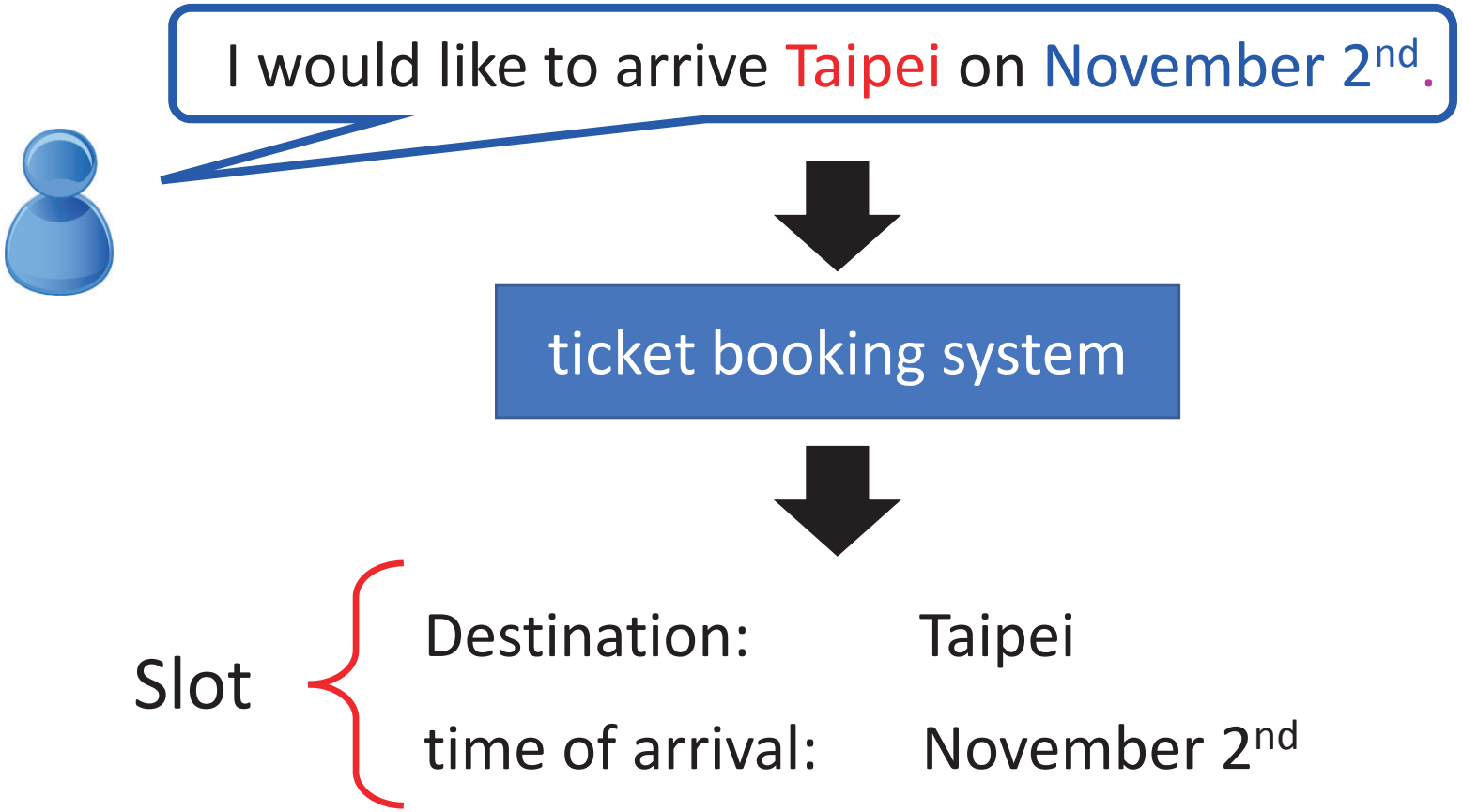
Recurrent Neural Network (RNN)

Example Application

- Slot Filling

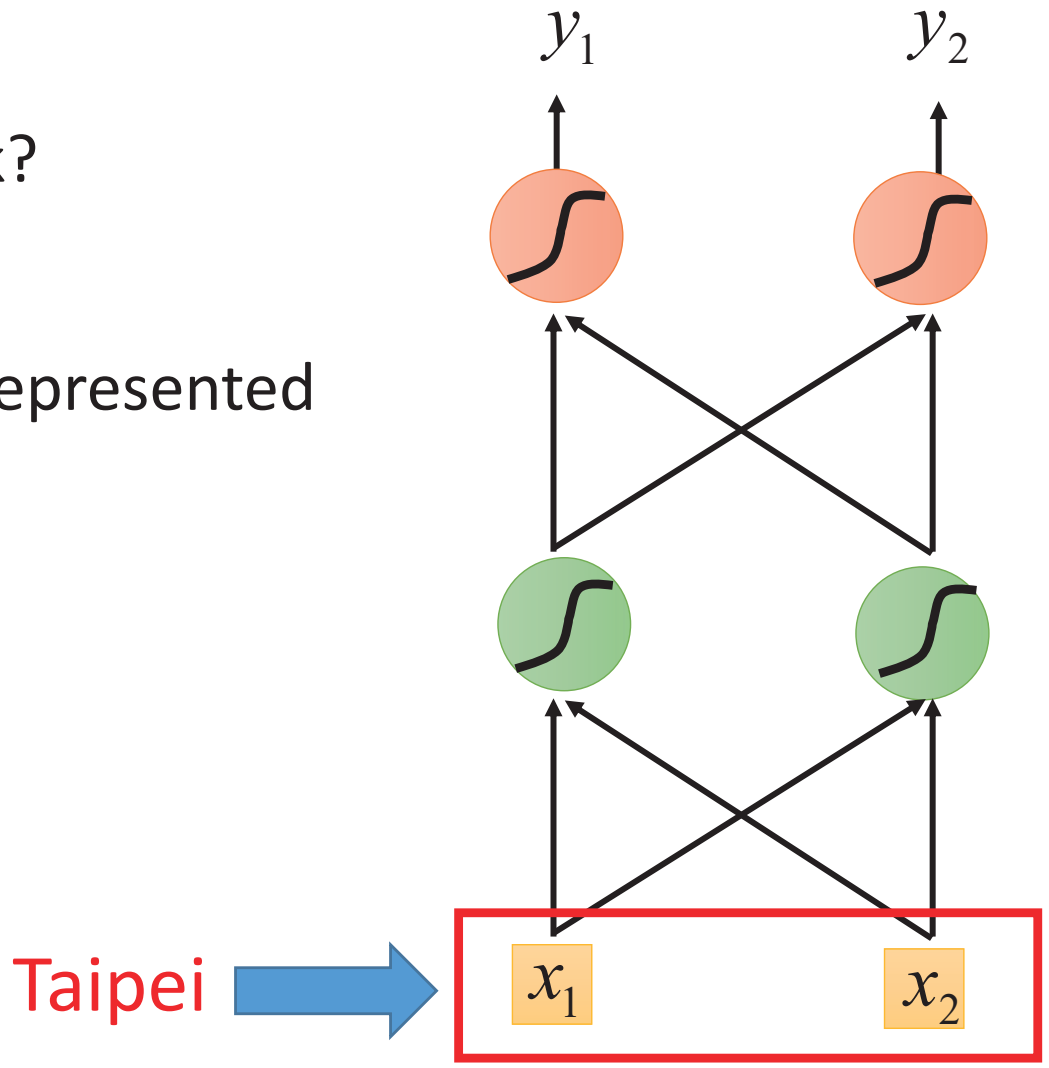


Example Application

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.

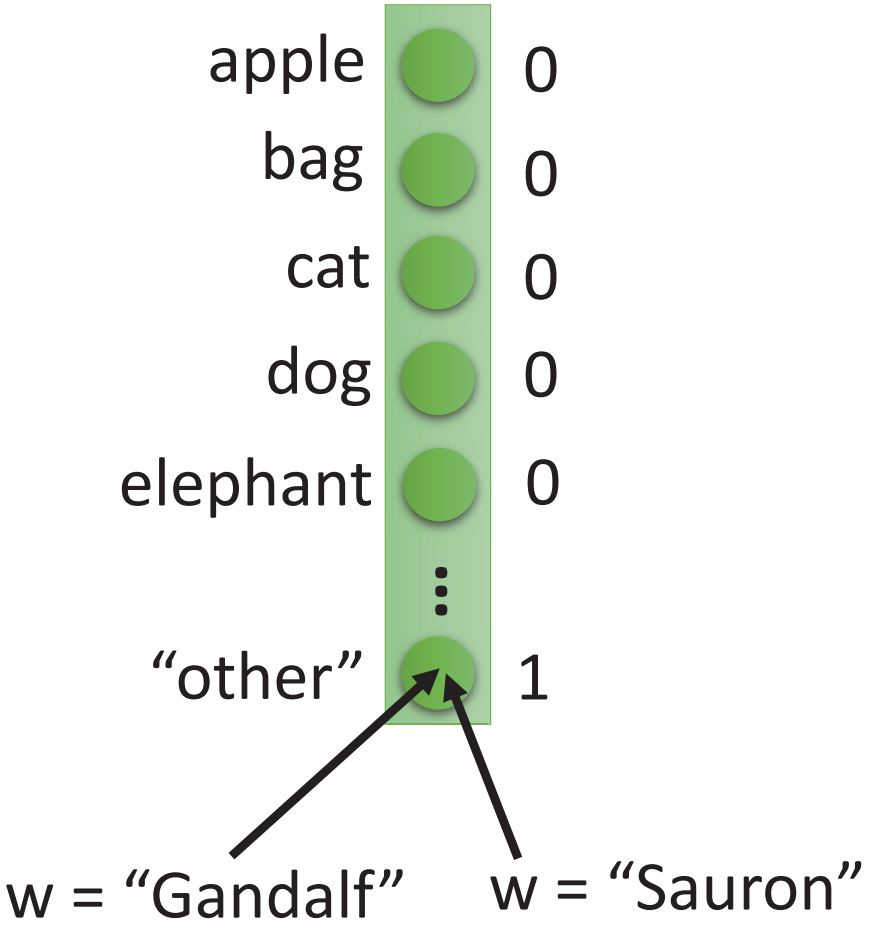
Each dimension corresponds to a word in the lexicon

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

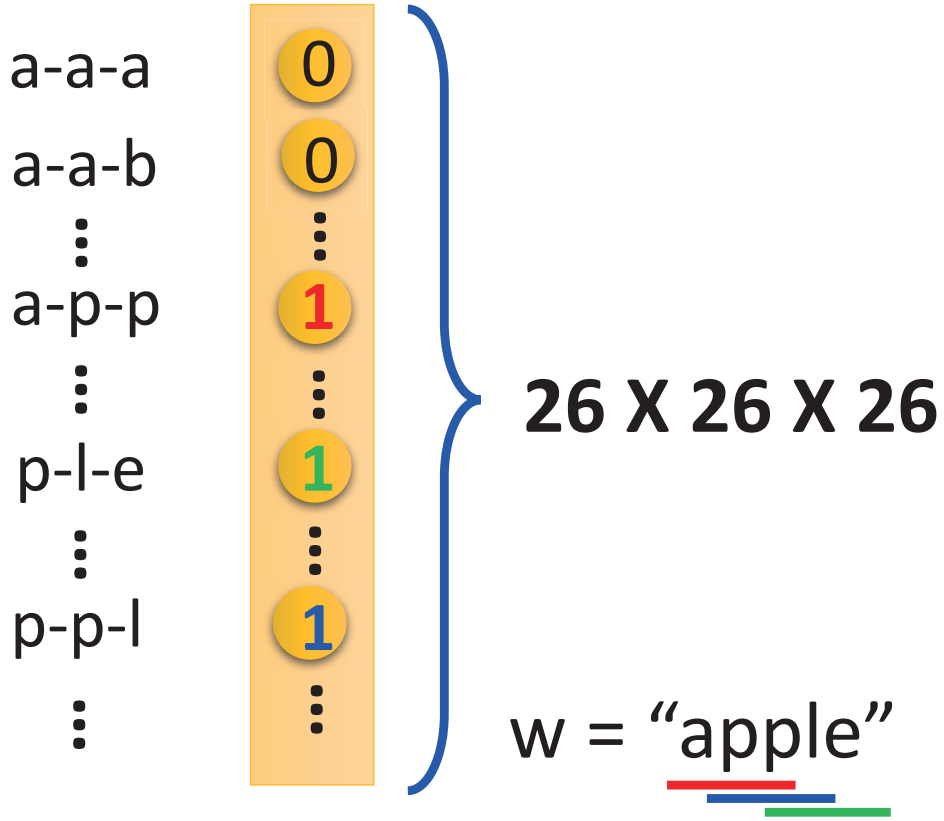
apple = [1 0 0 0 0]
bag = [0 1 0 0 0]
cat = [0 0 1 0 0]
dog = [0 0 0 1 0]
elephant = [0 0 0 0 1]

Beyond 1-of-N encoding

Dimension for "Other"



Word hashing



Example Application

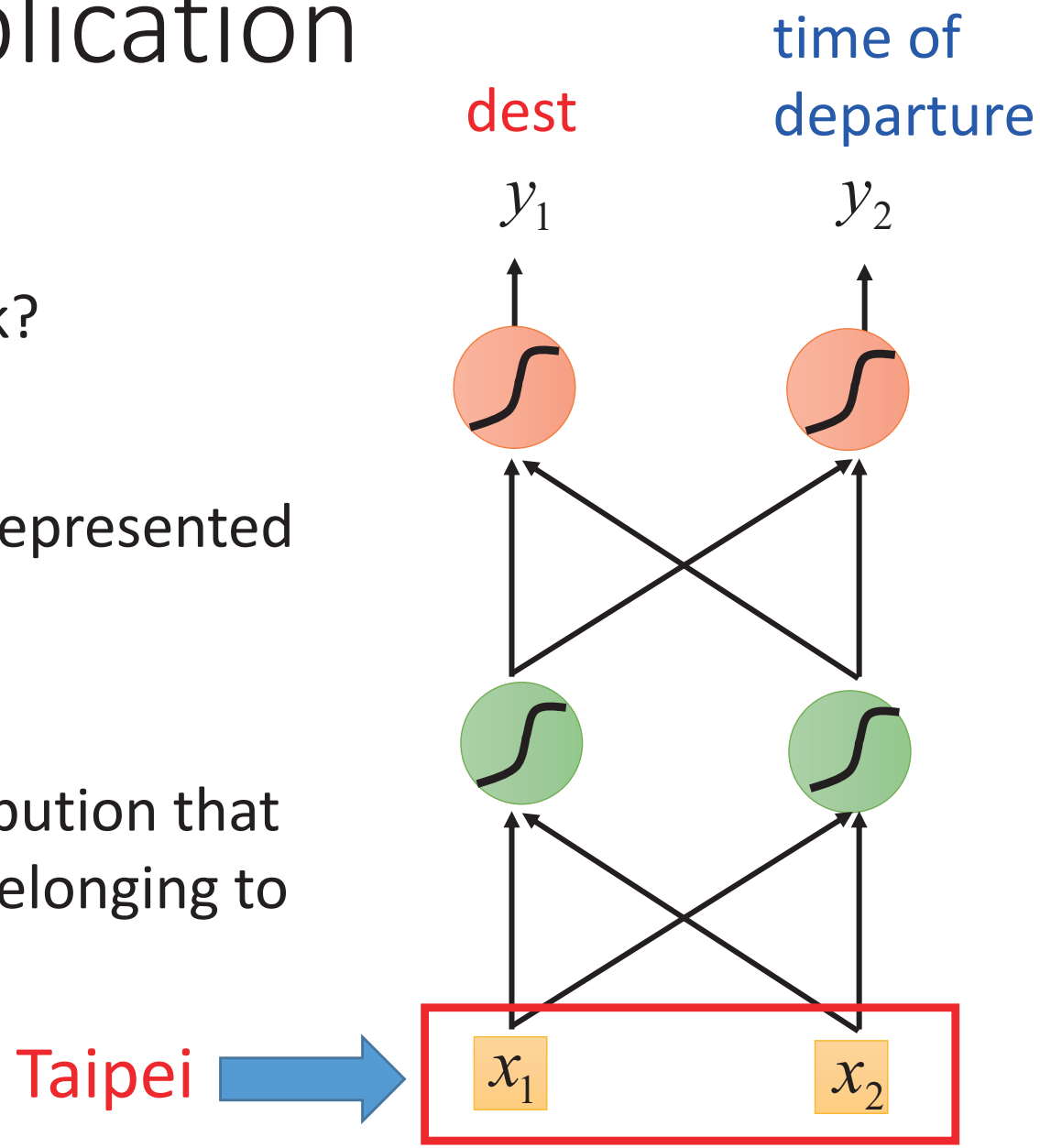
Solving slot filling by Feedforward network?

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

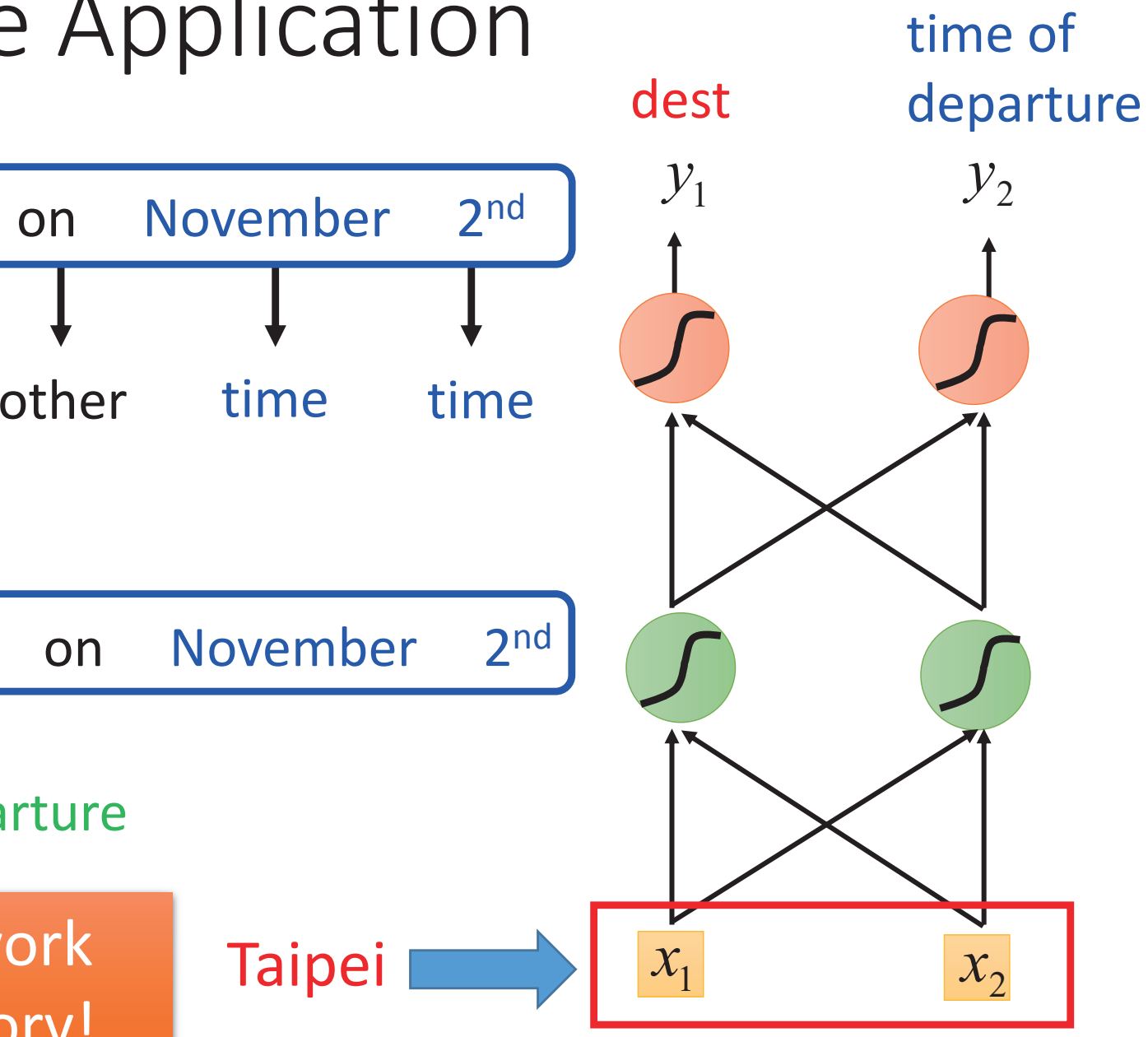
other dest other time time

Problem?

leave Taipei on November 2nd

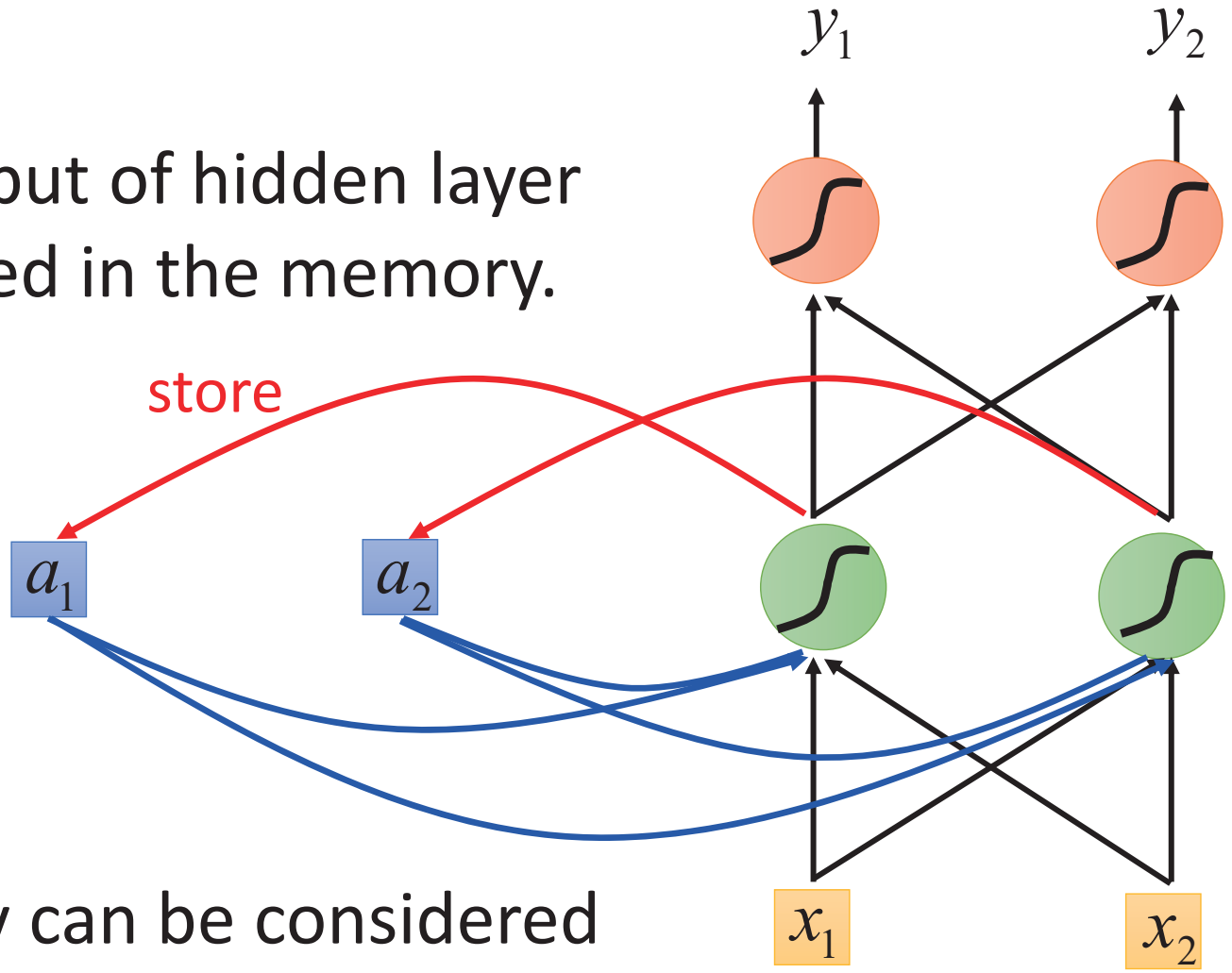
place of departure

Neural network needs memory!



Recurrent Neural Network (RNN)

The output of hidden layer are stored in the memory.

