Variants of Neural Networks

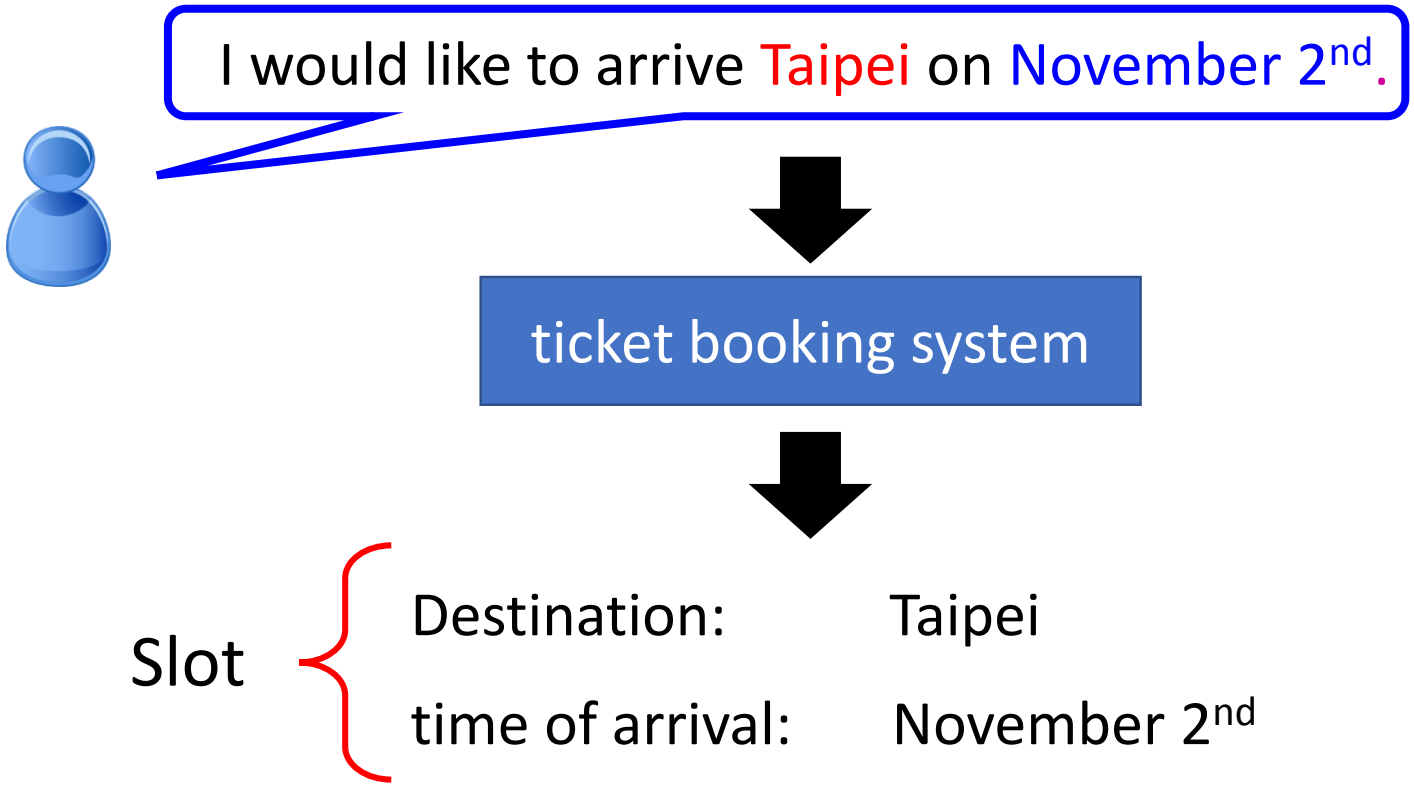
Convolutional Neural Network (CNN)

Recurrent Neural Network (RNN)

Neural Network with Memory

Example Application

- Slot Filling

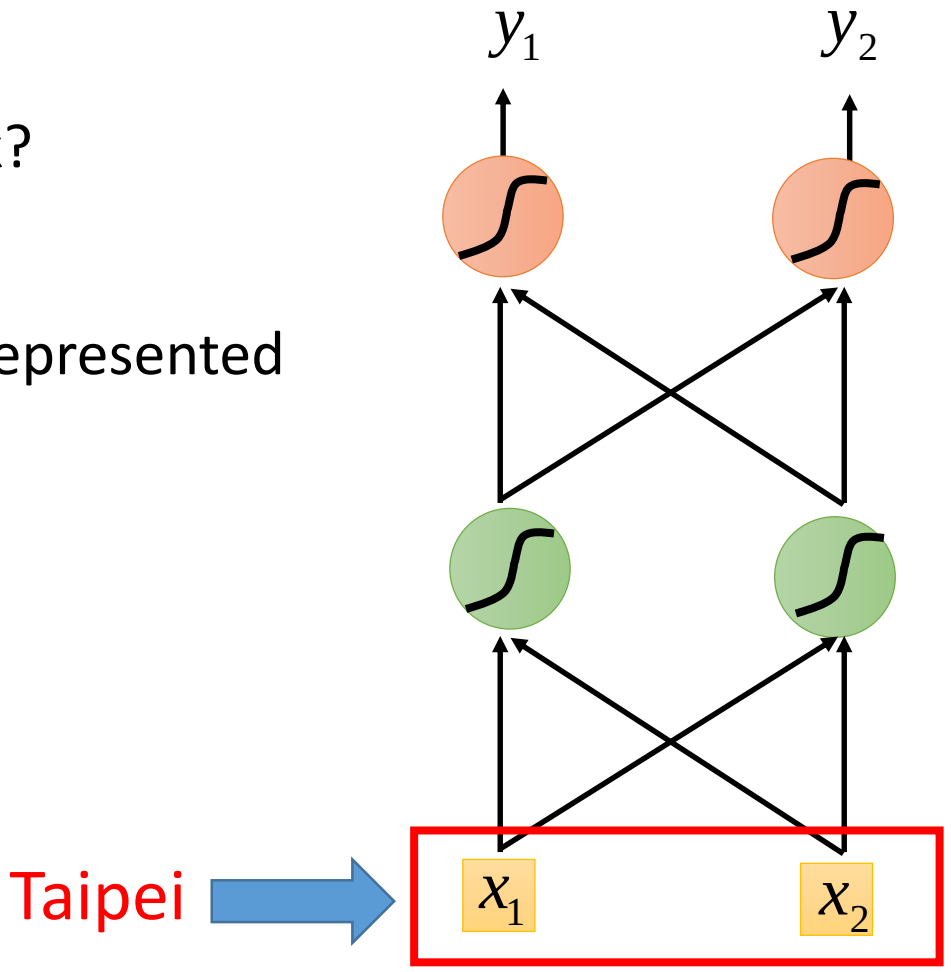


Example Application

Solving slot filling by Feedforward network?

Input: a word

(Each word is represented as a vector)



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]

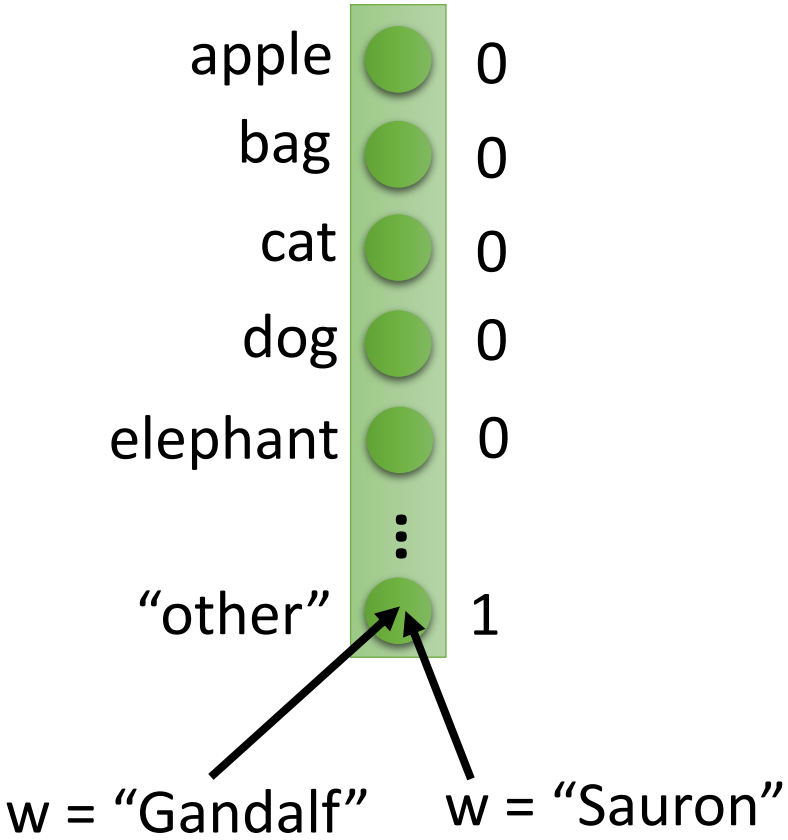
cat = [0 0 1 0 0]

dog = [0 0 0 1 0]

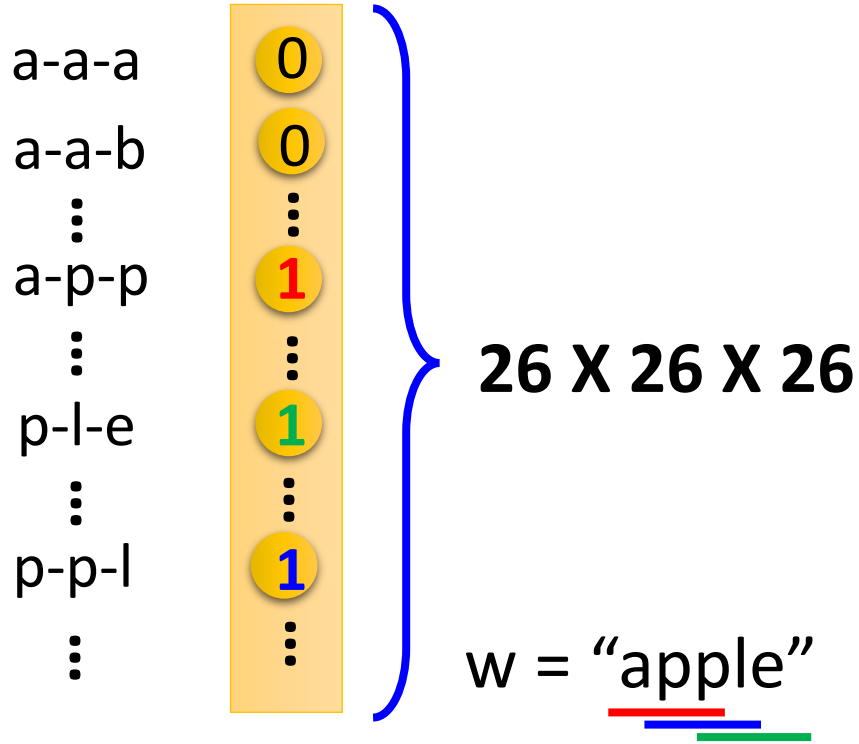
elephant = [0 0 0 0 1]

Beyond 1-of-N encoding

Dimension for "Other"



