Sequence Generation

Outline

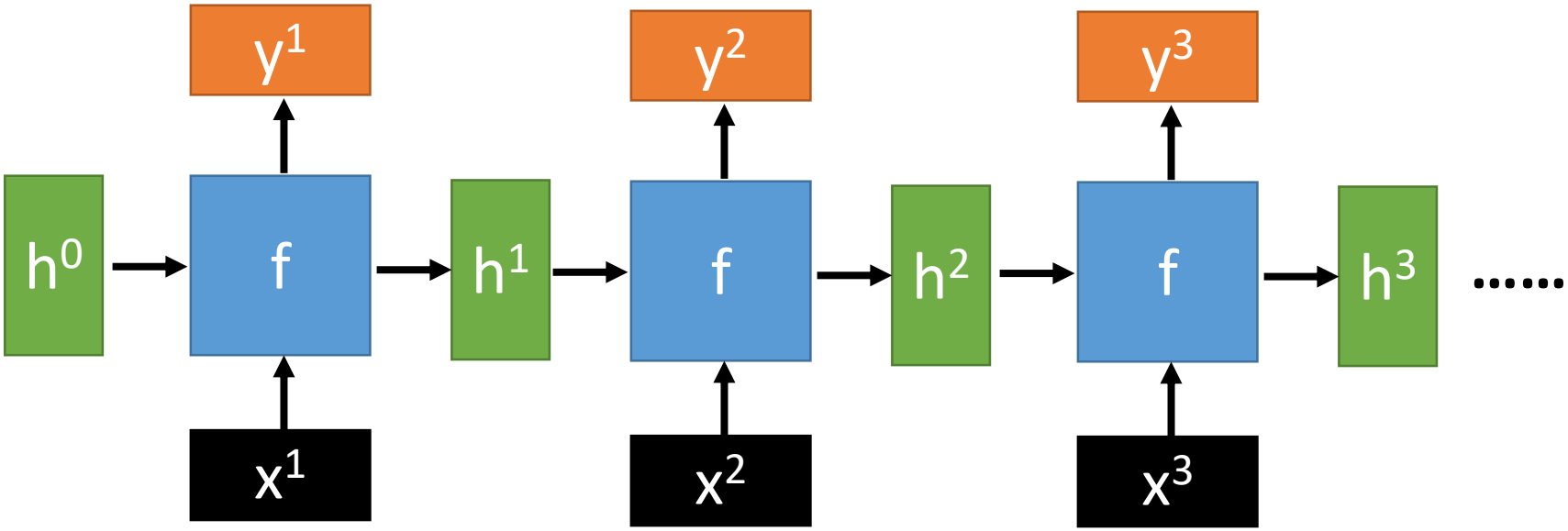
- RNN with Gated Mechanism
- Sequence Generation
- Conditional Sequence Generation

RNN with Gated Mechanism

Recurrent Neural Network

- Given function f: $h', y = f(h, x)$

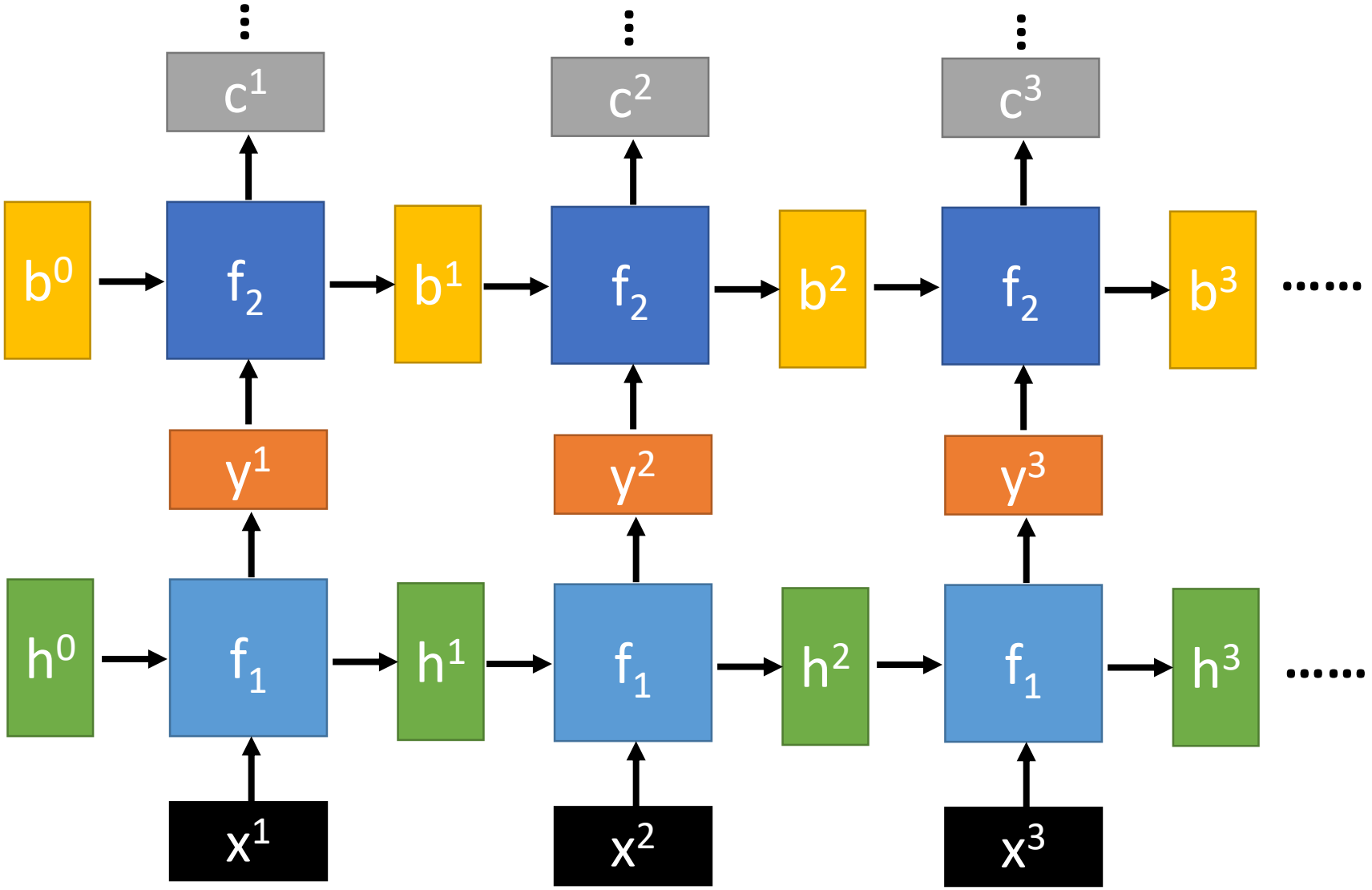
h and h' are vectors with the same dimension