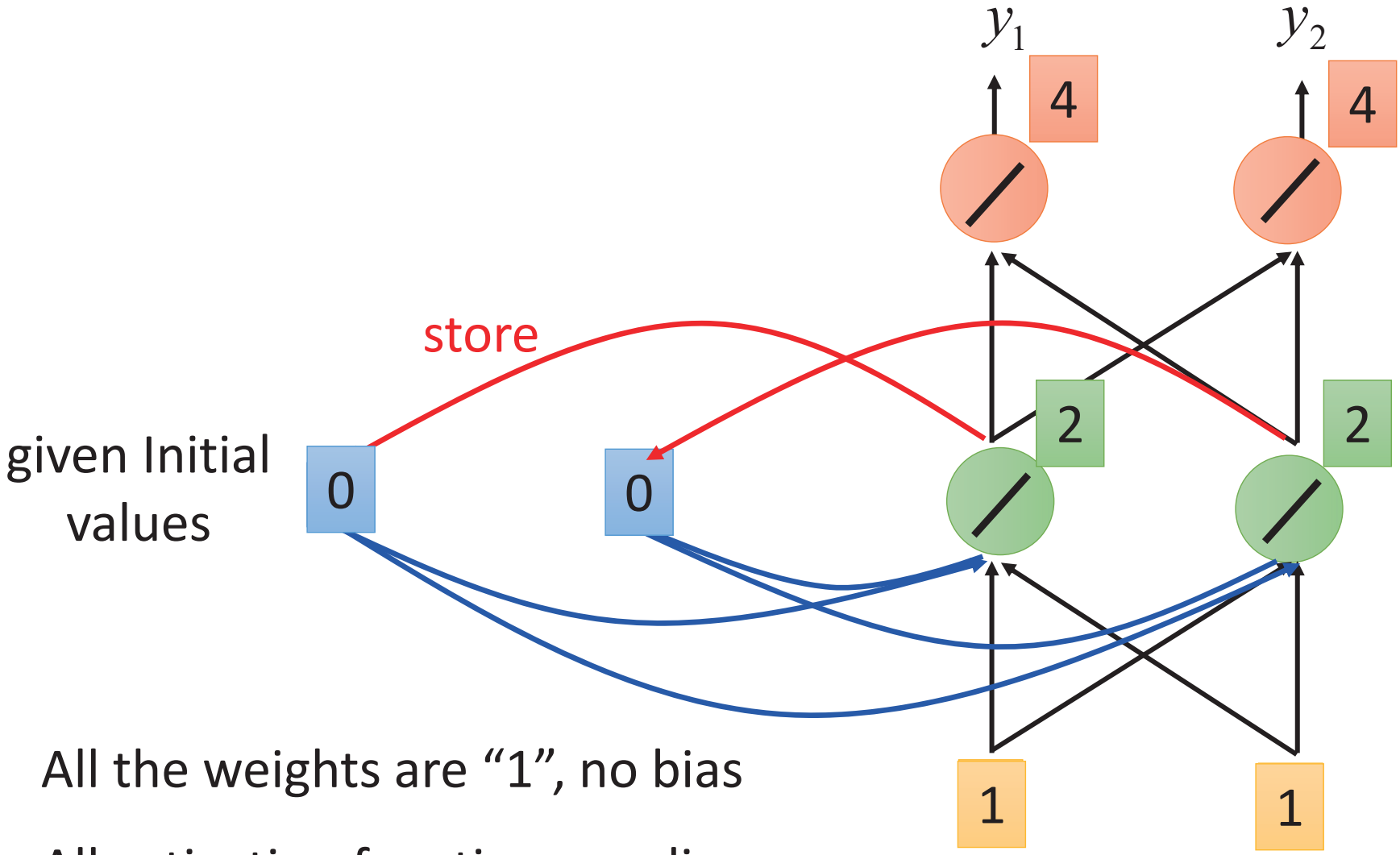


Memory can be considered as another input.

Example

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

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



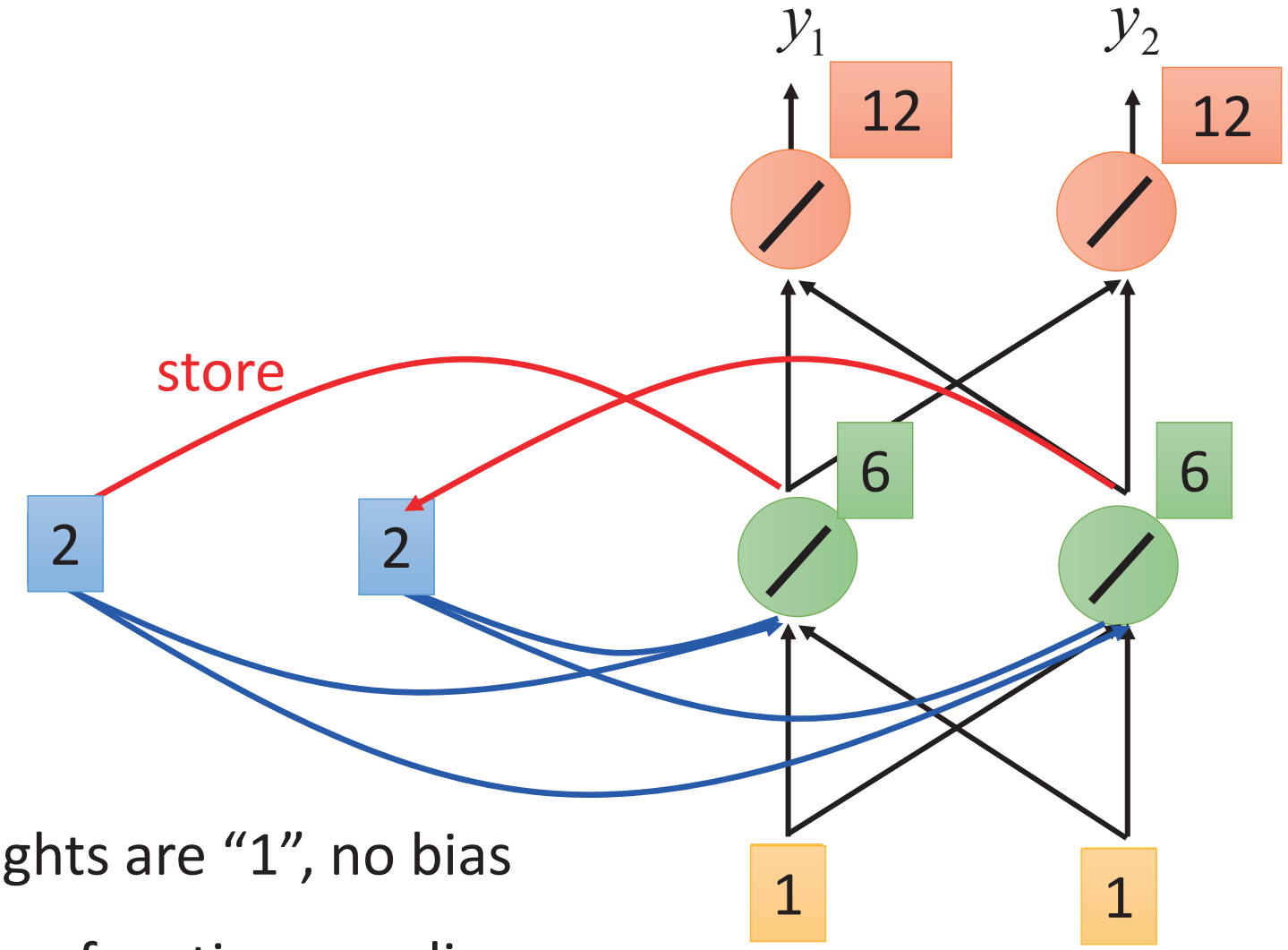
All the weights are “1”, no bias

All activation functions are linear

Example

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

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



All the weights are “1”, no bias

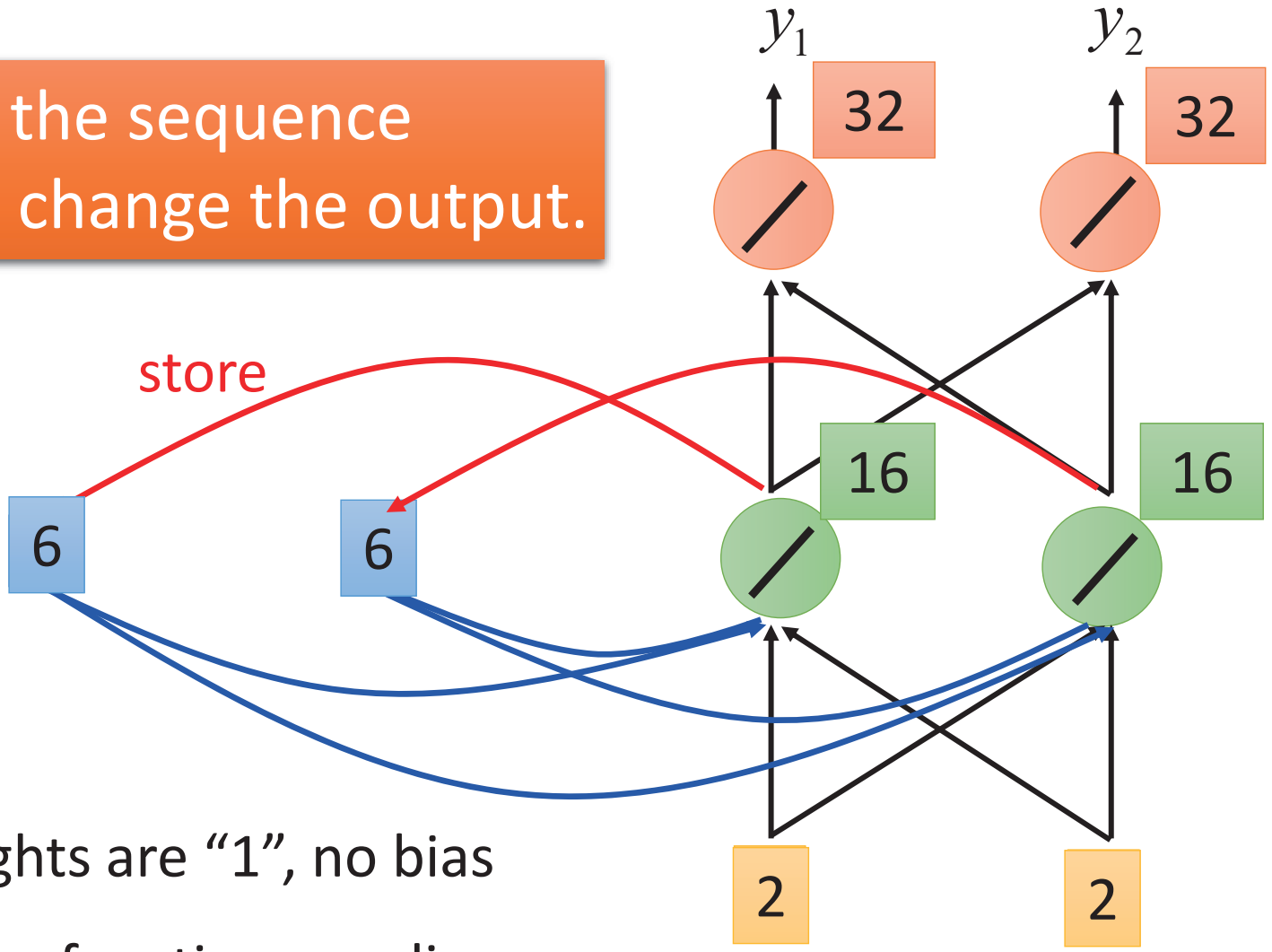
All activation functions are linear

Example

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

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

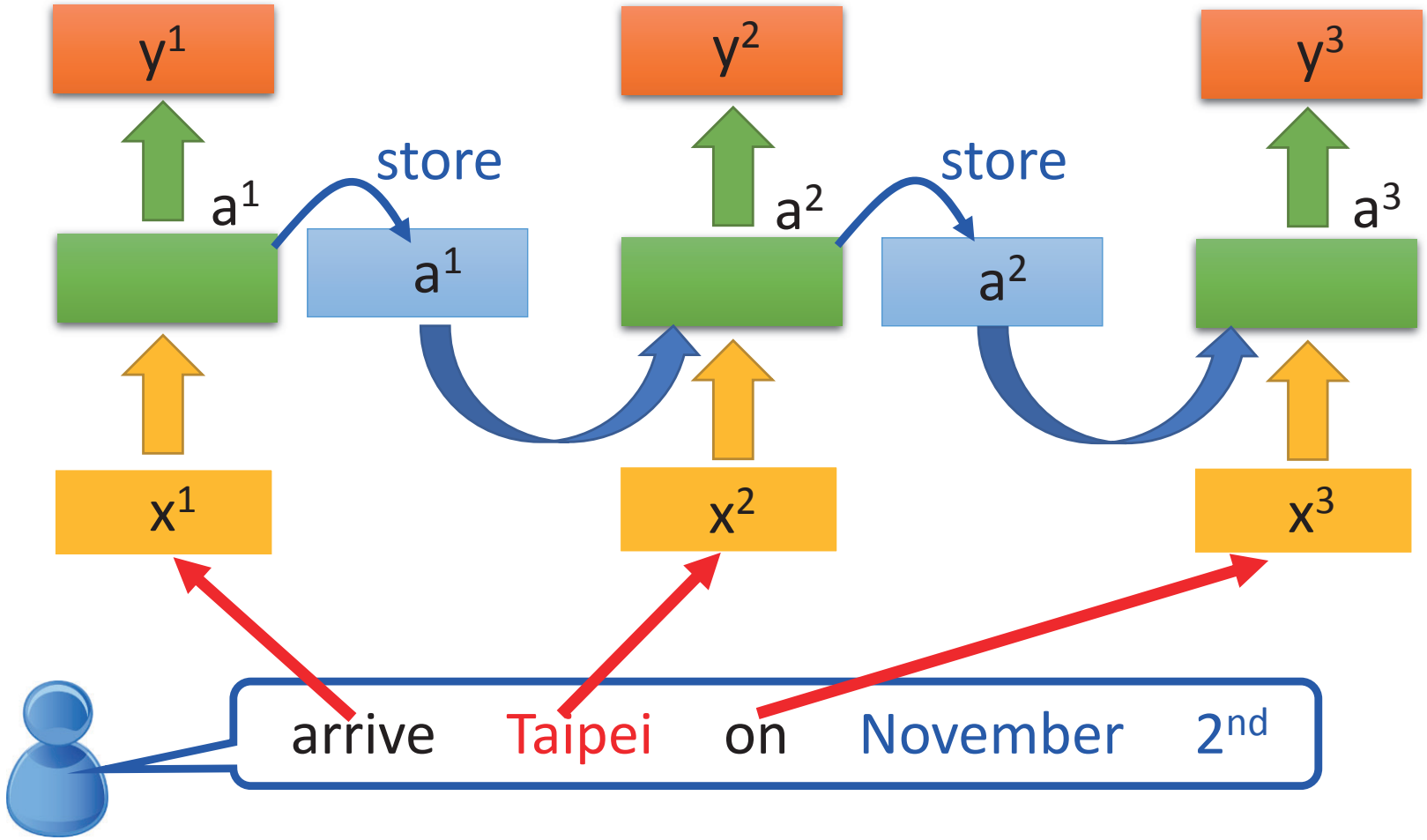
RNN

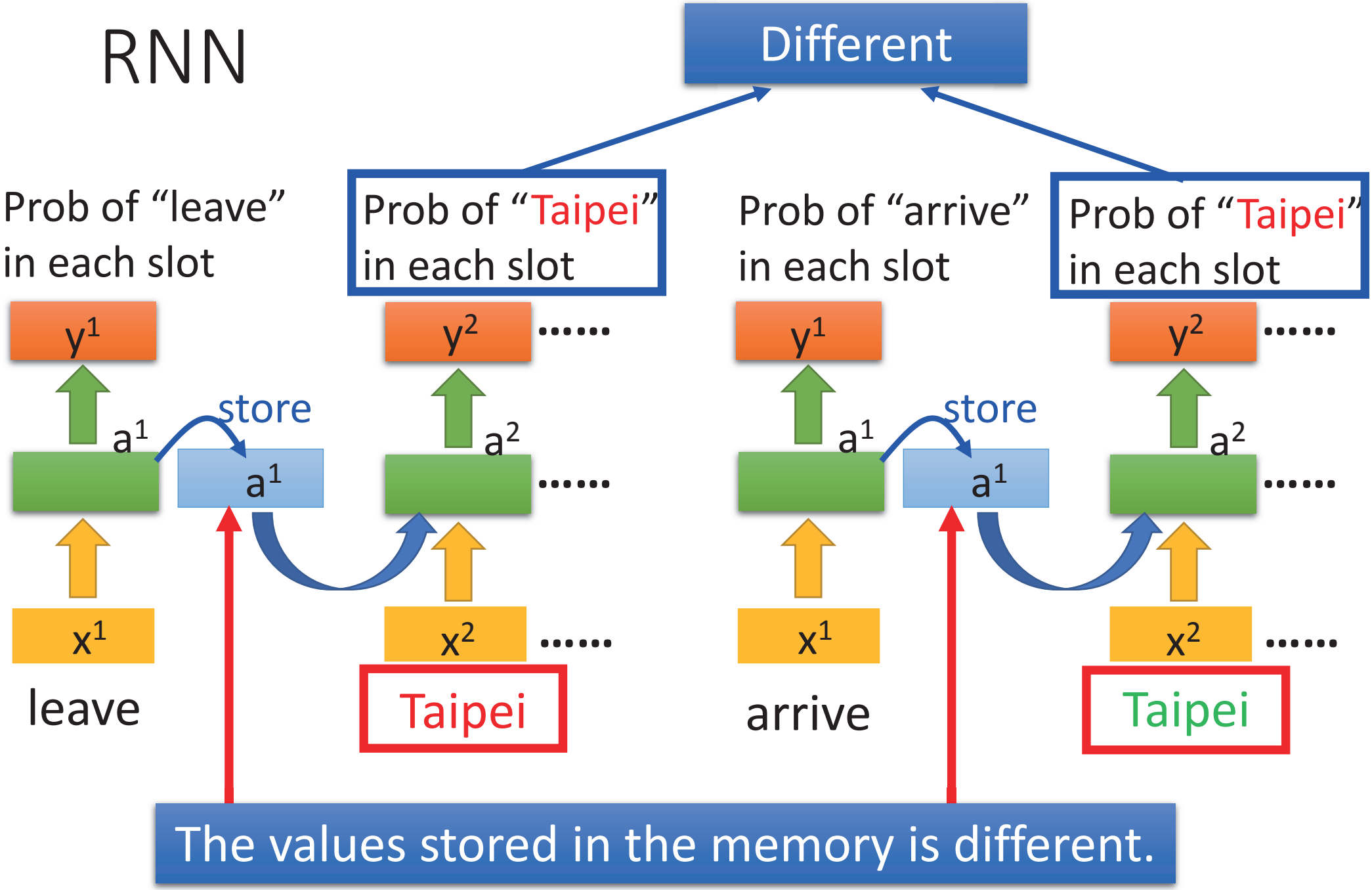
The same network is used again and again.

Probability of "arrive" in each slot

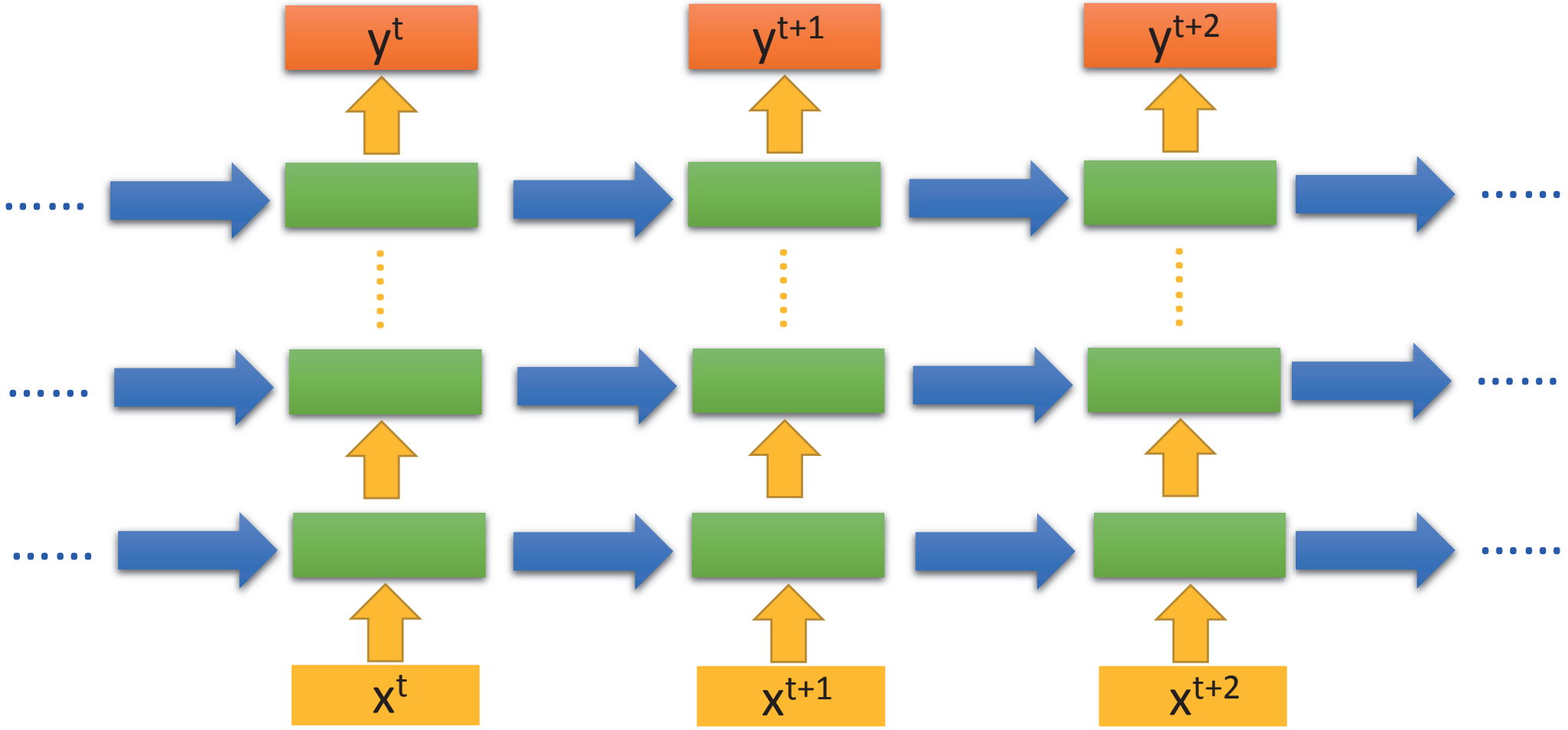
Probability of "Taipei" in each slot

Probability of "on" in each slot

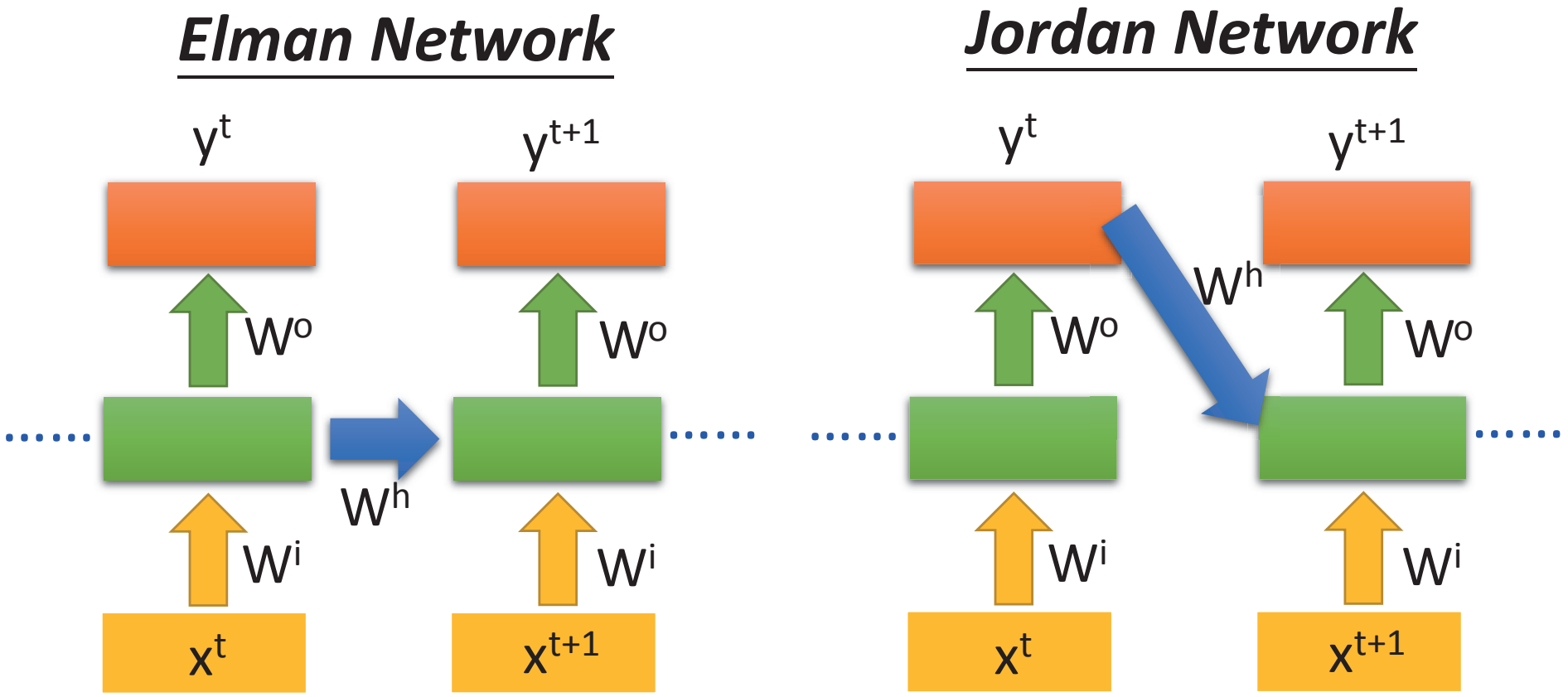



RNN
Different
Prob of "leave" in each slot
Prob of "Taipei" in each slot
Prob of "arrive" in each slot
Prob of "Taipei" in each slot
y1
y2
.....
a1
store
a1
a2
x1
x2
leave
Taipei
arrive
Taipei
The values stored in the memory is different.