Word hashing



Example Application

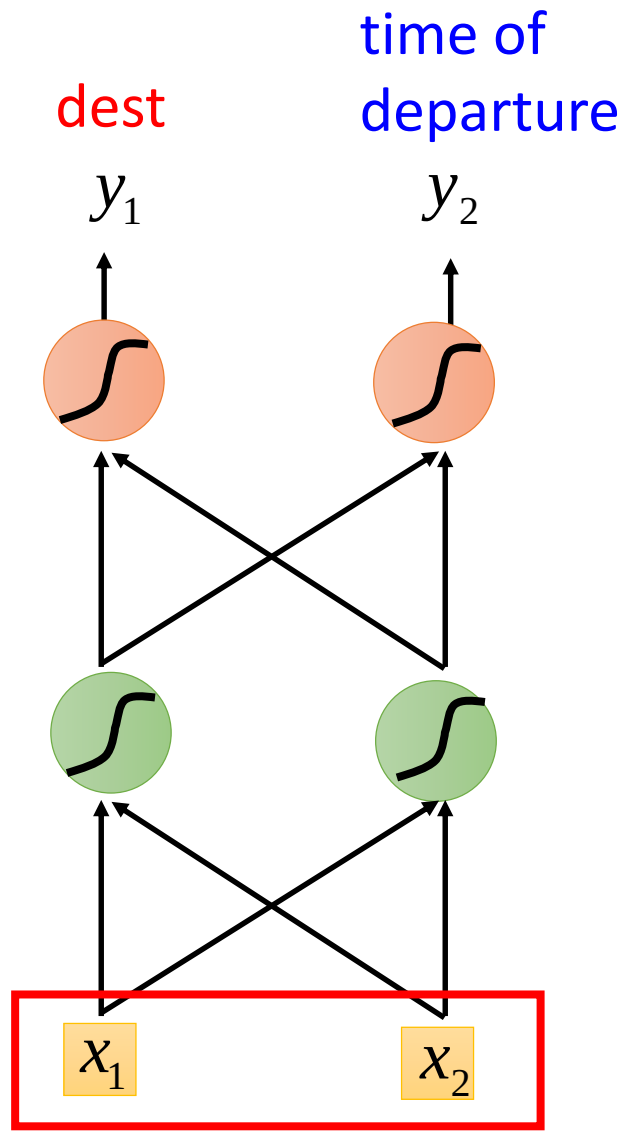
Solving slot filling by Feedforward network?

Input: a word

(Each word is represented as a vector)

Output:

Probability distribution that the input word belonging to the slots



Example Application

arrive Taipei on November 2nd

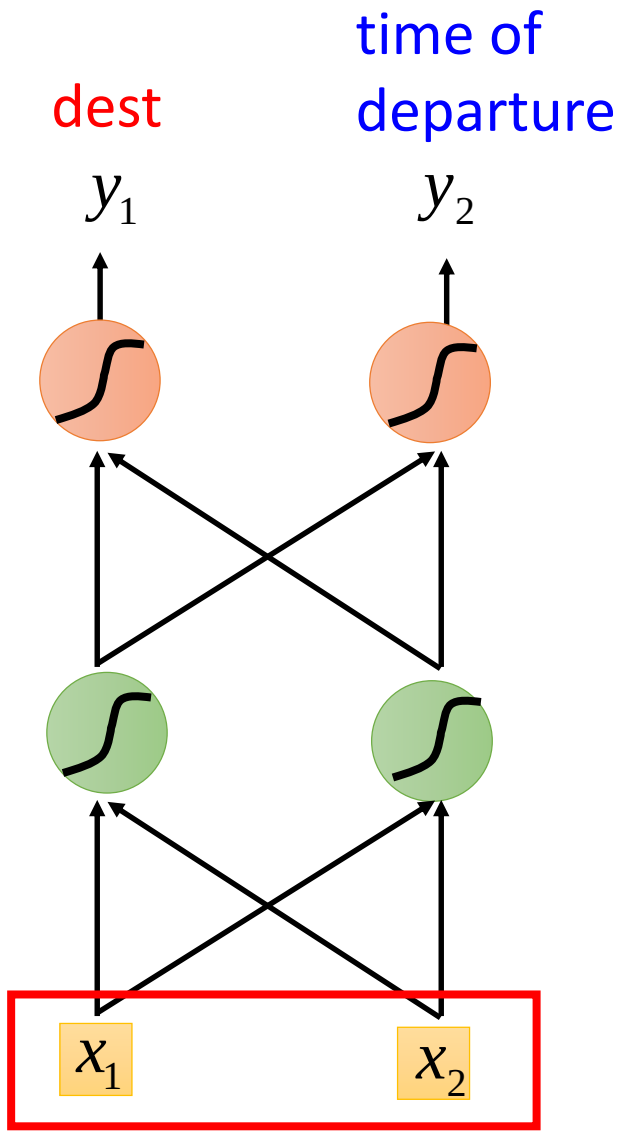
other dest other time time

Problem?

leave Taipei on November 2nd

place of departure

Neural network needs memory!



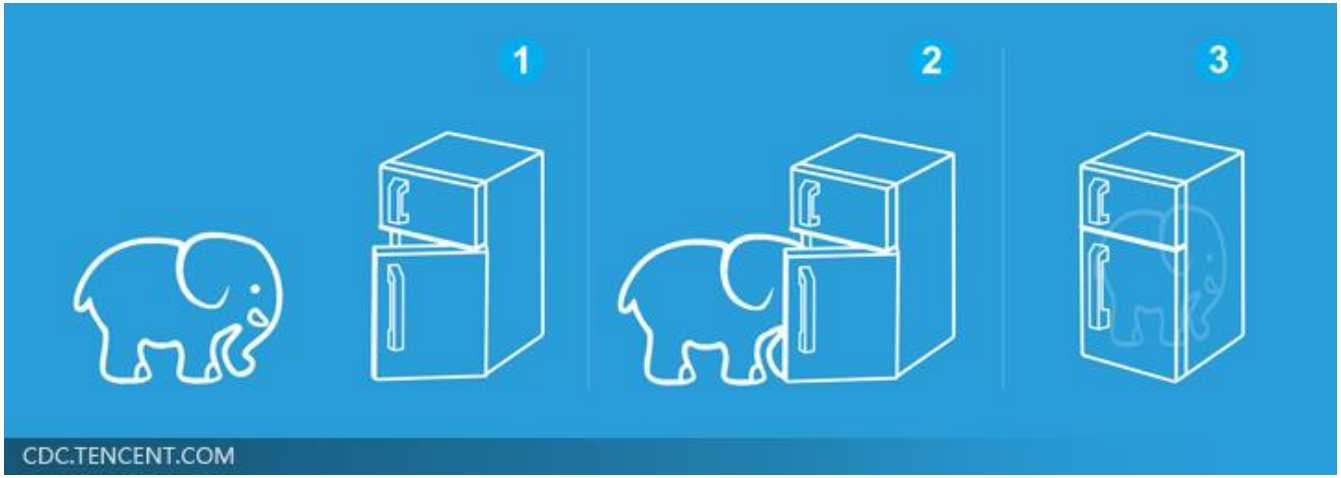
Three Steps for Deep Learning

Step 1:
Recurrent Neural Network

Step 2: goodness of function

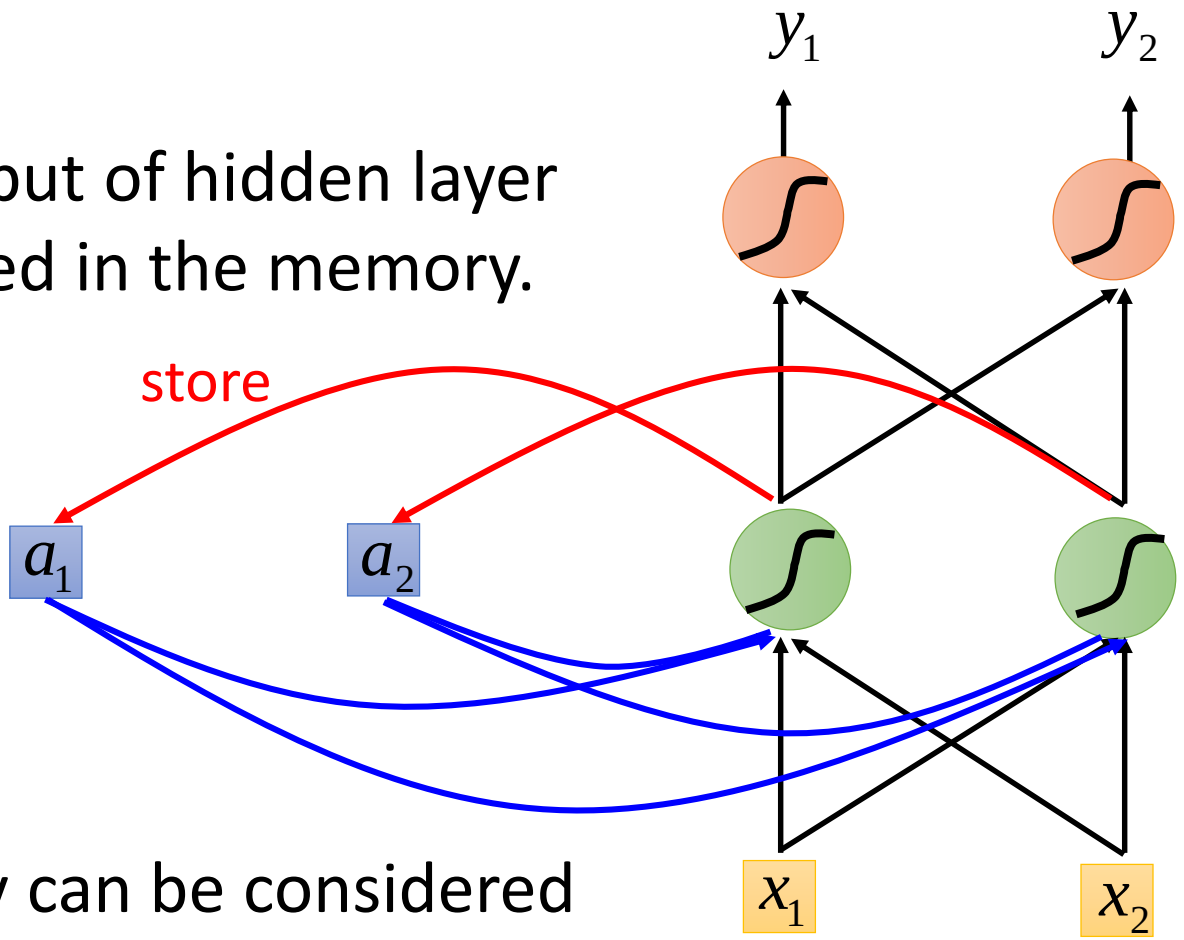
Step 3: pick the best function

Deep Learning is so simple ……



Recurrent Neural Network (RNN)

The output of hidden layer are stored in the memory.



Memory can be considered as another input.

