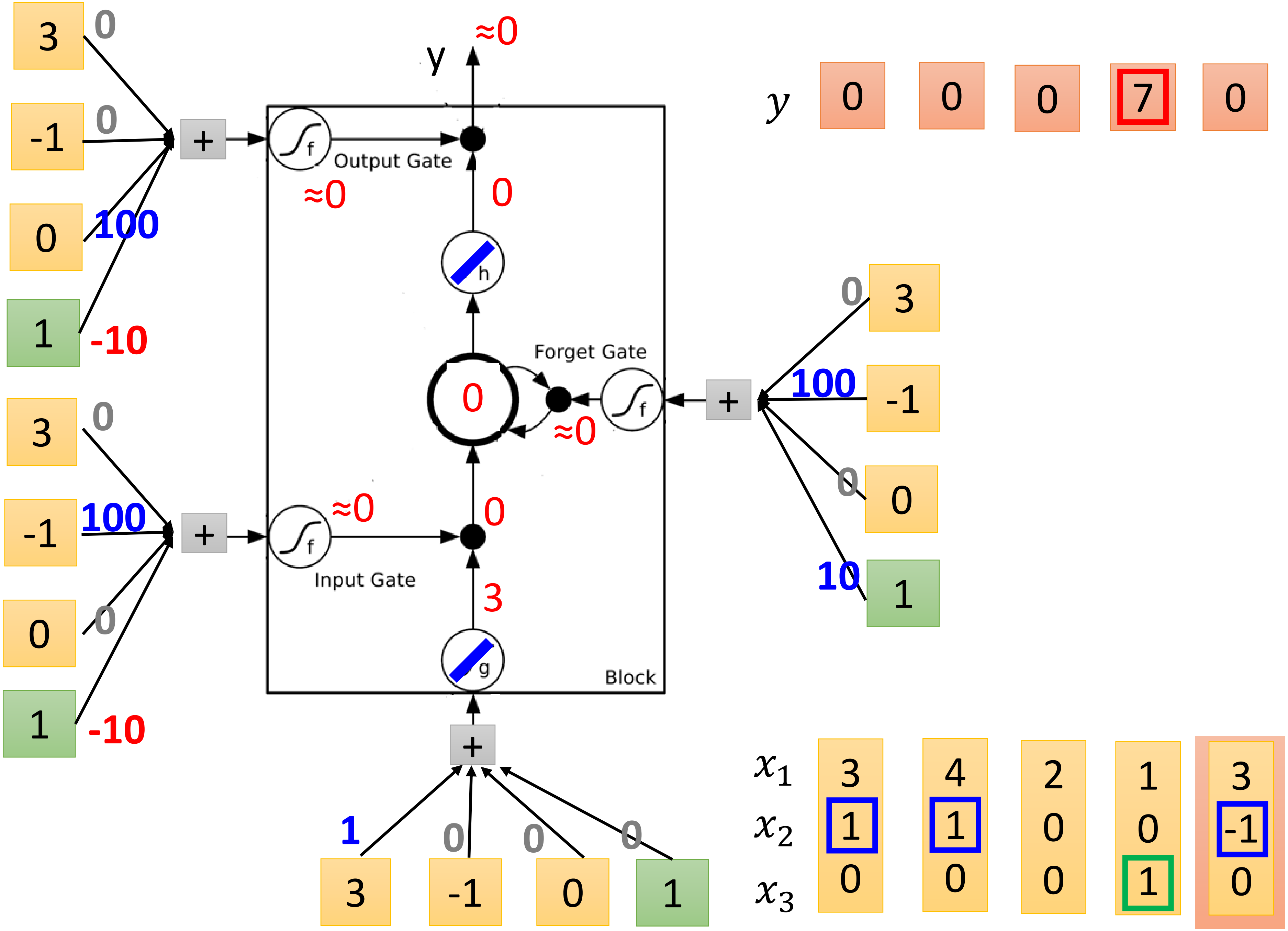


y
≈0
Output Gate
≈0
7
Forget Gate
7
≈1
Input Gate
≈1
4
4
Block
4
0
1
0
0
100
1
-10
4
0
1
100
0
0
1
-10
0
4
100
1
0
0
10
1
1
4
0
1
0
0
0
1
y
0
0
0
7
0
x_1
x_2
x_3
3 1 0
4 1 0
2 0 0
1 0 1
3 -1 0


y
≈0
Output Gate
≈0
7
Forget Gate
7
≈1
≈0
0
Input Gate
2
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 0 0 0
2 0 0 1
y 0 0 0 7 0
x_1 3 4 2 1 3
x_2 1 1 0 0 -1
x_3 0 0 0 1 0



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



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

Outline

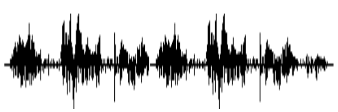
Recurrent Neural Network (RNN)

Long short-term memory (LSTM)

Application on language modeling

Language model (LM)

- Language model: Estimated the probability of word sequence
  - Word sequence: $w_1, w_2, w_3, ..., w_n$
  - $P(w_1, w_2, w_3, ..., w_n)$
- Useful in speech recognition
  - Different word sequence can have the same pronunciation



recognize speech
or
wreck a nice beach

If P(recognize speech)
>P(wreck a nice beach)

Output =
"recognize speech"

Language model

P("wreck a nice beach")
=P(wreck|START)P(a|wreck)
P(nice|a)P(beach|nice)

- How to estimate $P(w_1, w_2, w_3, \ldots., w_n)$
- Collect a large amount of text data as training data
  - However, the word sequence $w_1, w_2, \ldots., w_n$ may not appear in the training data
- N-gram language model: $P(w_1, w_2, w_3, \ldots., w_n) = P(w_1|START)P(w_2|w_1) \ldots\ldots P(w_n|w_{n-1})$
- Estimate P(beach|nice) from training data

$$P(\text{beach}|\text{nice}) = \frac{C(nice\ beach)}{C(nice)}$$

Count of "nice beach" in the training data

Count of "nice" in the training data

Language model - Smoothing

- Training data:
  - The dog ran ……
  - The cat jumped ……

This is called **language model smoothing**.