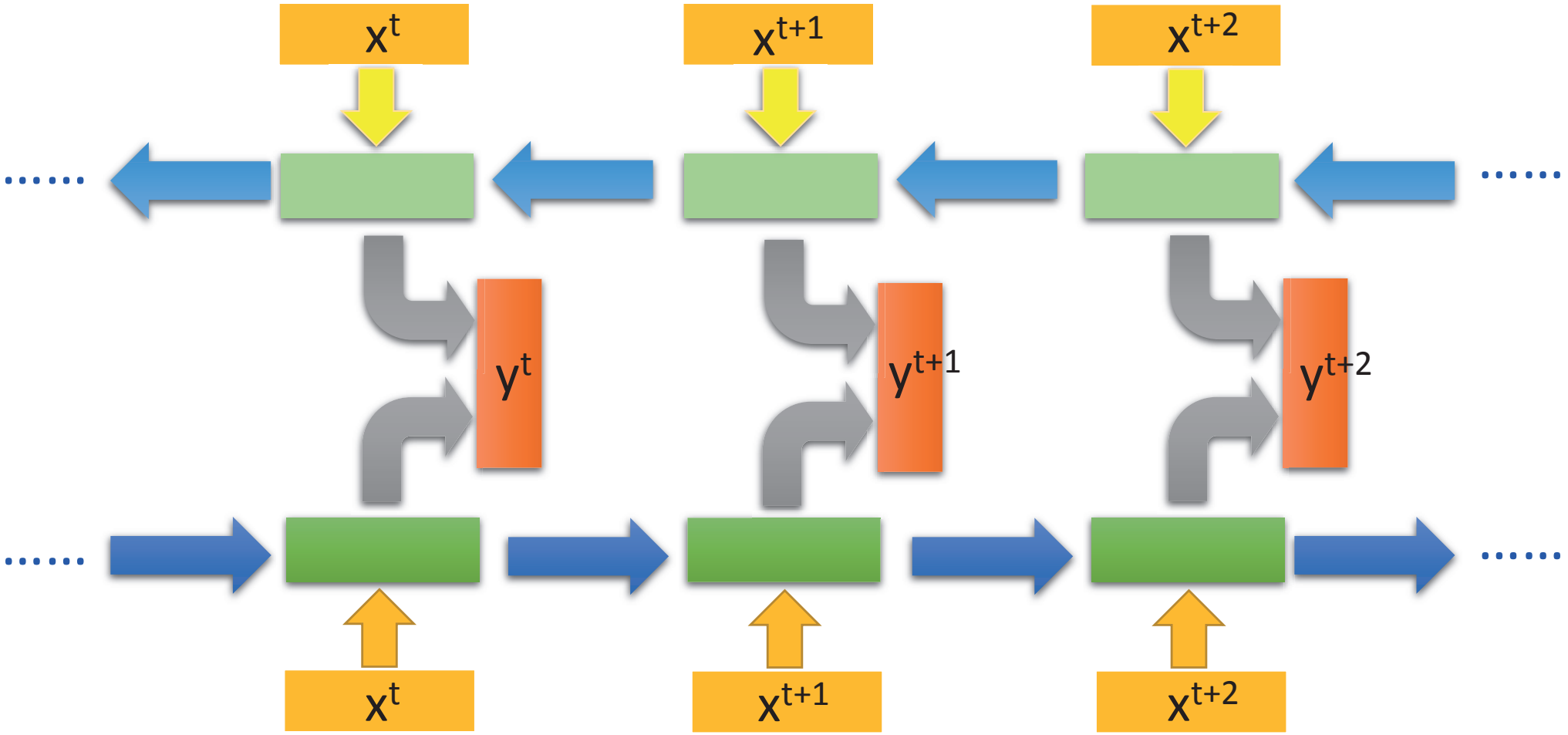
Of course it can be deep …



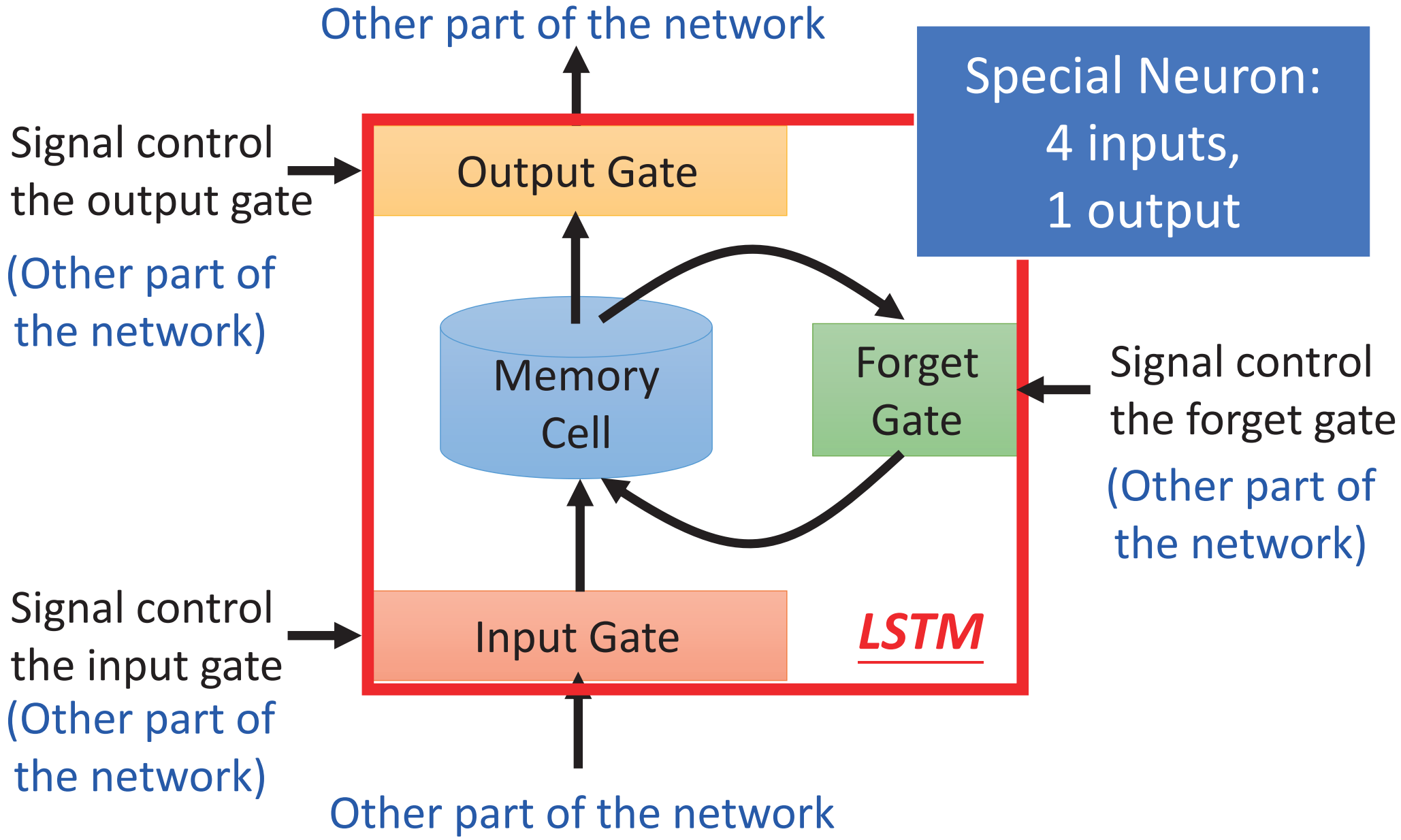
Elman Network & Jordan Network

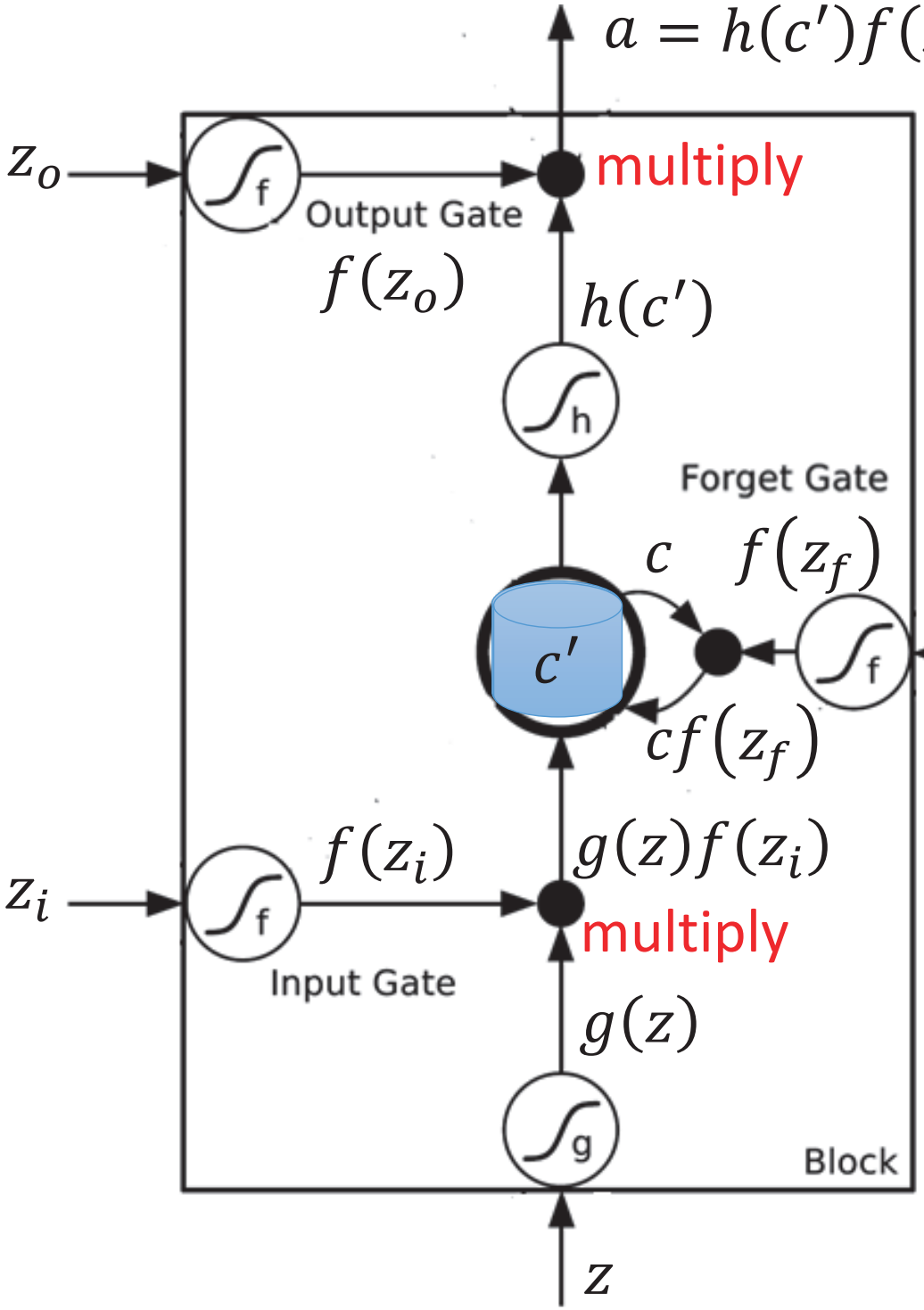


Bidirectional RNN



Long Short-term Memory (LSTM)





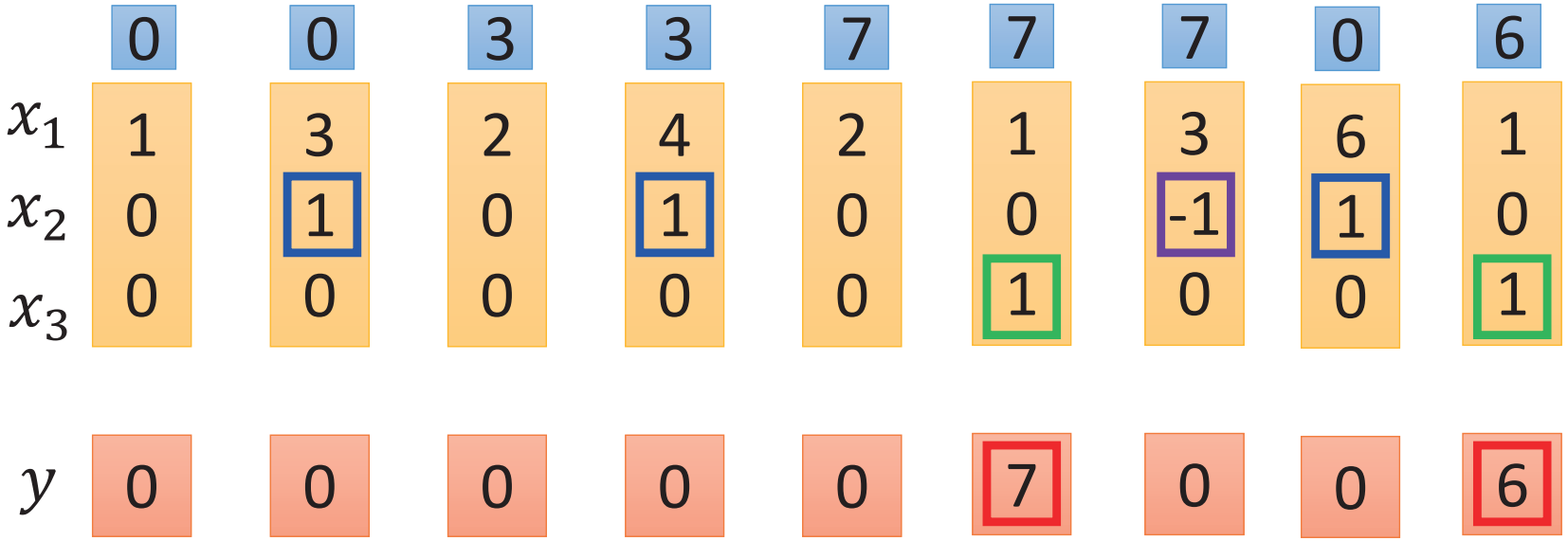
Activation function f is usually a sigmoid function

Between 0 and 1

Mimic open and close gate

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

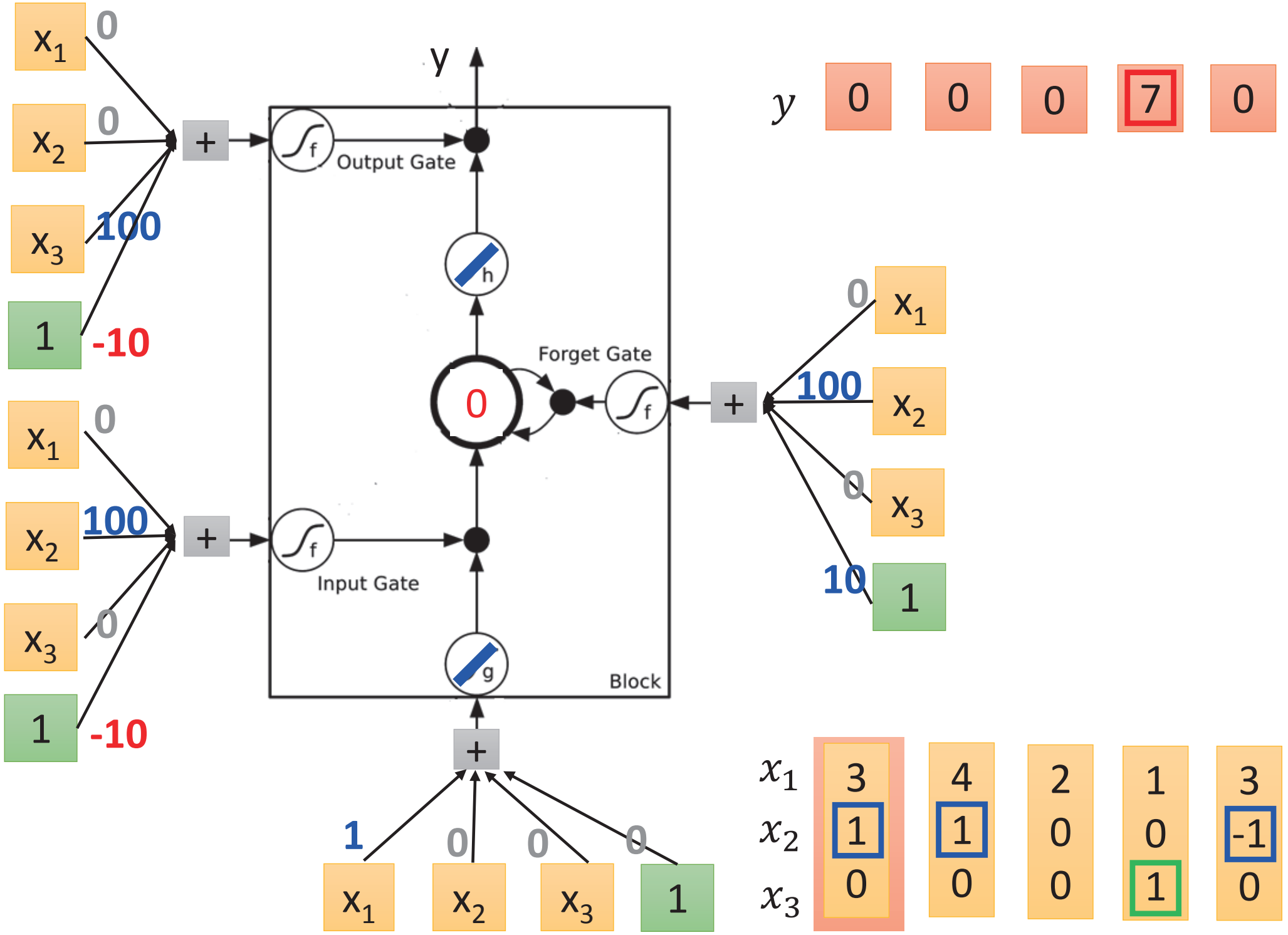
LSTM - Example

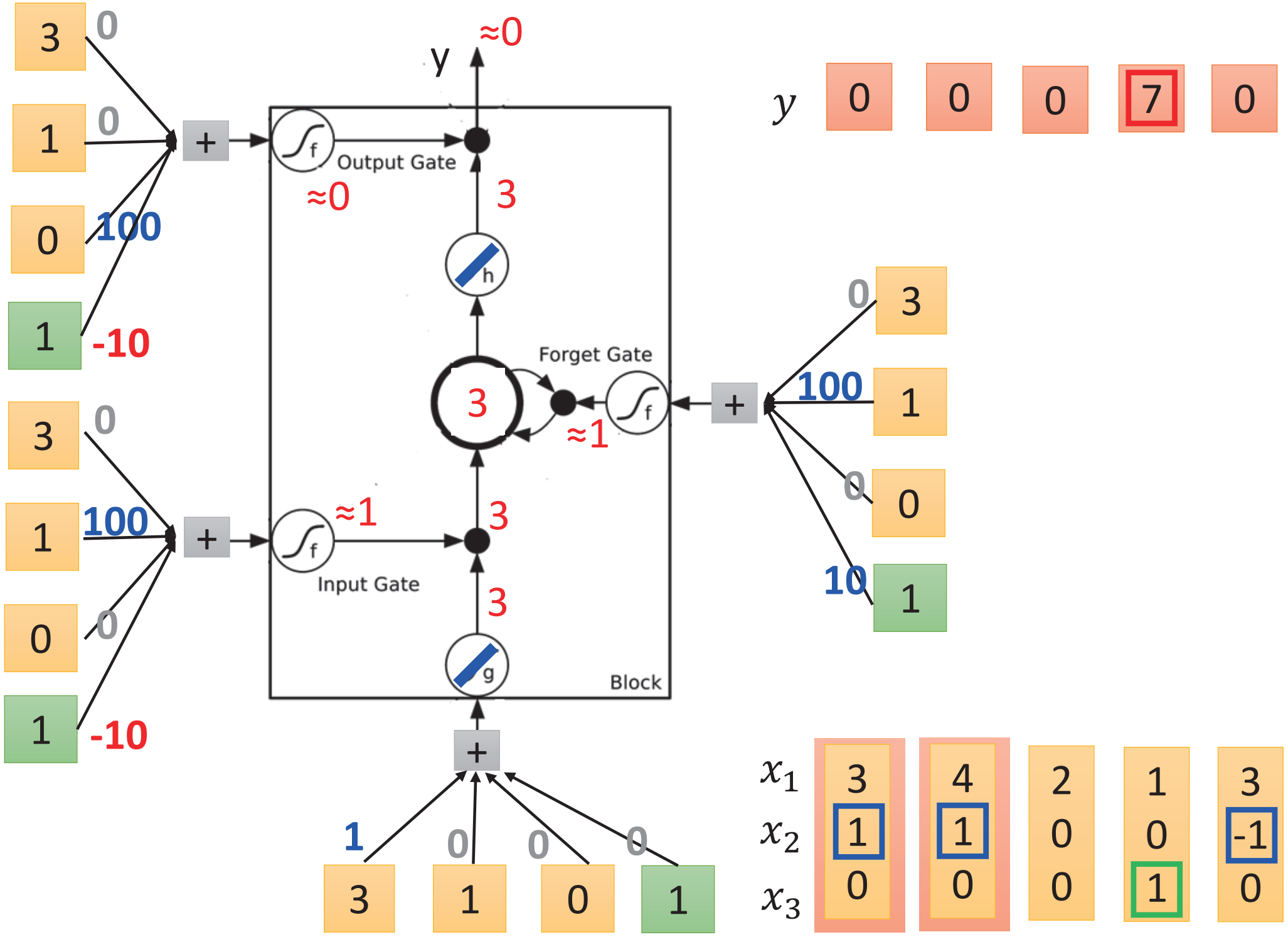


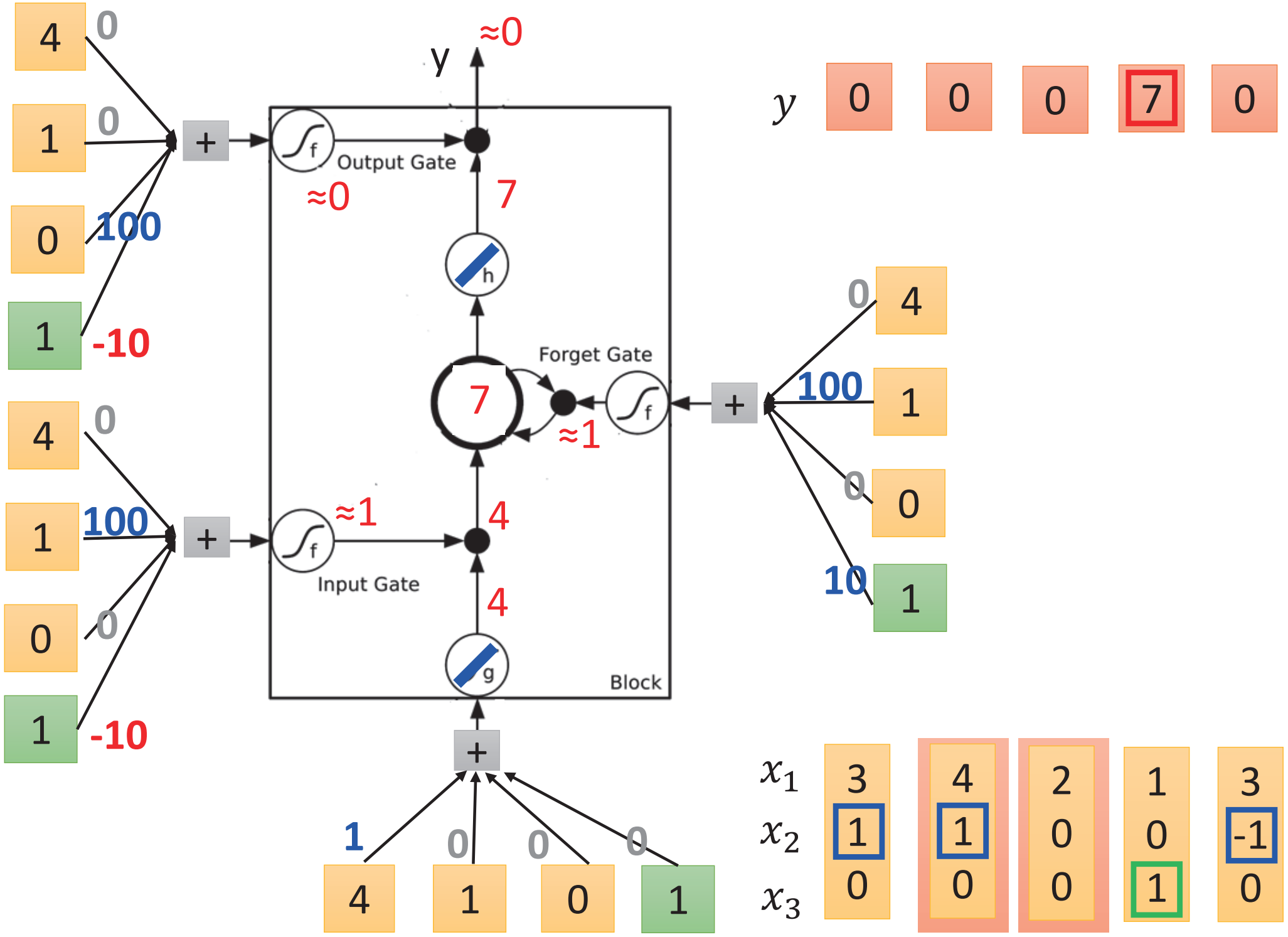
When x_2 = 1, add the numbers of x_1 into the memory