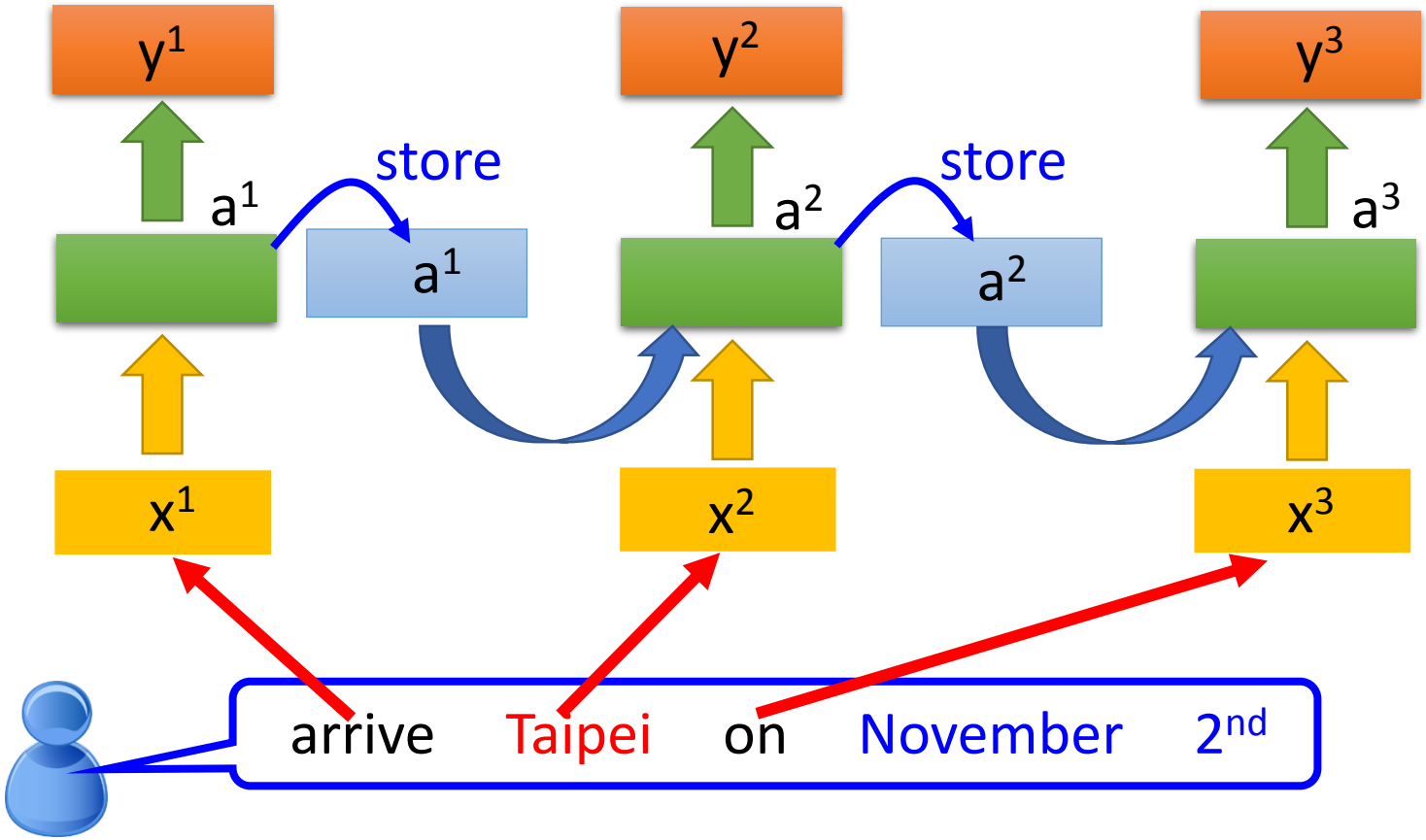
RNN

The same network is used again and again.