P(jumped | dog) = ~~0~~ 0.0001

P(ran | cat) = ~~0~~ 0.0001

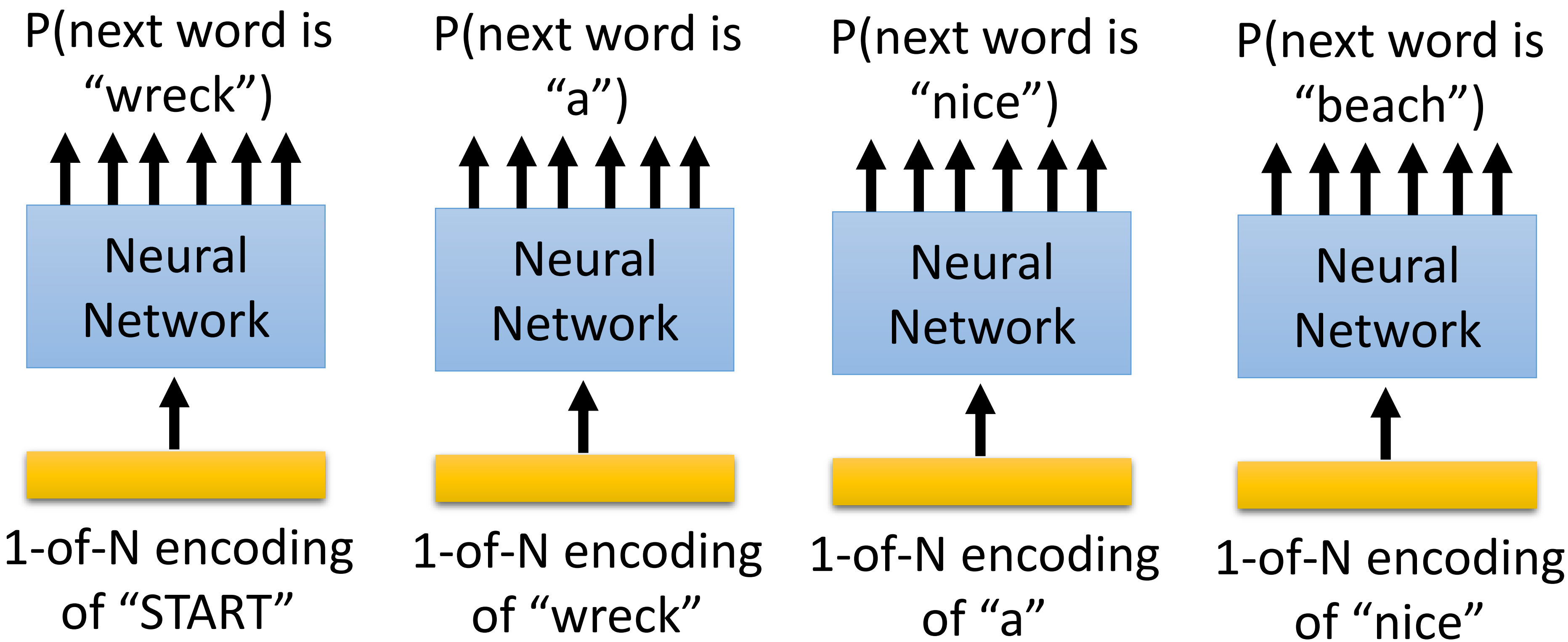
Give some small probability

- ➢ The probability is not accurate.
- ➢ The phenomenon happens because we cannot collect all the possible text in the world as training data.

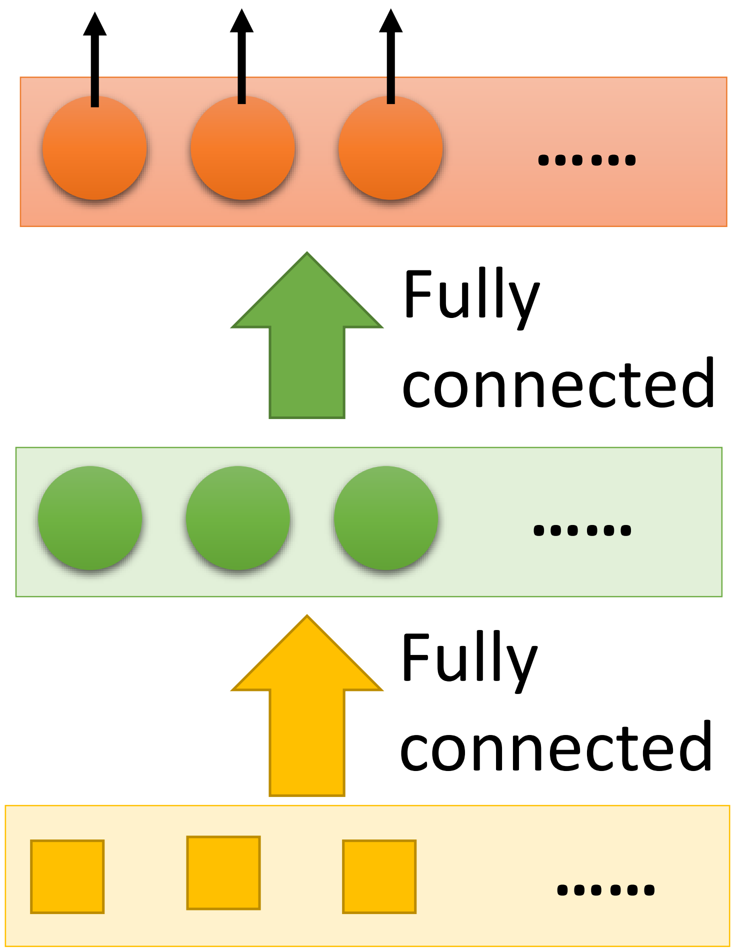
Neural-network based LM

P("wreck a nice beach")
=P(wreck|START)P(a|wreck)P(nice|a)P(beach|nice)

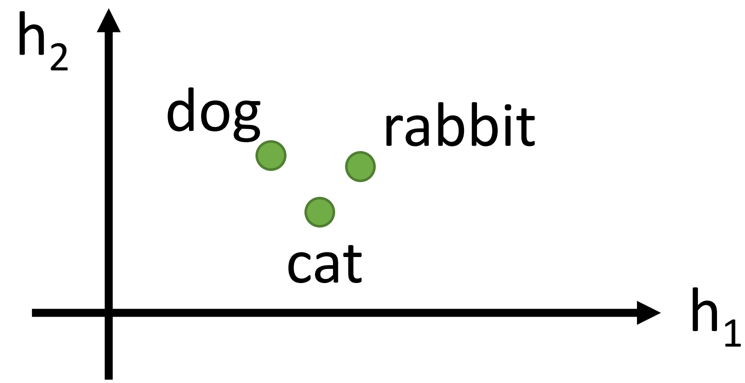
P(b|a): not from count, but the NN that can predict the next word.