No matter how long the input/output sequence is, we only need one function f

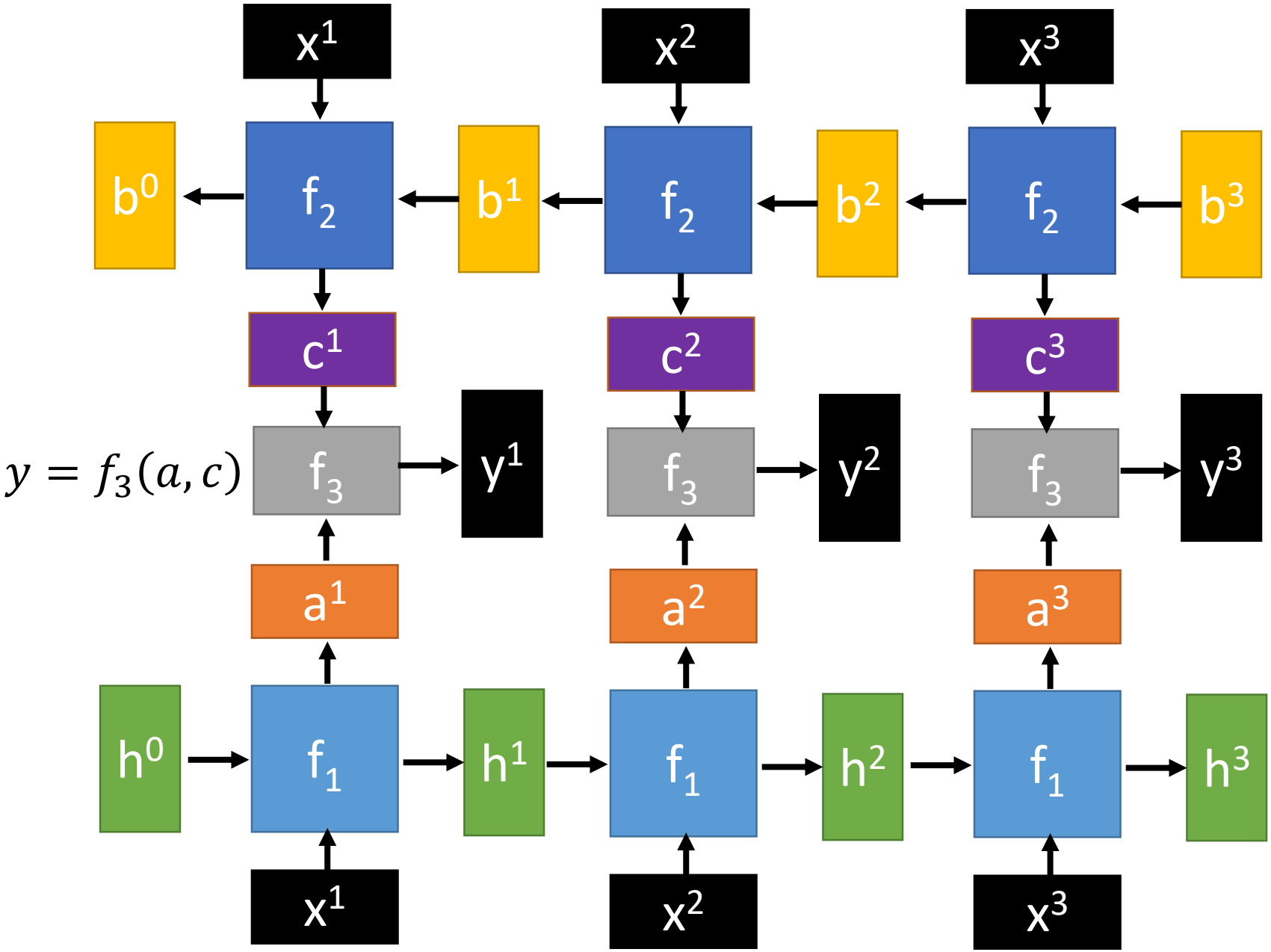
Deep RNN

$$h', y = f_1(h, x) \quad b', c = f_2(b, y) \quad \cdots$$



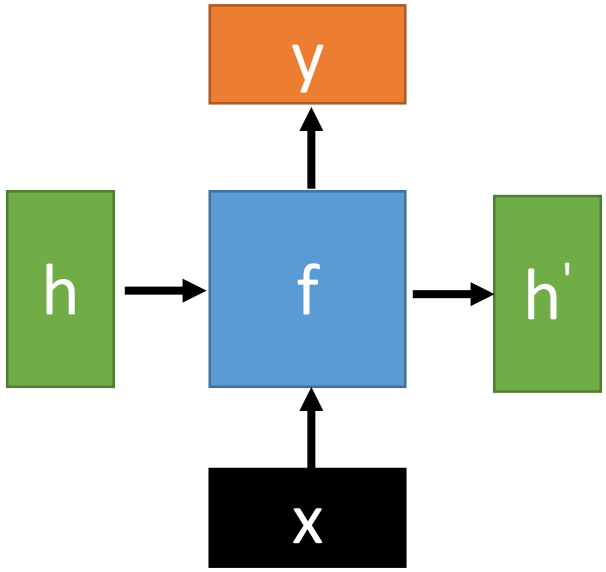
Bidirectional RNN

$h', a = f_1(h, x)$ $b', c = f_2(b, x)$



Naïve RNN

- Given function f: $h', y = f(h, x)$



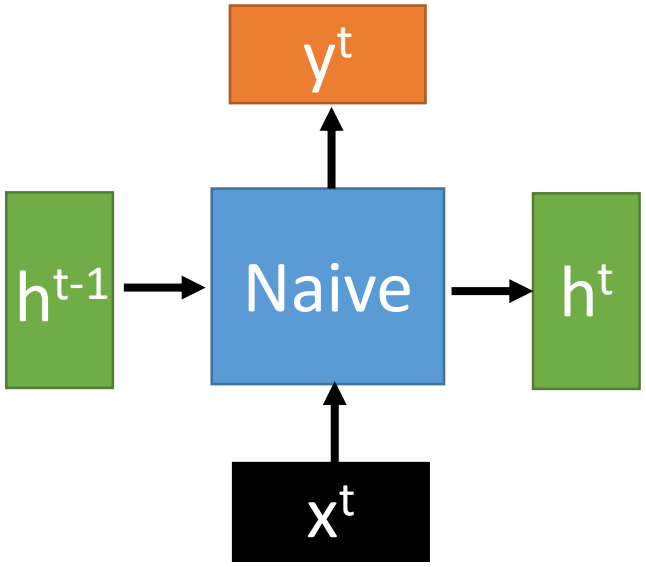
$$h' = \sigma\left(W^h h + W^i x \right)$$

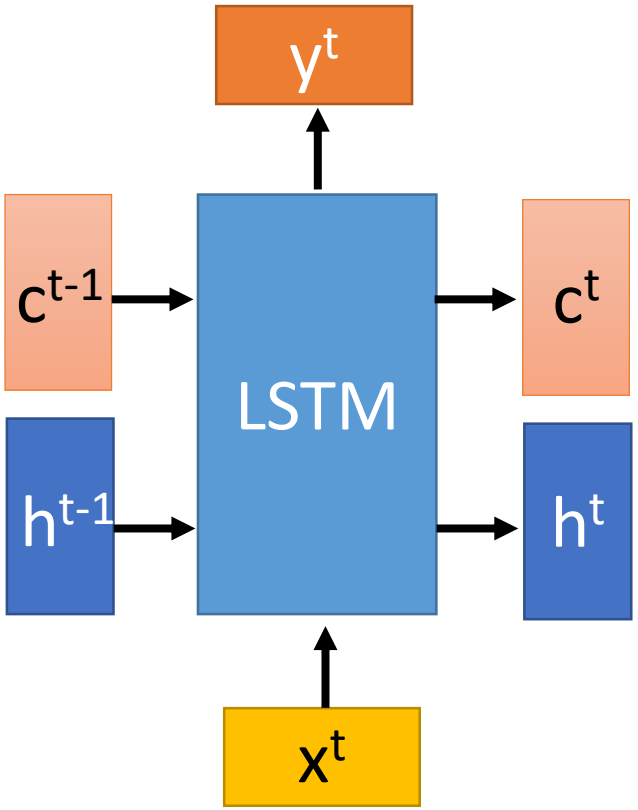
$$y = \sigma\left(W^o h' \right)$$

softmax

Ignore bias here

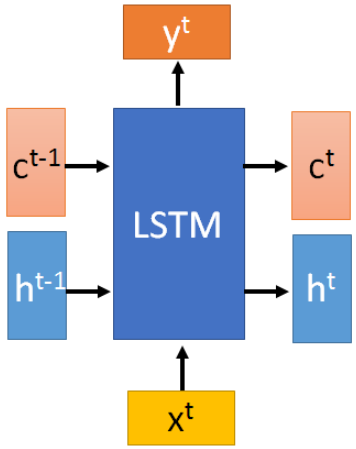
LSTM



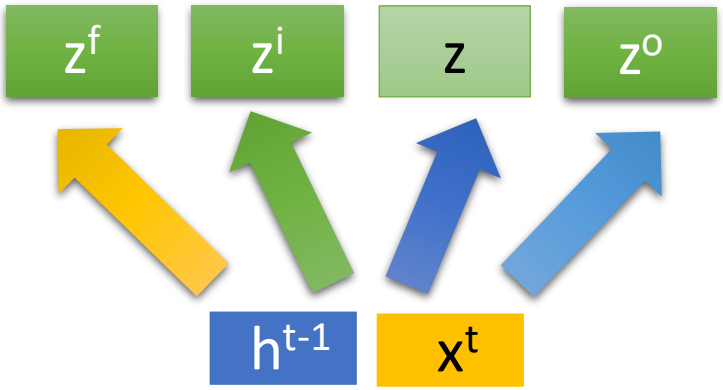


c changes slowly ➡ c^t is c^{t-1} added by something

h changes faster ➡ h^t and h^{t-1} can be very different



c^{t-1}



$$z = tanh\left(W \begin{matrix} x^t \\ h^{t-1} \end{matrix} \right)$$

$$z^i = \sigma\left(W^i \begin{matrix} x^t \\ h^{t-1} \end{matrix} \right)$$

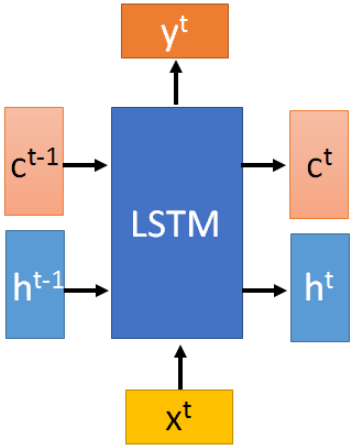
Input gate

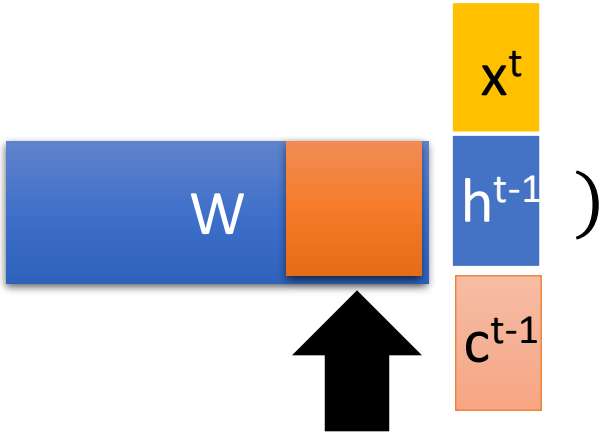
$$z^f = \sigma\left(W^f \begin{matrix} x^t \\ h^{t-1} \end{matrix} \right)$$