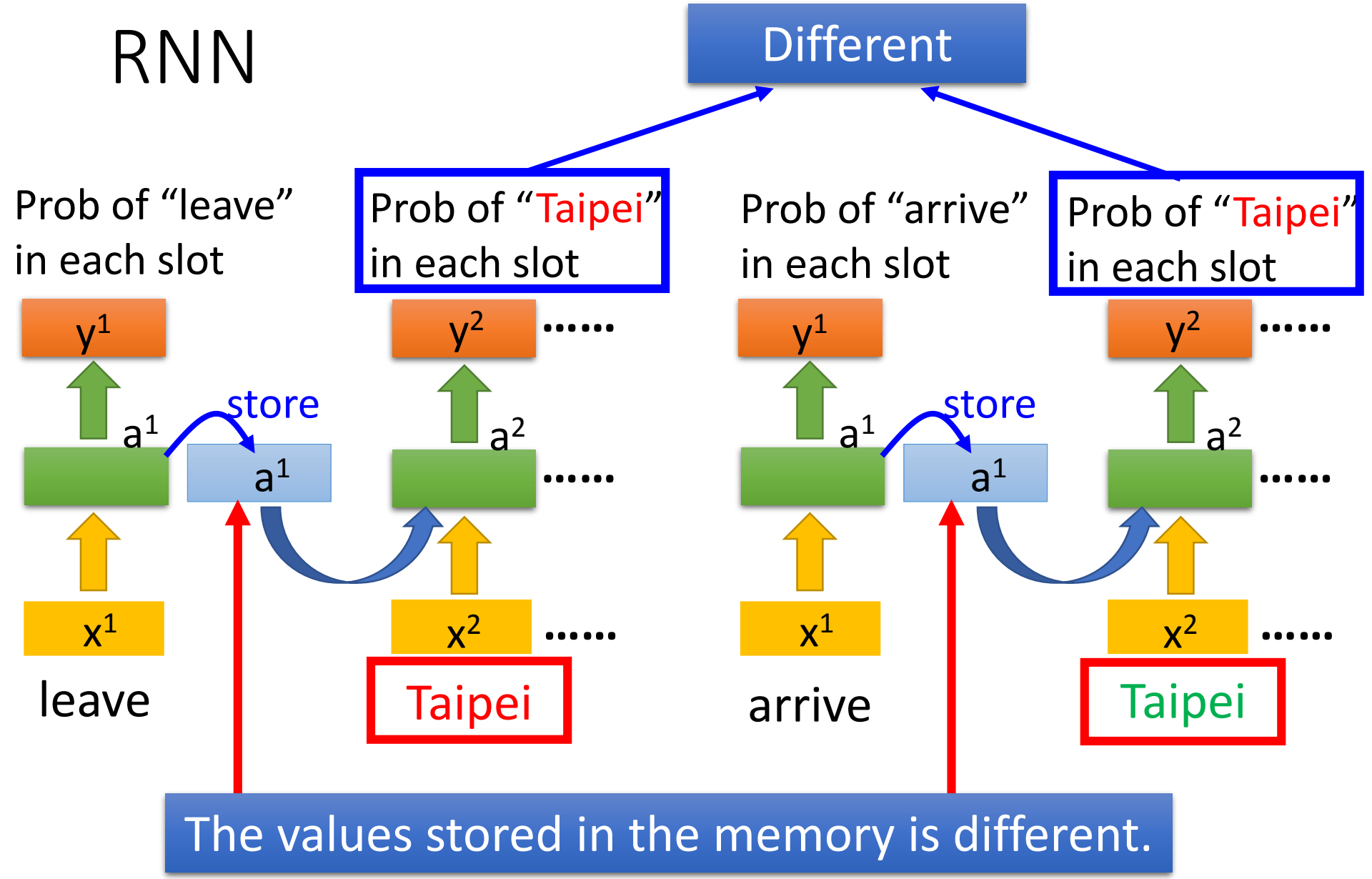
Probability of "arrive" in each slot

Probability of "Taipei" in each slot

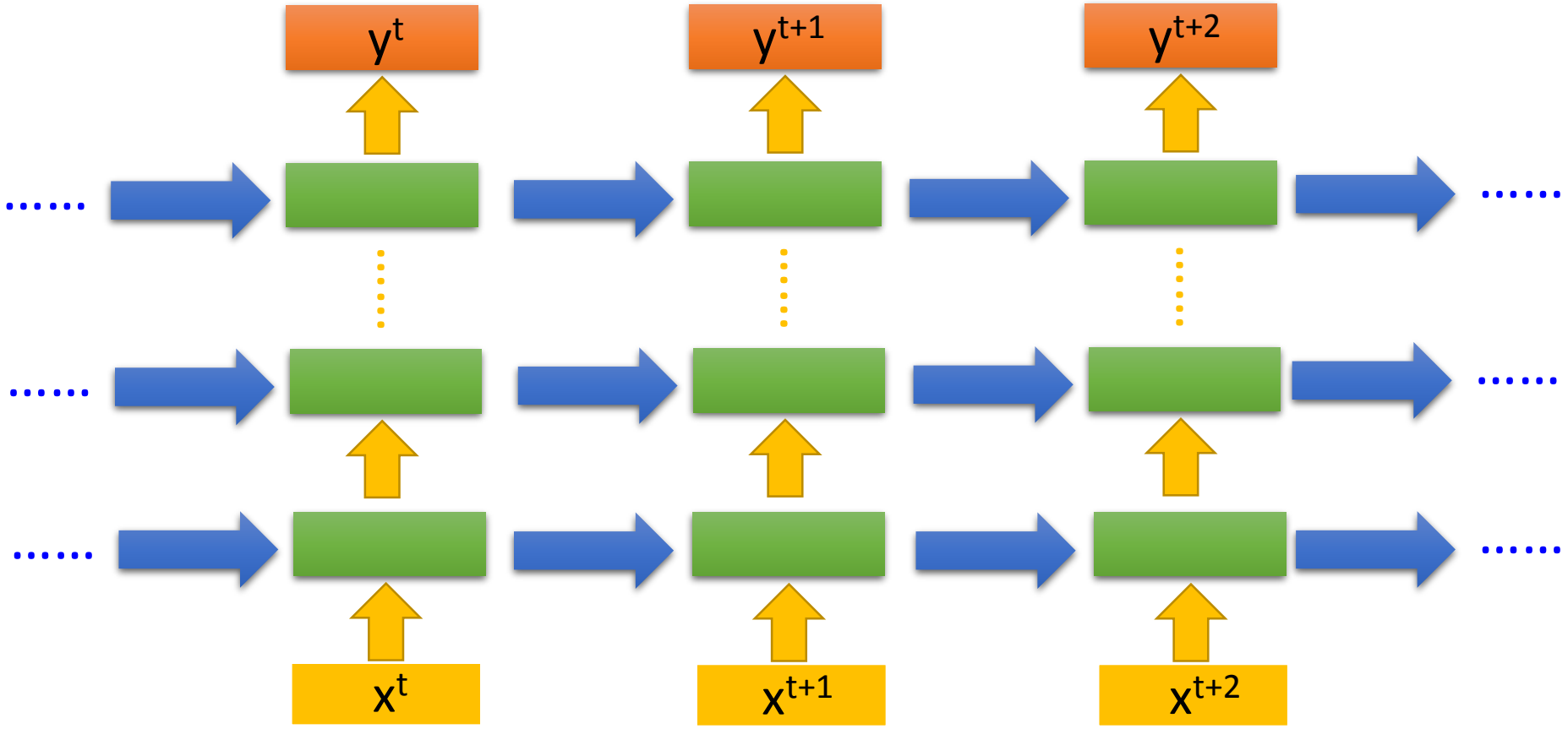
Probability of "on" in each slot



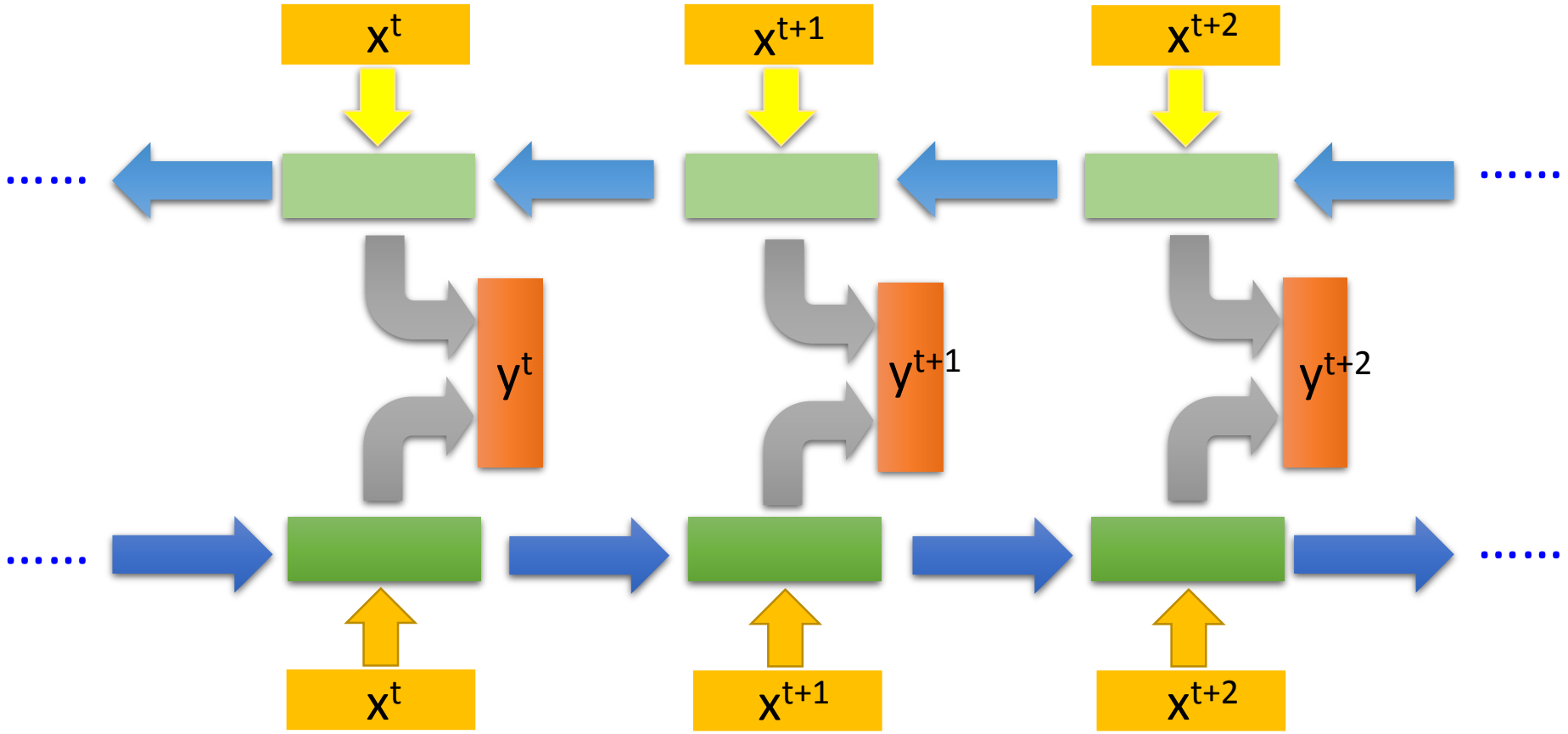
RNN


Different
Prob of "leave" in each slot
Prob of "Taipei" in each slot
Prob of "arrive" in each slot
Prob of "Taipei" in each slot
y^1
y^2
a^1
a^2
store
x^1
x^2
leave
Taipei
arrive
Taipei
The values stored in the memory is different.

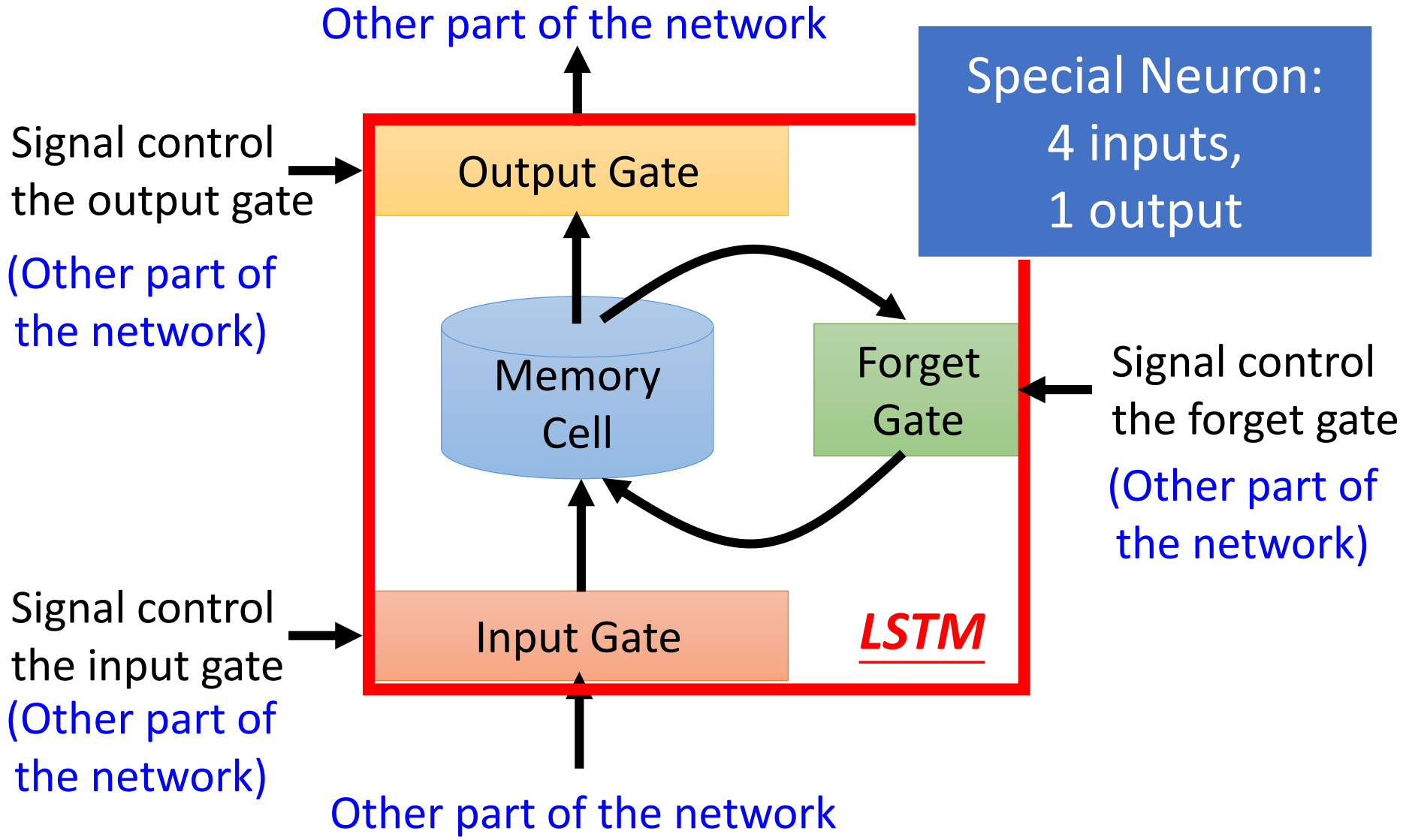
Of course it can be deep …

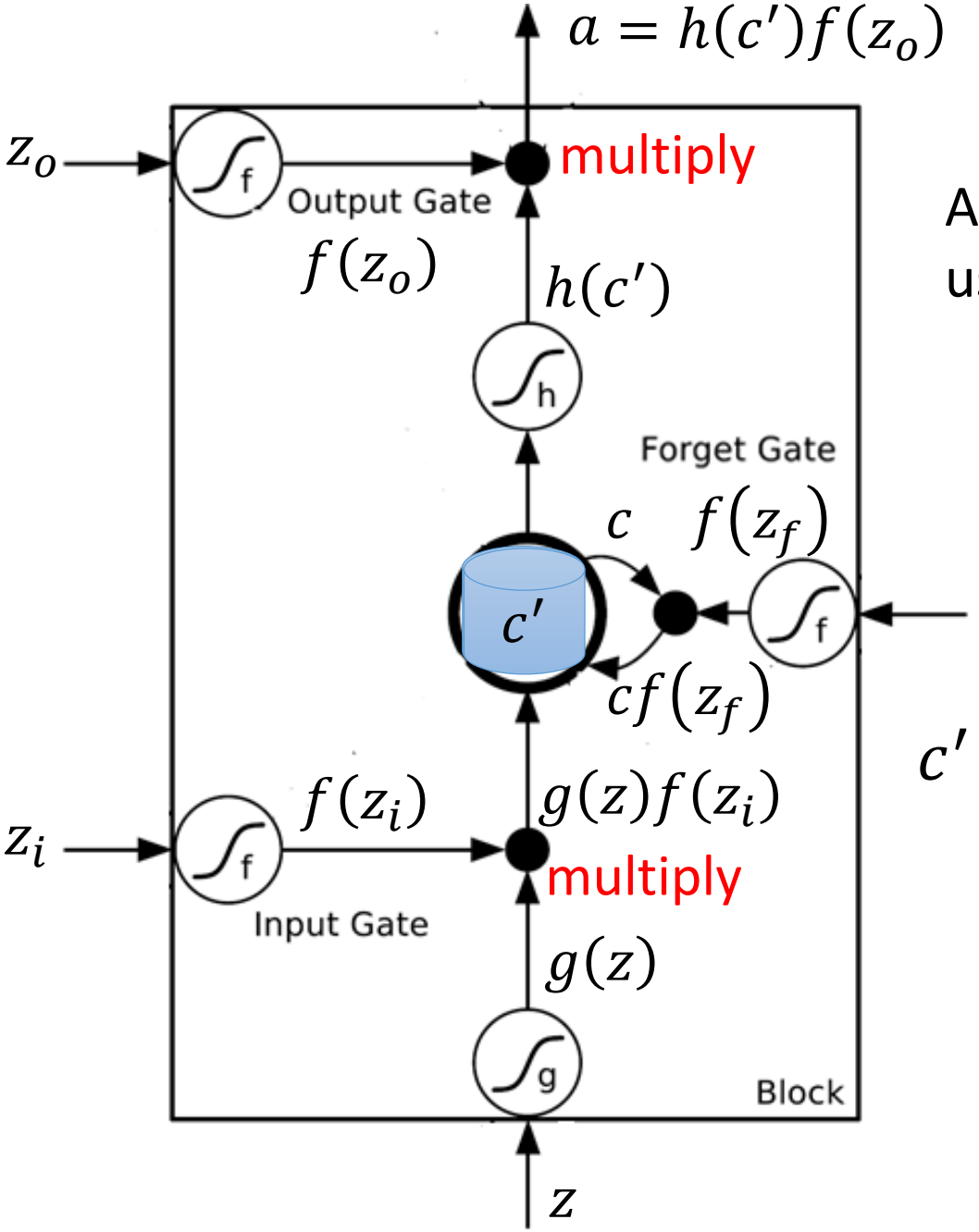


Bidirectional RNN



Long Short-term Memory (LSTM)