Neural-network based LM



The hidden layer of the related words are close.



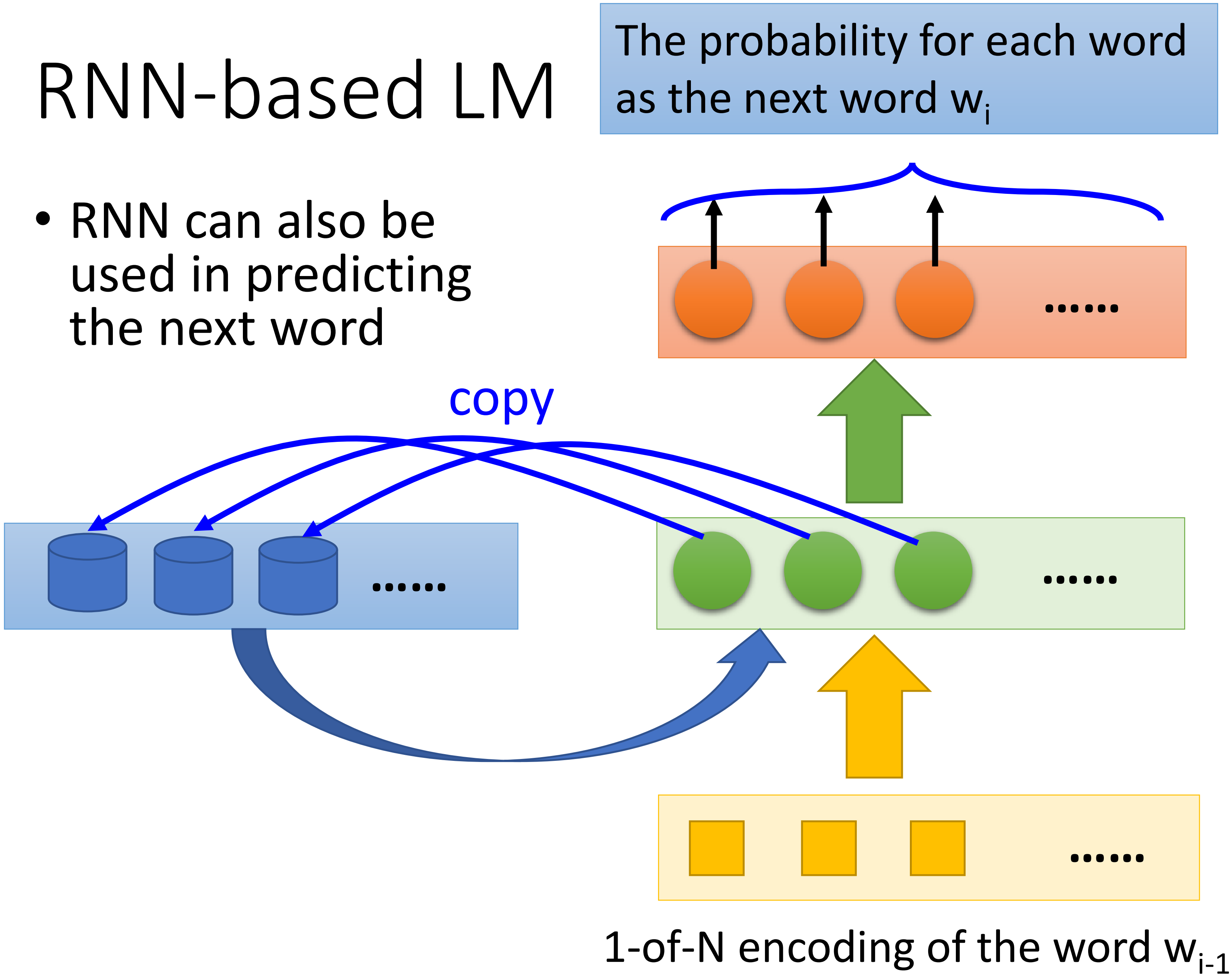
If P(jump|dog) is large, then P(jump|cat) increase accordingly.

(even there is not “… cat jump …” in the data)