forget gate

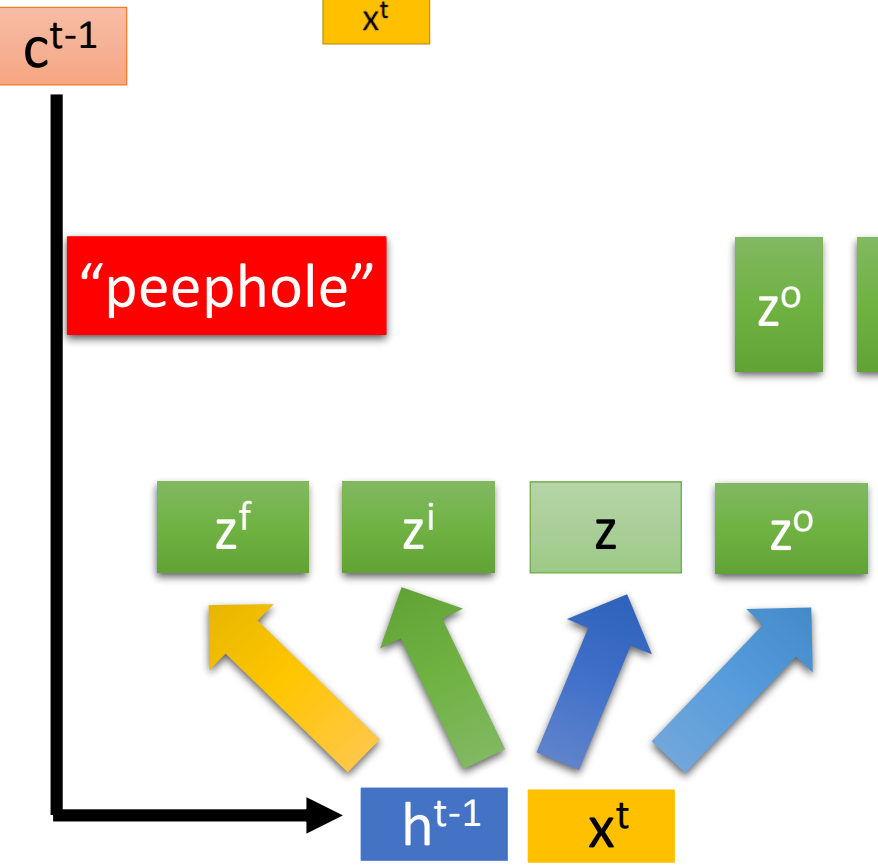
$$z^o = \sigma\left(W^o \begin{matrix} x^t \\ h^{t-1} \end{matrix} \right)$$

