

Topic 5

Well Known DL Architecture RNN, LSTM

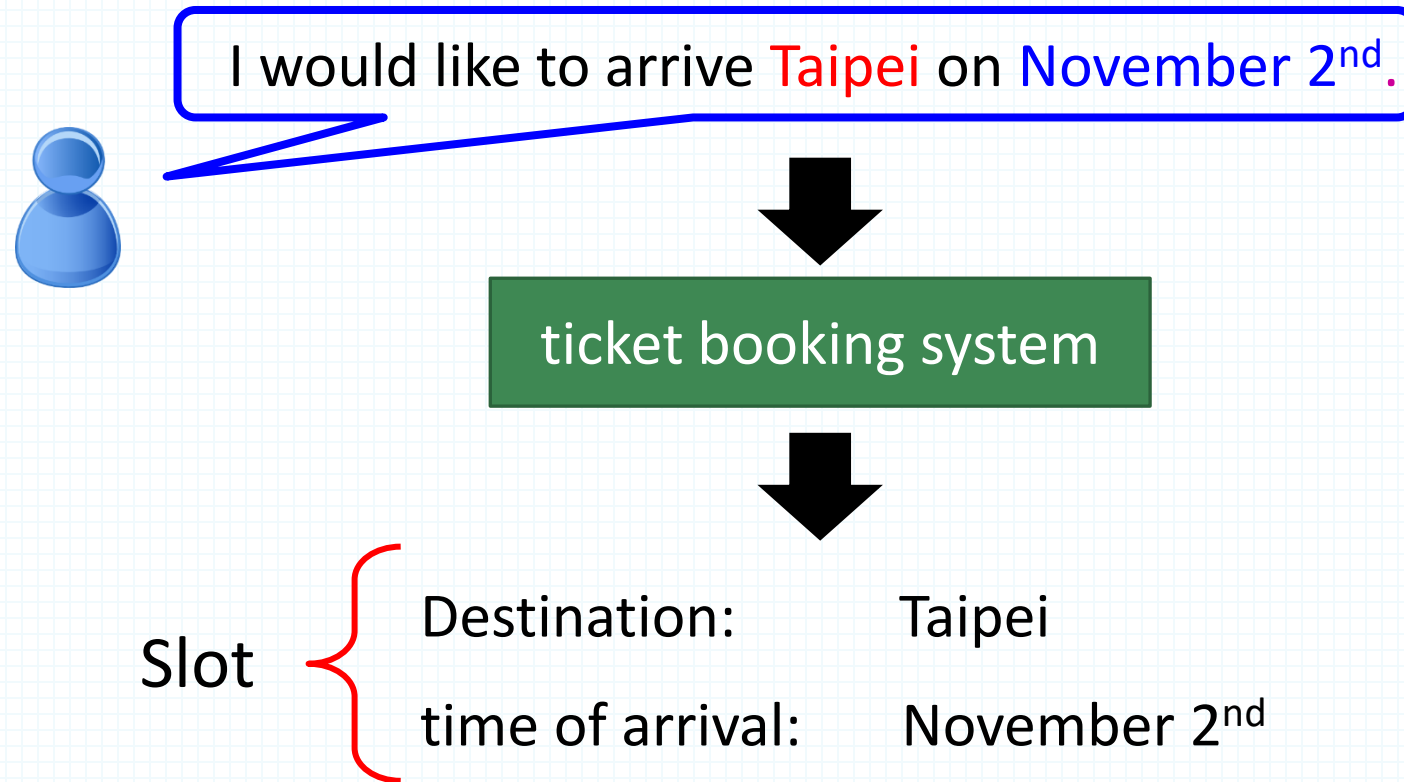
د. رياض سنبل

الشبكات العصبونية التكرارية

Recurrent Neural Networks

Example Application

- Slot Filling



Example Application

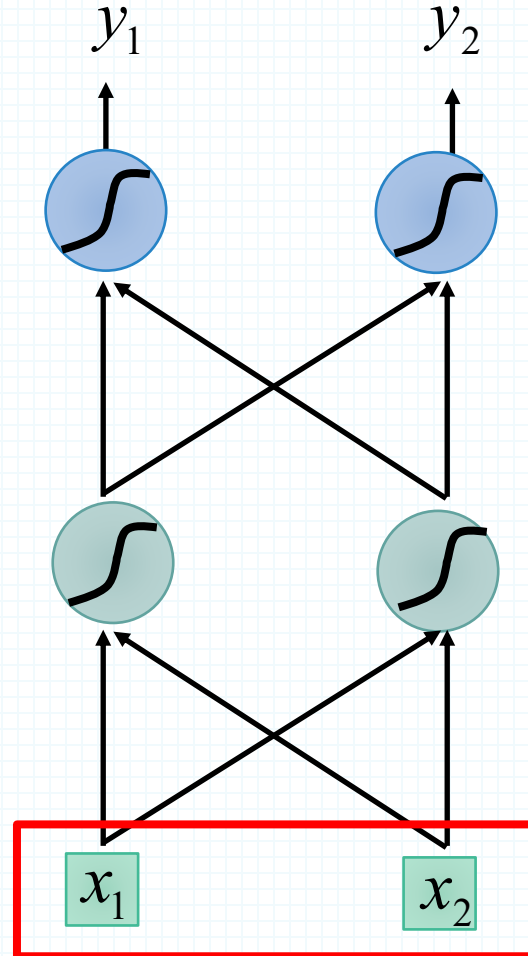
Solving slot filling by
Feedforward network?

Input: a word

(Each word is represented
as a vector)

How can we represent each word?

Taipei



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]

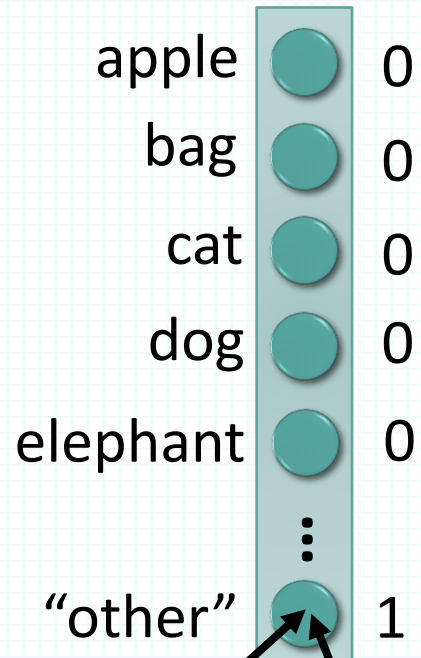
cat = [0 0 1 0 0]

dog = [0 0 0 1 0]

elephant = [0 0 0 0 1]