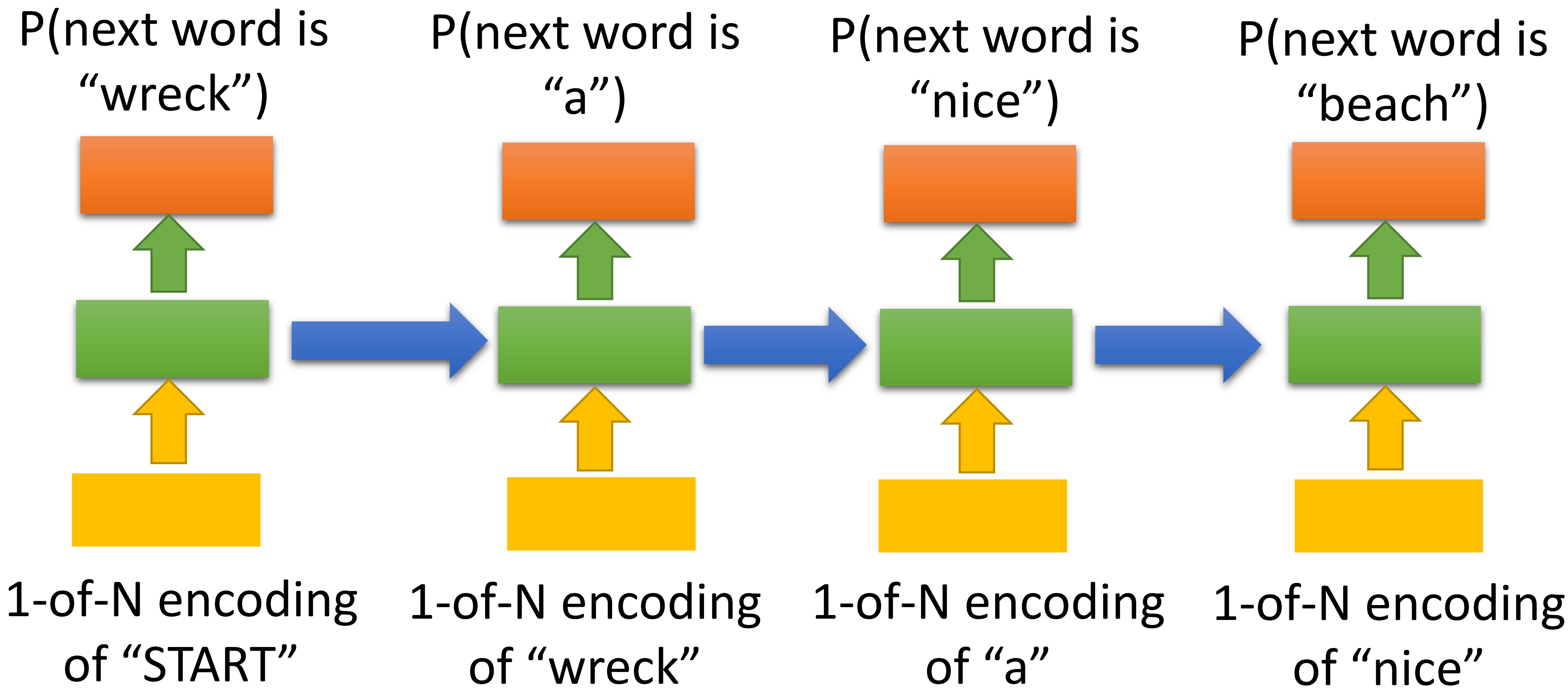
Smoothing is automatically done.

RNN-based LM

- RNN can also be used in predicting the next word



RNN-based LM

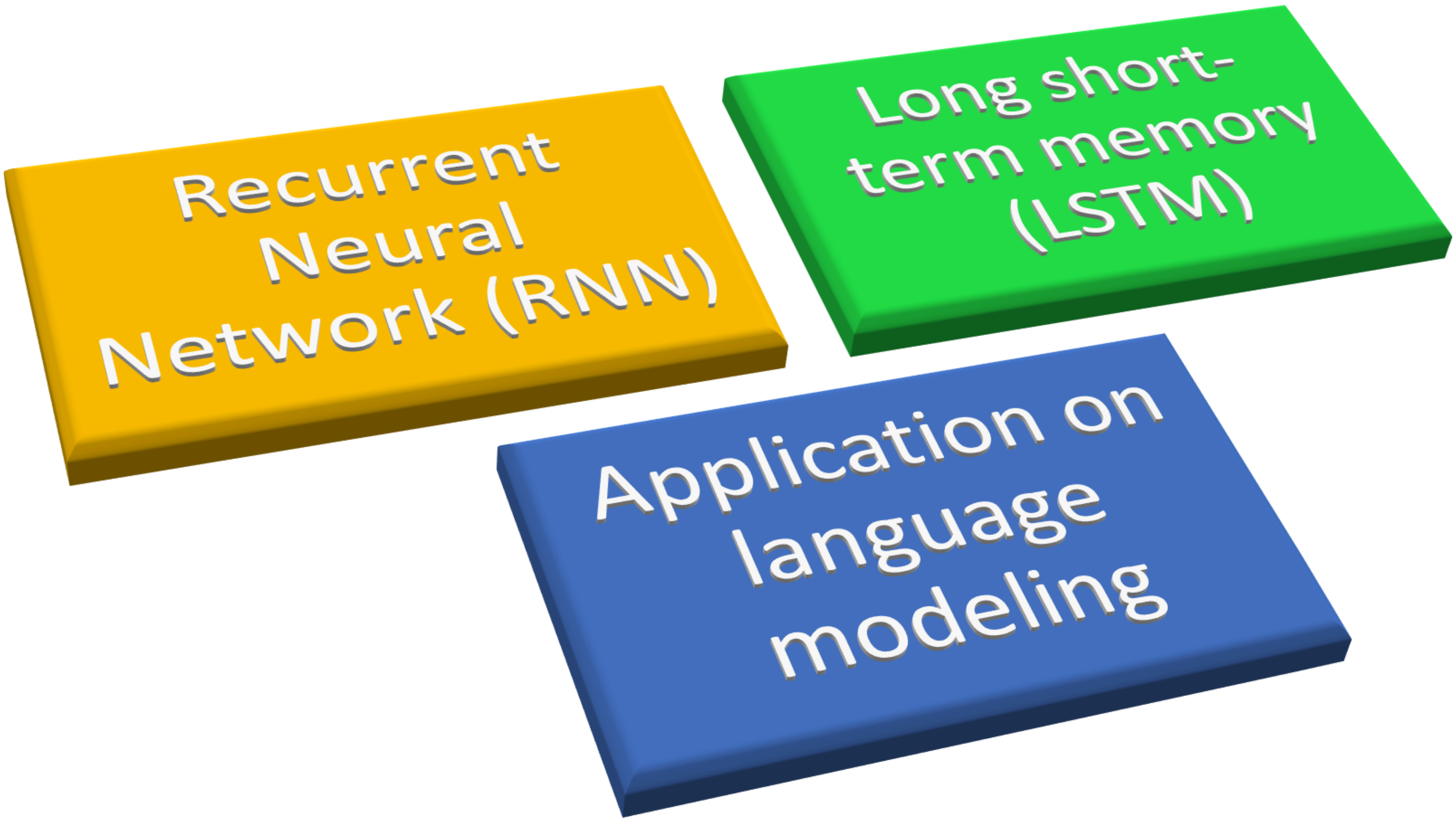


- Model long-term information
- Can also consider LSTM

Another Applications

- Composer
  - http://people.idsia.ch/~juergen/blues/
- Sentence generation
  - http://www.cs.toronto.edu/~ilya/rnn.html