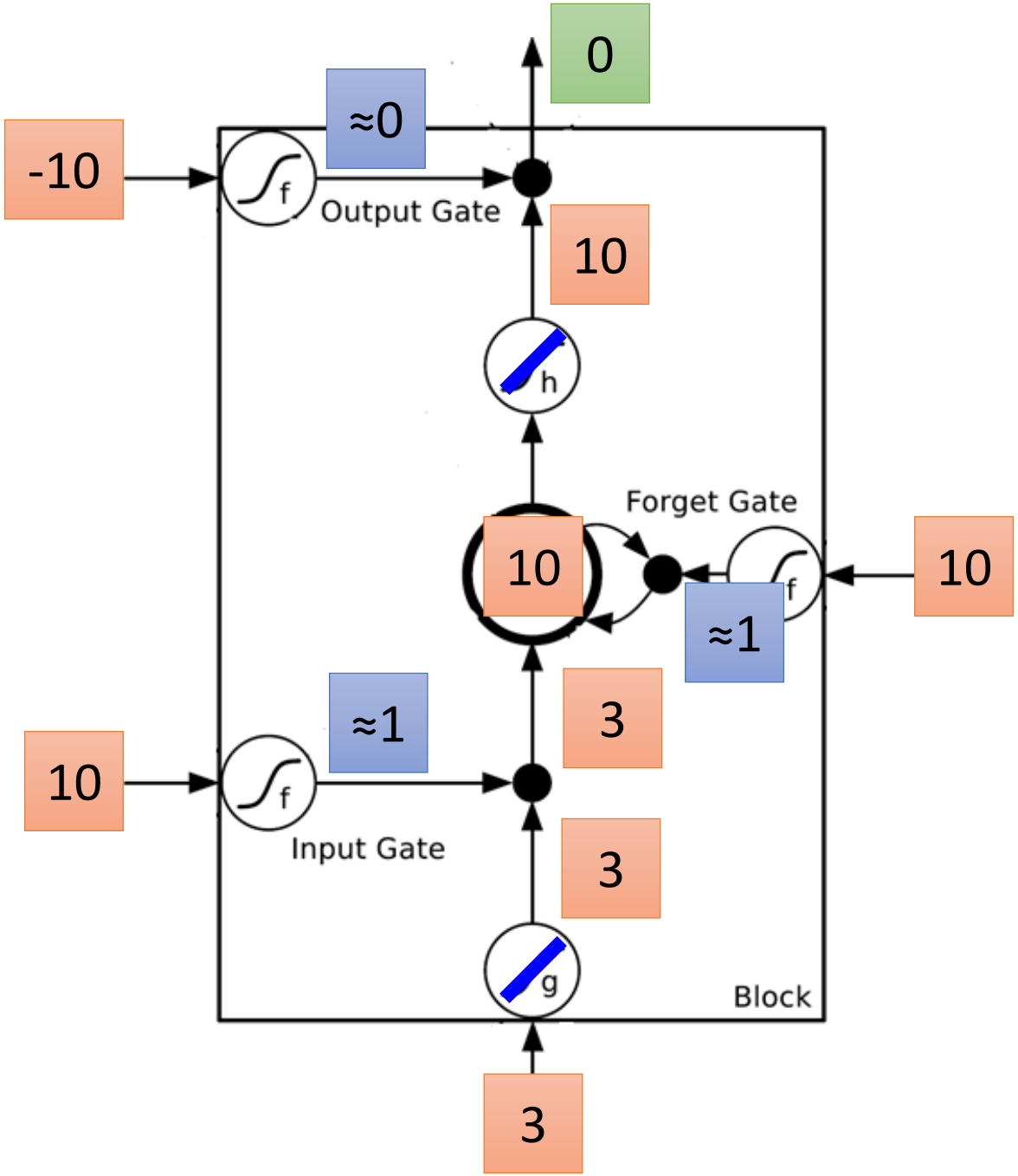


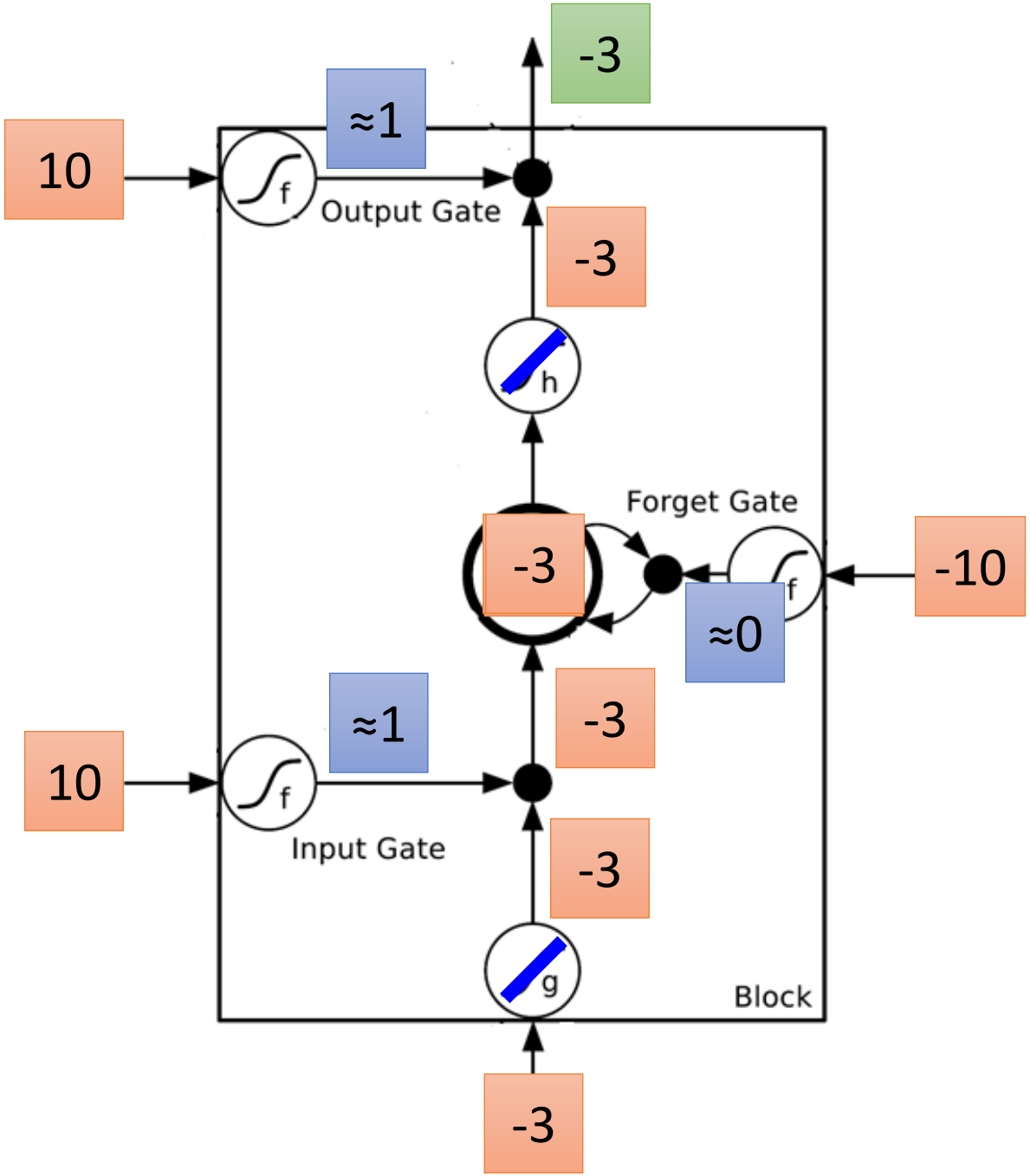
Activation function f is usually a sigmoid function