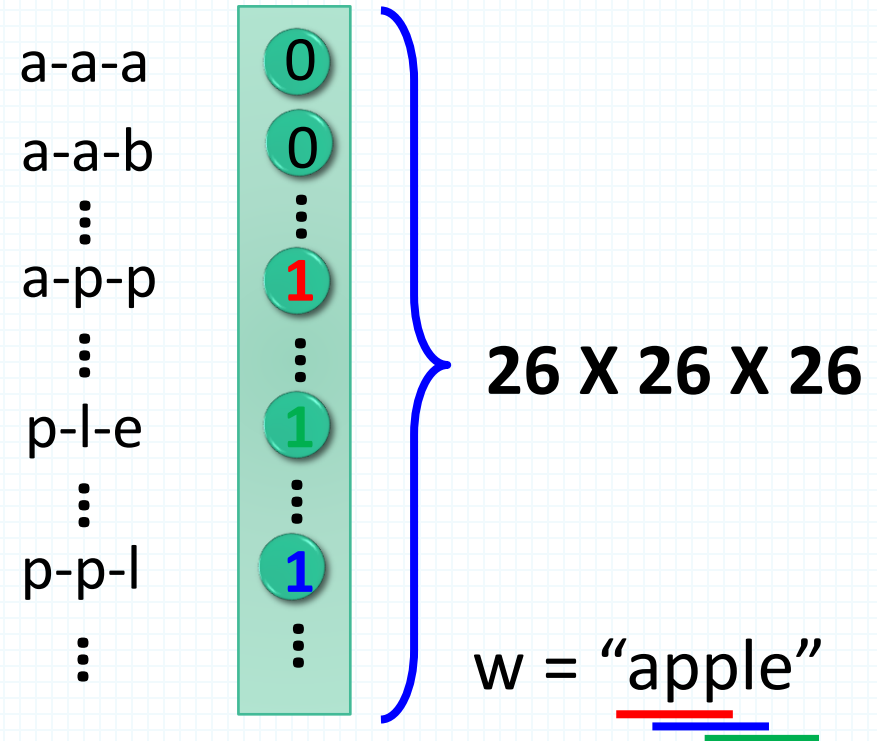
Beyond 1-of-N encoding

Dimension for "Other"



w = "Gandalf" w = "Sauron"

Word hashing



Example Application

Solving slot filling by
Feedforward network?

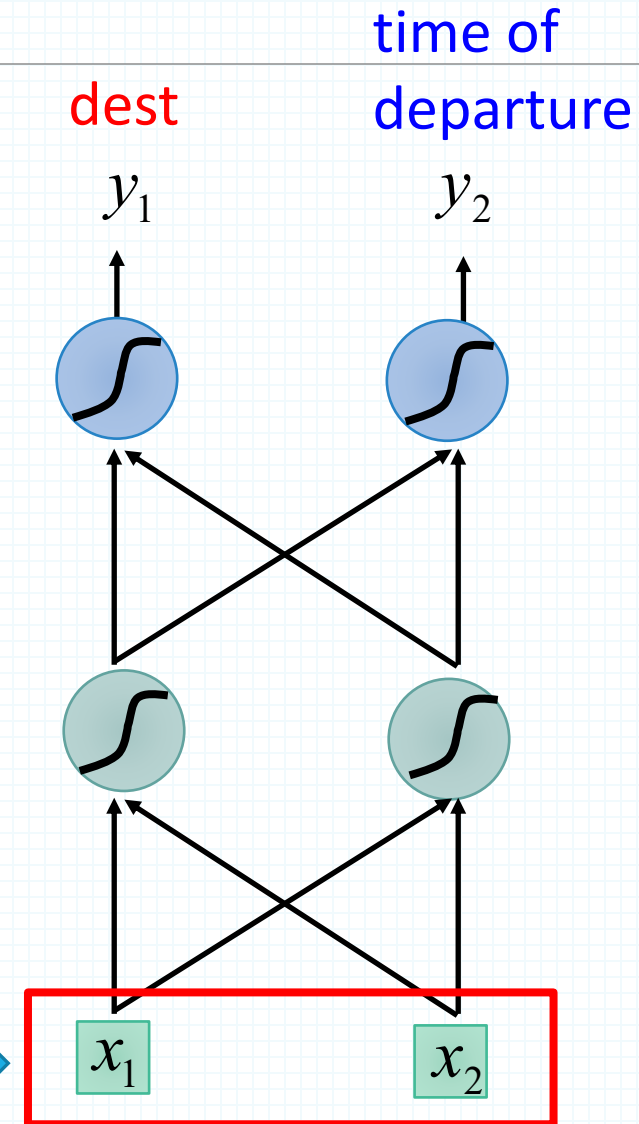
Input: a word

(Each word is represented
as a vector)

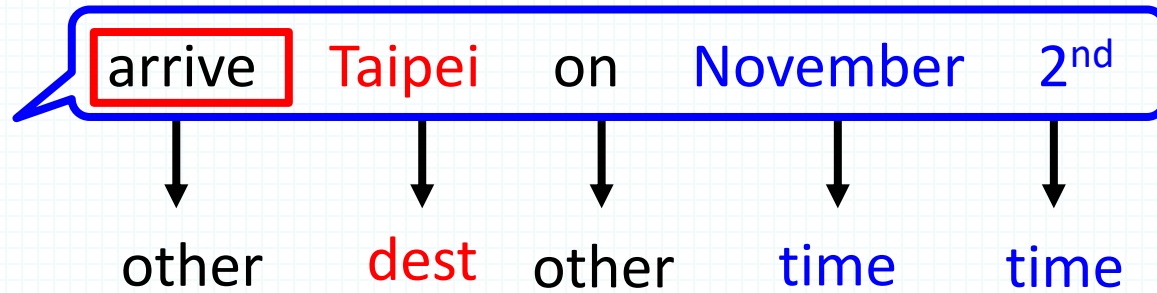
Output:

Probability distribution that
the input word belonging to
the slots

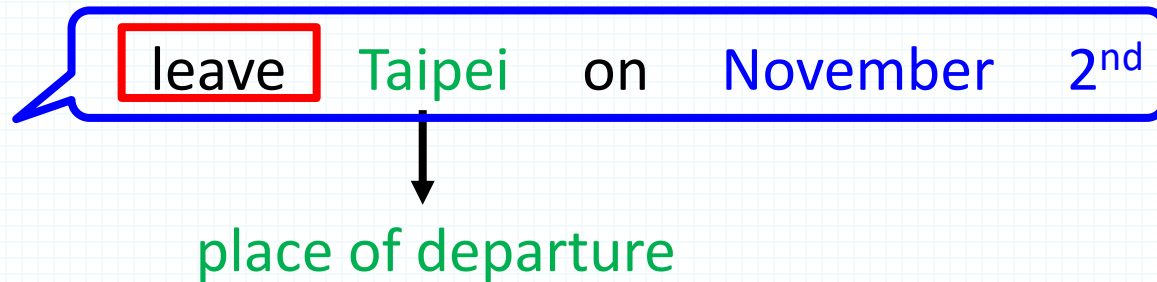
Taipei



Example Application

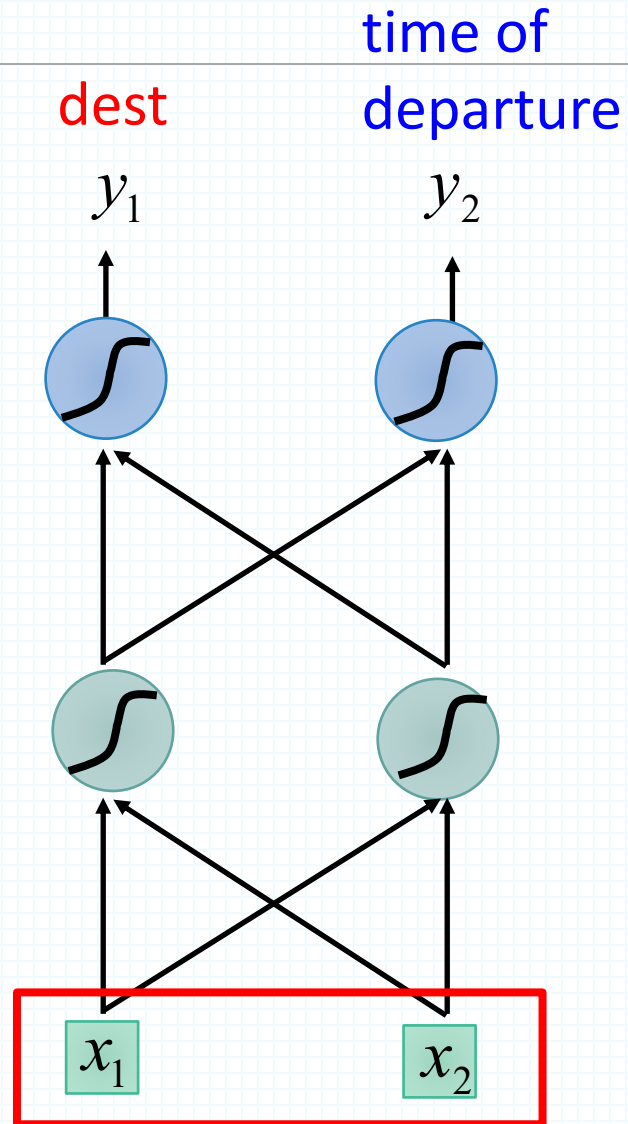


Problem?