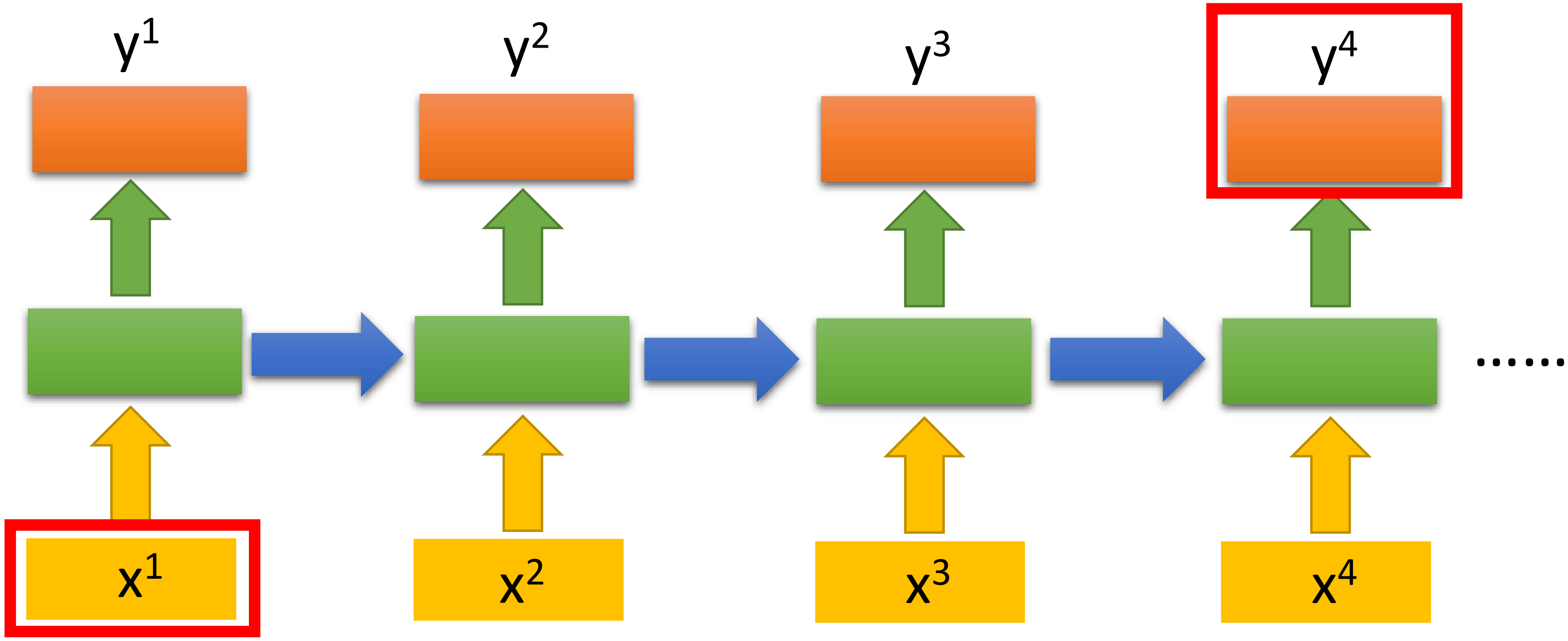
Summary – Network with Memory




Thank You

Appendix

Long term memory

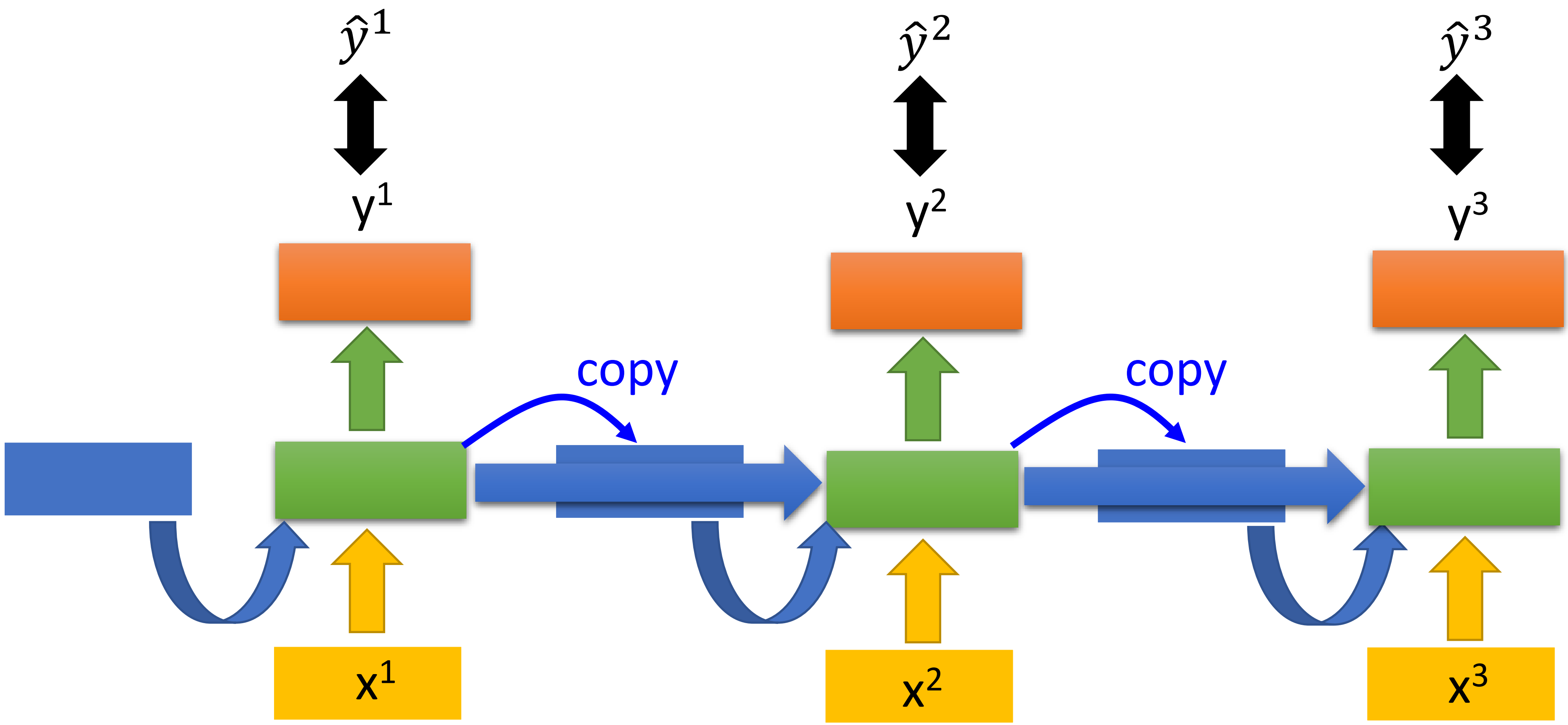


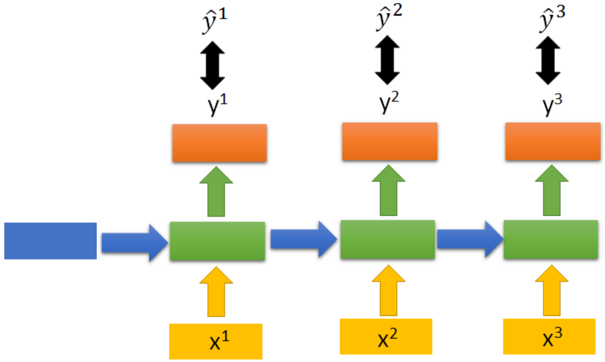
RNN - Training