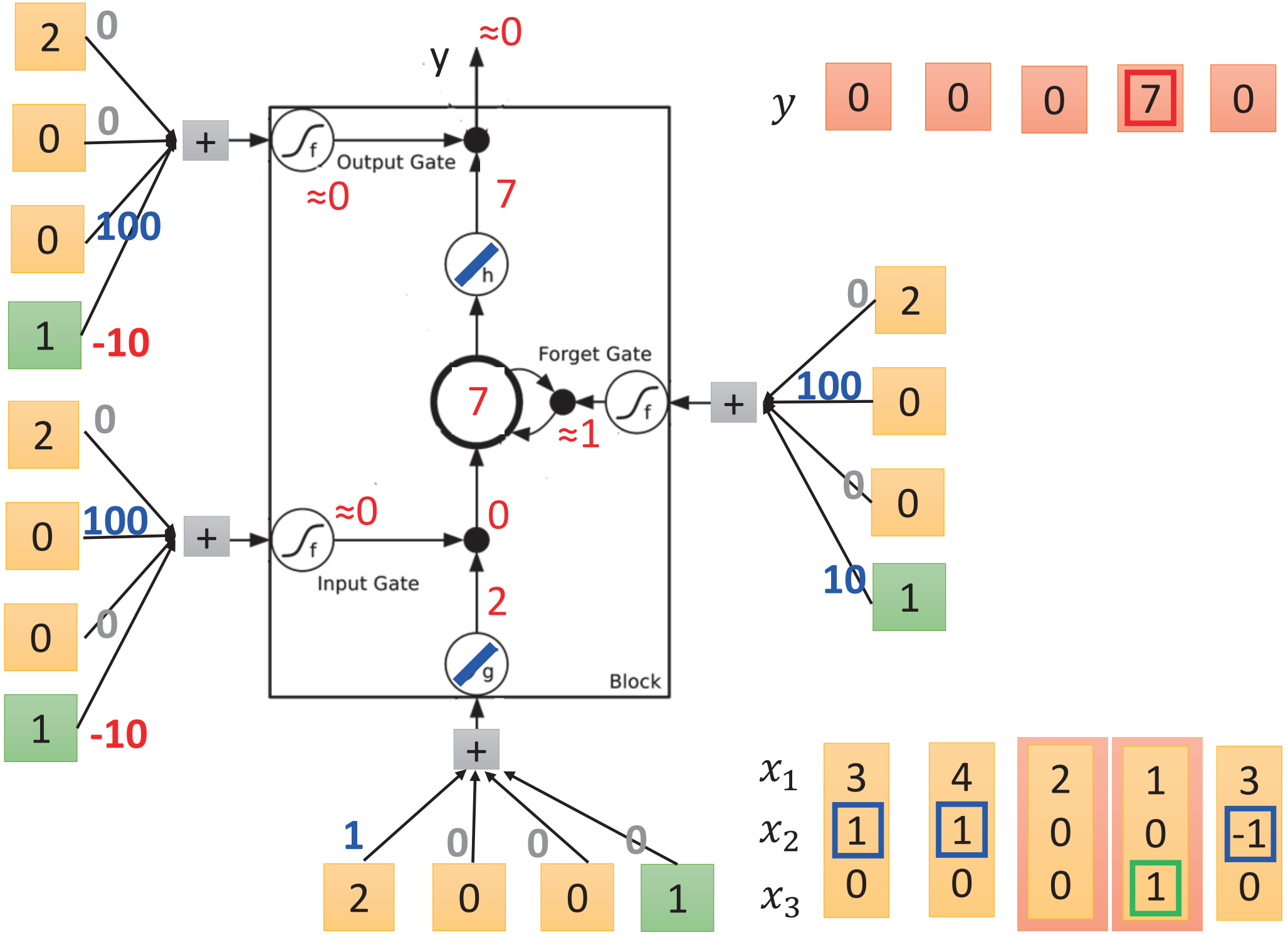
When x_2 = -1, reset the memory

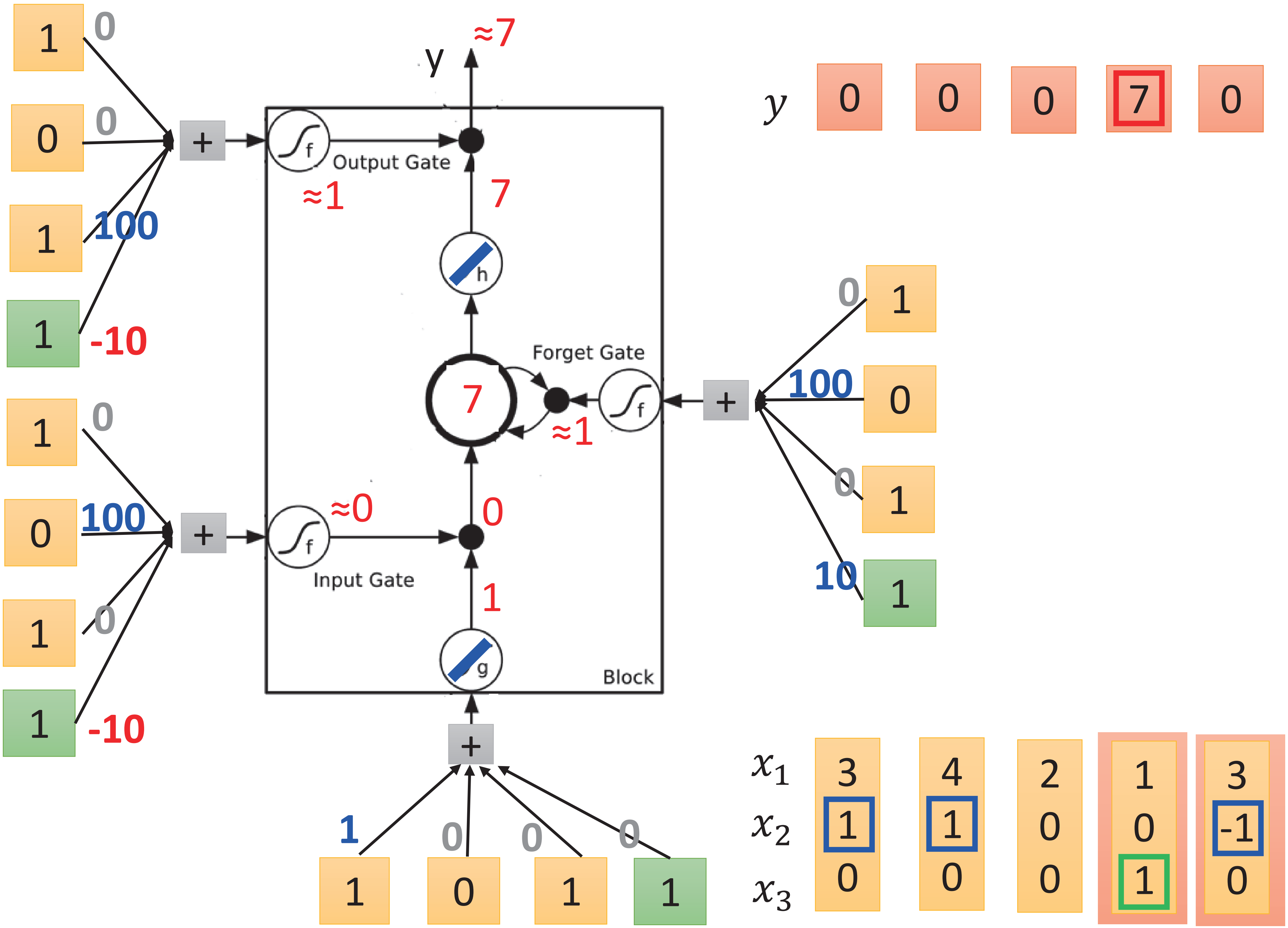
When x_3 = 1, output the number in the memory.


y
Output Gate
Forget Gate
Input Gate
Block
f
h
g
+
0
100
-10
1
x_1
x_2
x_3
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

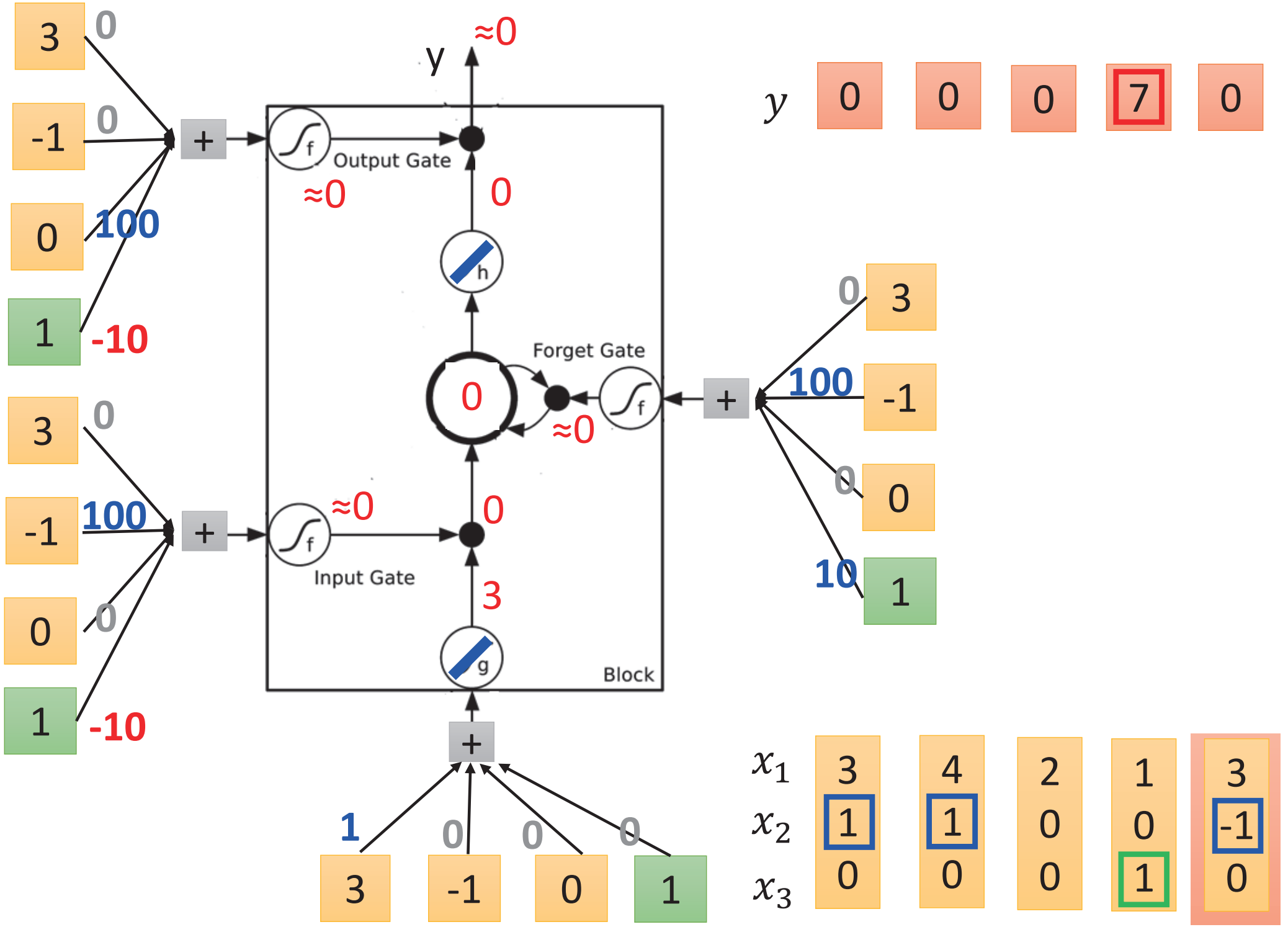

y
≈0
Output Gate
≈0
3
Forget Gate
3
≈1
Input Gate
≈1
3
3
Block
3
0
1
0
100
1
-10
3
0
1
100
0
0
1
-10
0
3
100
1
0
0
10
1
1
0
0
0
3
1
0
1
y 0 0 0 7 0
x_1 3 4 2 1 3
x_2 1 1 0 0 -1
x_3 0 0 0 1 0


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

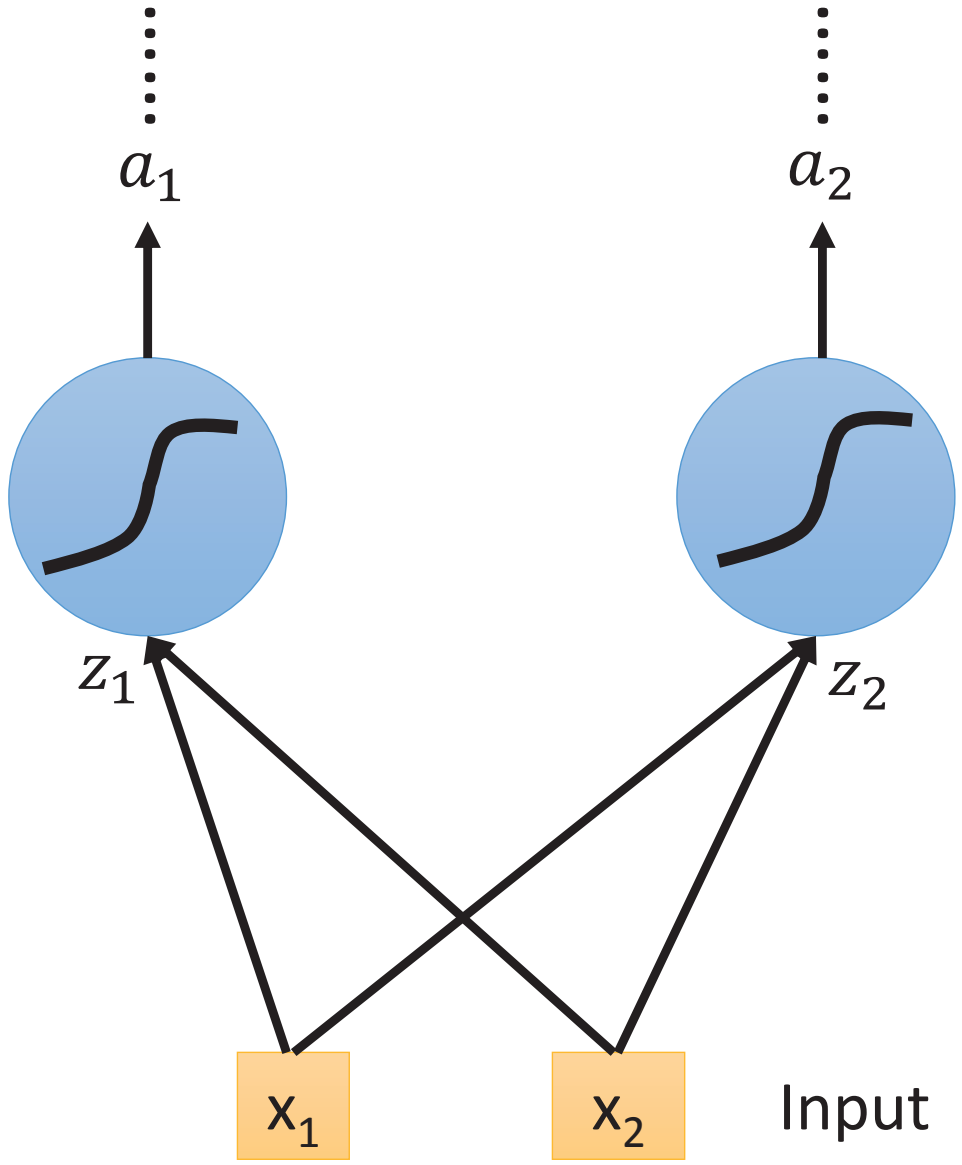


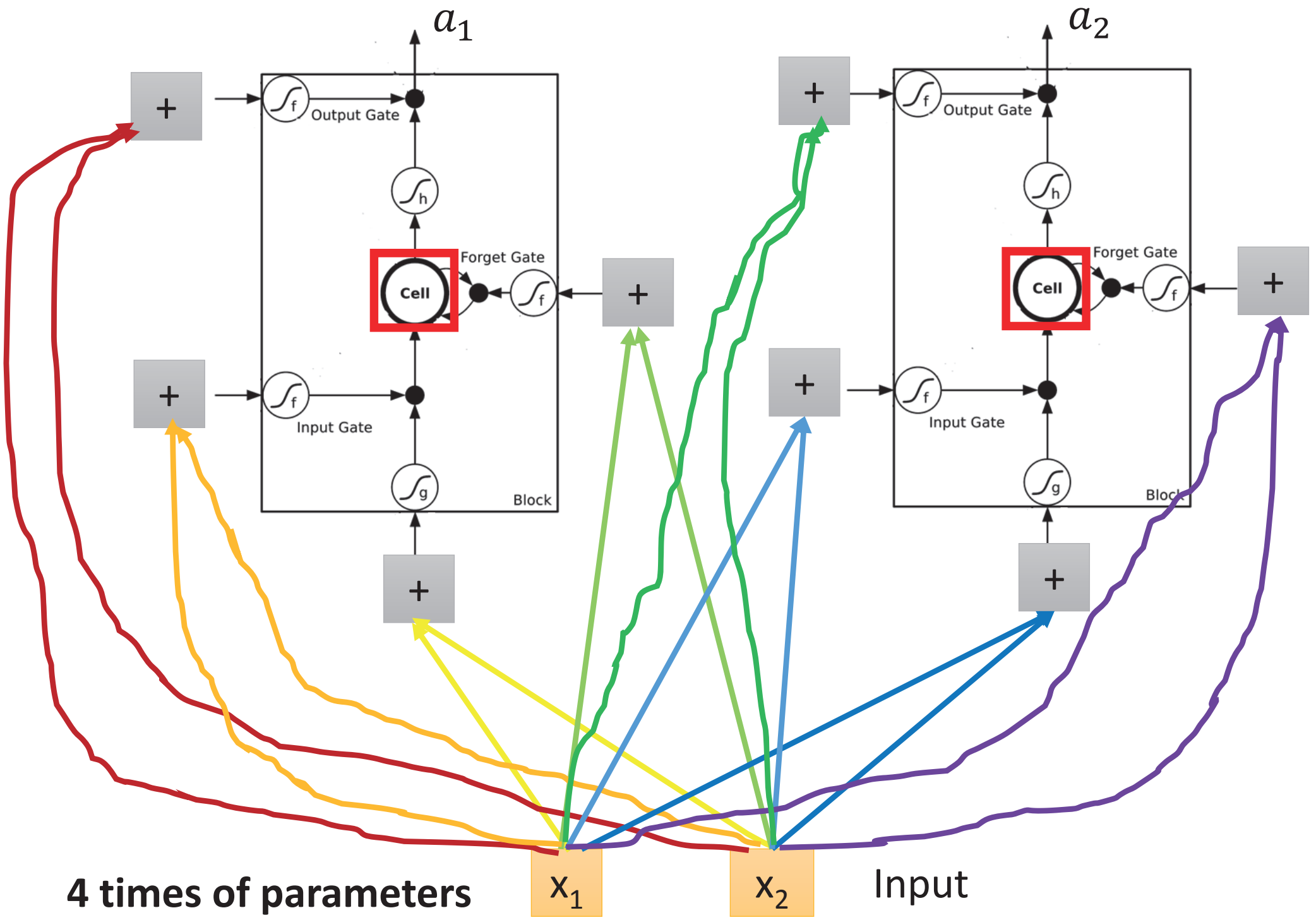
y
≈7
Output Gate
≈1
7
h
Forget Gate
7
≈1
Input Gate
≈0
0
1
g
Block
100
-10
10
y
0 0 0 7 0
x1
x2
x3
3 1 0
4 1 0
2 0 0
1 0 1
3 -1 0


y
≈0
Output Gate
≈0
0
Forget Gate
0
≈0
Input Gate
≈0
0
3
Block
3
0
-1
0
100
1
-10
3
0
-1
100
0
0
1
-10
0
3
100
-1
0
0
10
1
+
+
+
+
f
f
f
h
g
1
0
0
0
3
-1
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

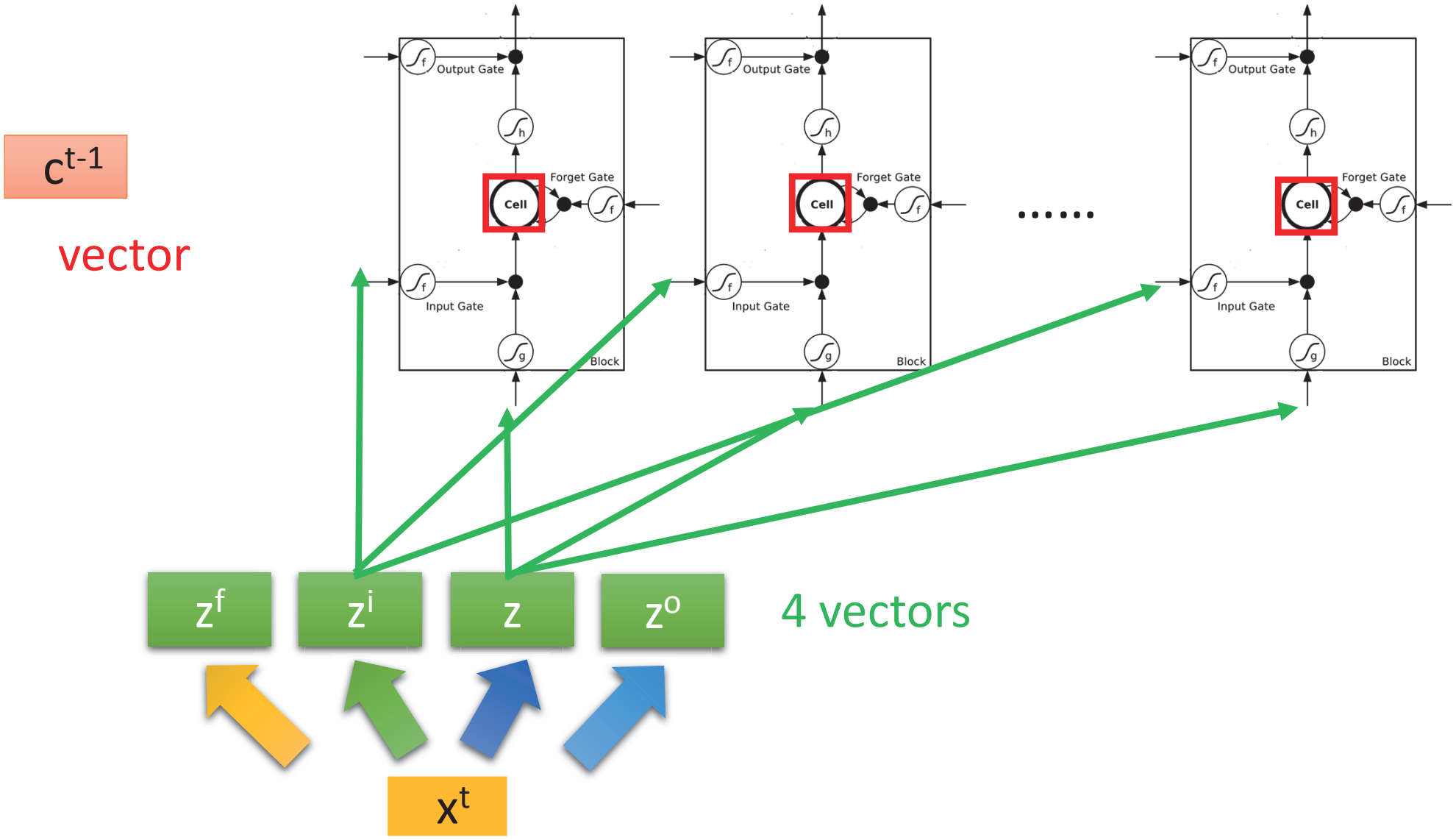
Original Network:

- Simply replace the neurons with LSTM

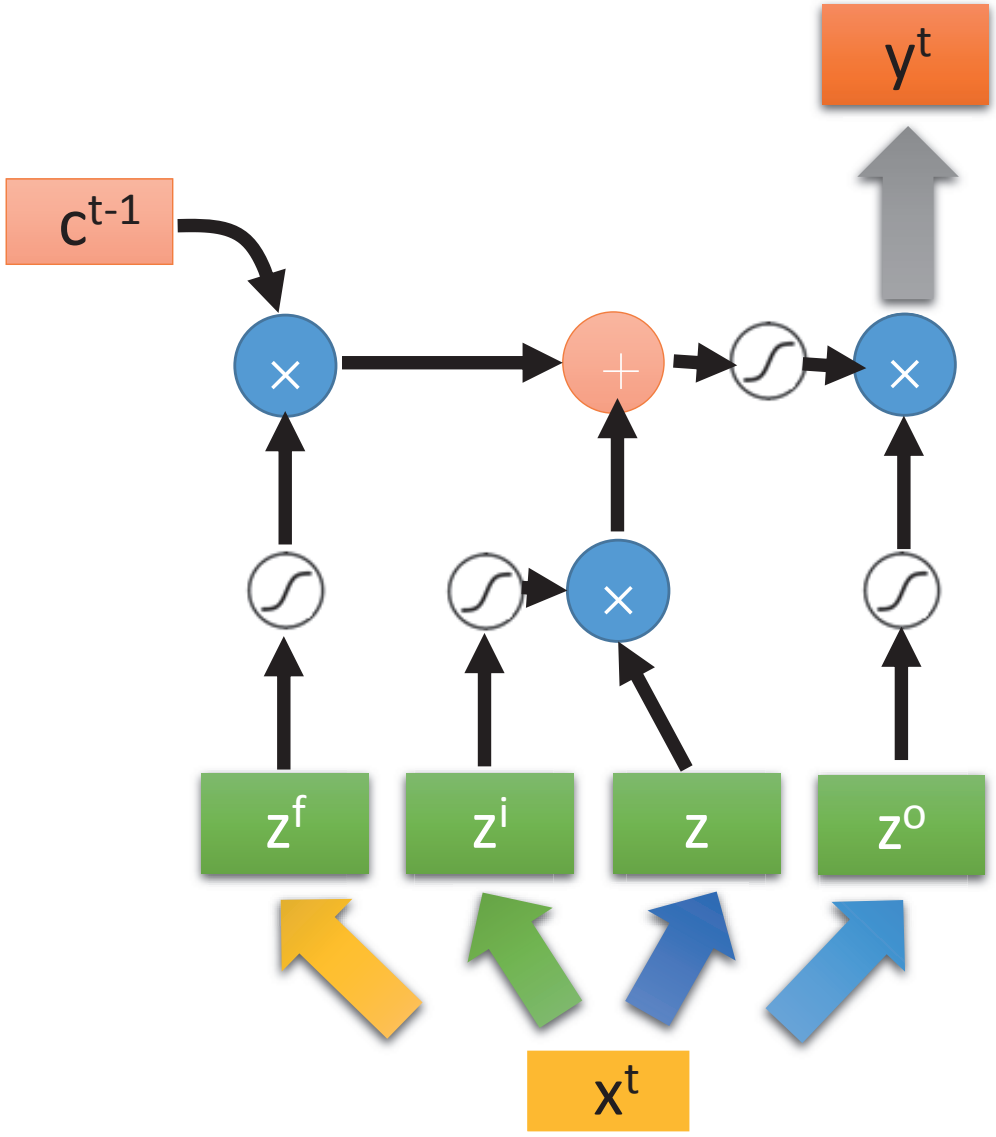


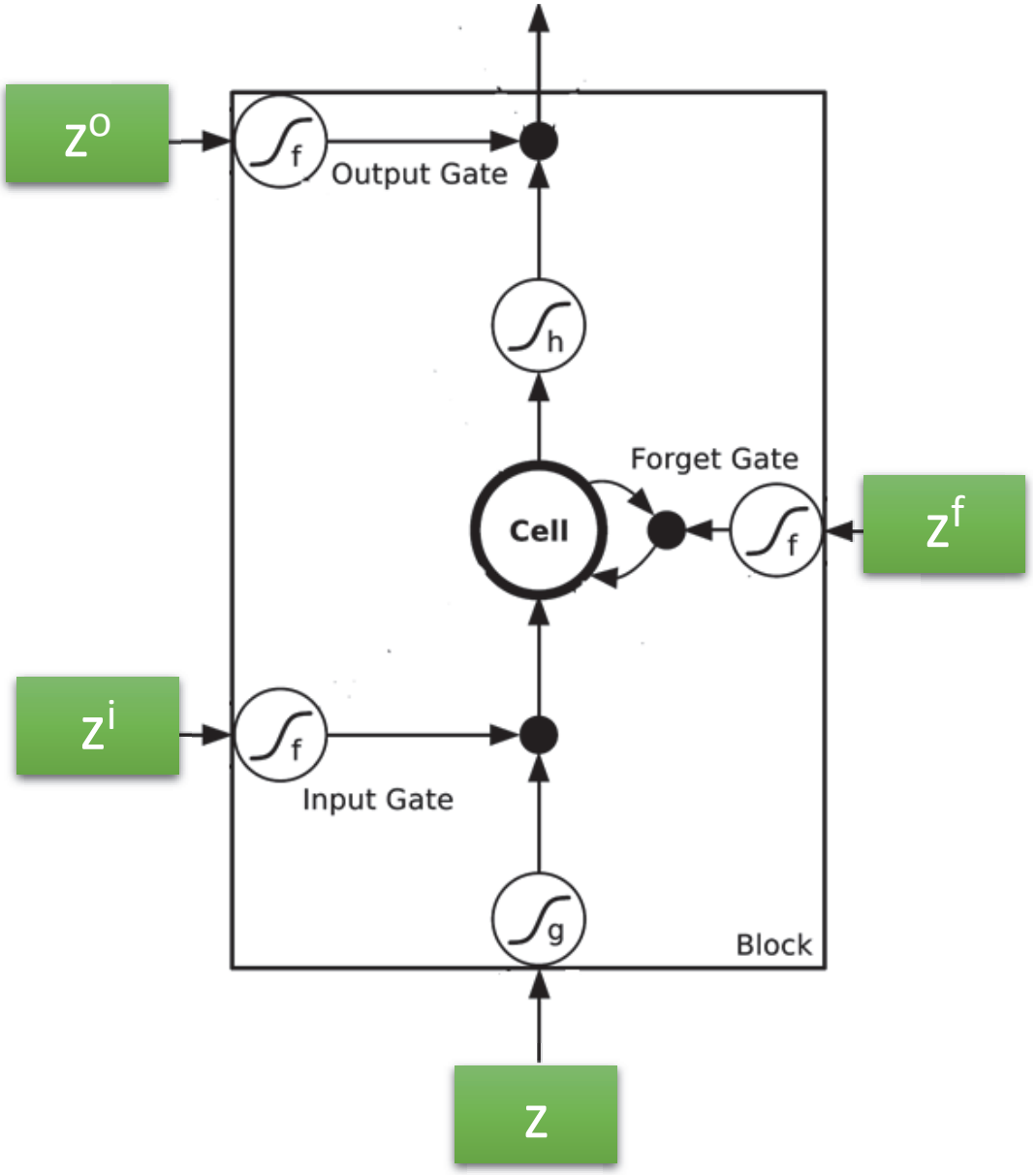

a_1
a_2
Output Gate
Forget Gate
Input Gate
Cell
Block
x_1
x_2
Input
4 times of parameters

LSTM

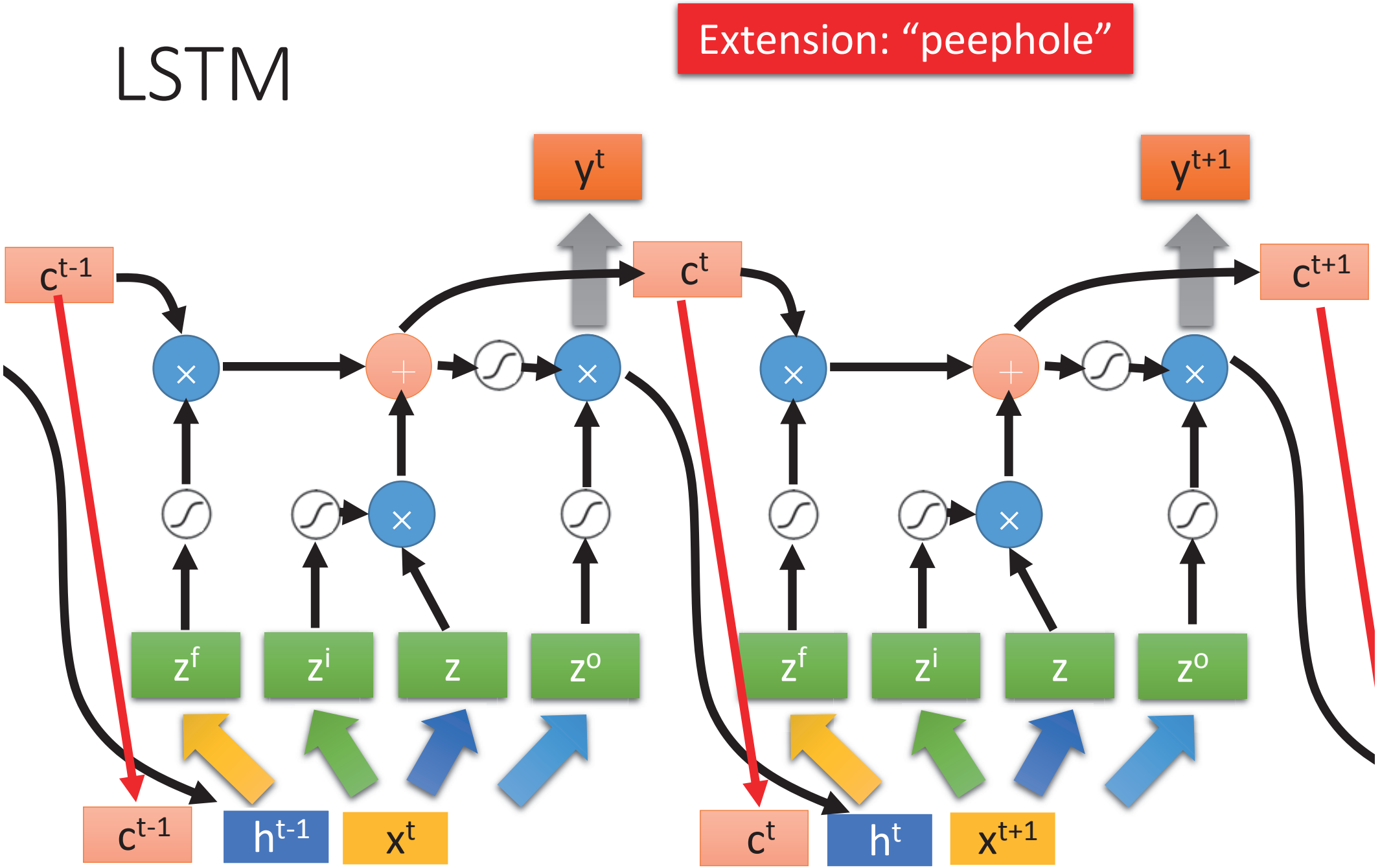

c^{t-1}
vector
Output Gate
Forget Gate
Cell
Input Gate
Block
.....
z^f
z^i
z
z^o
4 vectors
x^t

LSTM

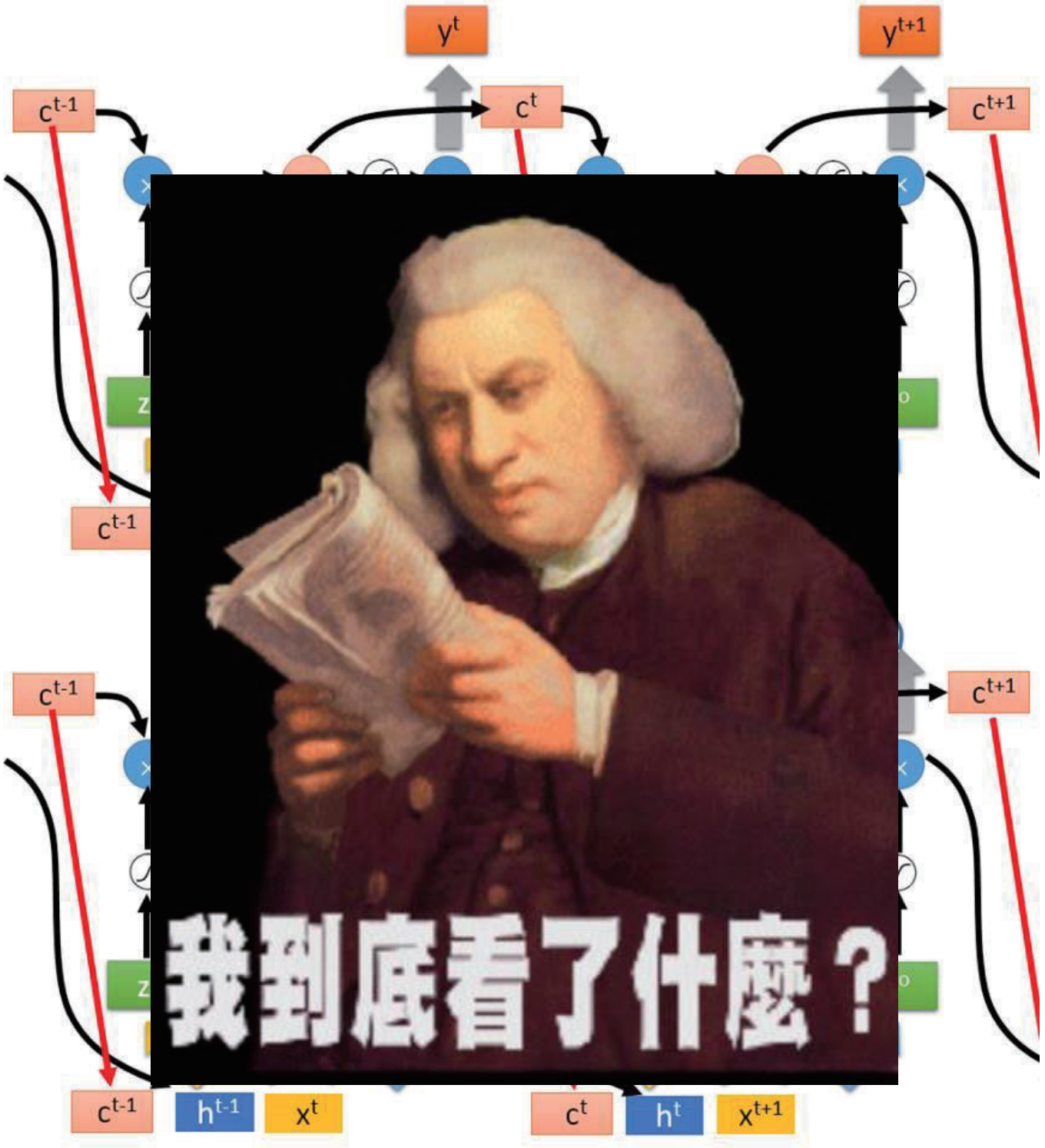

y^t
c^{t-1}
z^f
z^i
z
z^o
x^t


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

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}

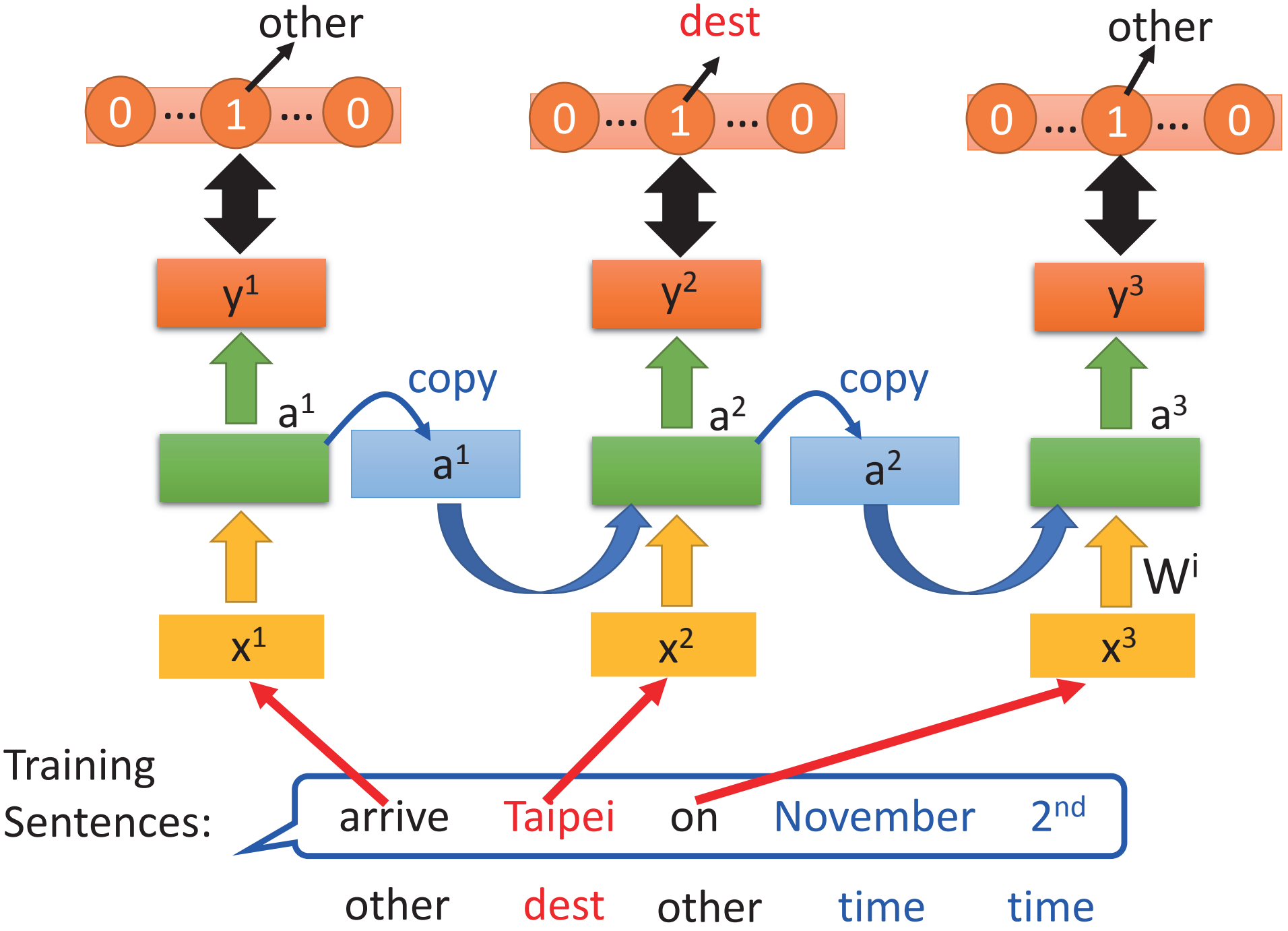
Multiple-layer LSTM



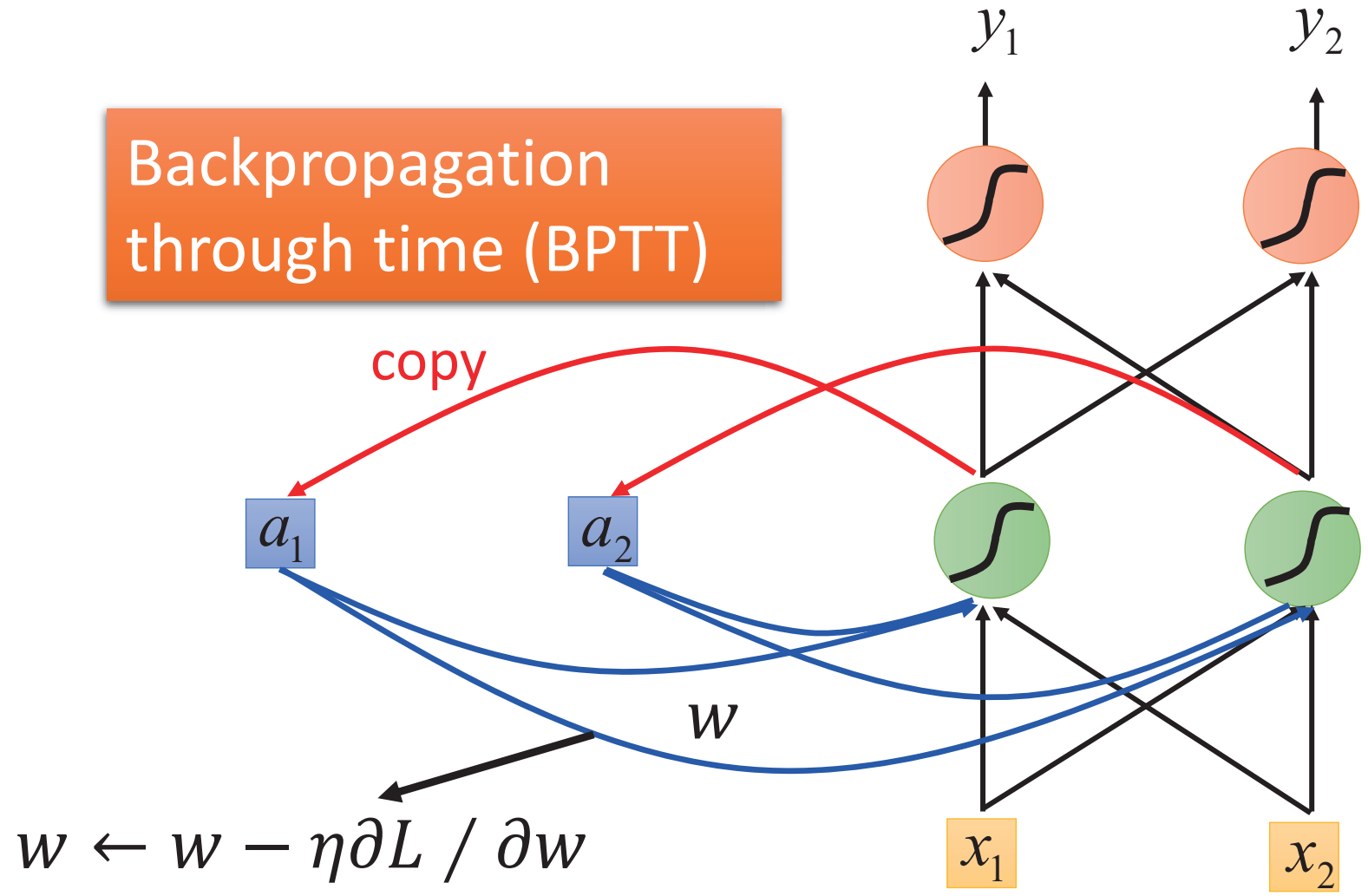
This is quite standard now.

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

Learning Target



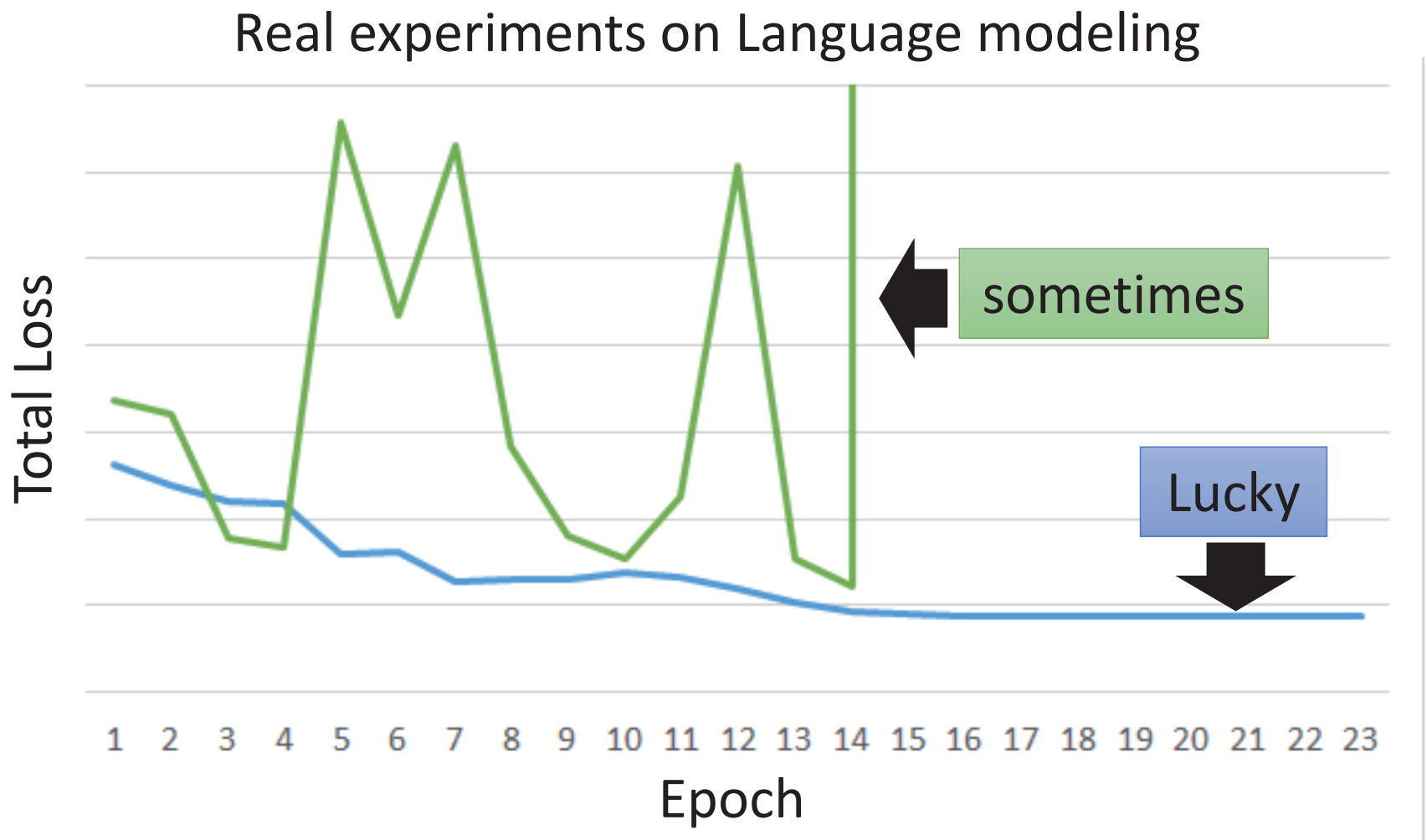
Learning


Backpropagation through time (BPTT)
copy
y_1
y_2
a_1
a_2
w
x_1
x_2
$w \leftarrow w - \eta \partial L / \partial w$

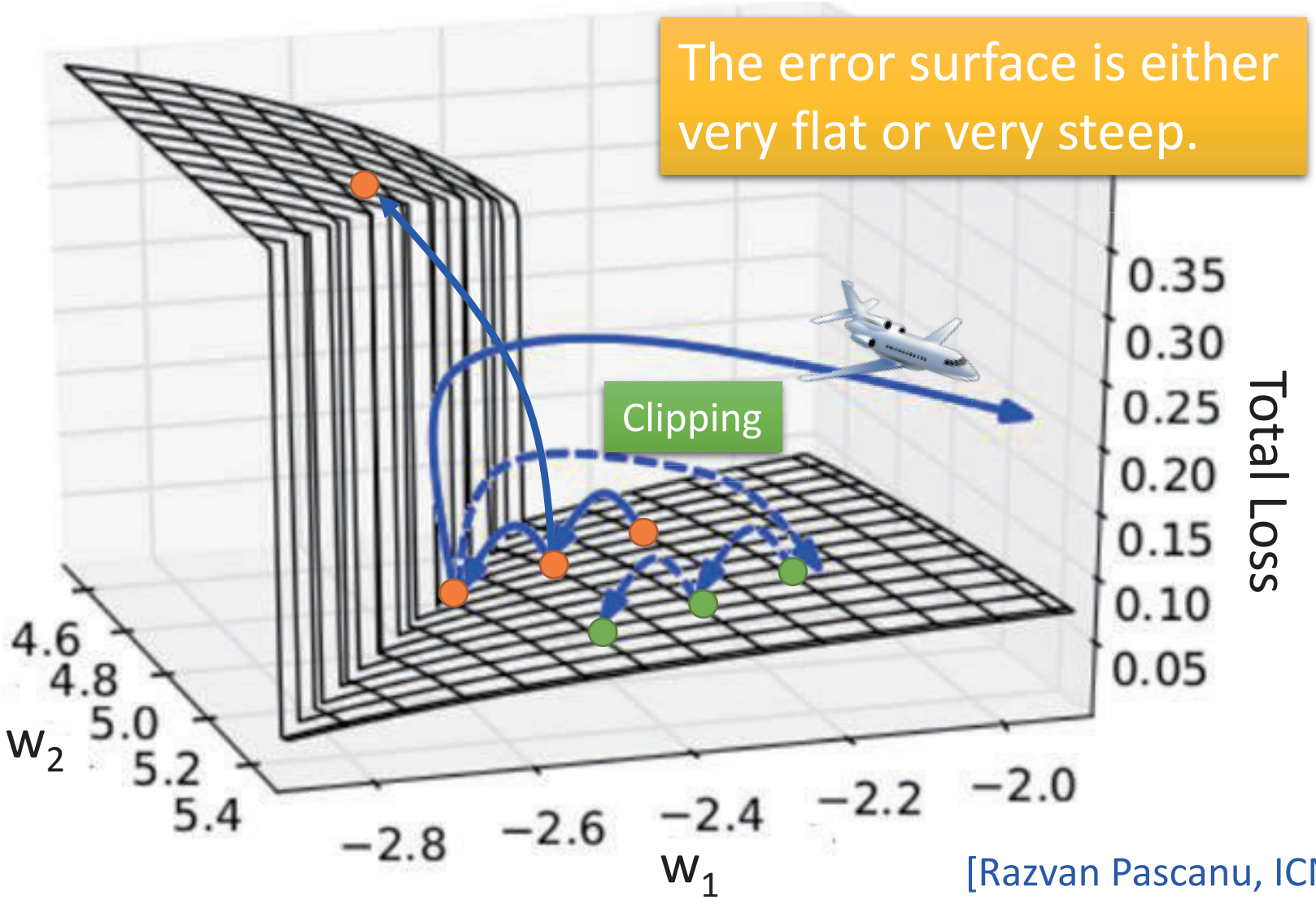
感謝 曾柏翔 同學
提供實驗結果

Unfortunately ……

- RNN-based network is not always easy to learn



The error surface is rough.