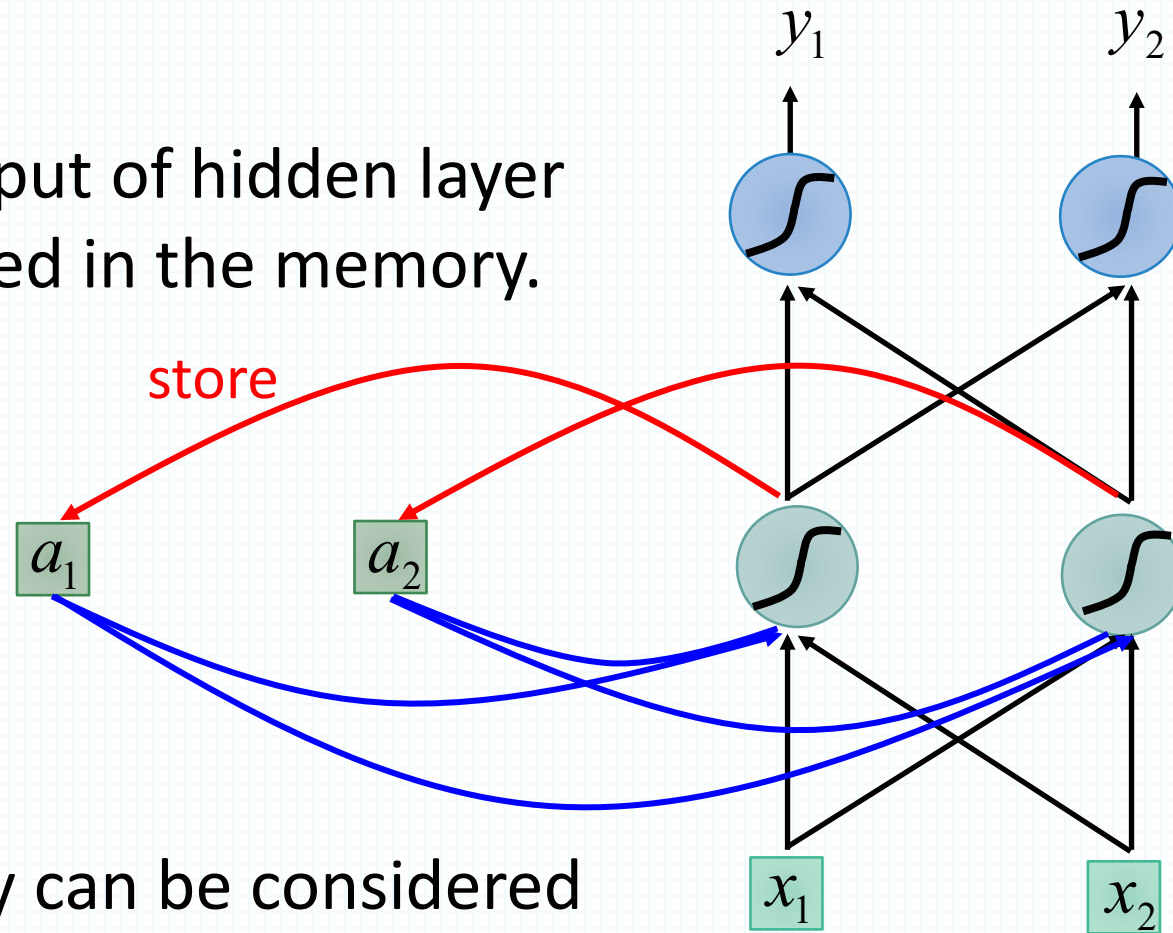
Neural network
needs memory!

Taipei



Recurrent Neural Network (RNN)

The output of hidden layer are stored in the memory.



Memory can be considered as another input.

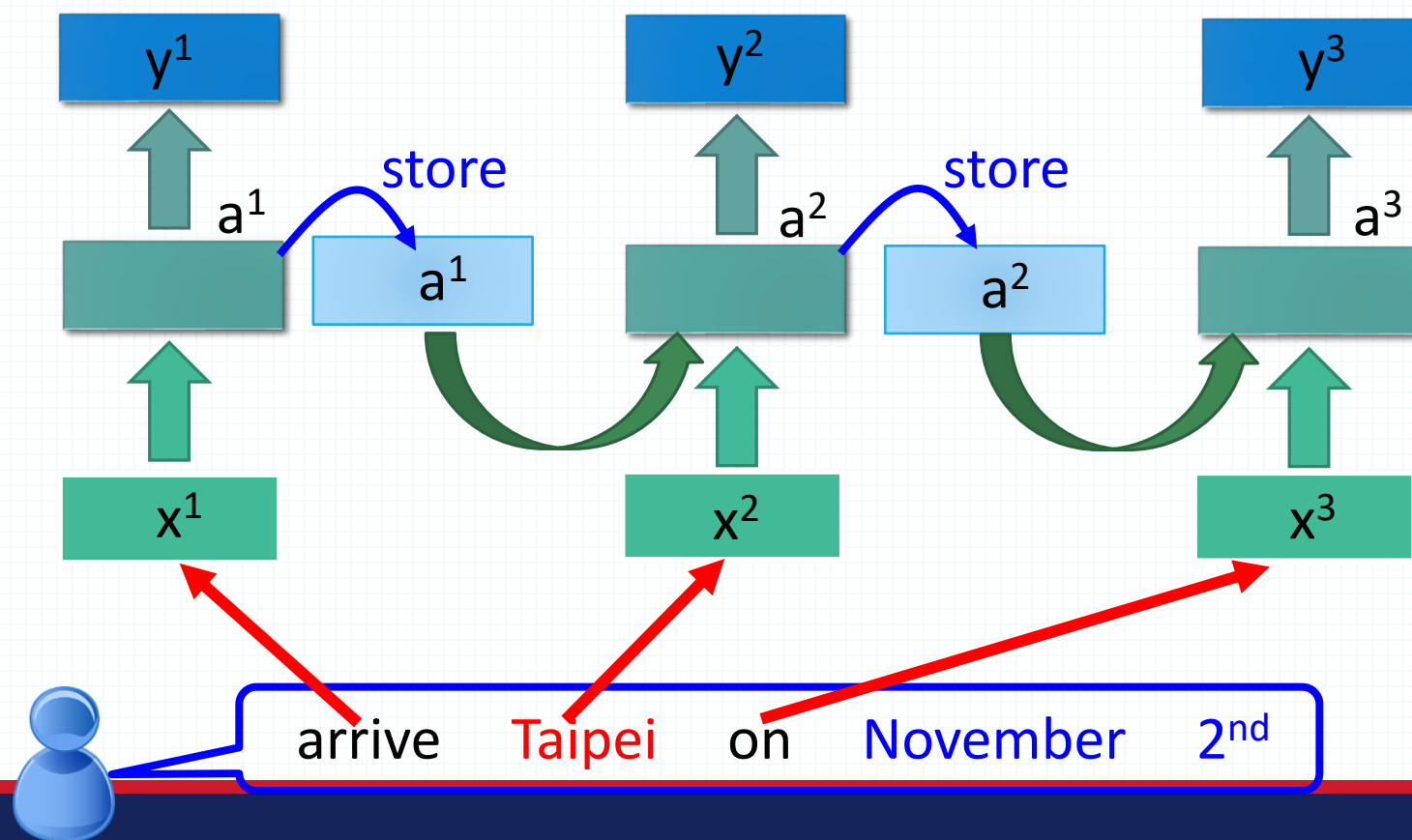
RNN

The same network is used again and again.