output gate

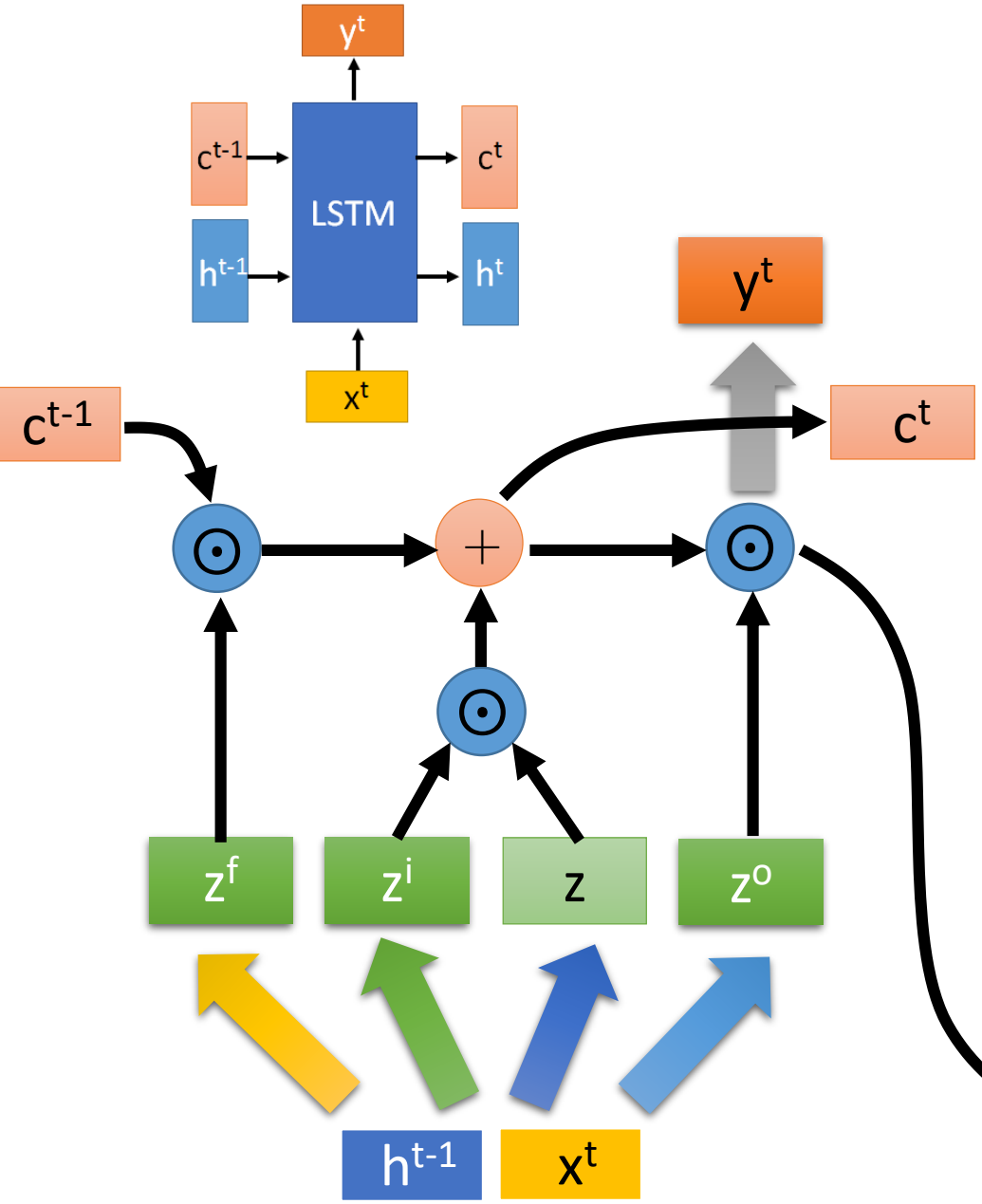


$z = tanh($  $)$

diagonal



z^o z^f z^i obtained by the same way

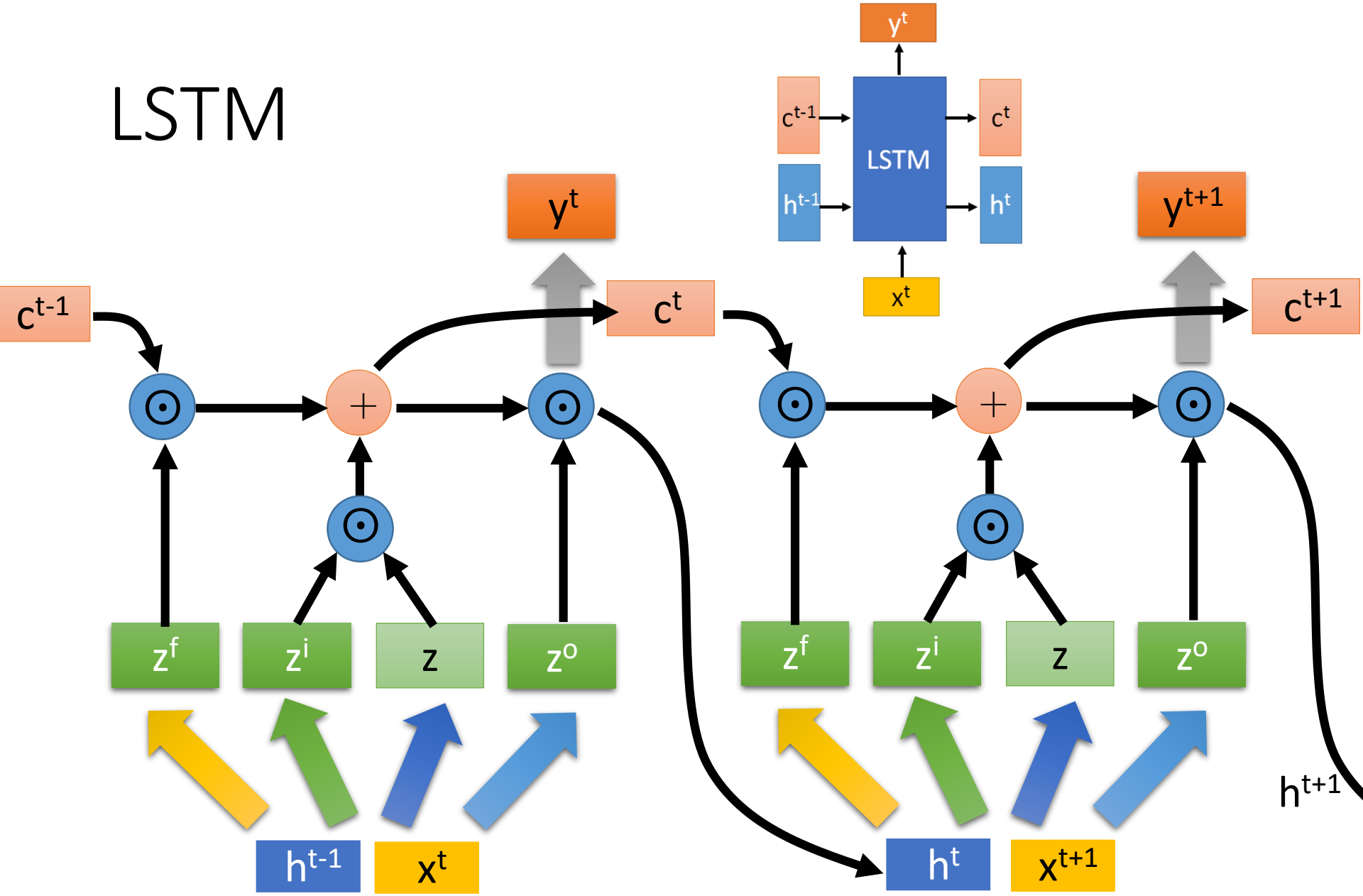


$$c^t = z^f \odot c^{t-1} + z^i \odot z$$

$$h^t = z^o \odot tanh(c^t)$$

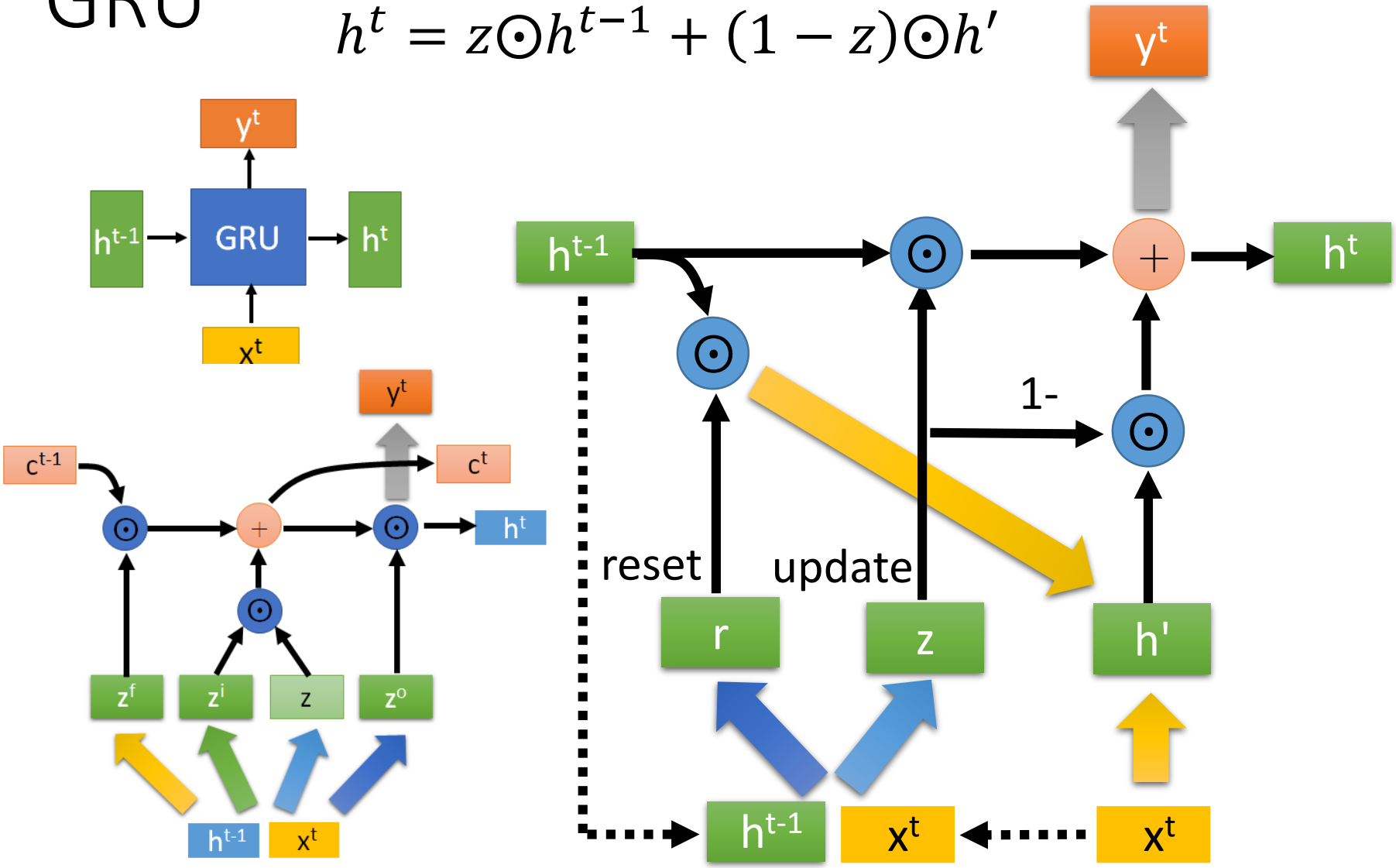
$$y^t = \sigma(W' h^t)$$

LSTM


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

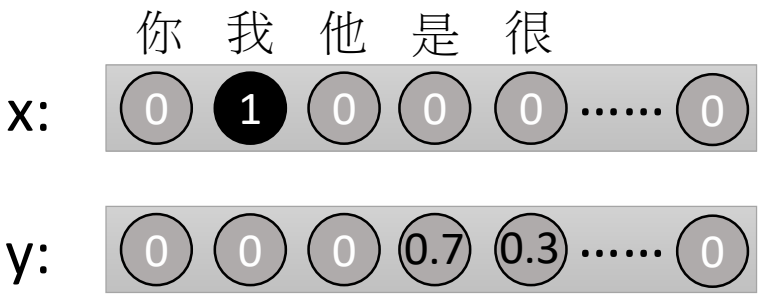
GRU

$$h^t = z \odot h^{t-1} + (1 - z) \odot h'$$

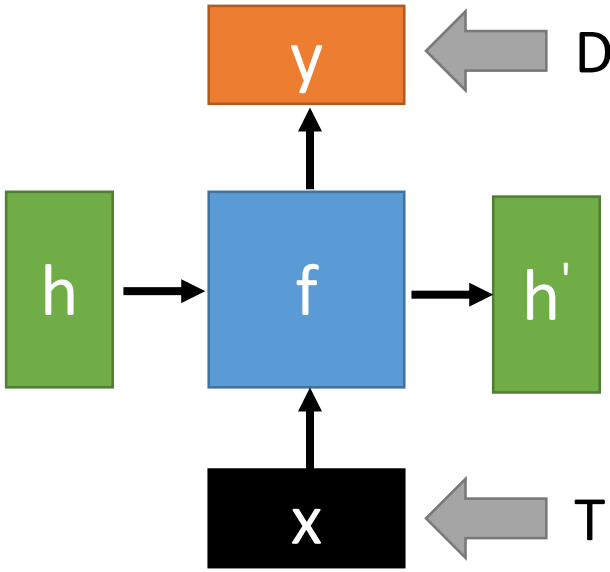


Sequence Generation

Generation



- Sentences are composed of characters/words
- Generating a character/word at each time by RNN



Distribution over the token

(sampling from the distribution to generate a token)

The token generated at the last time step.

(represented by 1-of-N encoding)