[Razvan Pascanu, ICML'13]

Why?

$w = 1 \rightarrow y^{1000} = 1$

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

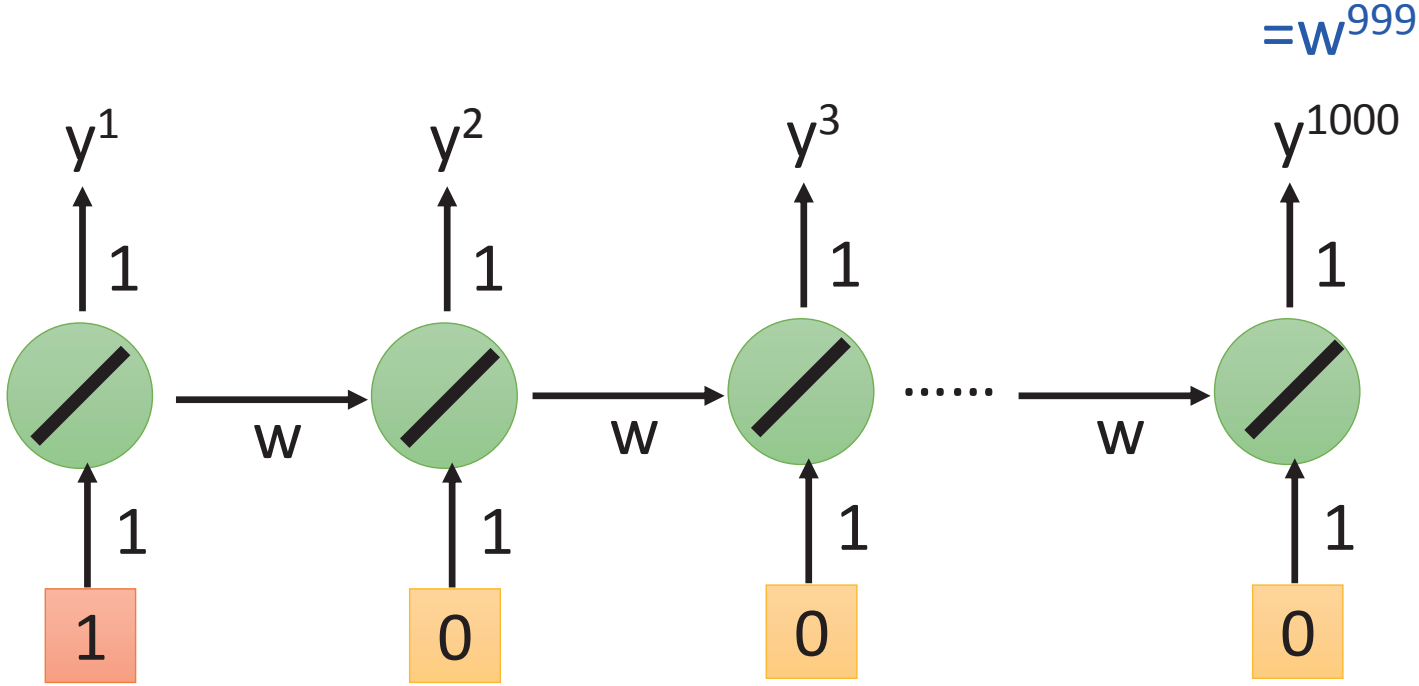
Large $\partial L/\partial w$ → Small Learning rate?

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

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

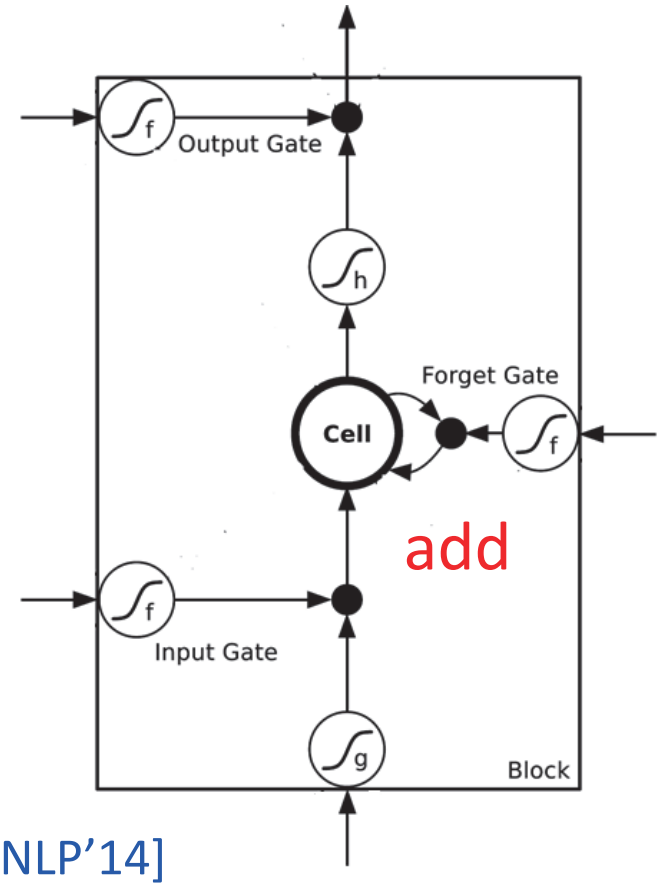
small $\partial L/\partial w$ → Large Learning rate?

Toy Example



Helpful Techniques

- Long Short-term Memory (LSTM)
  - Can deal with gradient vanishing (not gradient explode)
    - Memory and input are ***added***
    - The influence never disappears unless forget gate is closed



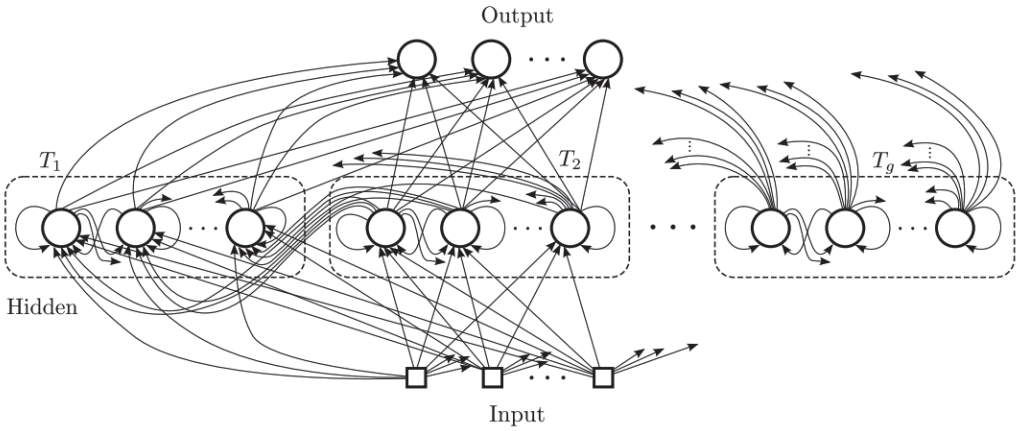
⇒ No Gradient vanishing
(If forget gate is opened.)

Gated Recurrent Unit (GRU): simpler than LSTM

[Cho, EMNLP'14]

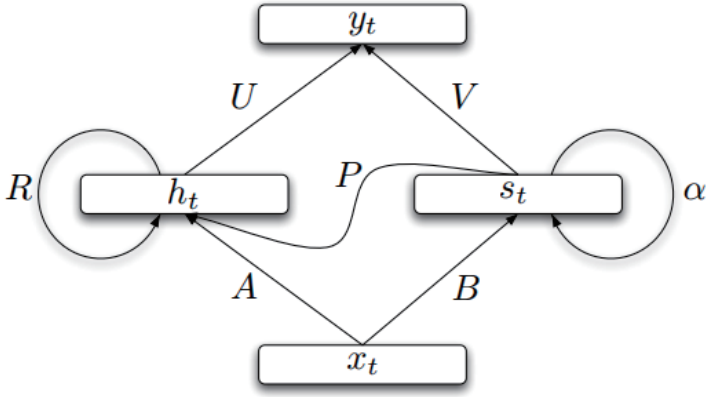
Helpful Techniques

Clockwise RNN



[Jan Koutnik, JMLR'14]

Structurally Constrained Recurrent Network (SCRN)



[Tomas Mikolov, ICLR'15]

Vanilla RNN Initialized with Identity matrix + ReLU activation function [Quoc V. Le, arXiv'15]

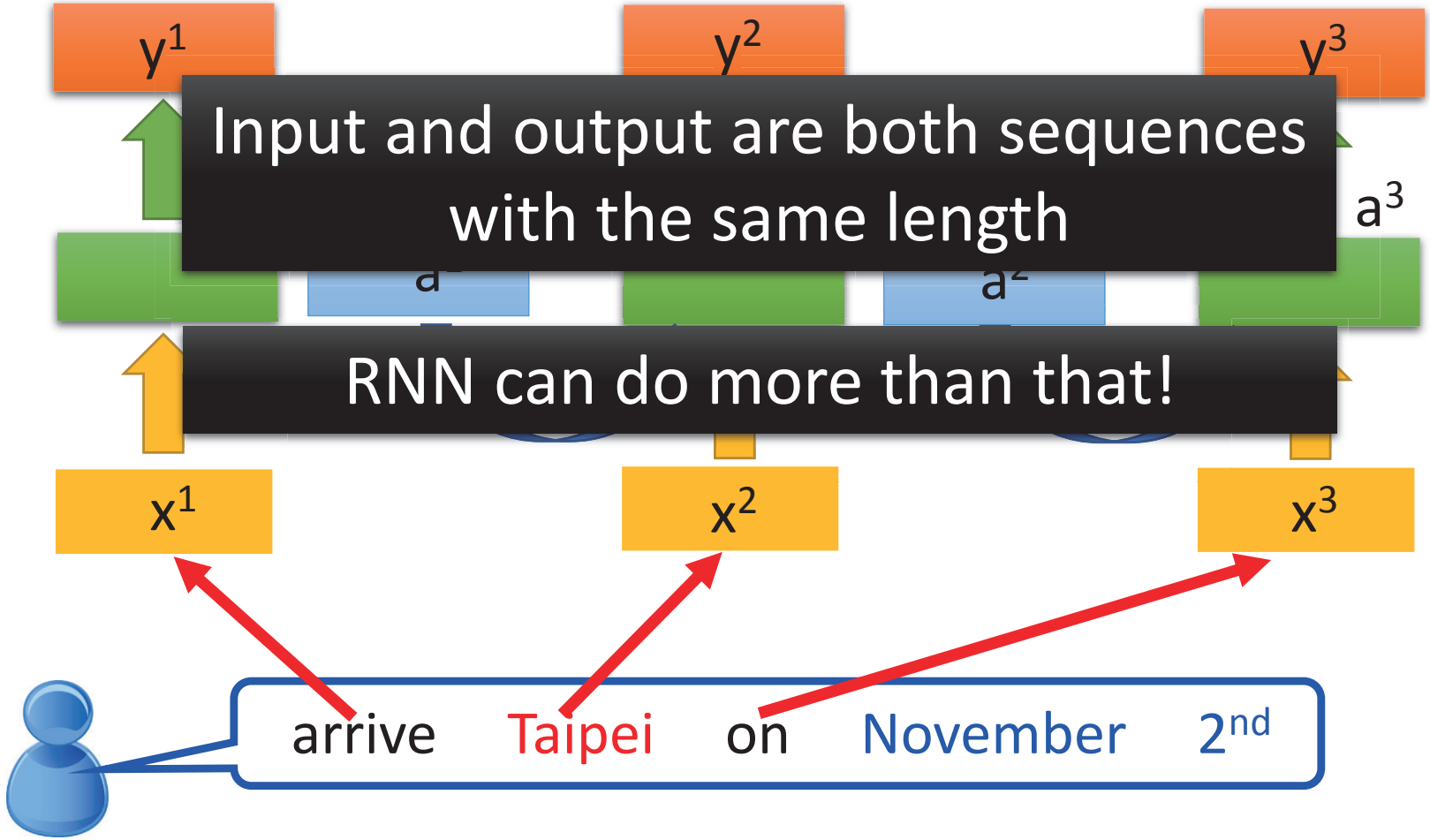
- Outperform or be comparable with LSTM in 4 different tasks

More Applications ……

Probability of "arrive" in each slot

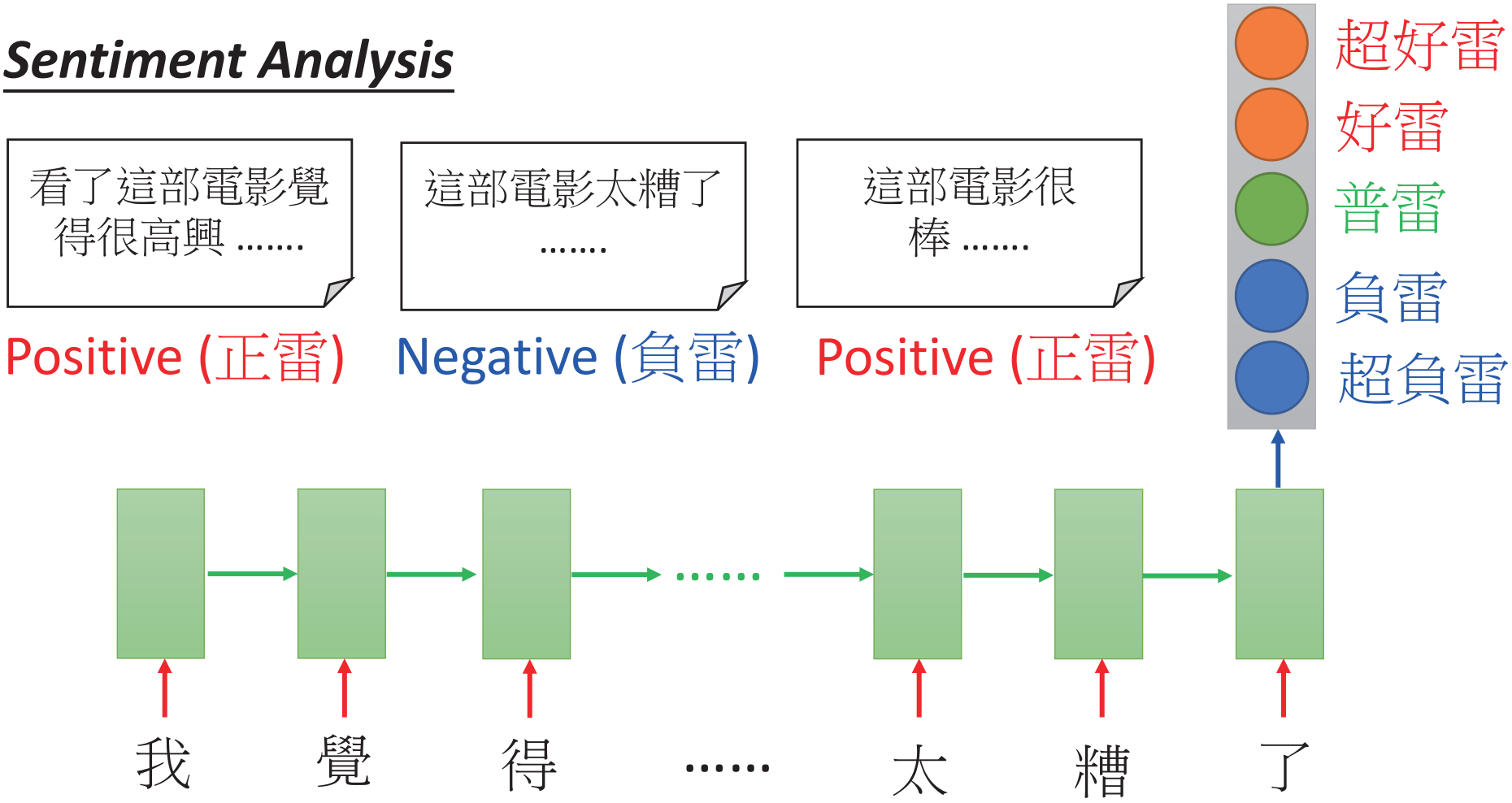
Probability of "Taipei" in each slot

Probability of "on" in each slot



Many to one

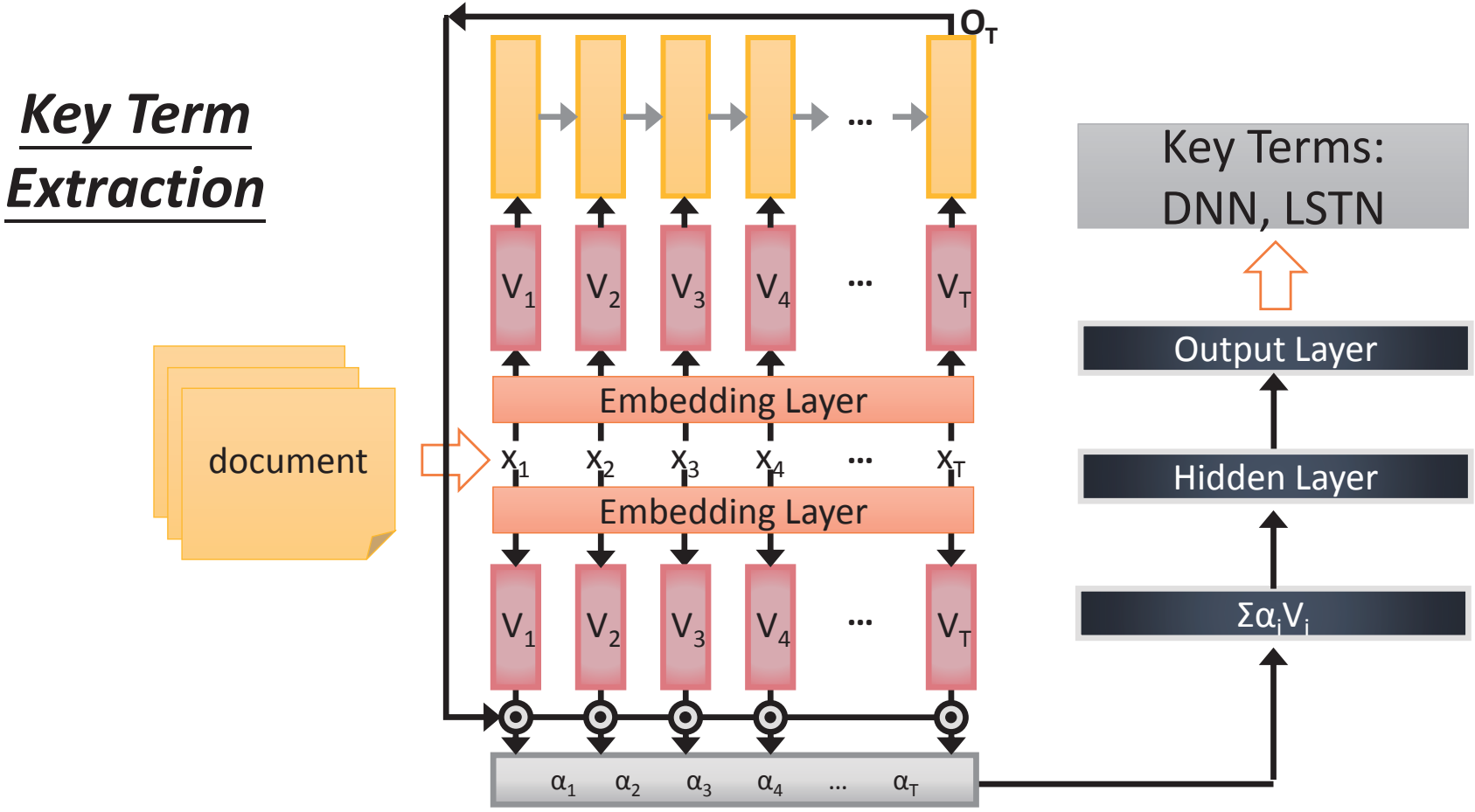
- Input is a vector sequence, but output is only one vector



Many to one

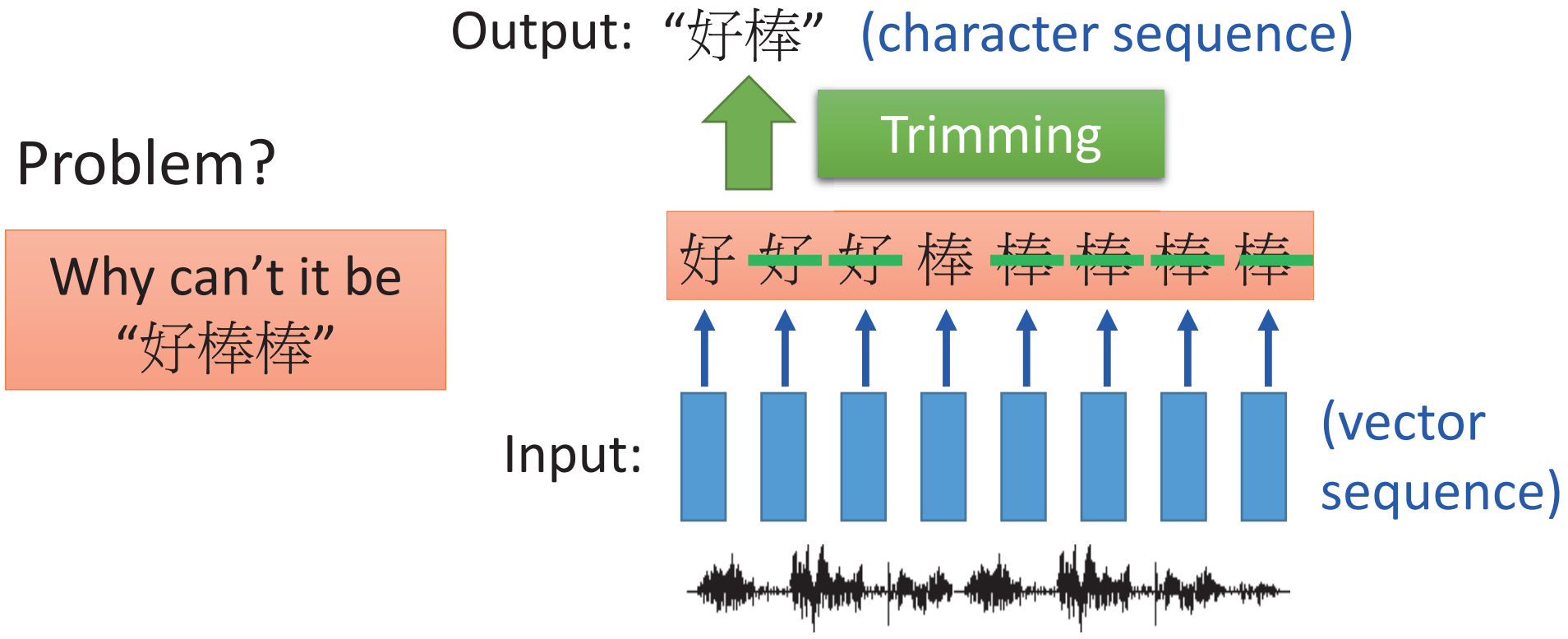
[Shen & Lee, Interspeech 16]