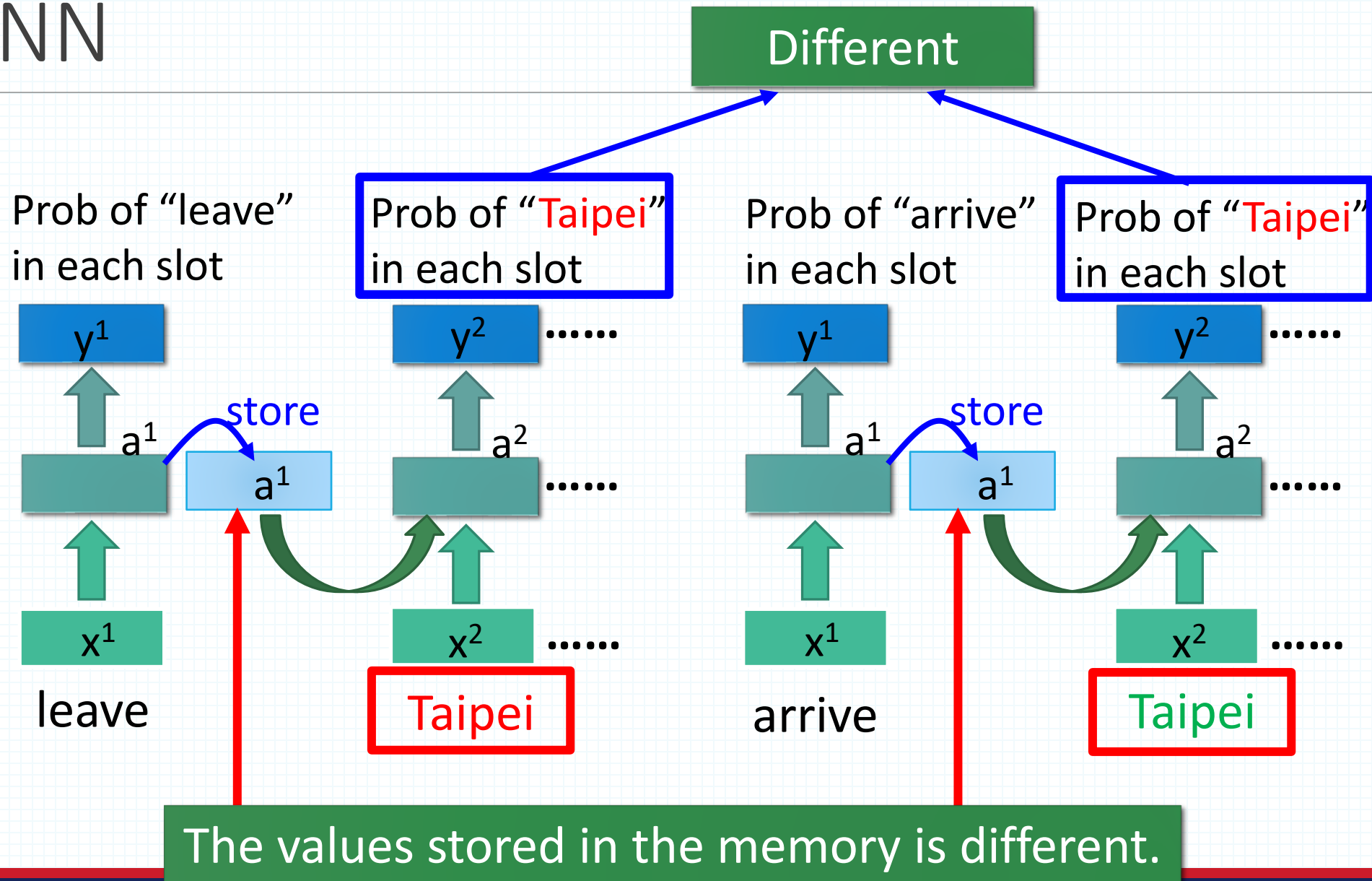
Probability of
“arrive” in each slot

Probability of
“**Taipei**” in each slot

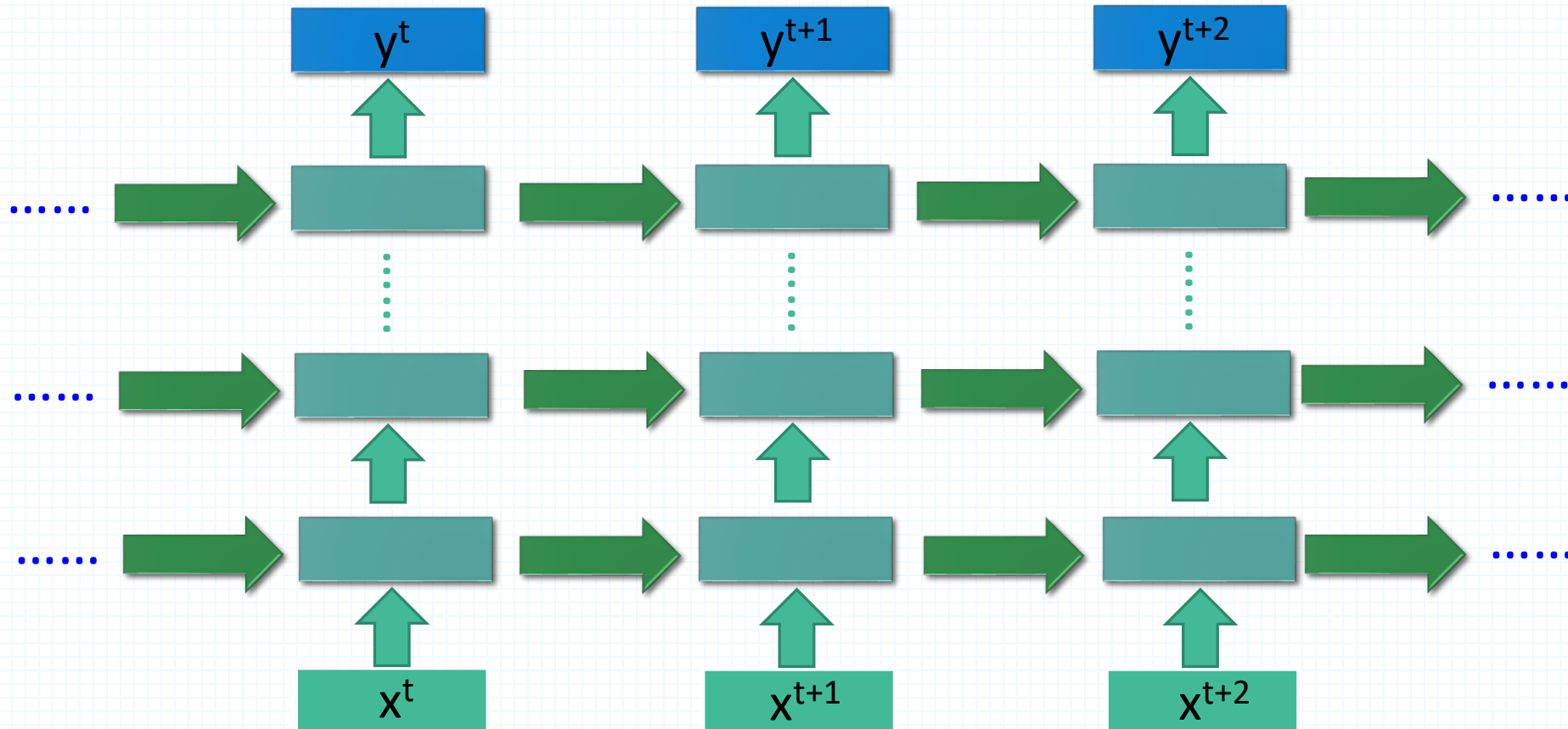
Probability of
“on” in each slot



RNN

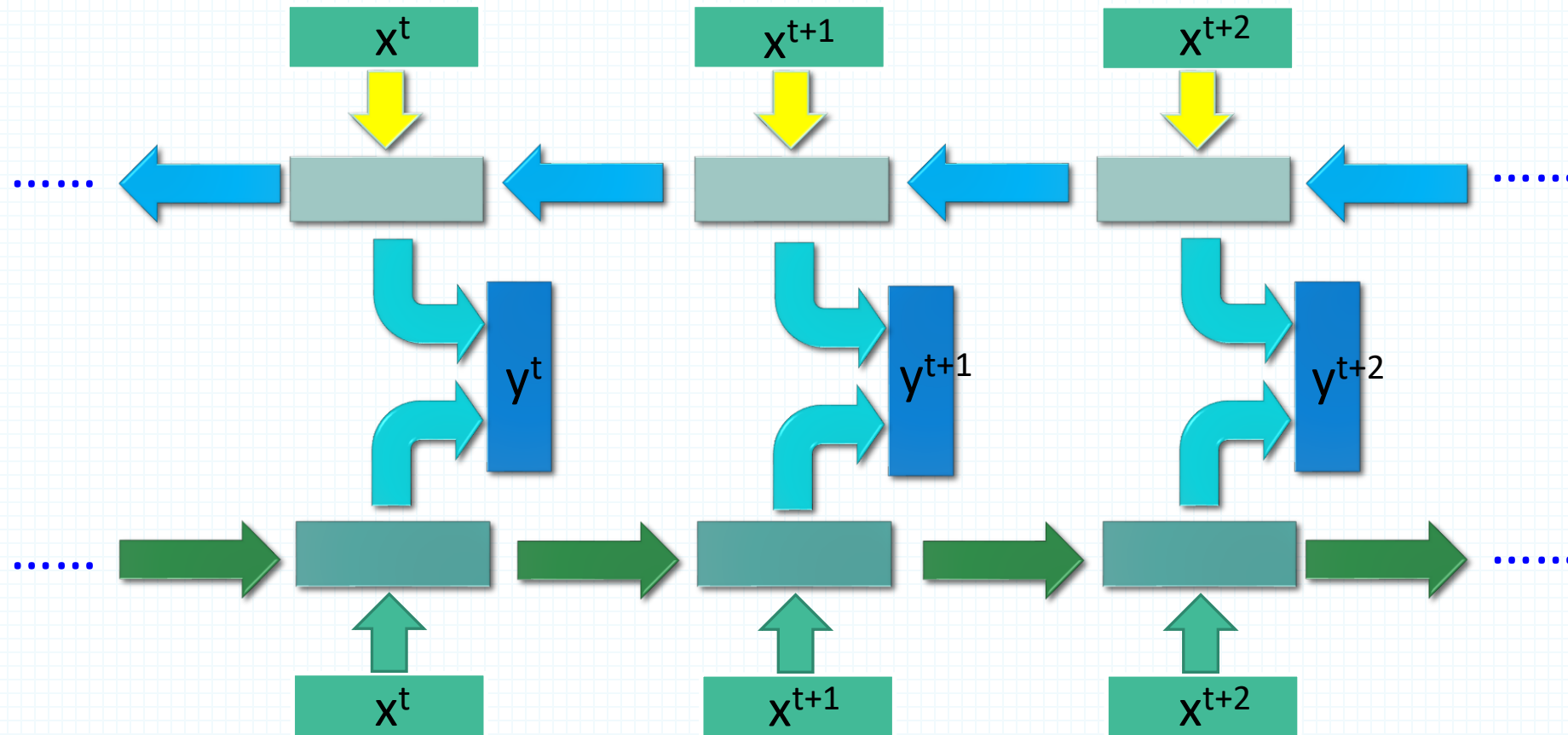


Of course it can be deep ...