Training data: x^1 x^2 x^3 ......

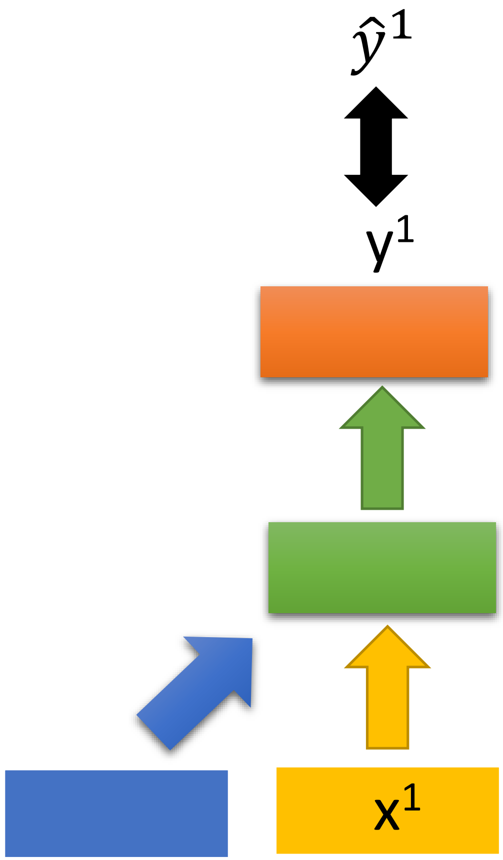
$\hat{y}^1$ $\hat{y}^2$ $\hat{y}^3$ ......

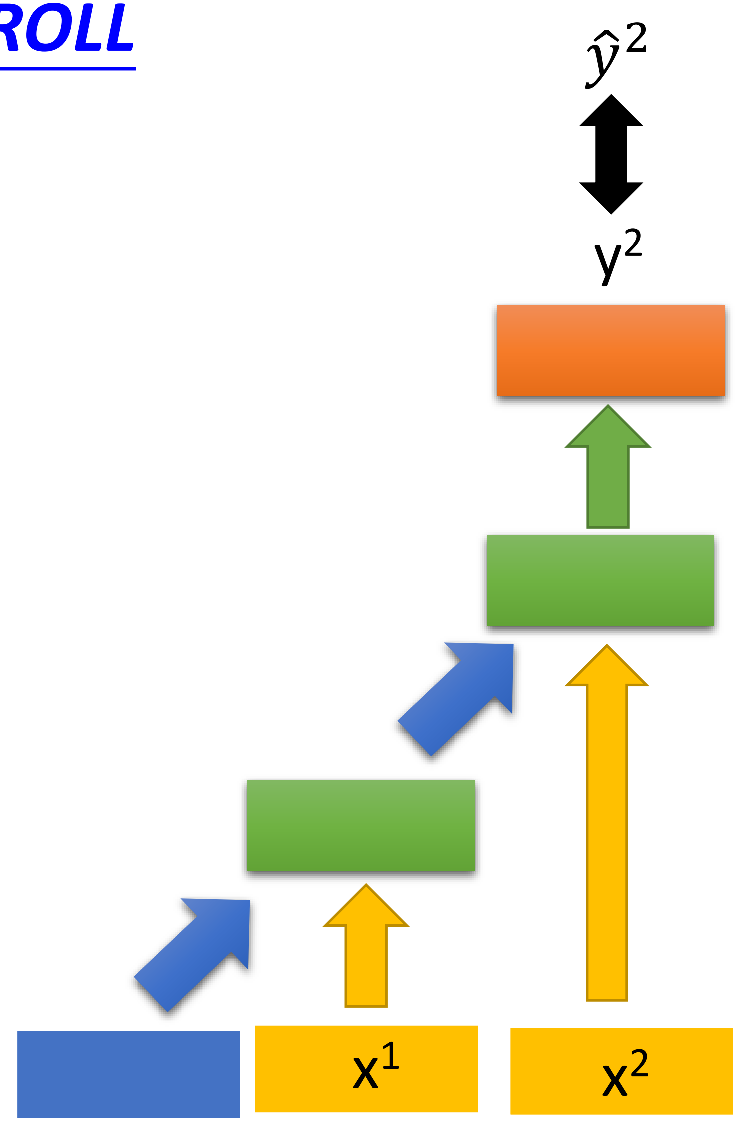
Training the parameters to let y close to $\hat{y}$