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



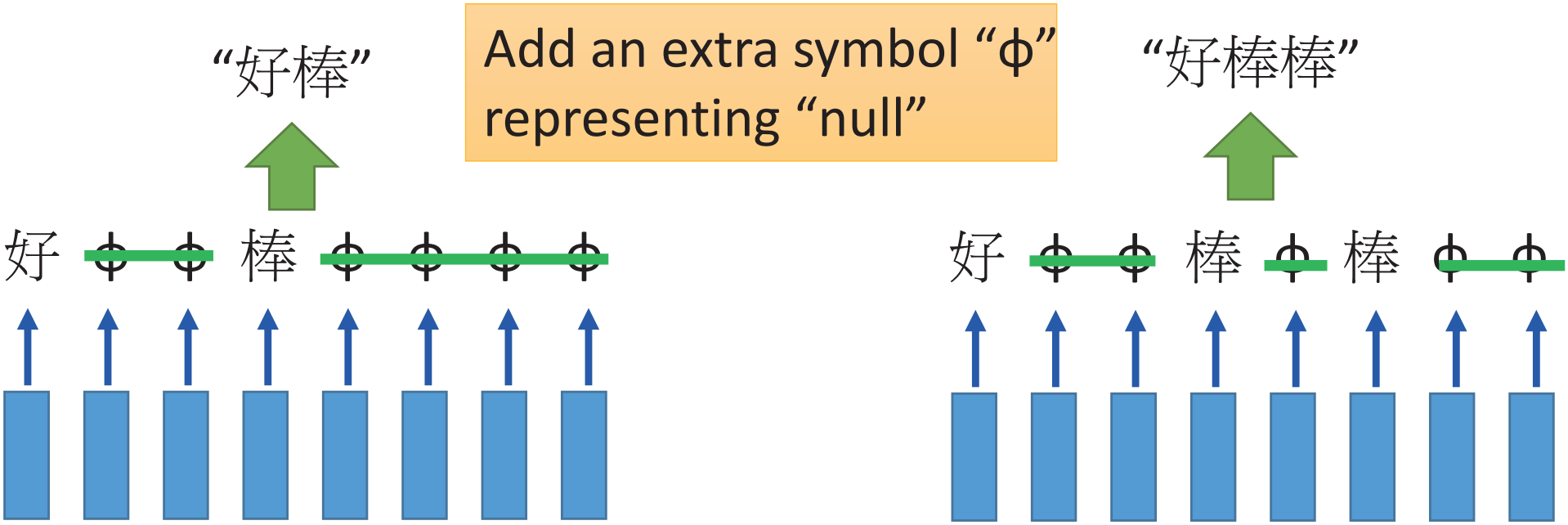
Many to Many (Output is shorter)

- Both input and output are both sequences, ***but the output is shorter.***
  - E.g. ***Speech Recognition***



Many to Many (Output is shorter)

- Both input and output are both sequences, ***but the output is shorter.***
- Connectionist Temporal Classification (CTC) [Alex Graves, ICML'06][Alex Graves, ICML'14][Haşim Sak, Interspeech'15][Jie Li, Interspeech'15][Andrew Senior, ASRU'15]



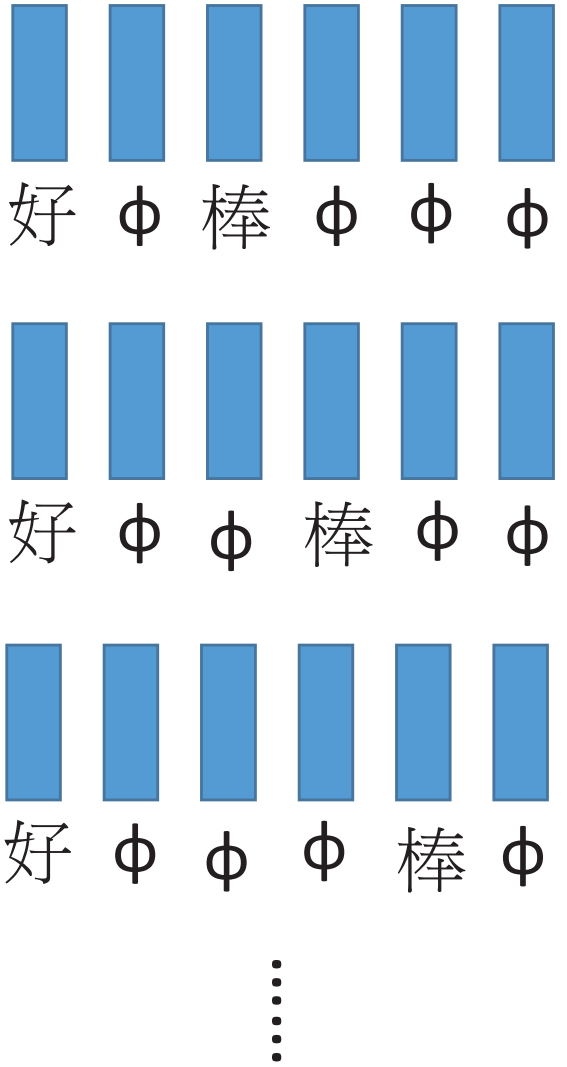
Many to Many (Output is shorter)

- CTC: Training

Acoustic Features:

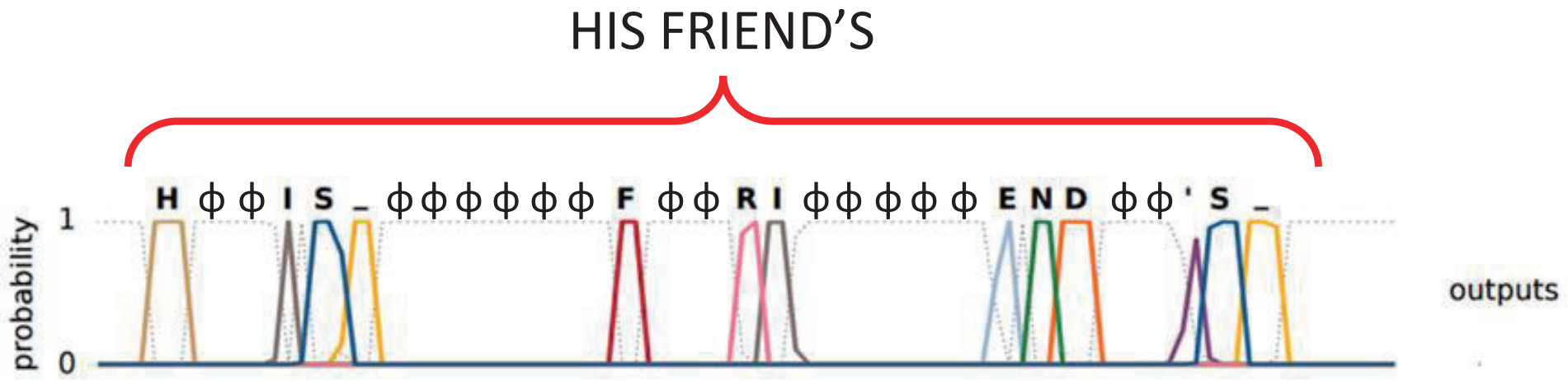
Label: 好 棒

All possible alignments are considered as correct.



Many to Many (Output is shorter)

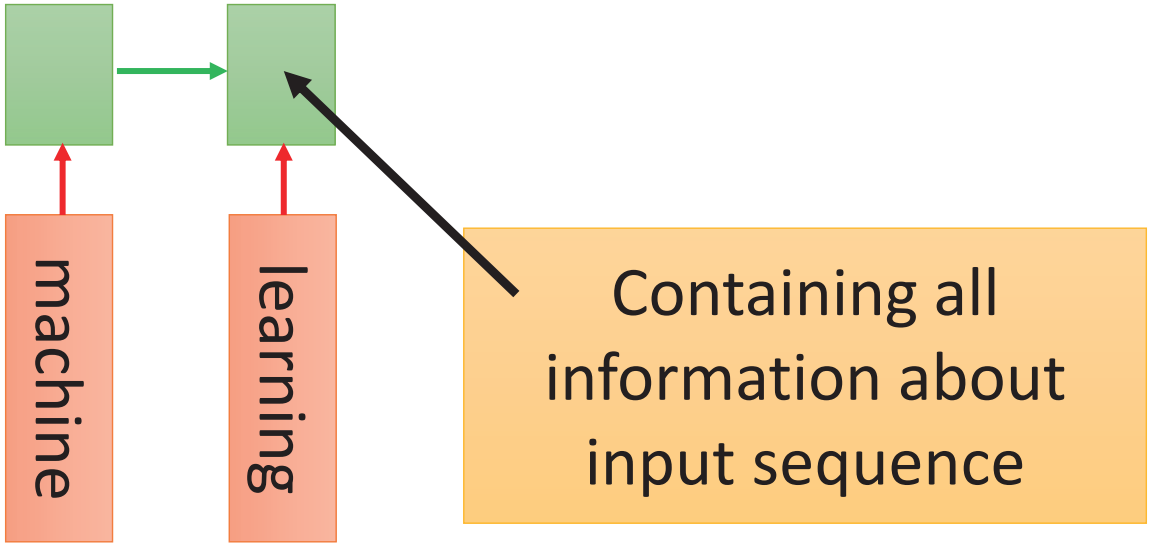
- CTC: example



Graves, Alex, and Navdeep Jaitly. "Towards end-to-end speech recognition with recurrent neural networks." *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 2014.

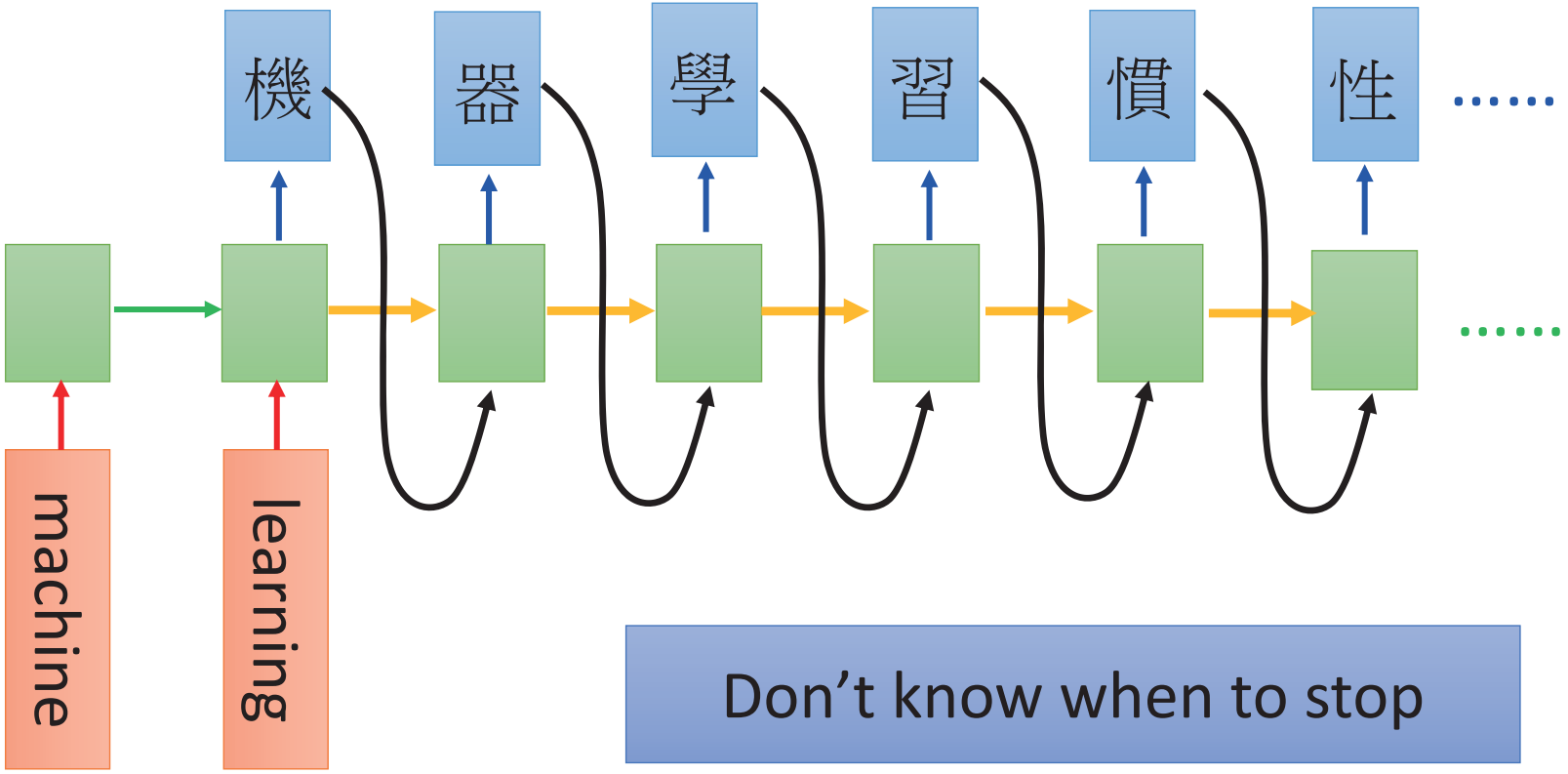
Many to Many (No Limitation)

- Both input and output are both sequences ***with different lengths***. → ***Sequence to sequence learning***
  - E.g. ***Machine Translation*** (machine learning→機器學習)



Many to Many (No Limitation)

- Both input and output are both sequences ***with different lengths***. → ***Sequence to sequence learning***
  - E.g. ***Machine Translation*** (machine learning→機器學習)

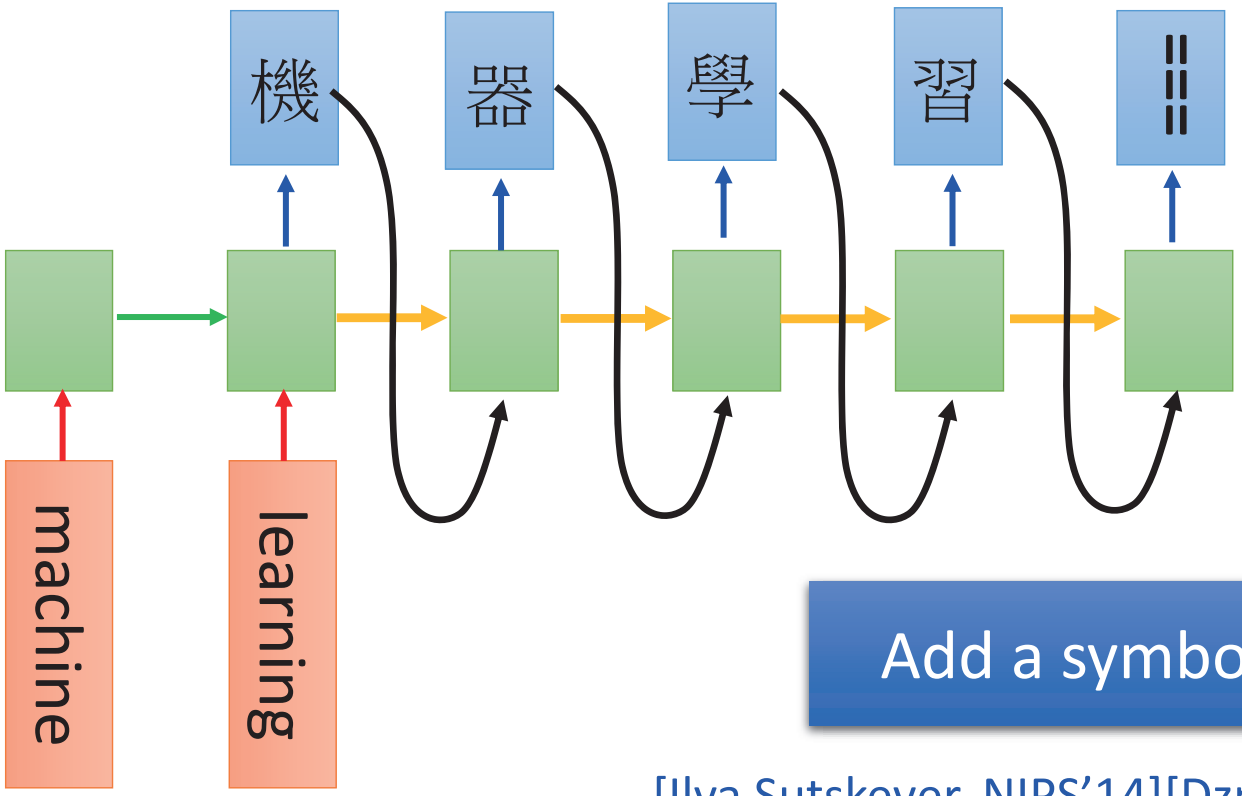


Many to Many (No Limitation)

接龍推文是ptt在推文中的一種趣味玩法，與推齊有些類似但又有所不同，是指在推文中接續上一樓的字句，而推出連續的意思。該類玩法確切起源已不可知(鄉民百科)

Many to Many (No Limitation)

- Both input and output are both sequences ***with different lengths***. → ***Sequence to sequence learning***
  - E.g. ***Machine Translation*** (machine learning→機器學習)



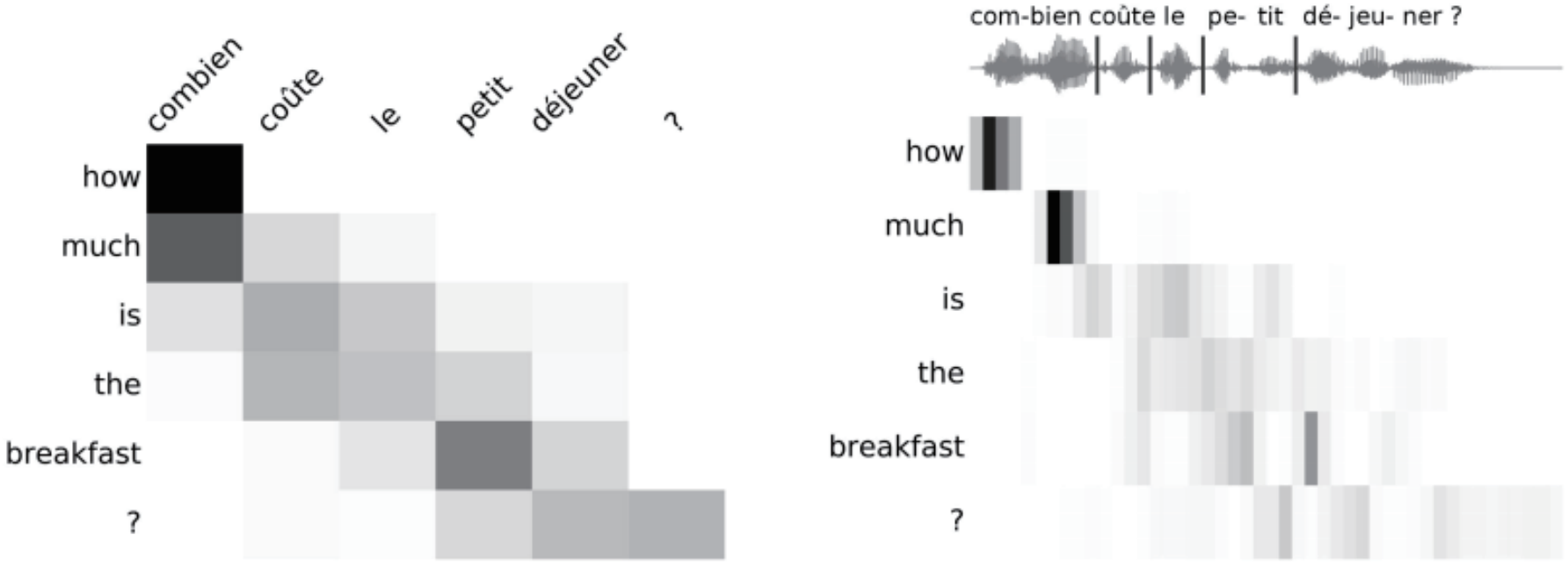
Add a symbol "===" (斷)

[Ilya Sutskever, NIPS'14][Dzmitry Bahdanau, arXiv'15]

https://arxiv.org/pdf/1612.01744v1.pdf

Many to Many (No Limitation)

- Both input and output are both sequences ***with different lengths***. → ***Sequence to sequence learning***
  - E.g. ***Machine Translation*** (machine learning→機器學習)



(a) Machine translation alignment

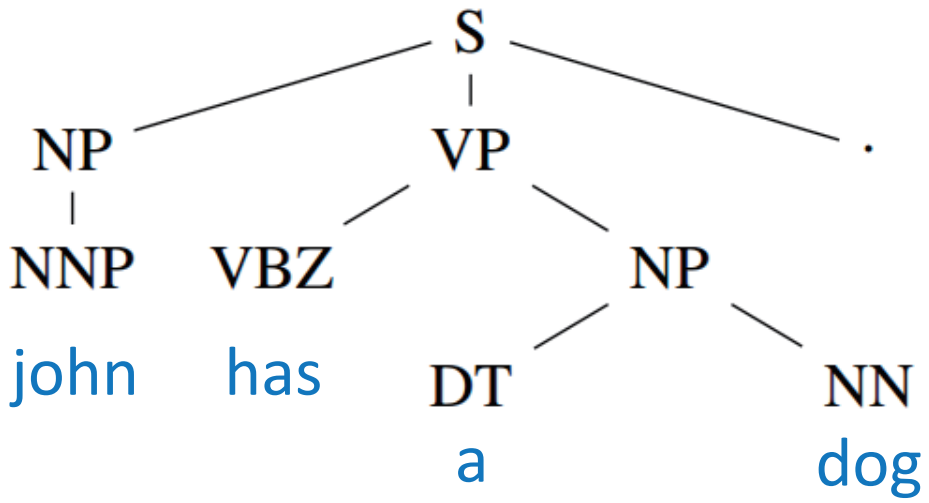
(b) Speech translation alignment

Figure 1: Alignments performed by the attention model during training

Beyond Sequence

- Syntactic parsing

John has a dog . →

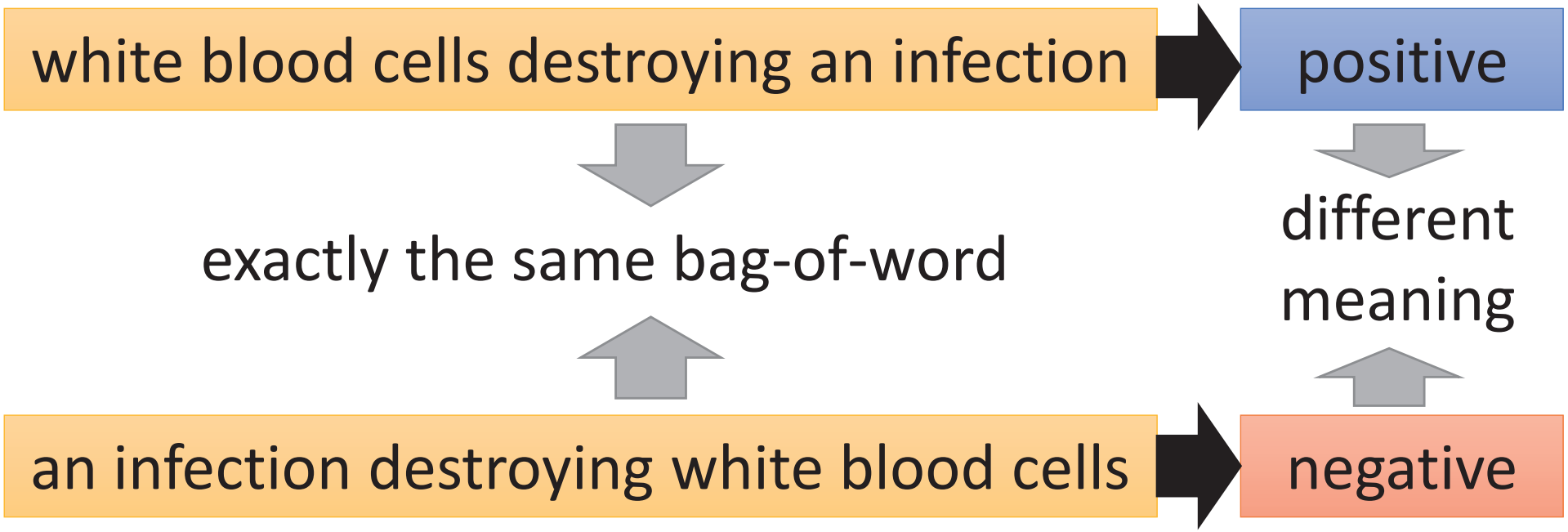


John has a dog . → (S (NP NNP)$_{NP}$ (VP VBZ (NP DT NN)$_{NP}$)$_{VP}$.)$_{S}$

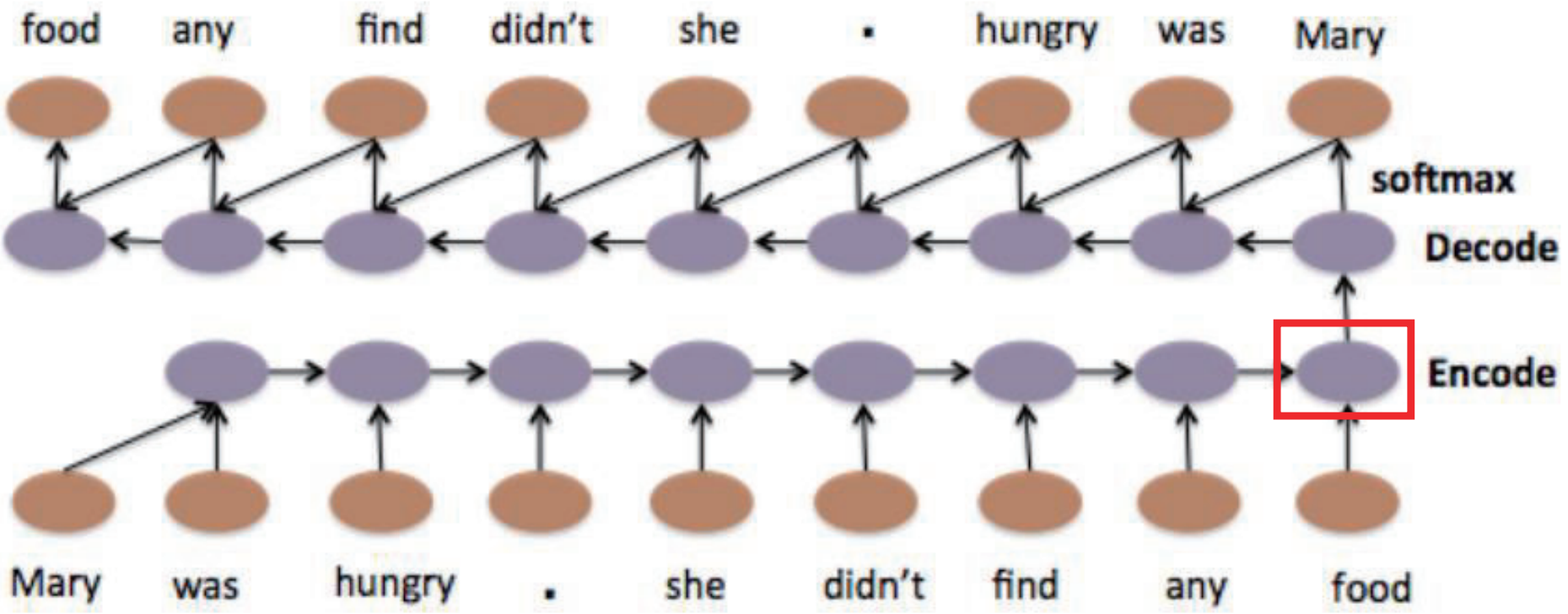
Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, Geoffrey Hinton, Grammar as a Foreign Language, NIPS 2015

Sequence-to-sequence Auto-encoder - Text

- To understand the meaning of a word sequence, the order of the words can not be ignored.