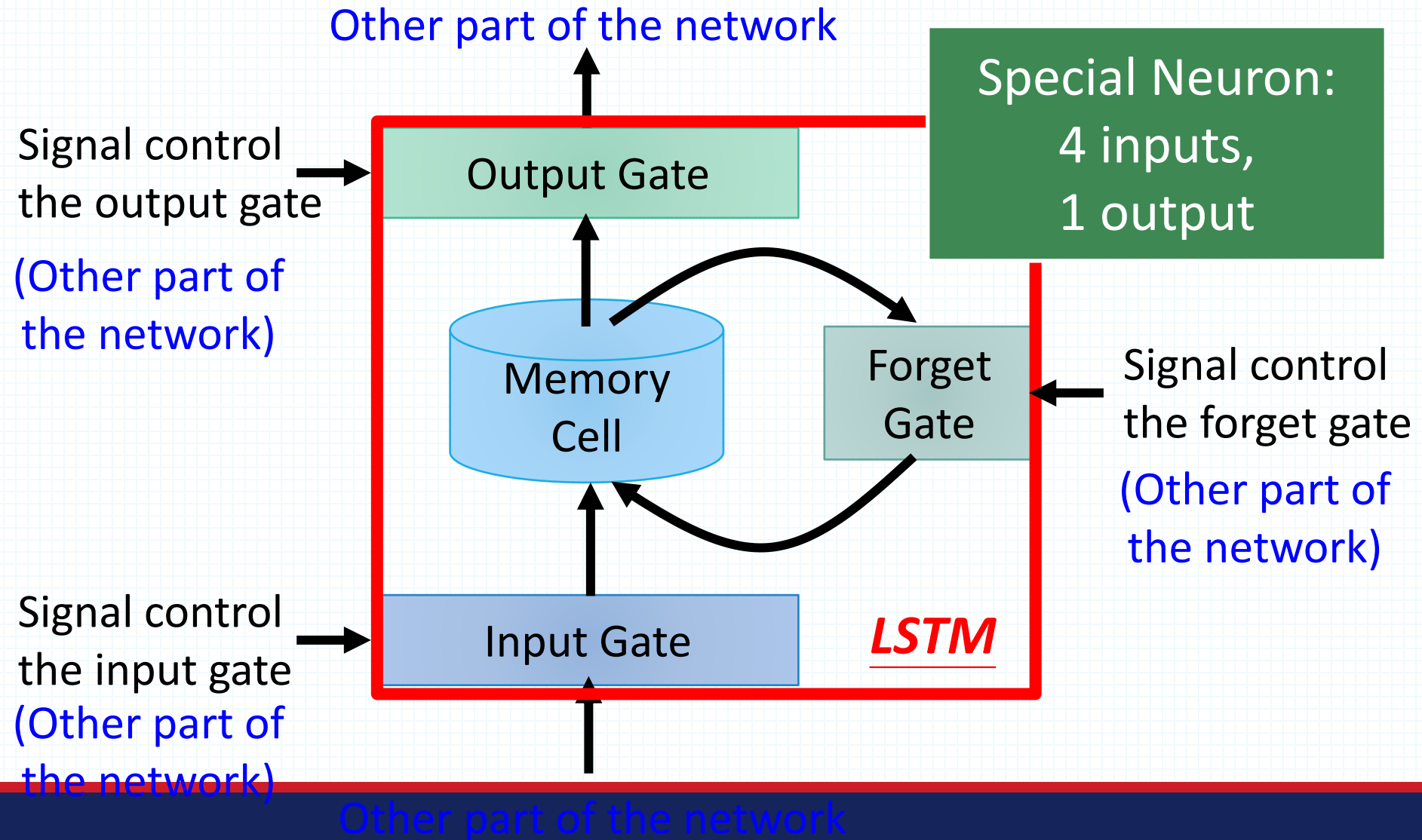
Bidirectional RNN

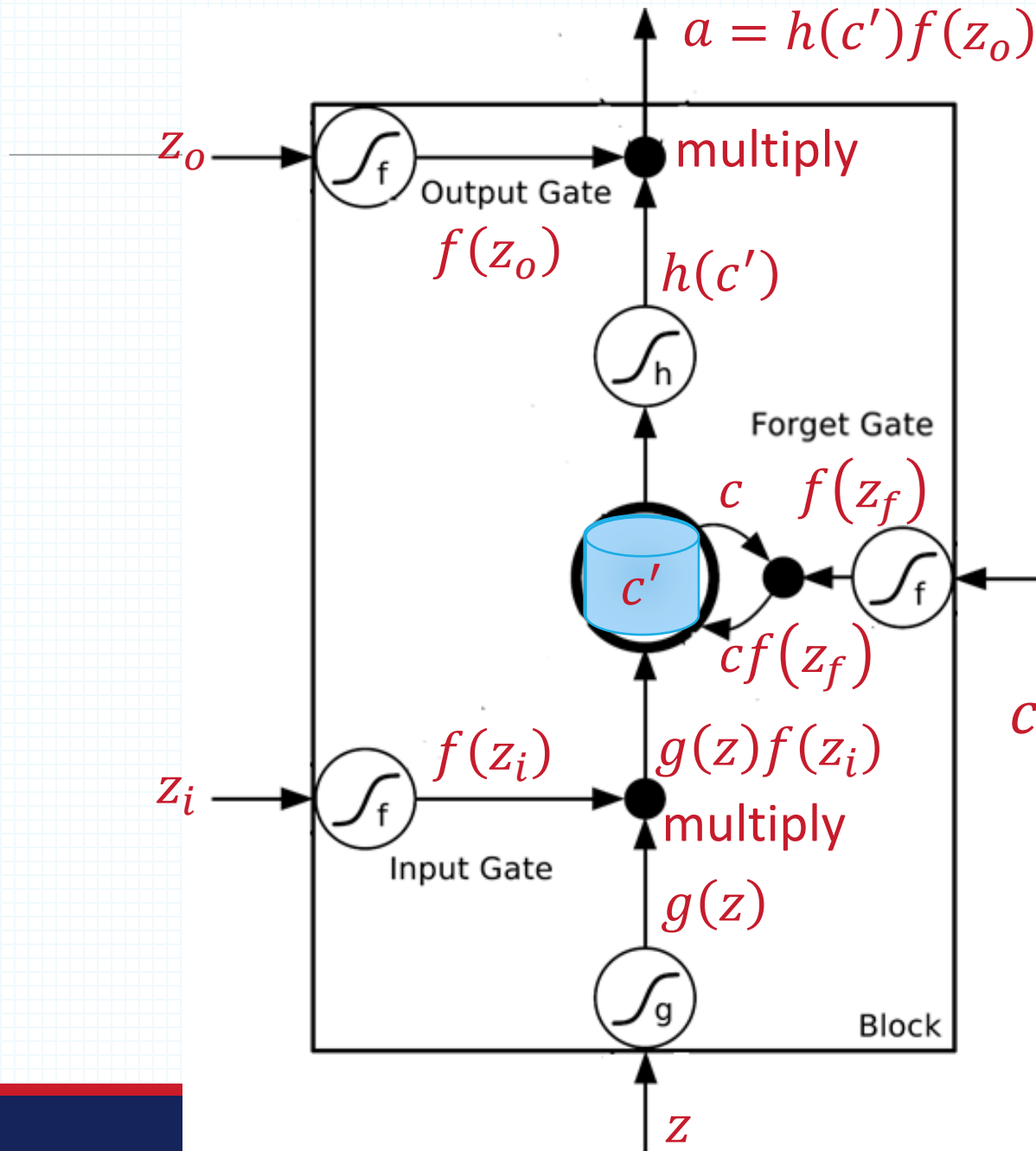
We can combine past and future evidence in separate chains