Between 0 and 1

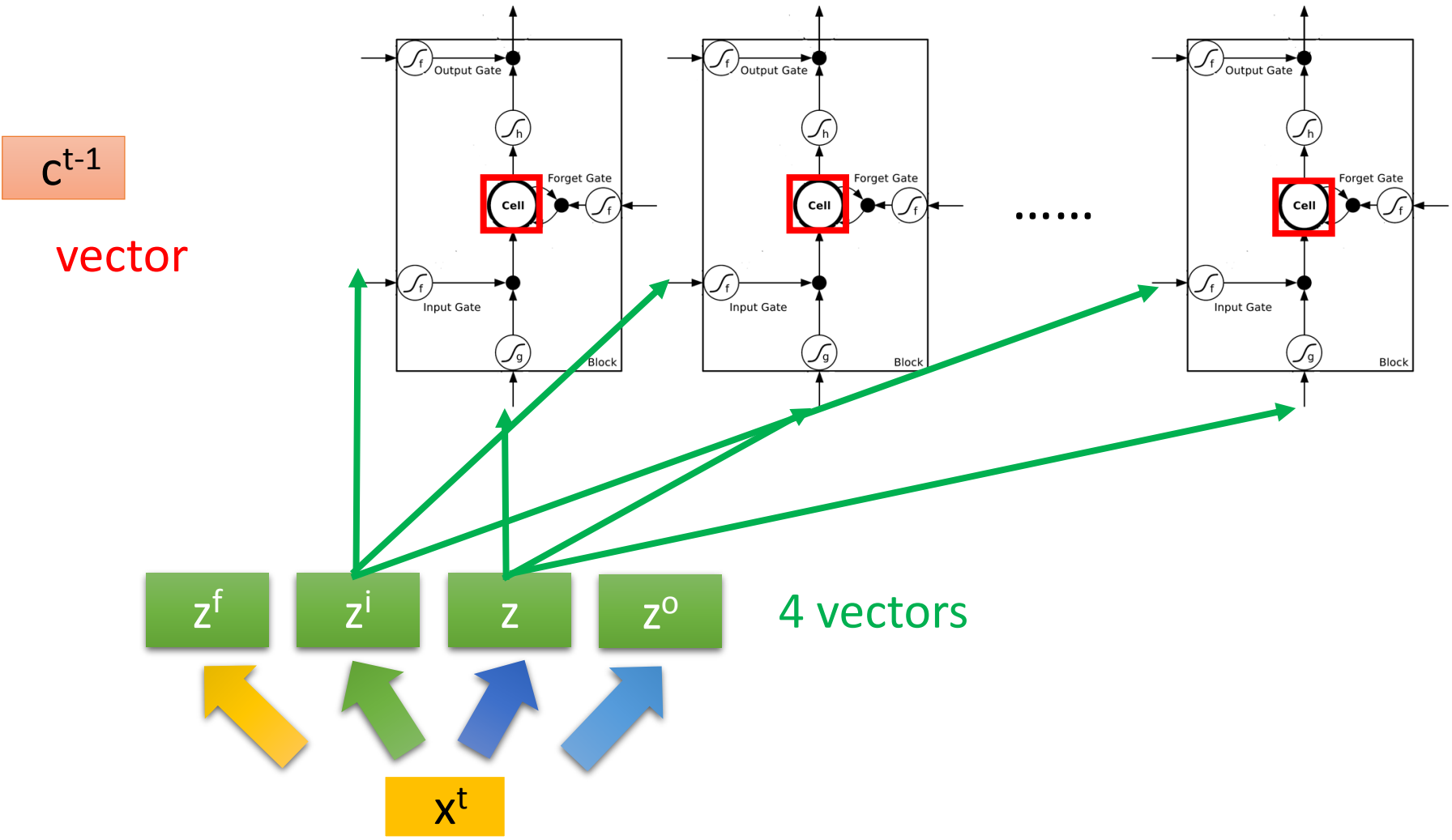
Mimic open and close gate

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

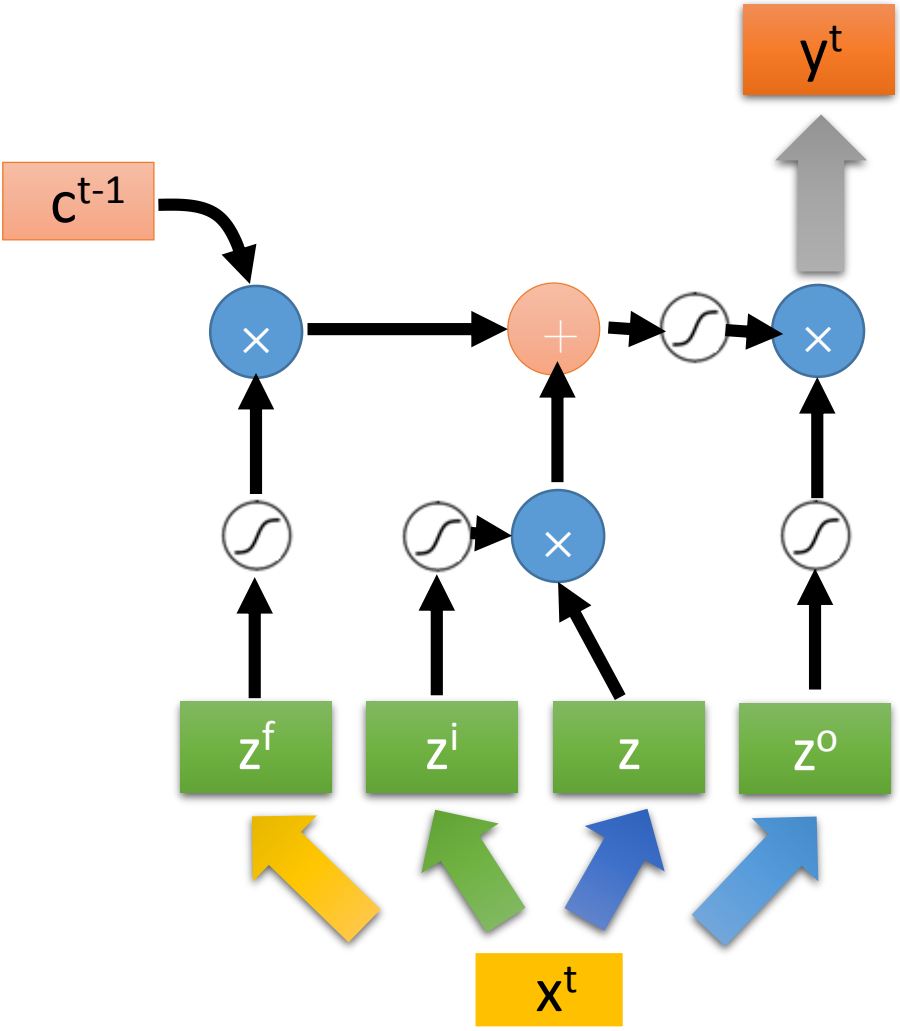

0
≈0
-10
f
Output Gate
10
h
Forget Gate
10
10
f
≈1
≈1
3
10
f
Input Gate
3
g
Block
3

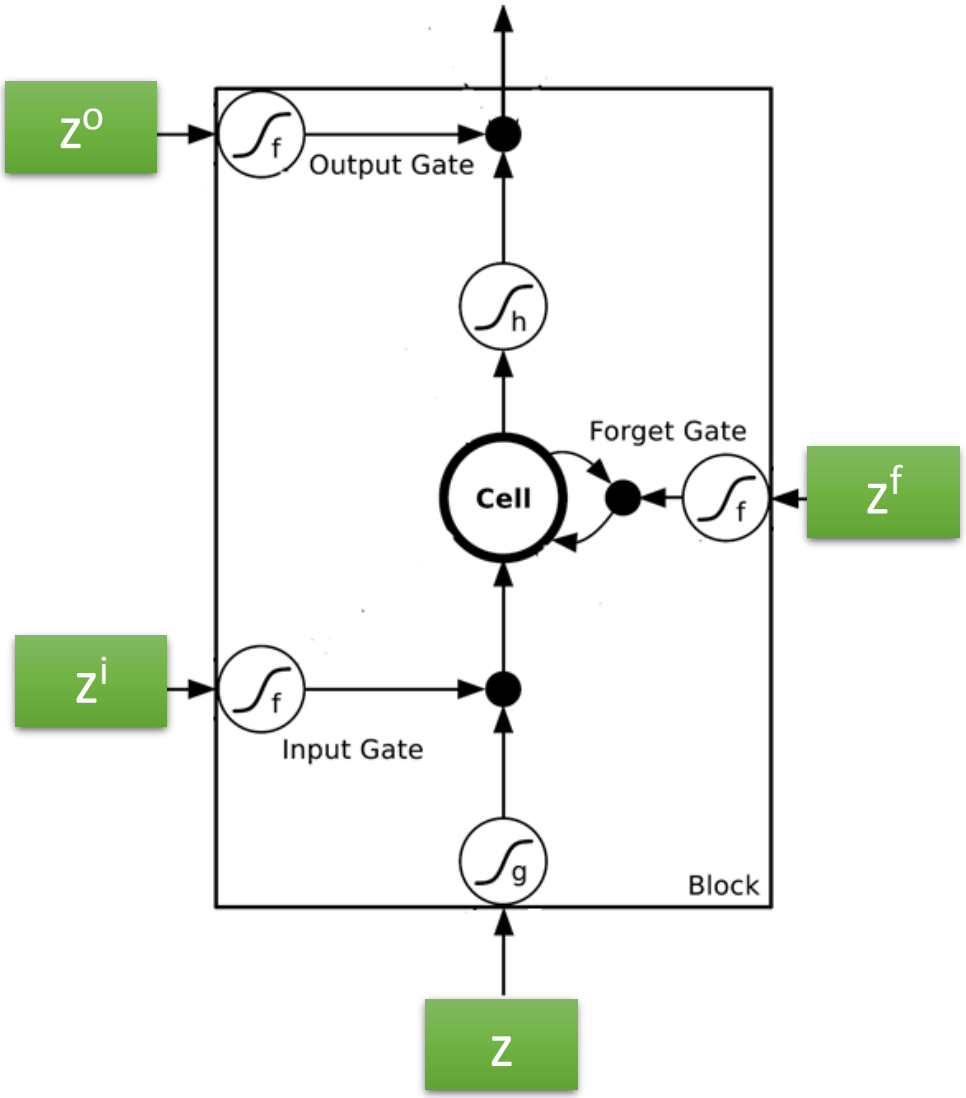

-3
≈1
10
f
Output Gate
-3
h
Forget Gate
-3
-10
f
≈0
≈1
-3
10
f
Input Gate
-3
g
Block
-3

LSTM


c^{t-1}
vector
Output Gate
Forget Gate
Cell
Input Gate
Block
.....
z^f
z^i
z
z^o
4 vectors
x^t