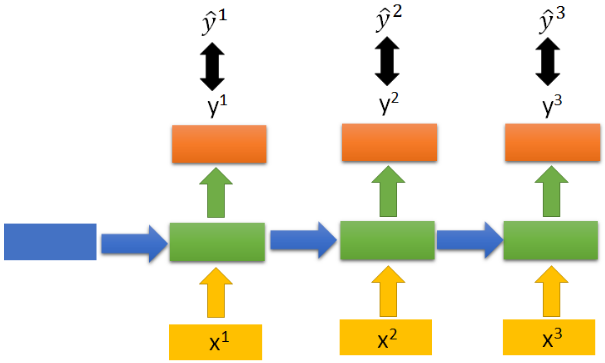




UNROLL





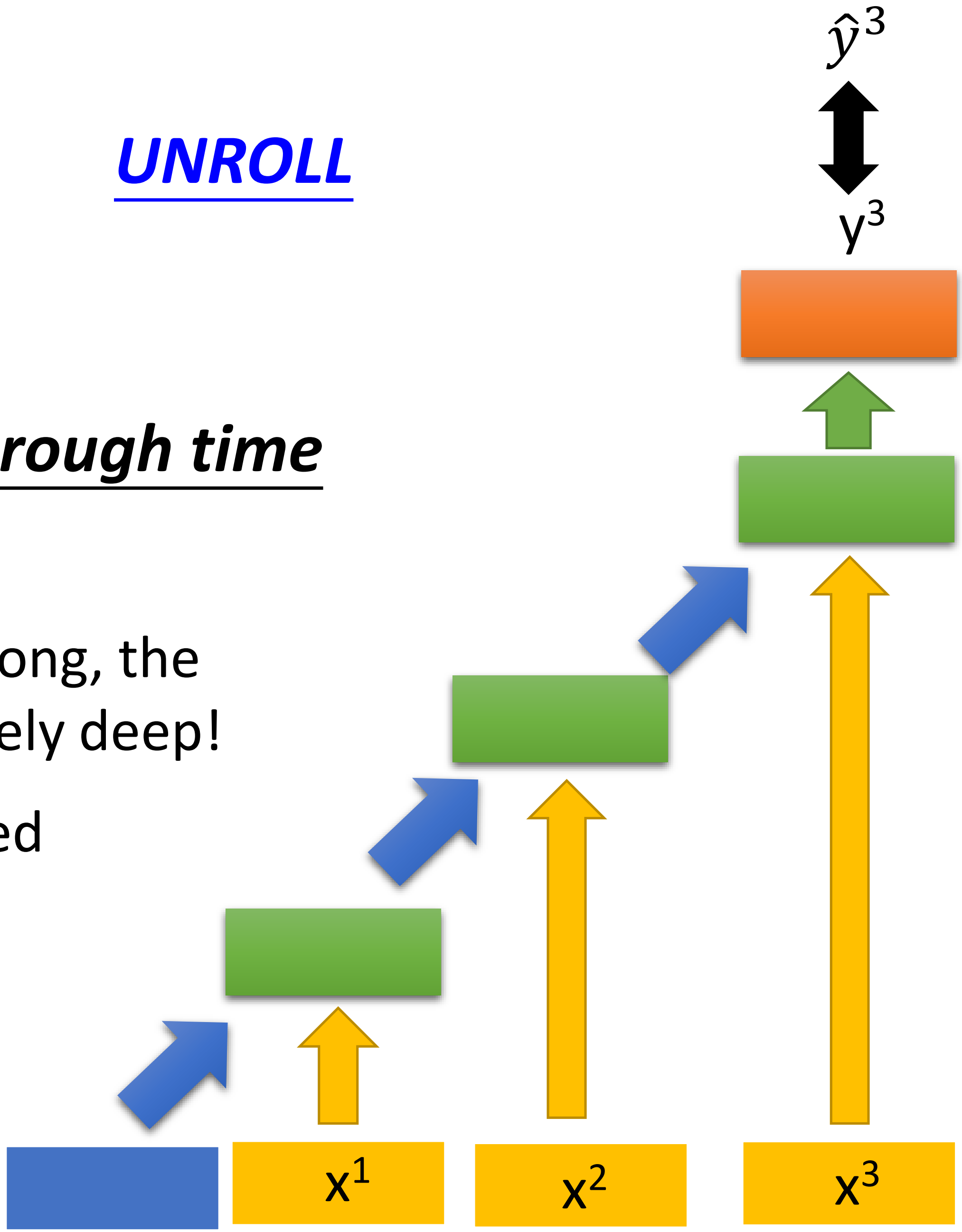


UNROLL

Backpropagation through time (BPTT)

When the sequence is long, the network can be extremely deep!