Long Short-term Memory (LSTM)

Long Short-term Memory (LSTM)





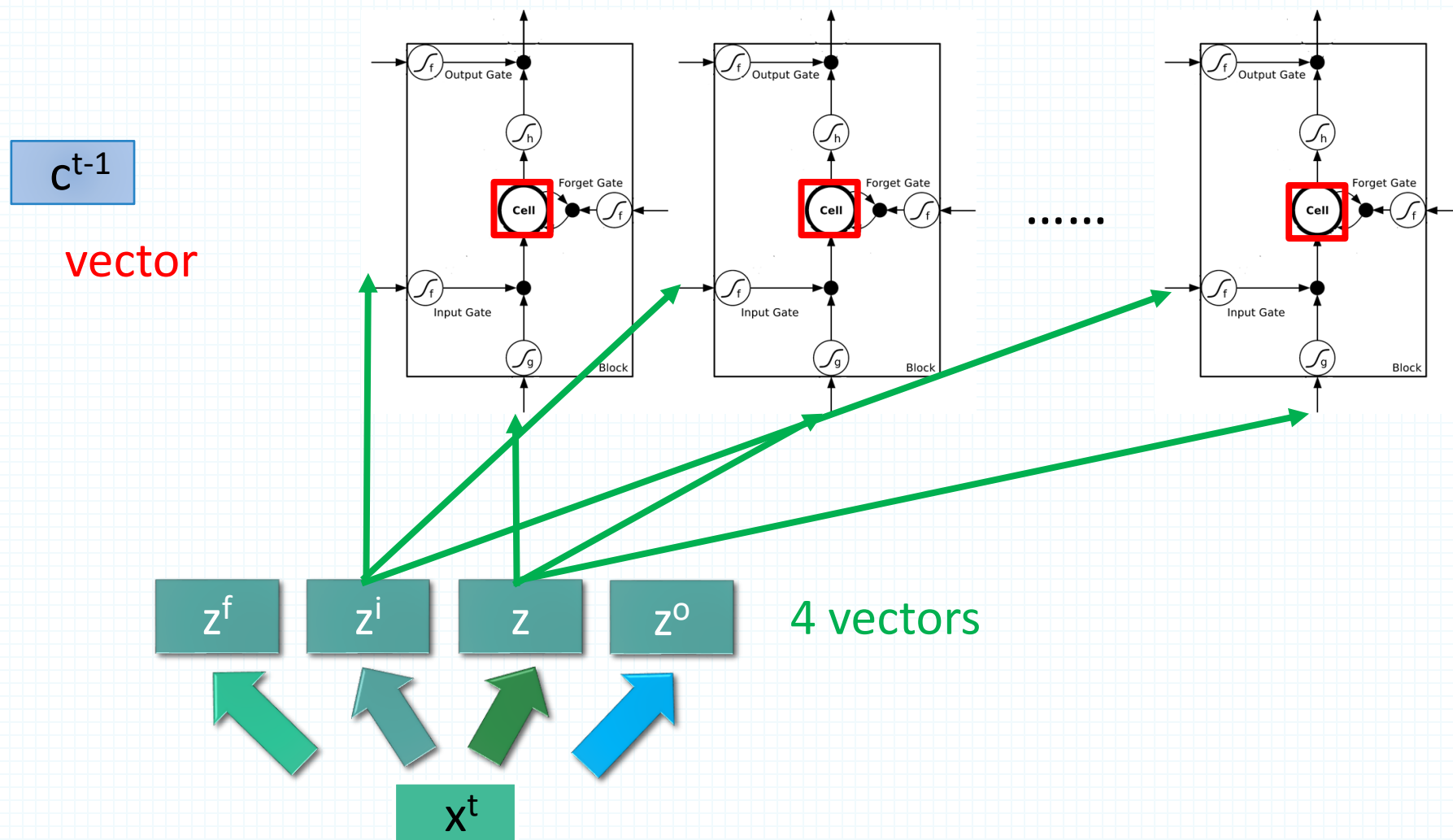
Activation function f is usually a sigmoid function