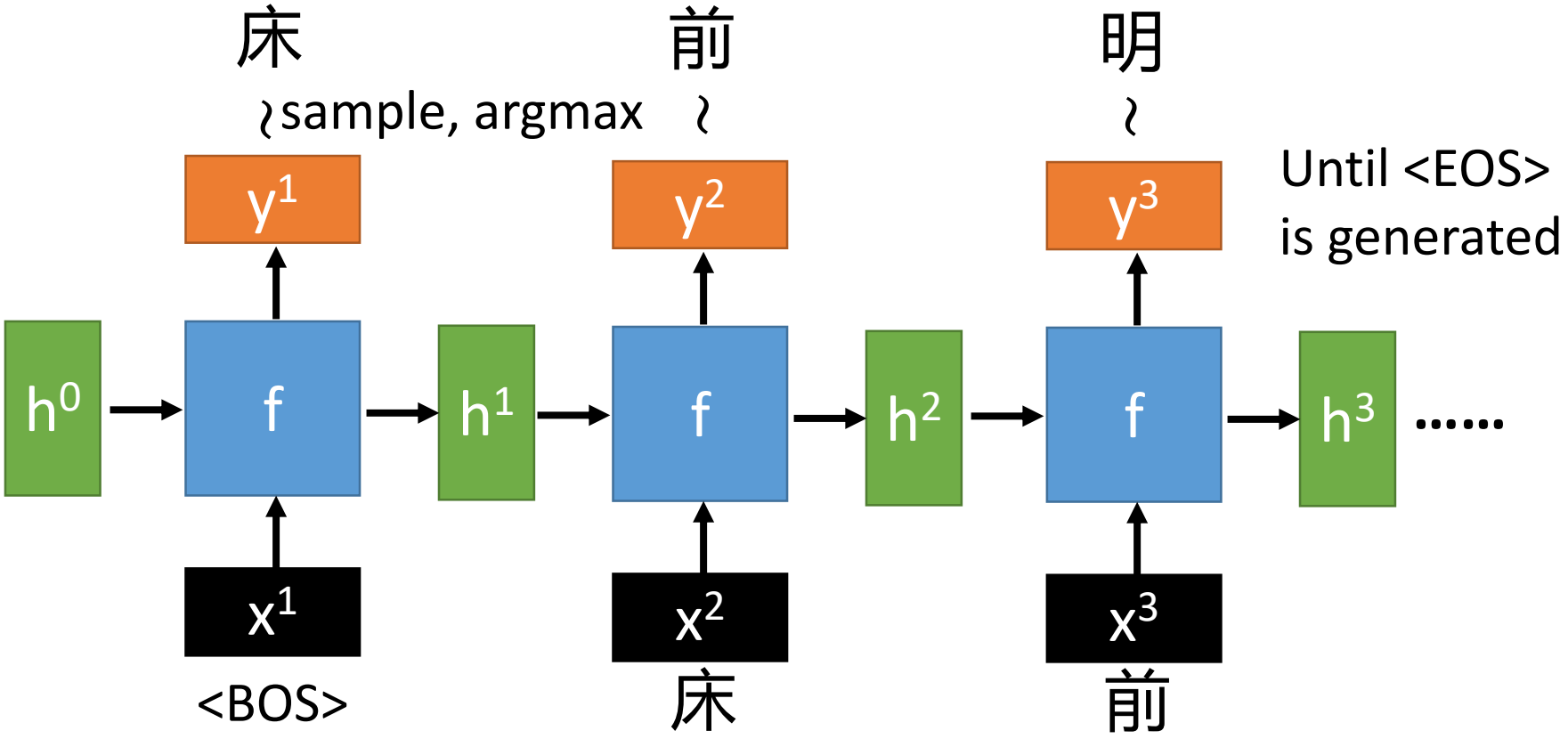
Generation

y^1: P(w|<BOS>)

y^2: P(w|<BOS>,床)

y^3: P(w|<BOS>,床,前)

- Sentences are composed of characters/words
- Generating a character/word at each time by RNN

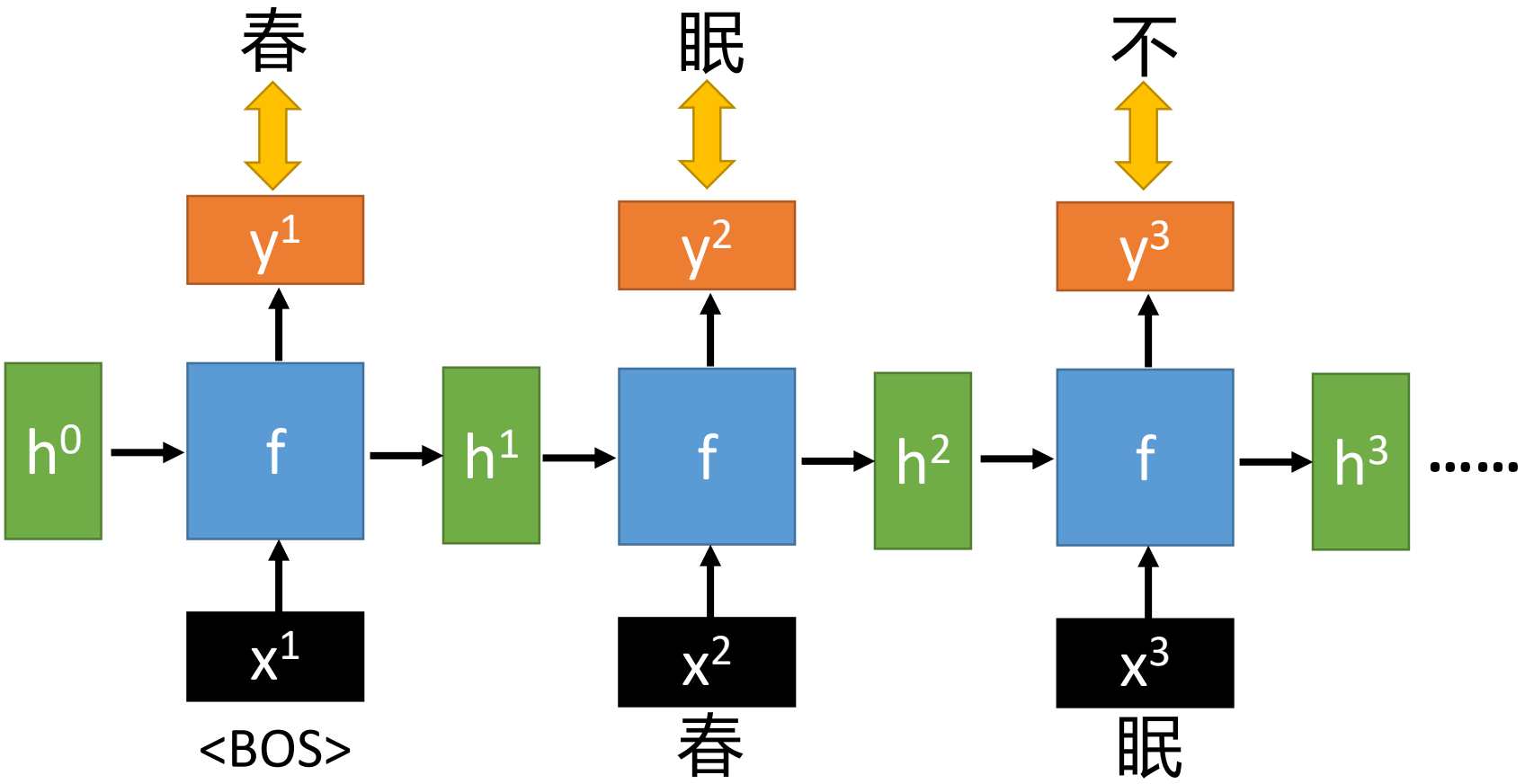


Generation

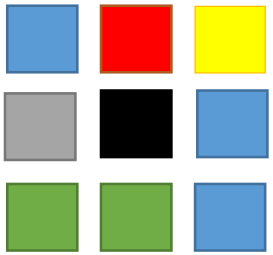
: minimizing cross-entropy

- ***Training***

Training data: 春眠不觉晓



Generation

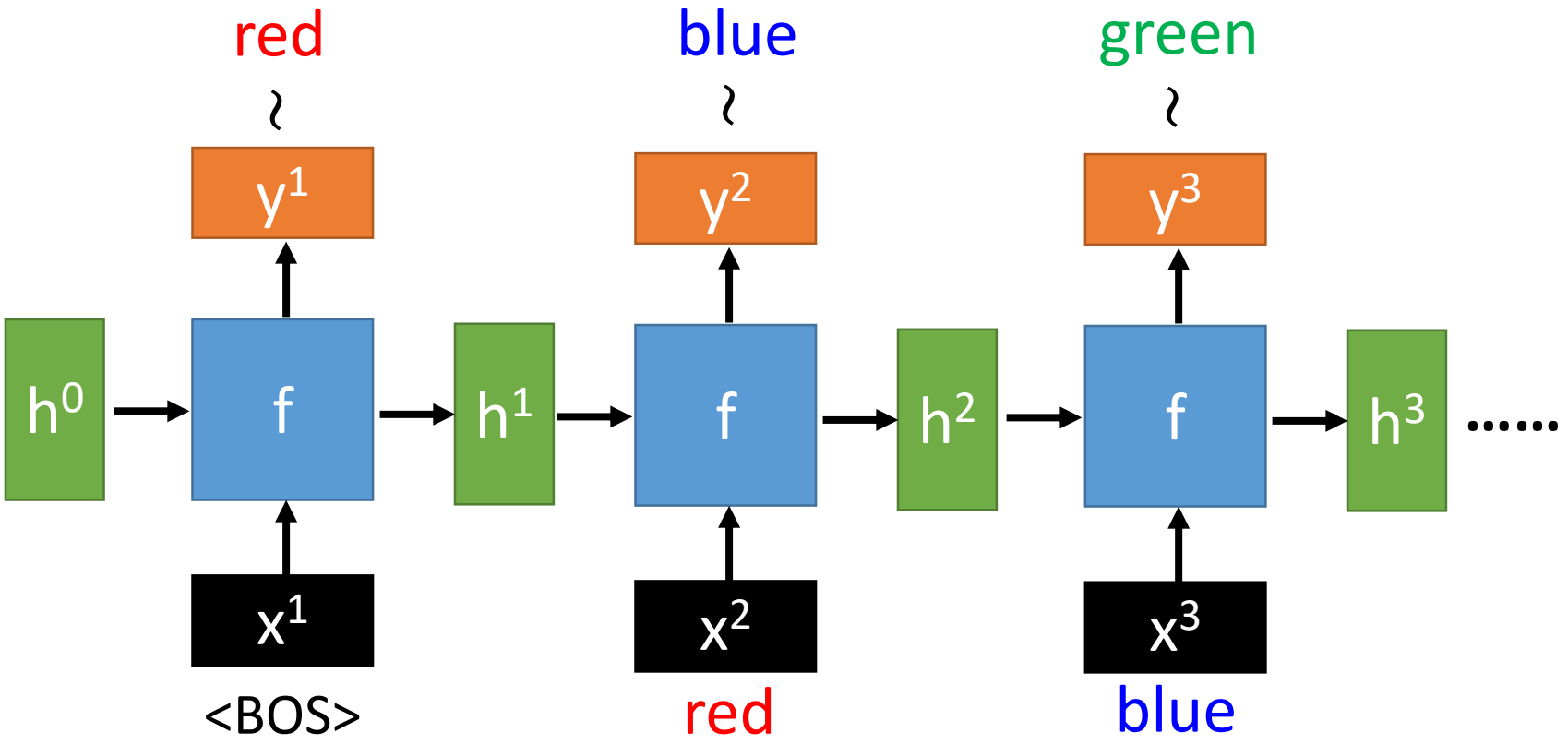


Consider as a sentence

blue red yellow gray ……

Train a RNN based on the "sentences"