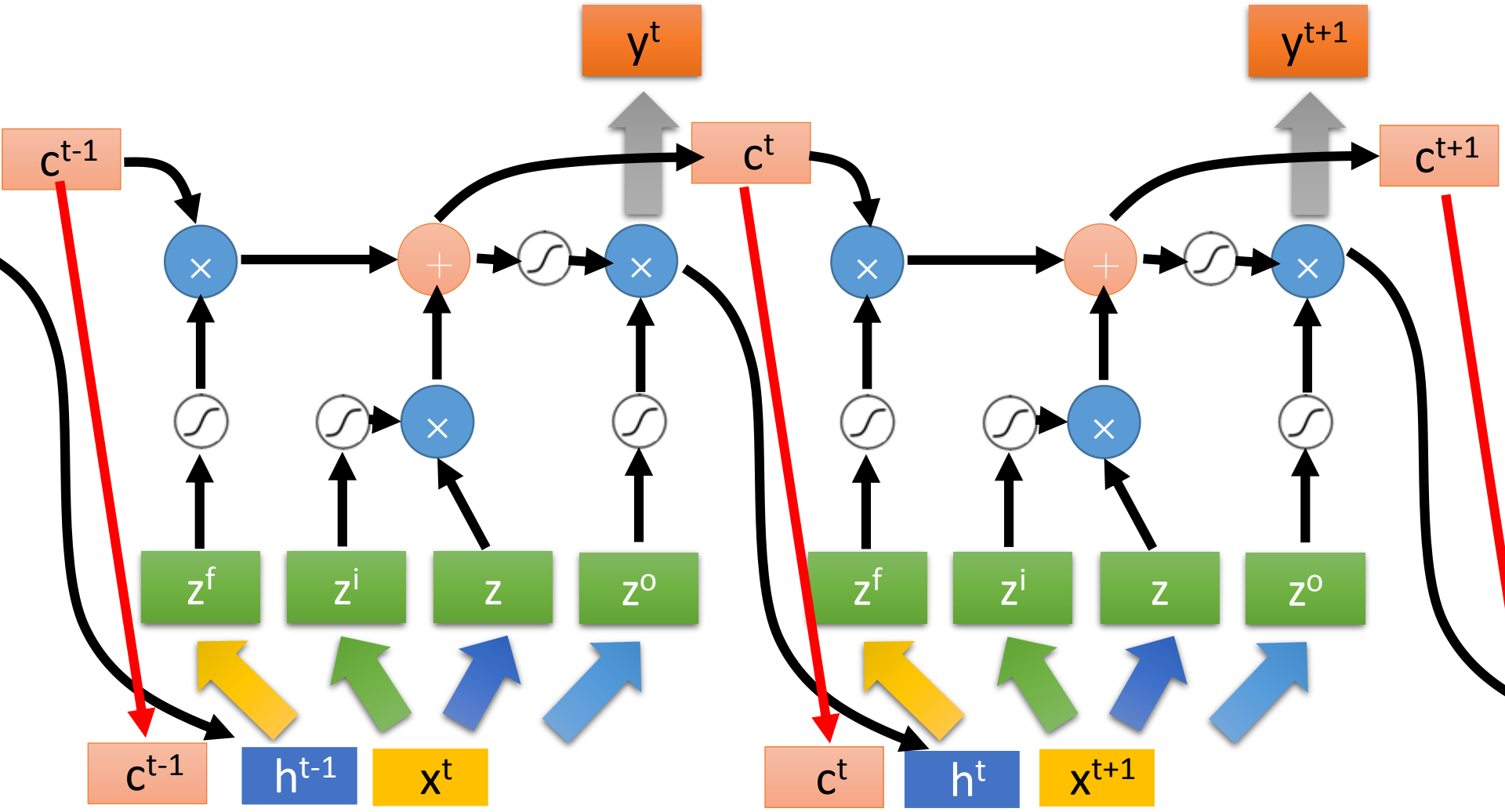
LSTM


y^t
c^{t-1}
z^f
z^i
z
z^o
x^t

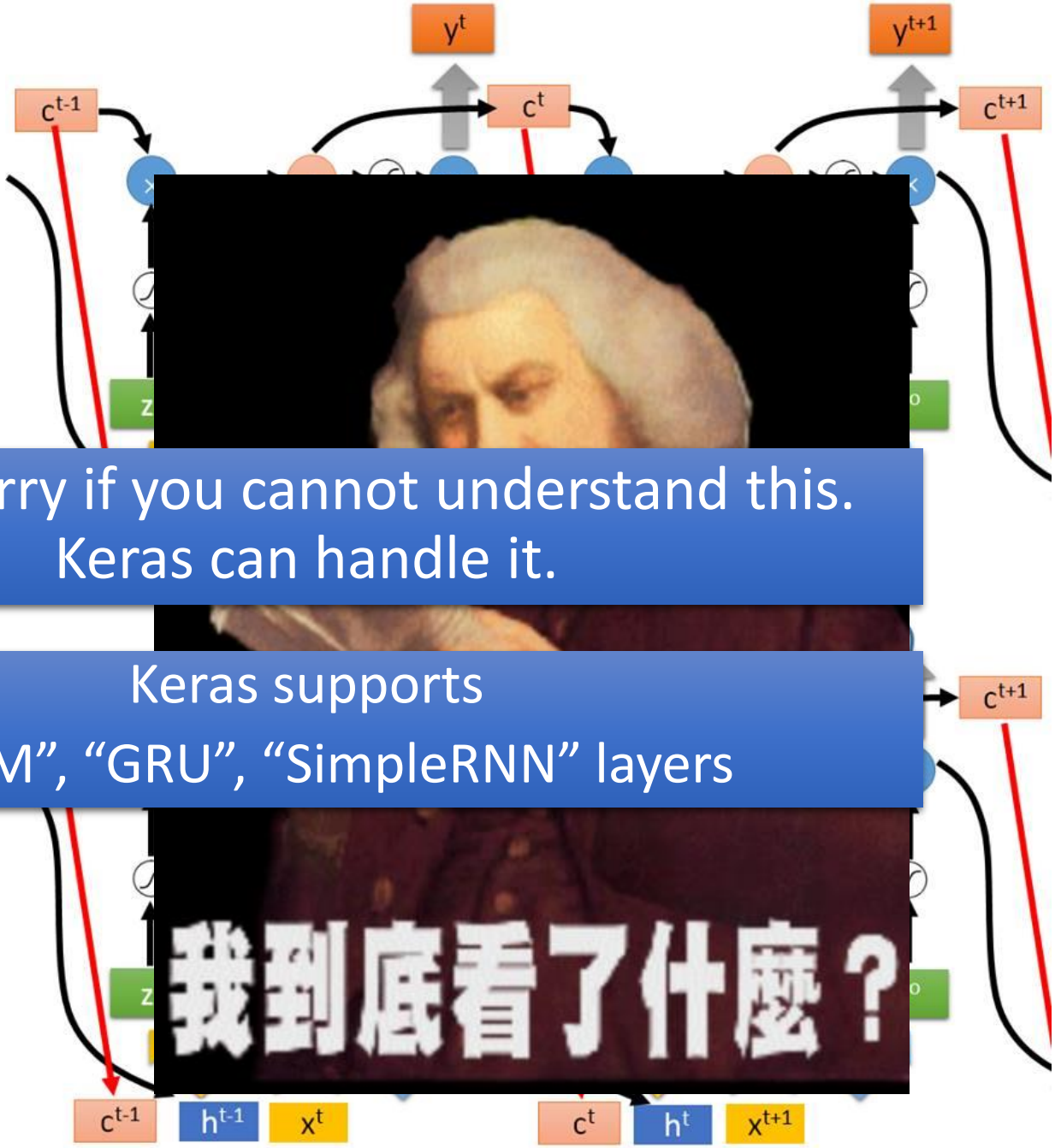

z^o
Output Gate
h
Forget Gate
Cell
z^f
z^i
Input Gate
g
Block
z

LSTM

Extension: “peephole”



Multiple-layer LSTM



Don’t worry if you cannot understand this.
Keras can handle it.

Keras supports
“LSTM”, “GRU”, “SimpleRNN” layers

This is quite standard now.