Some weights are shared

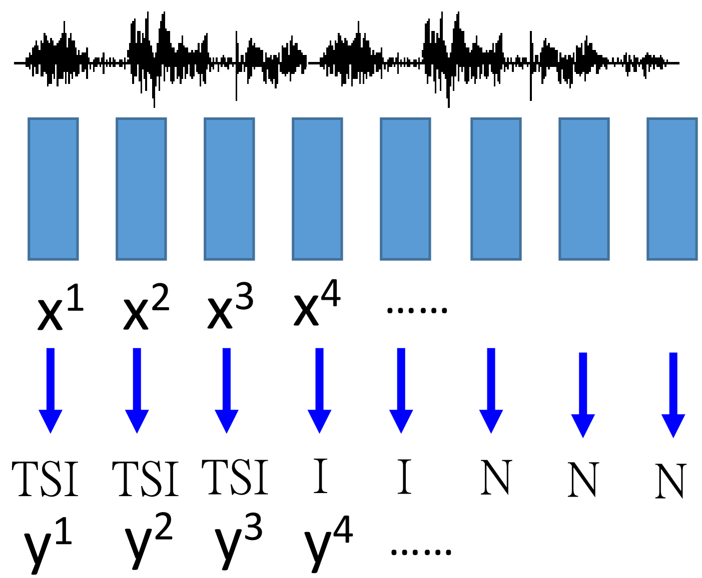


The task that needs memory
Structured Leargning v.s. RNN

POS Tagging

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

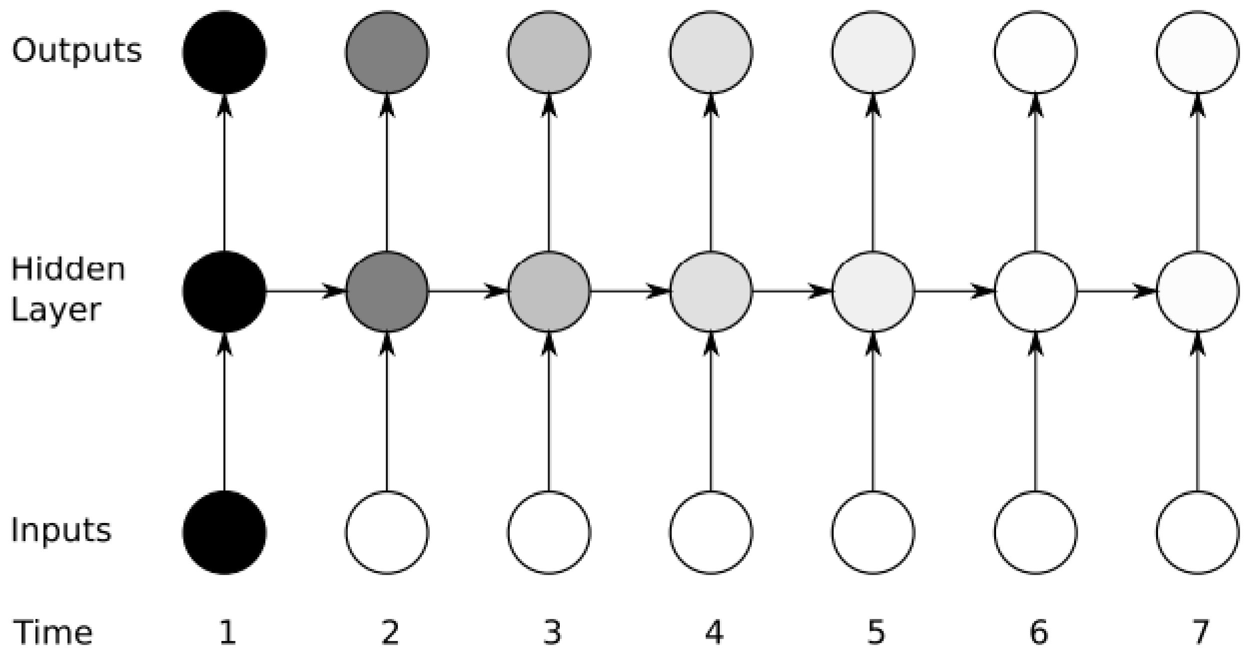
Speech Recognition

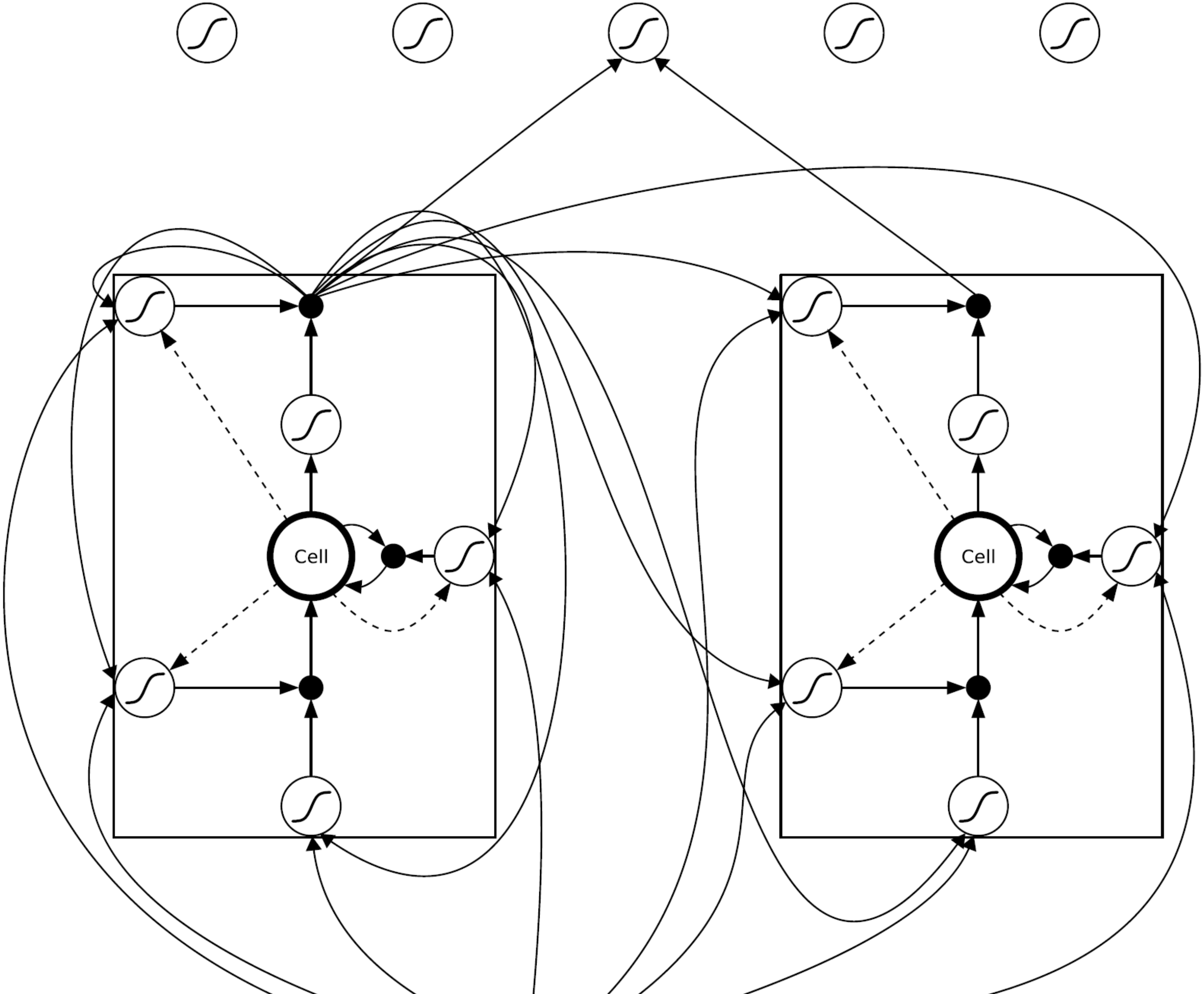


- ➢ Structured learning can also deal with these problems
- ➢ What are their difference?

Outlook

- Speech recognition
  - http://www.cs.toronto.edu/~fritz/absps/RNN13.pdf
- Structured + Deep
  - http://research.microsoft.com/pubs/210167/rcrf_v9.pdf


Outputs
Hidden
Layer
Inputs
Time
1
2
3
4
5
6
7


Cell
Cell

Neural-network based LM

- Training
  - Training data: “Hello how are you …… ”