- Images are composed of pixels
- Generating a pixel at each time by RNN



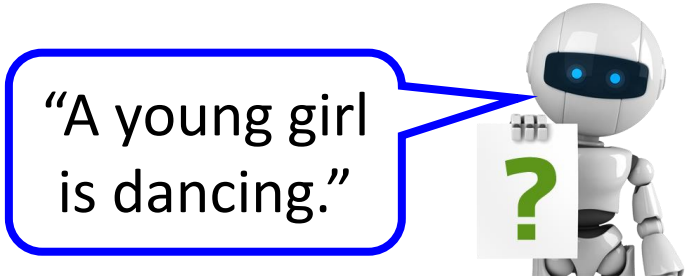
Conditional Sequence Generation

Conditional Generation

- We don't want to simply generate some random sentences.
- Generate sentences based on conditions:

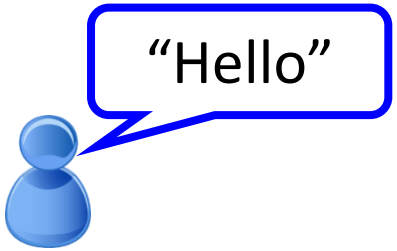
Caption Generation

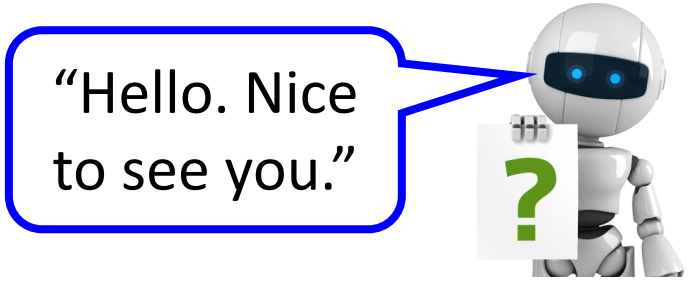
Given condition:



Chat-bot

Given condition:

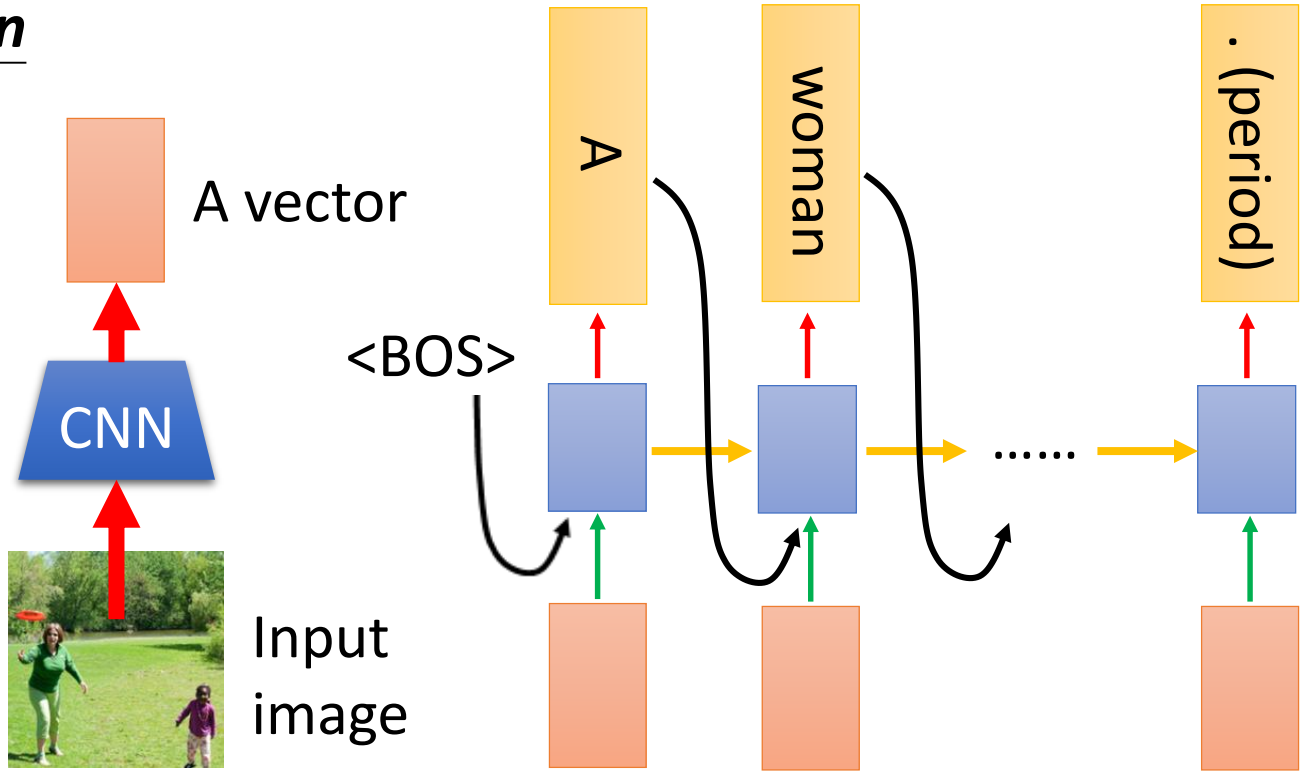




Conditional Generation

- Represent the input condition as a vector, and consider the vector as the input of RNN generator

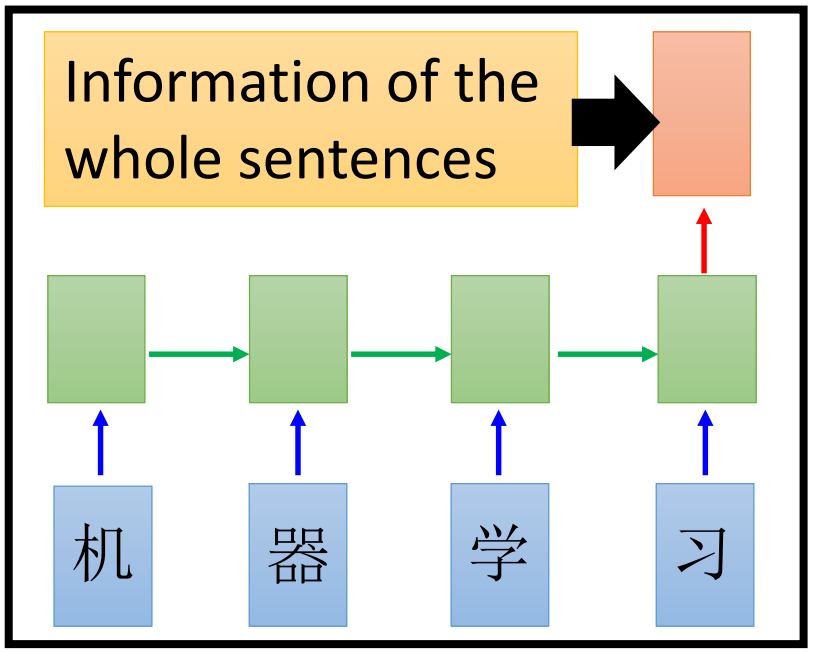
Image Caption Generation

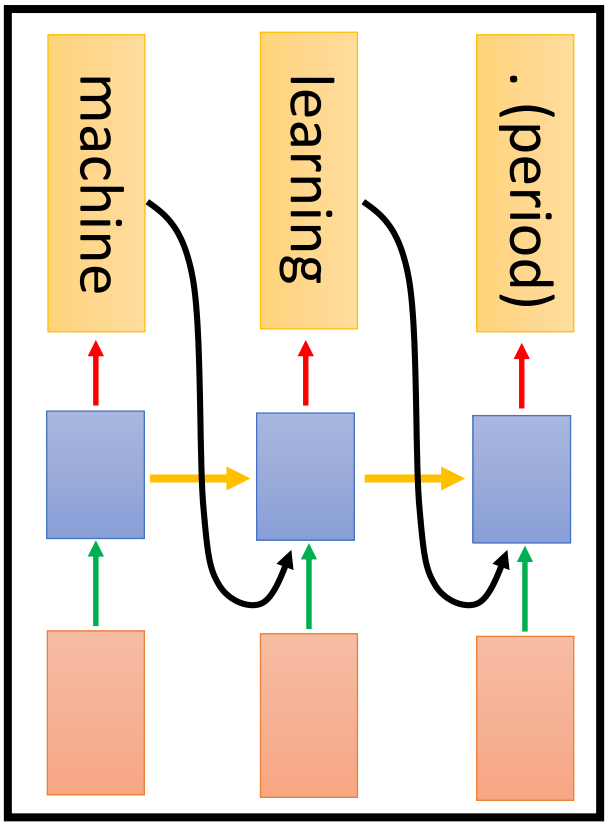


Conditional Generation

Sequence-to-sequence learning

- Represent the input condition as a vector, and consider the vector as the input of RNN generator
- E.g. Machine translation / Chat-bot