Between 0 and 1

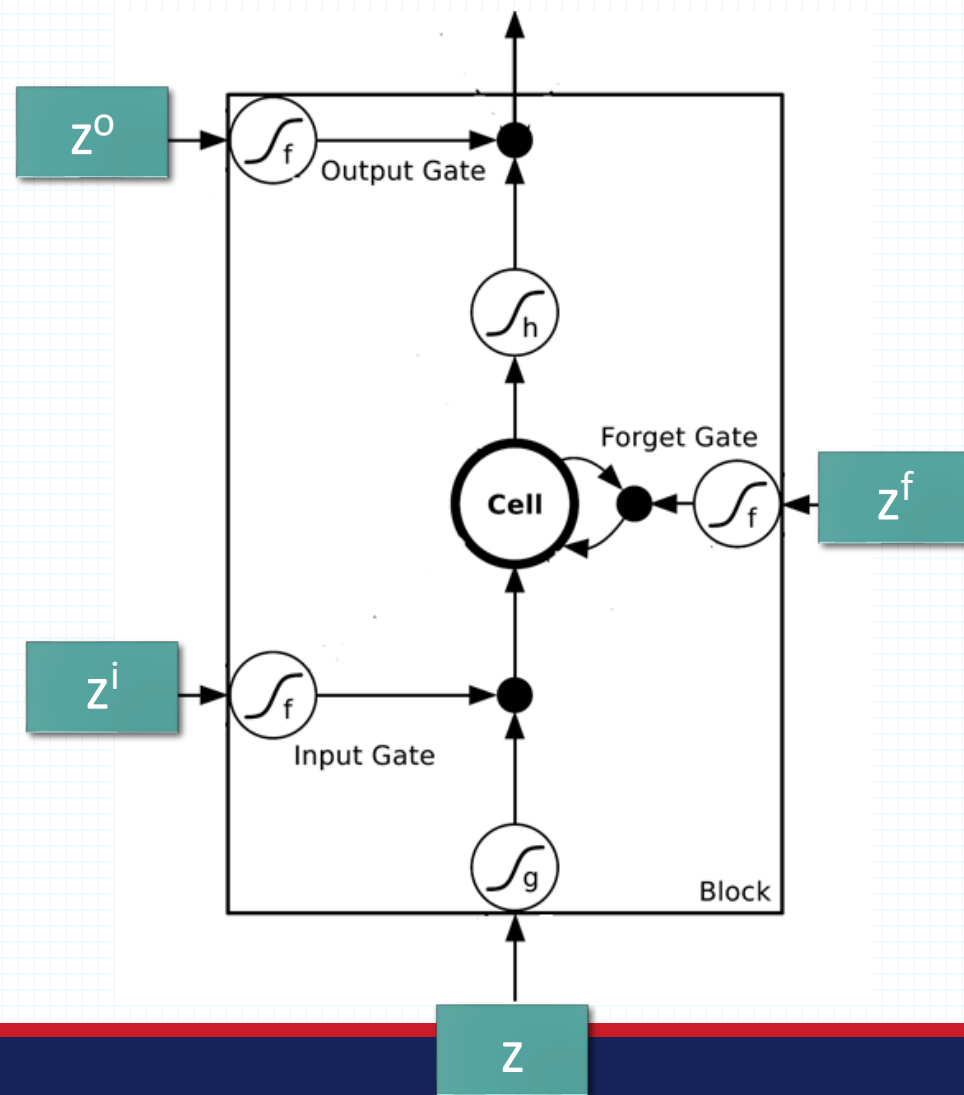
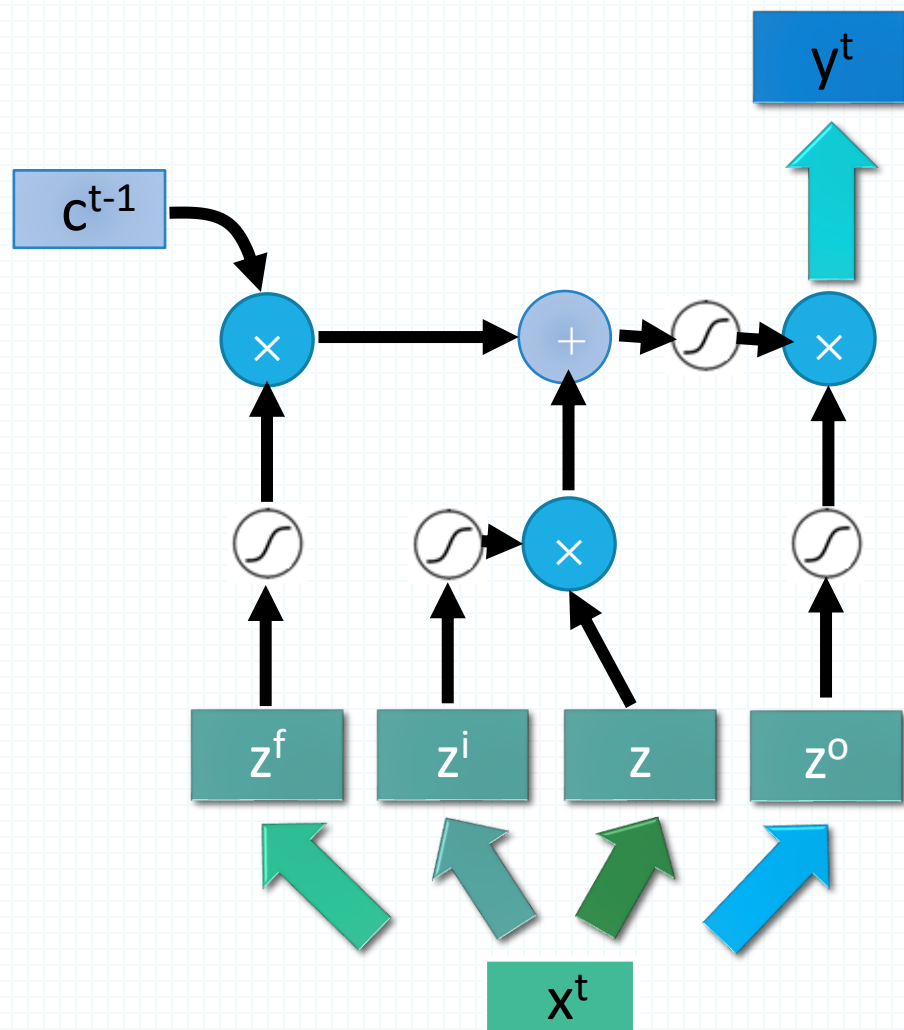
Mimic open and close gate