https://img.komicolle.org/2015-09-20/src/14426967627131.gif

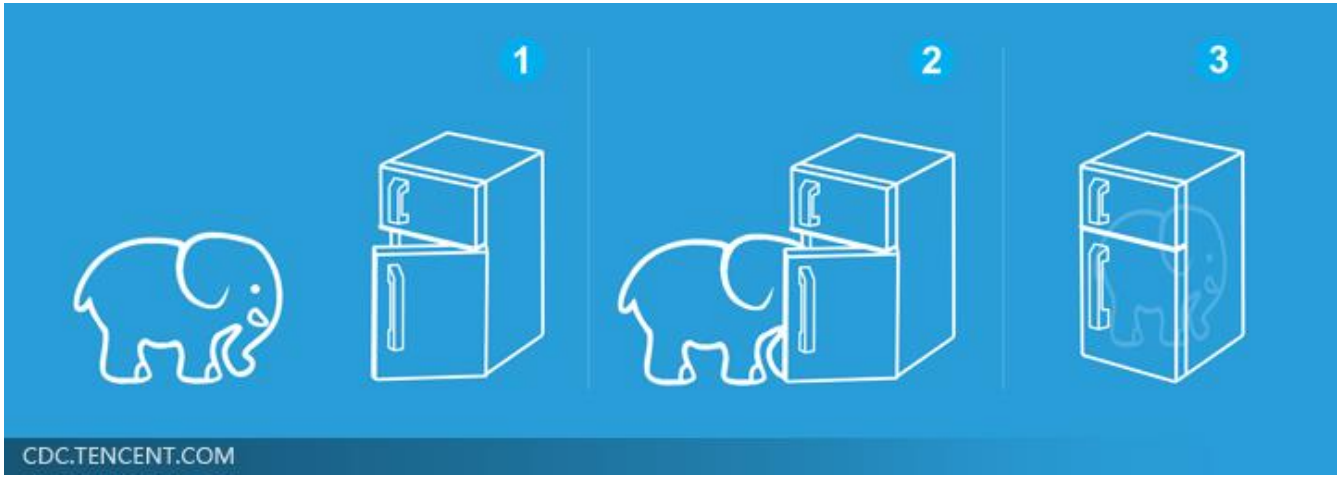
Three Steps for Deep Learning

Step 1: define a set of function

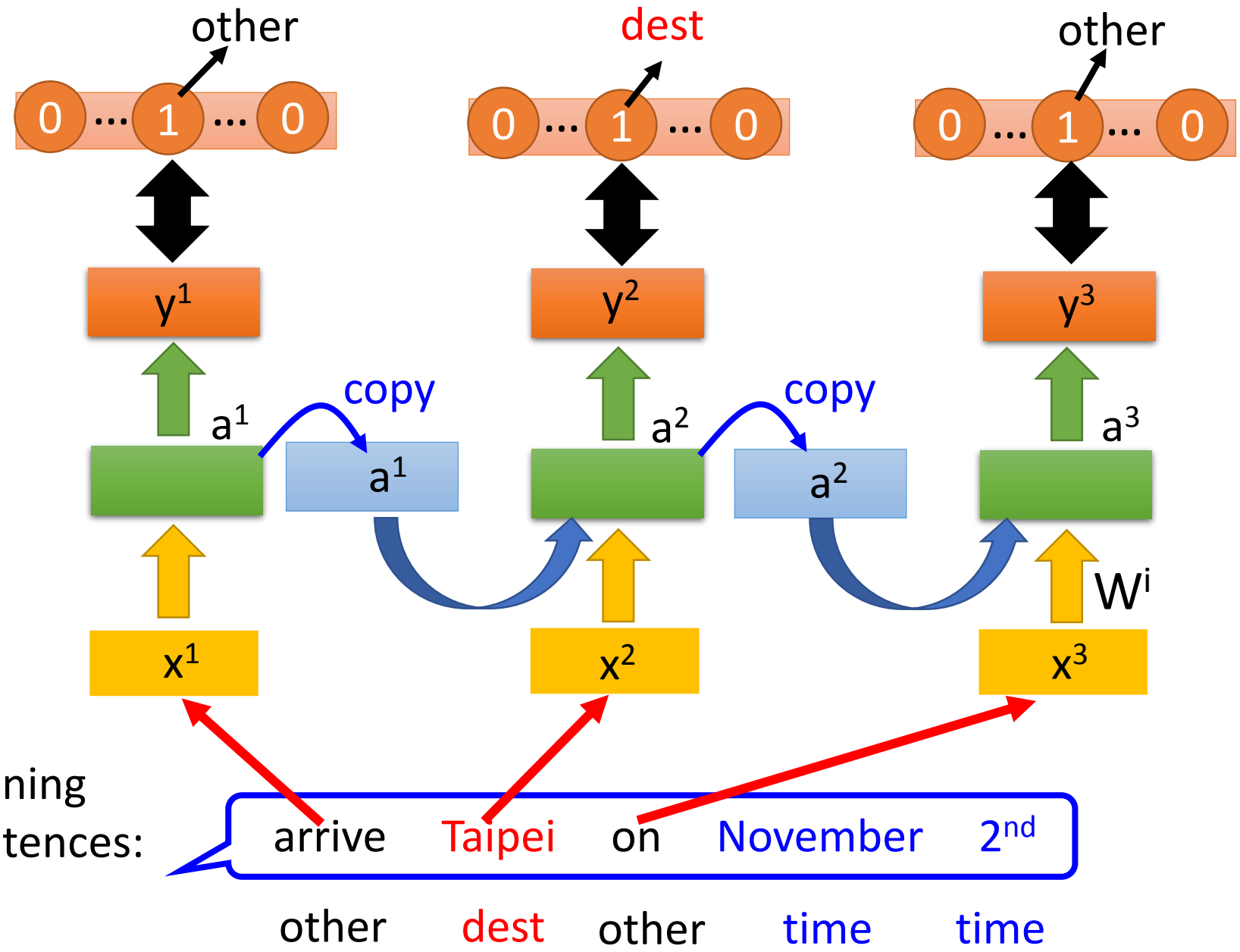
Step 2: goodness of function

Step 3: pick the best function

Deep Learning is so simple ……



Learning Target



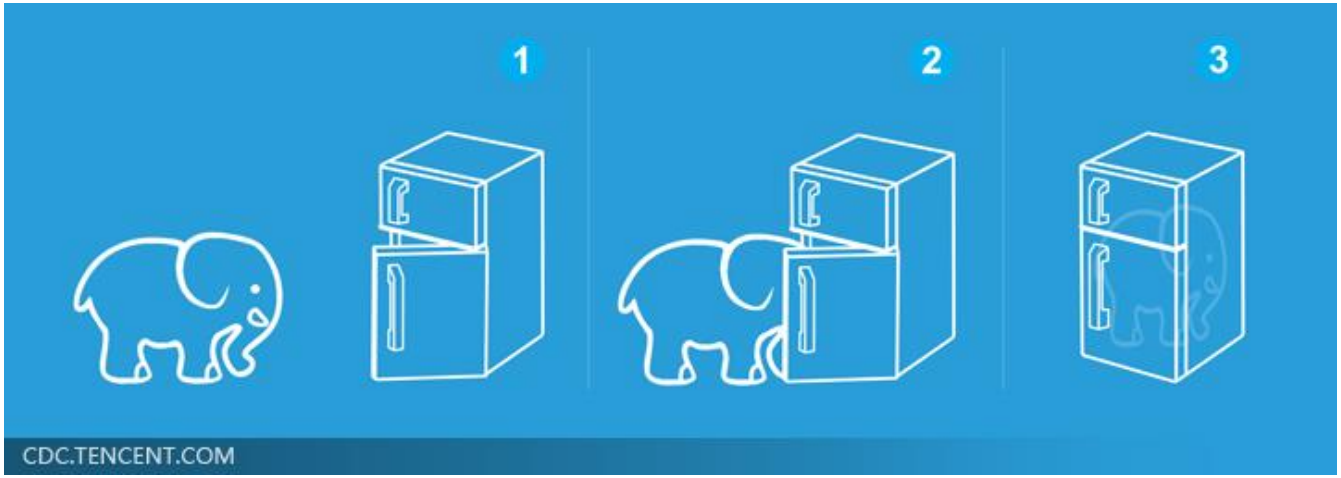
Three Steps for Deep Learning

Step 1: define a set of function

Step 2: goodness of function

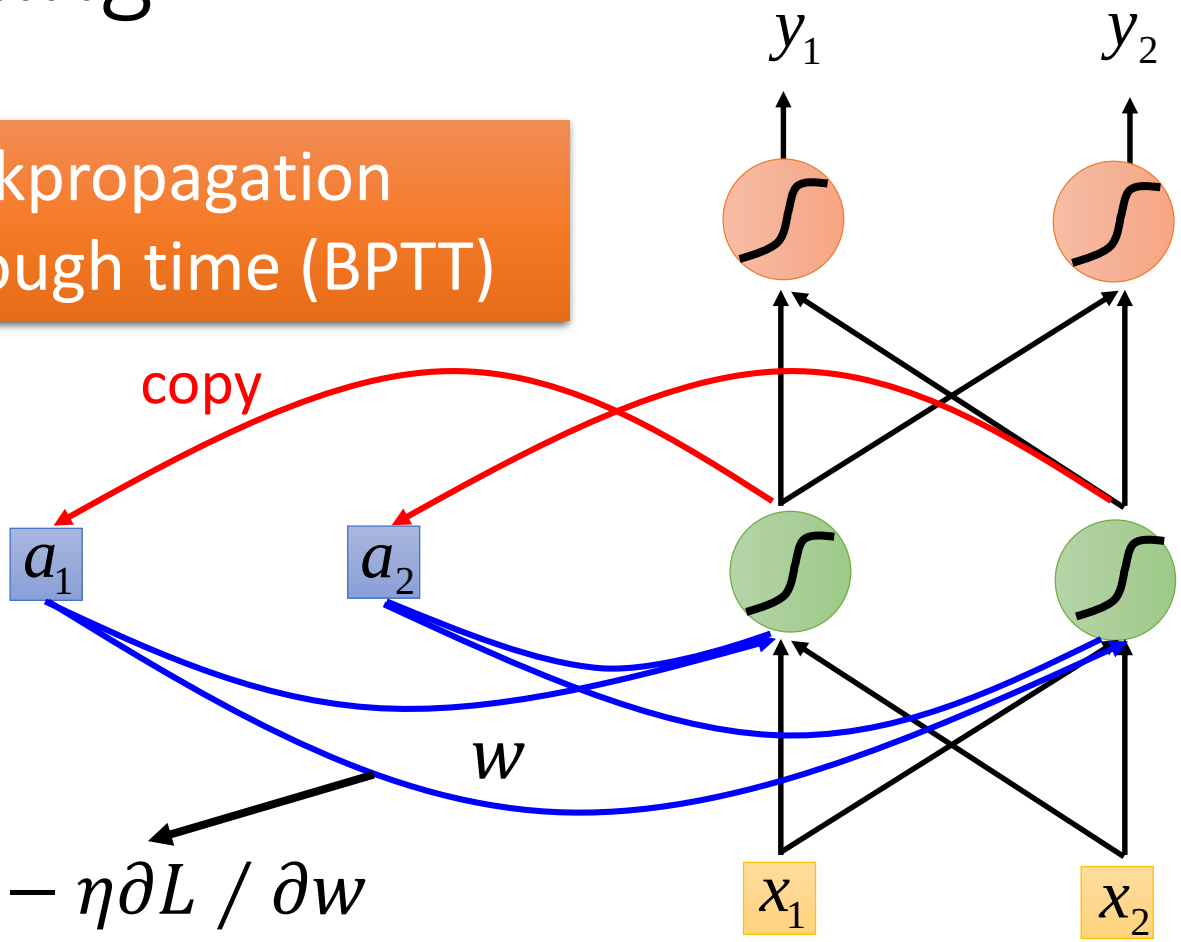
Step 3: pick the best function

Deep Learning is so simple ……



Learning

Backpropagation through time (BPTT)



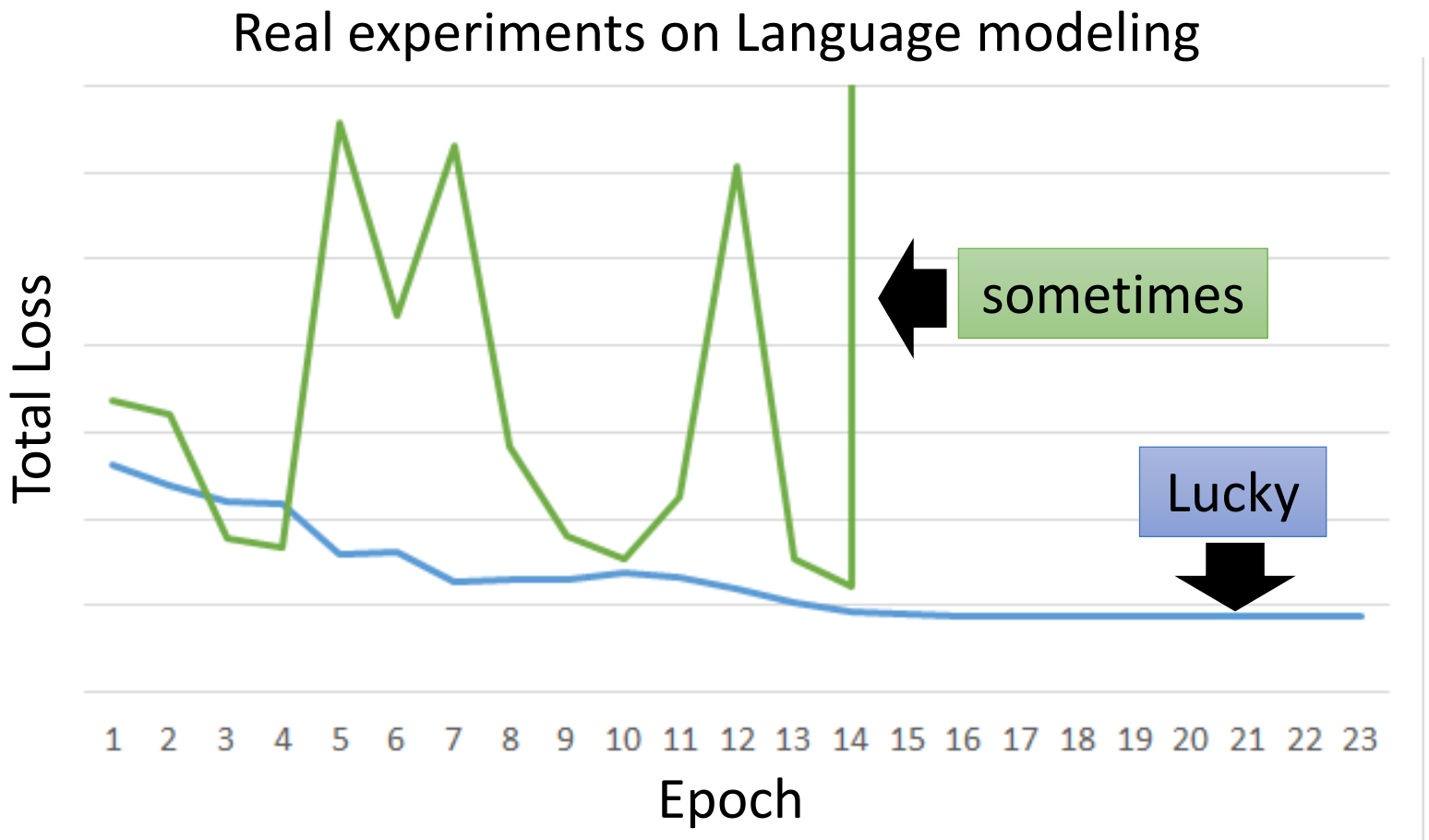
$$w \leftarrow w - \eta \partial L / \partial w$$

RNN Learning is very difficult in practice.

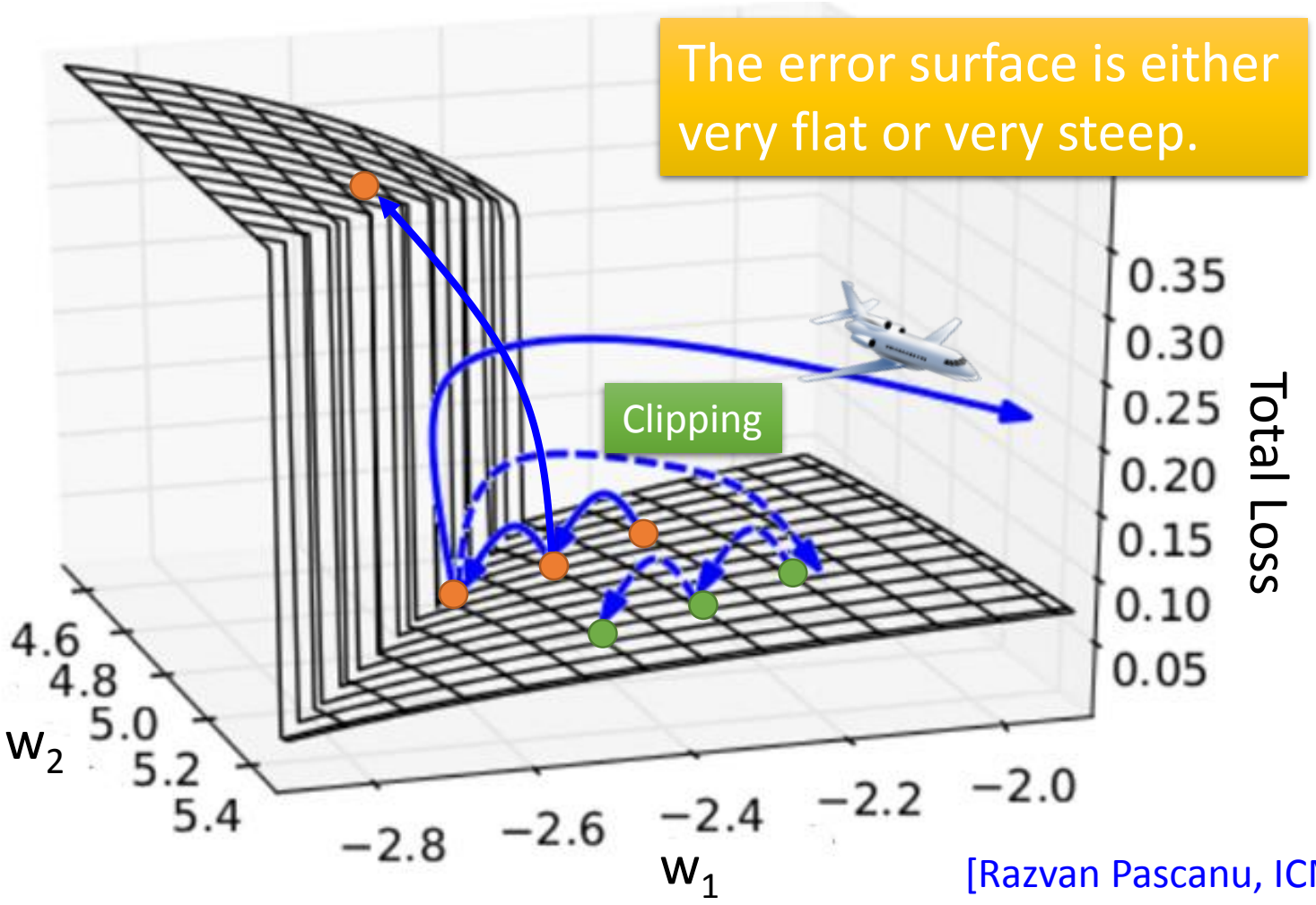
感謝 曾柏翔 同學
提供實驗結果

Unfortunately ……

- RNN-based network is not always easy to learn



The error surface is rough.



[Razvan Pascanu, ICML'13]

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?

Toy Example