Encoder ← Jointly train → ***Decoder***

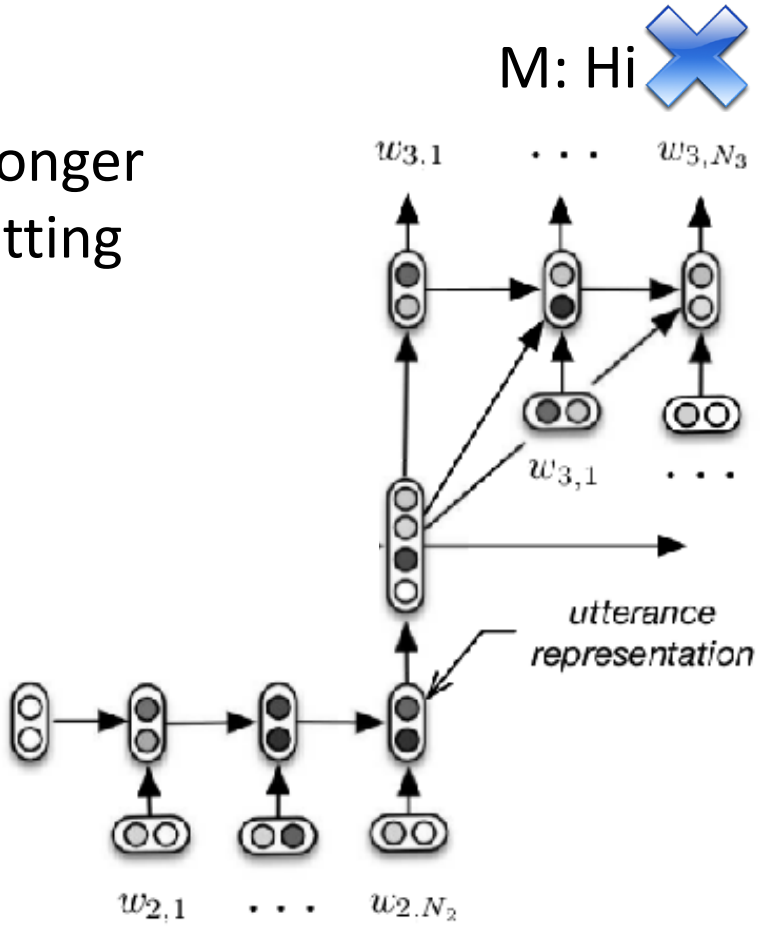
Conditional Generation

M: Hello

U: Hi

M: Hi

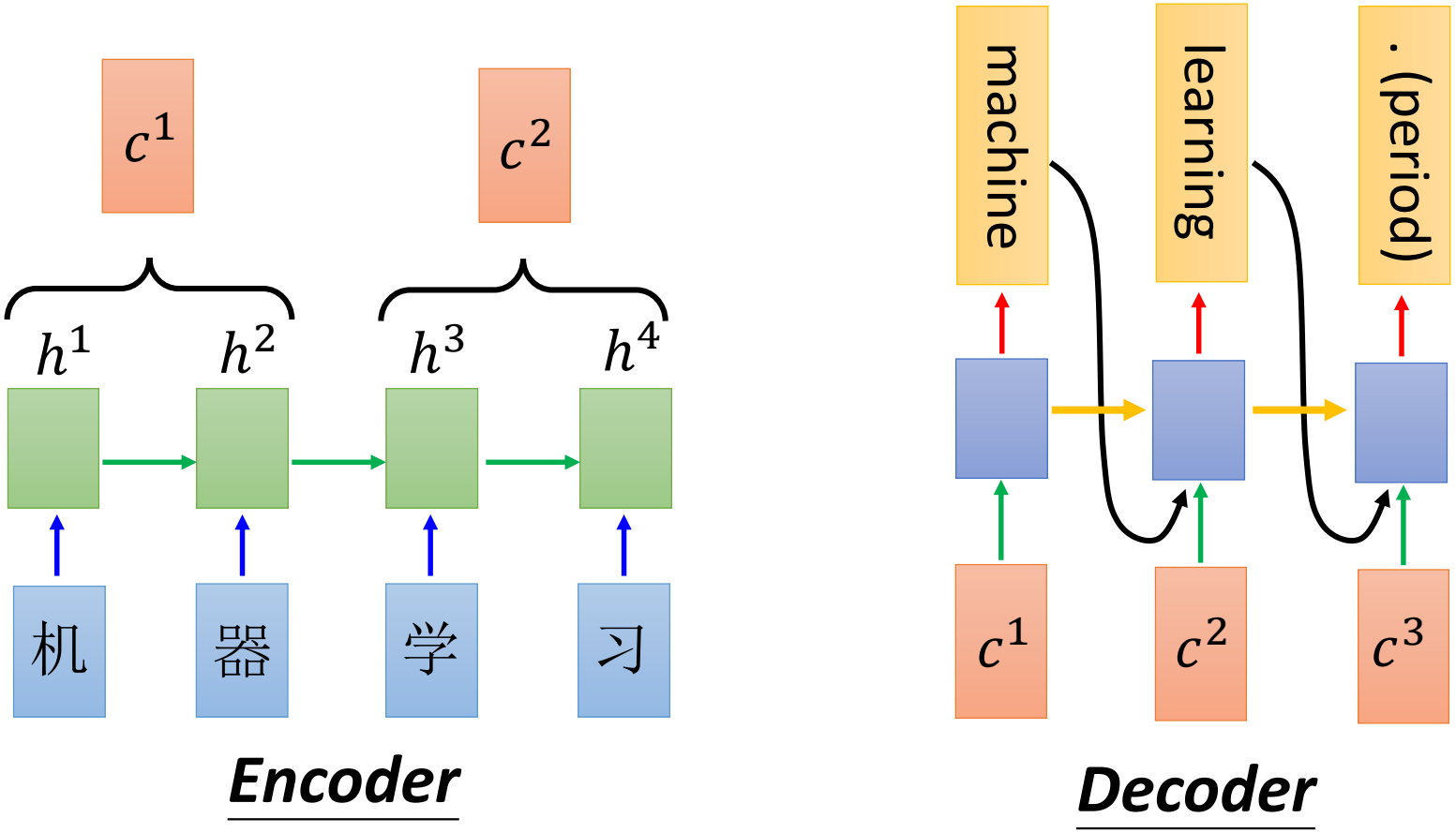
Need to consider longer context during chatting



M: Hello

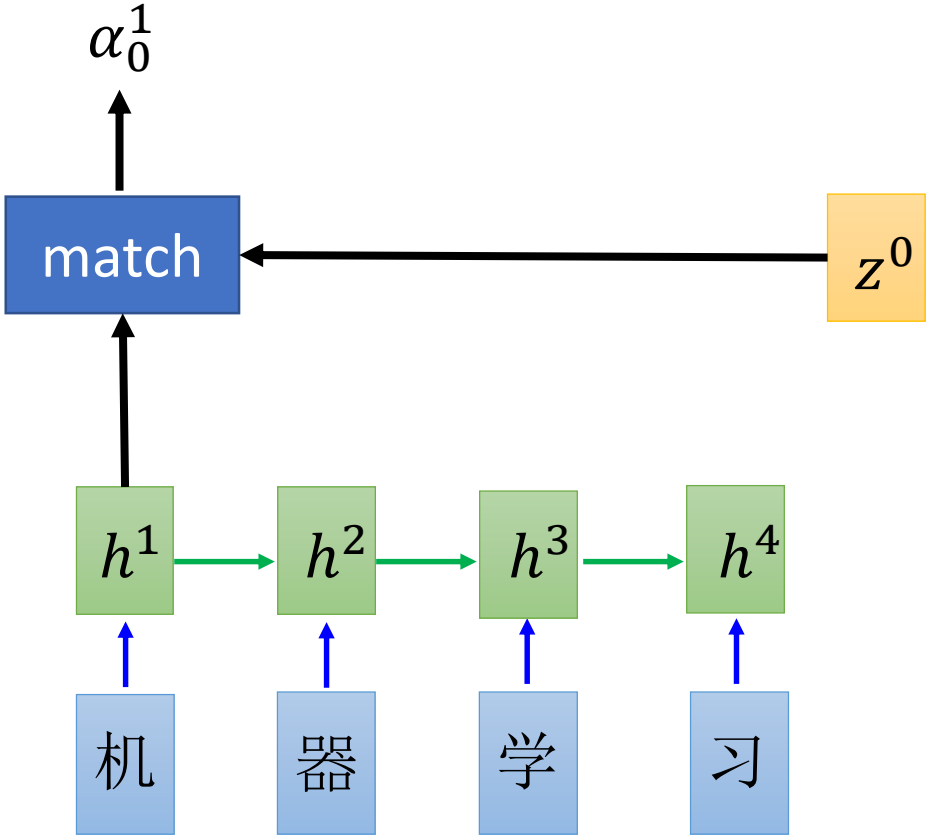
Serban, Iulian V., Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau, 2015 "Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models.