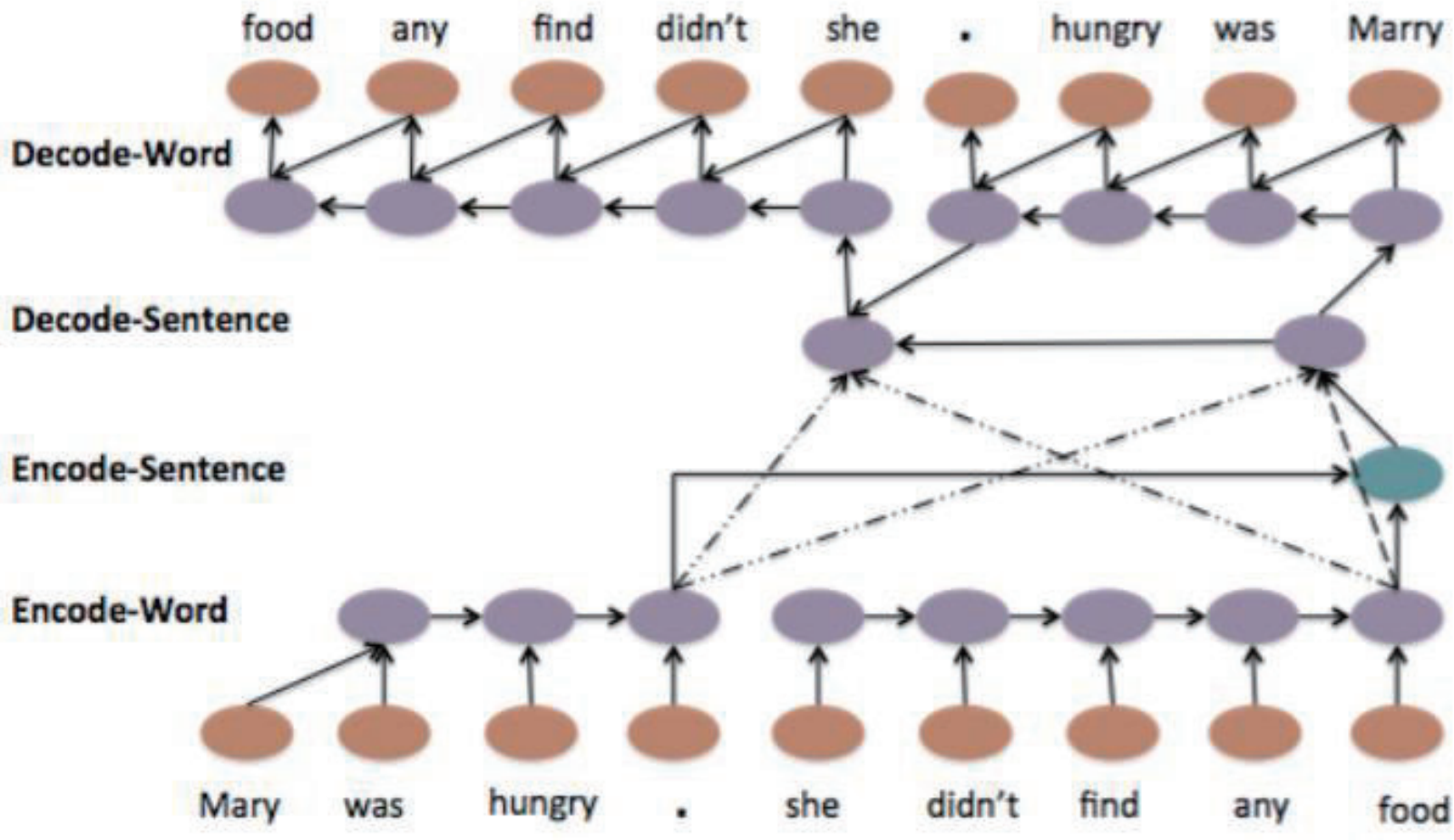


Sequence-to-sequence Auto-encoder - Text



Li, Jiwei, Minh-Thang Luong, and Dan Jurafsky. "A hierarchical neural autoencoder for paragraphs and documents." *arXiv preprint arXiv:1506.01057*(2015).

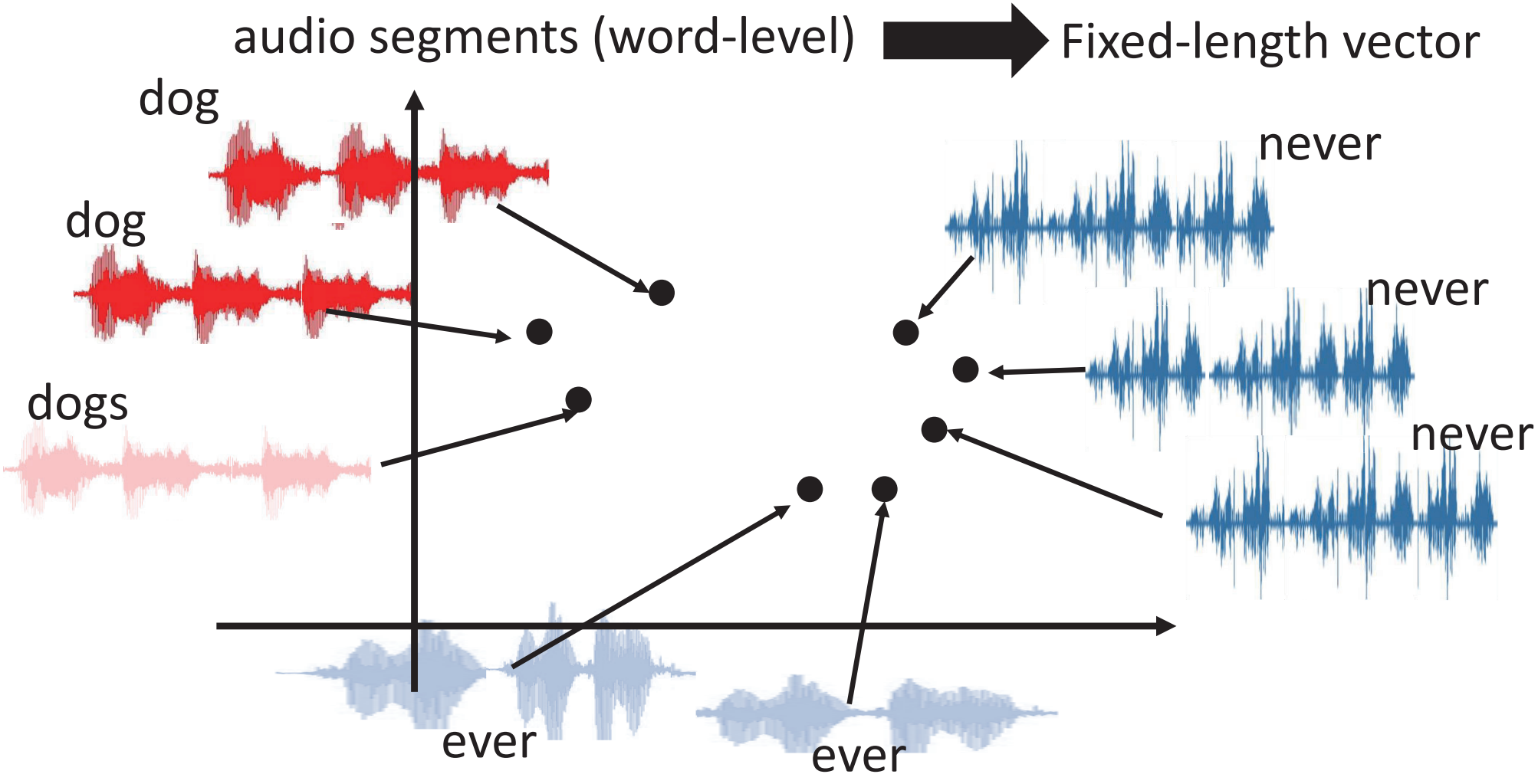
Sequence-to-sequence Auto-encoder - Text



Li, Jiwei, Minh-Thang Luong, and Dan Jurafsky. "A hierarchical neural autoencoder for paragraphs and documents." *arXiv preprint arXiv:1506.01057*(2015).

Sequence-to-sequence Auto-encoder - Speech

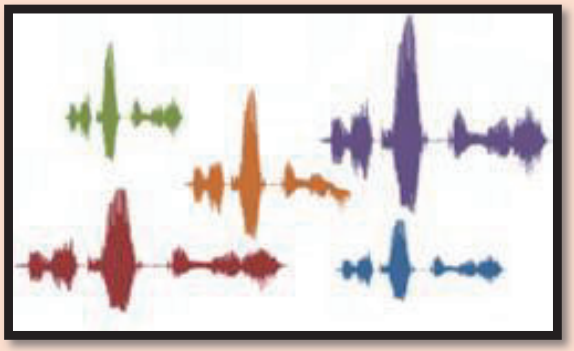
- Dimension reduction for a sequence with variable length



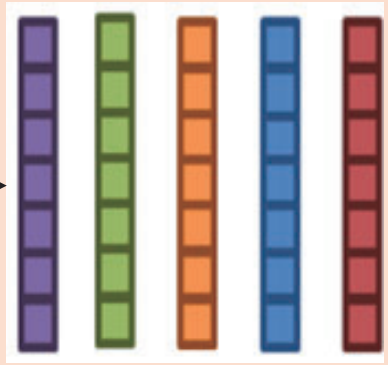
Sequence-to-sequence Auto-encoder - Speech

Audio archive divided into variable-length audio segments

Off-line



Audio Segment to Vector



Spoken Query

Audio Segment to Vector

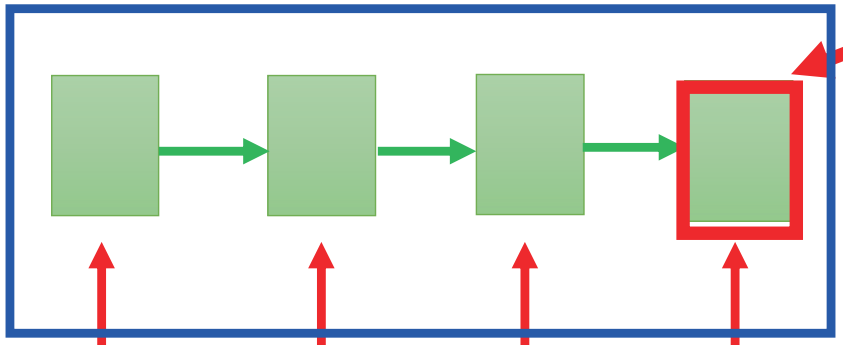
Similarity

Search Result

On-line

Sequence-to-sequence Auto-encoder - Speech

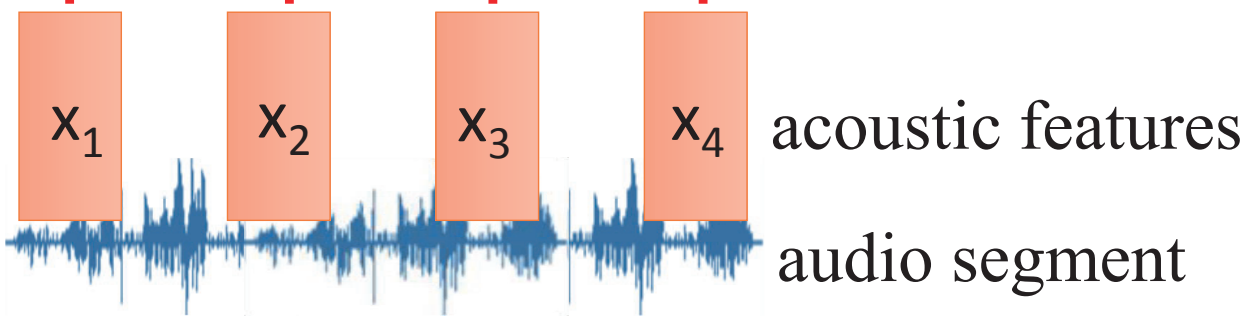
RNN Encoder



The values in the memory represent the whole audio segment

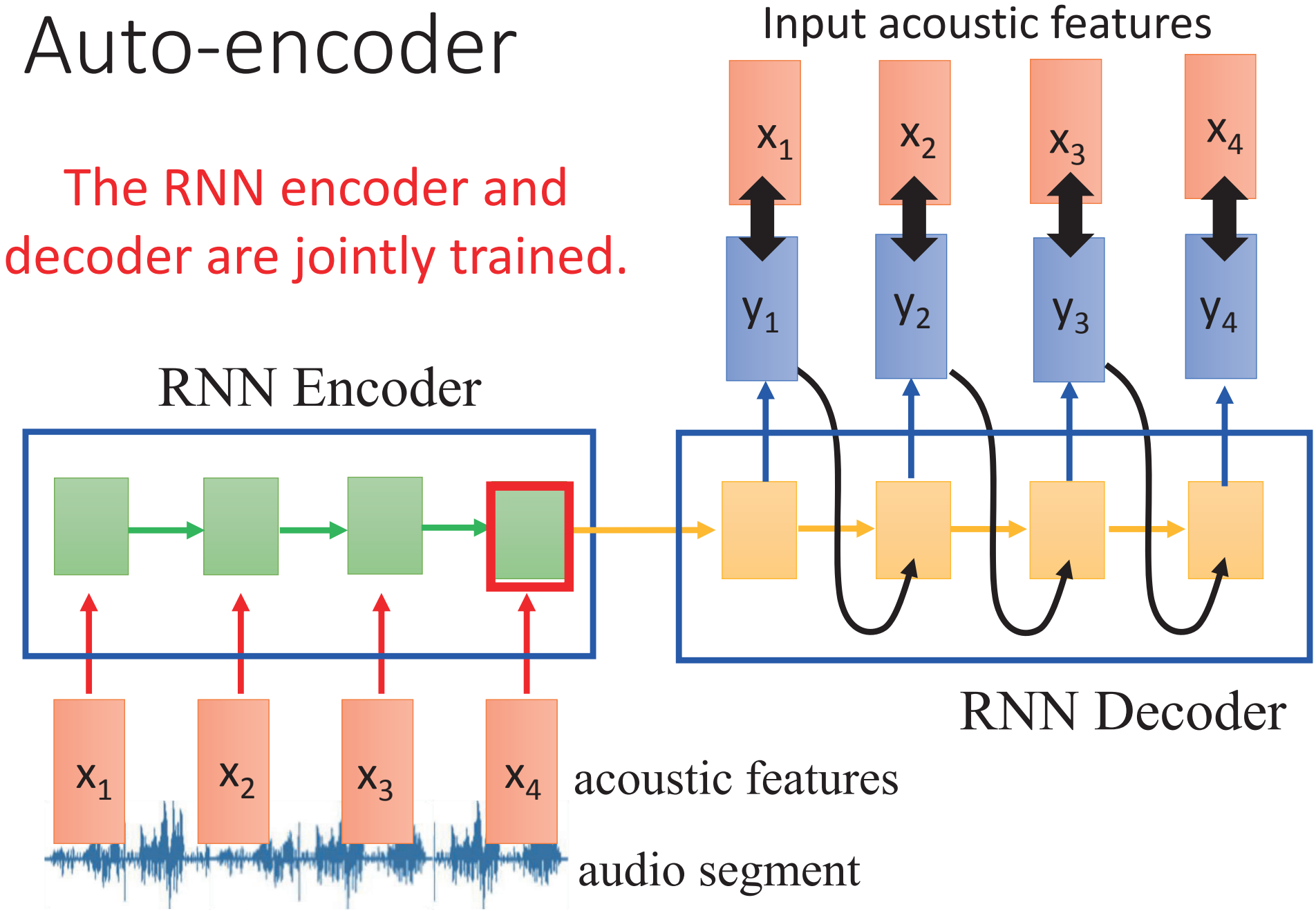
The vector we want

How to train RNN Encoder?



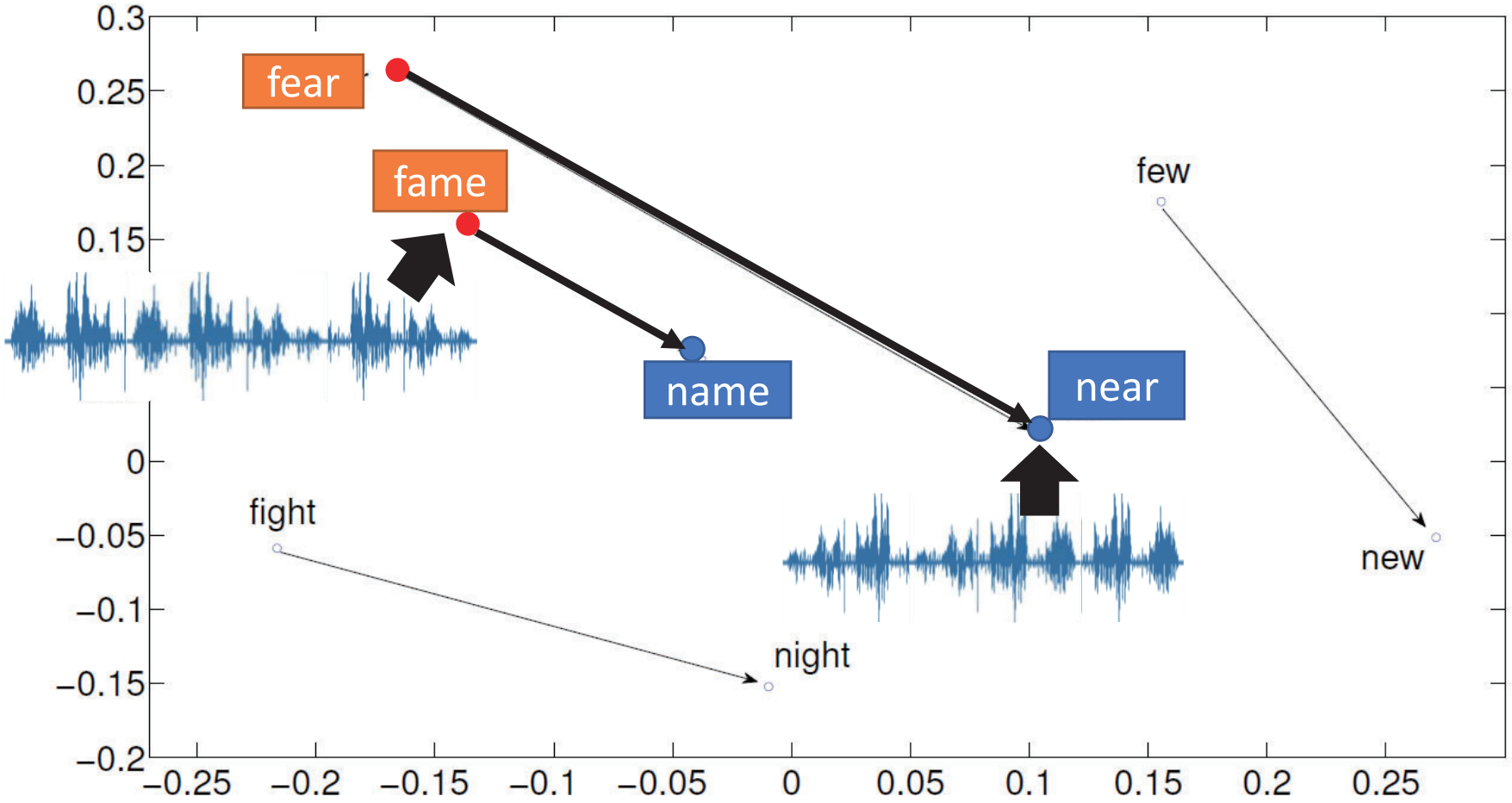
Sequence-to-sequence Auto-encoder

The RNN encoder and decoder are jointly trained.

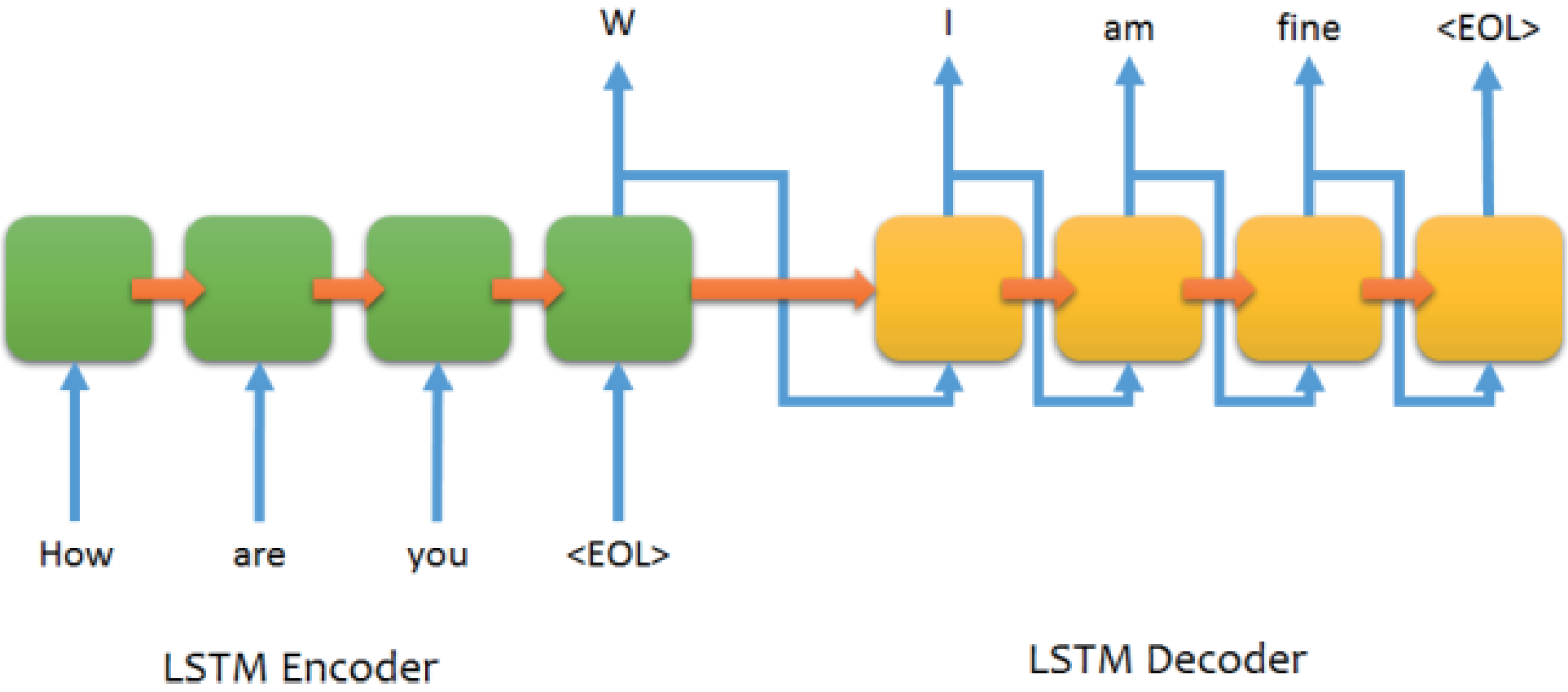


Sequence-to-sequence Auto-encoder - Speech

- Visualizing embedding vectors of the words



Demo: Chat-bot



電視影集 (~40,000 sentences)、美國總統大選辯論

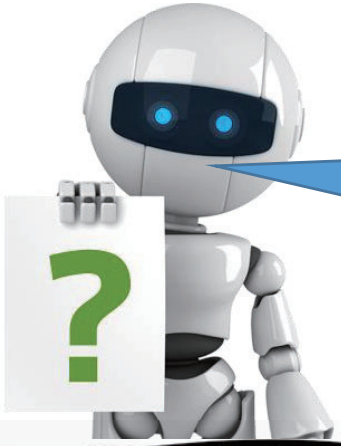
Demo: Chat-bot

- Develop Team
  - Interface design: Prof. Lin-Lin Chen & Arron Lu
  - Web programming: Shi-Yun Huang
  - Data collection: Chao-Chuang Shih
  - System implementation: Kevin Wu, Derek Chuang, & Zhi-Wei Lee (李致緯), Roy Lu (盧柏儒)
  - System design: Richard Tsai & Hung-Yi Lee

Demo: Video Caption Generation

Video

A group of people is knocked by a tree.



A girl is running.

A group of people is walking in the forest.