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

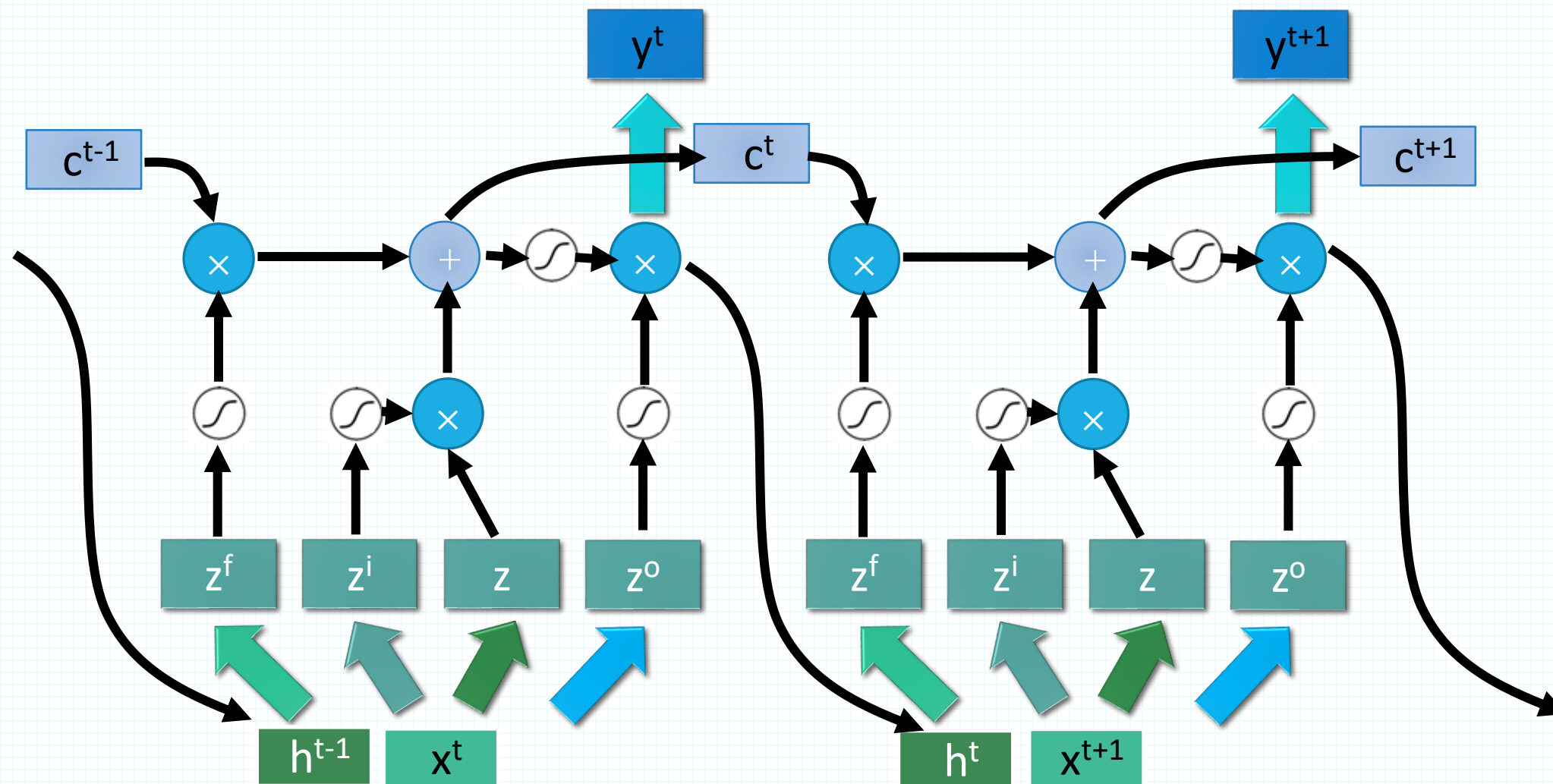
LSTM



LSTM



LSTM



Encoder-Decoder Model (Seq2Seq Model)

Encoder-Decoder Model

- $X^{(i)}$: i^{th} input
- $Y^{(i)}$: i^{th} output
- C : context (embedding)

Usage:

- Machine translation
- Question answering
- Dialog

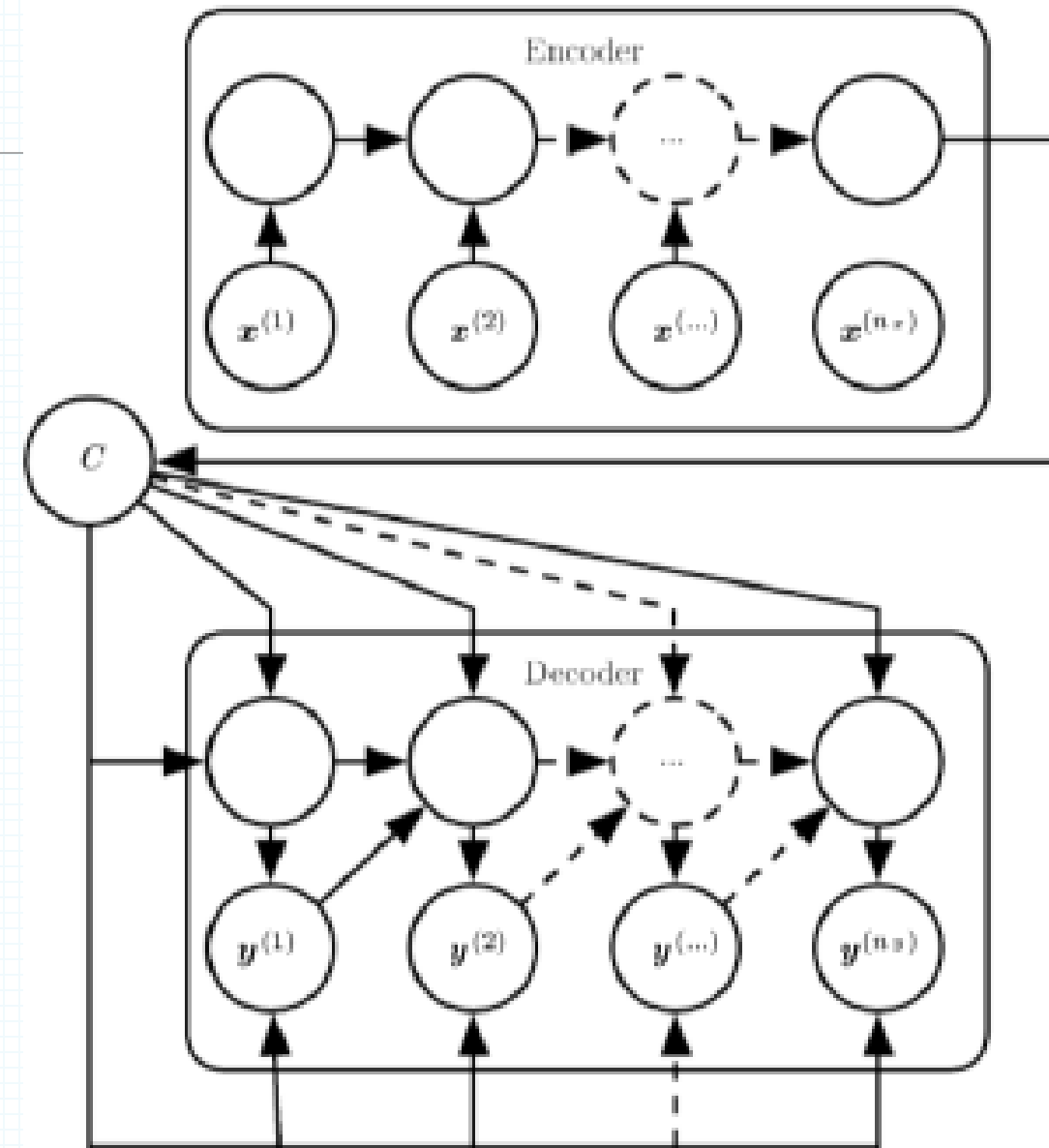


Image Caption Generation

- Input an image, but output a sequence of words

[Kelvin Xu, arXiv'15][Li Yao, ICCV'15]