Dynamic Conditional Generation

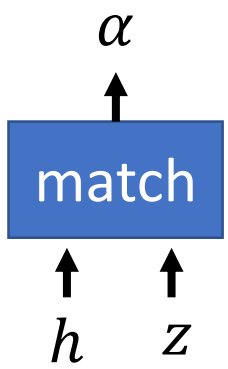


Machine Translation

- Attention-based model



Jointly learned with other part of the network



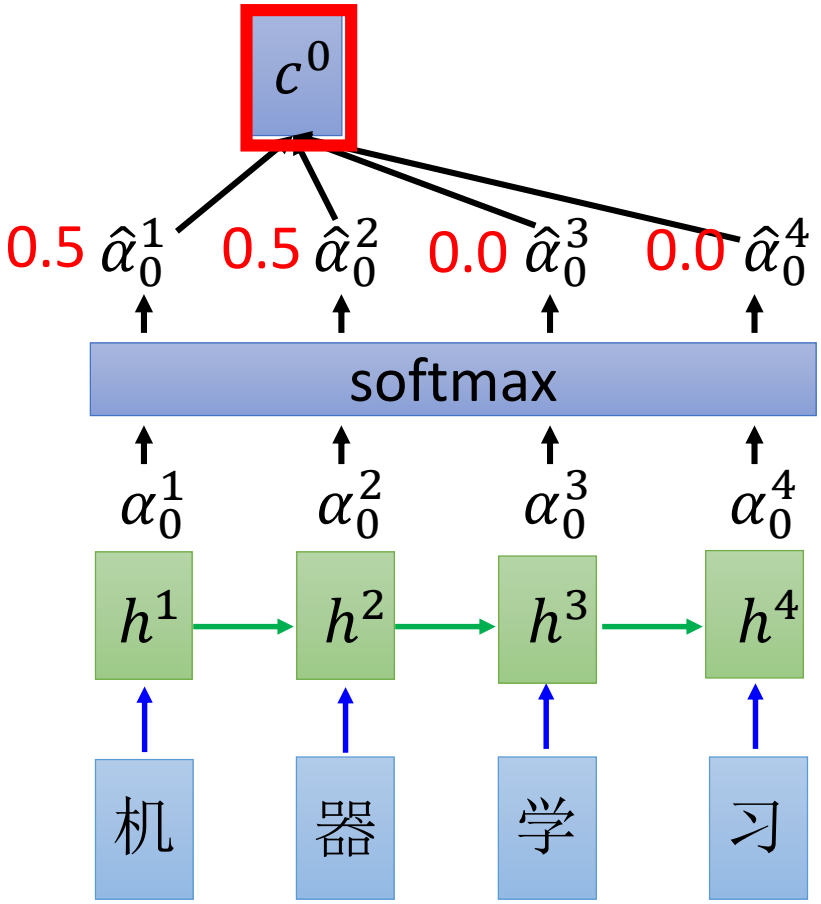
What is match ?

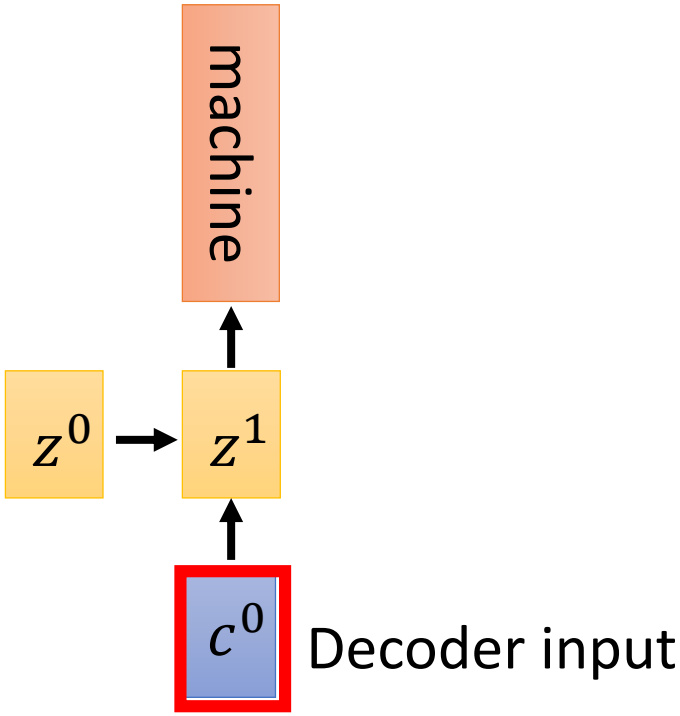
Design by yourself

- Cosine similarity of z and h
- Small NN whose input is z and h, output a scalar
- $\alpha = h^T W z$

Machine Translation

- Attention-based model



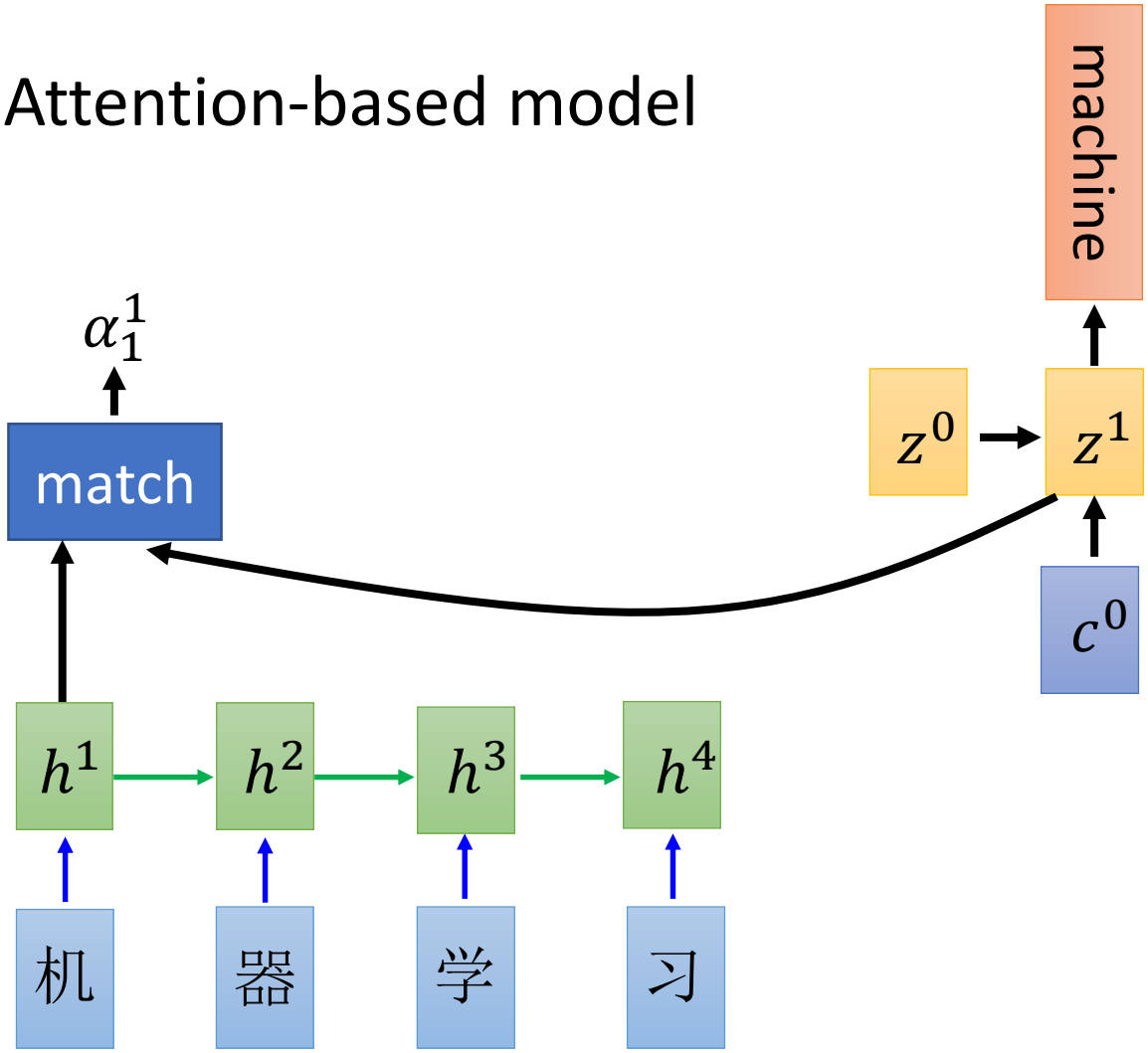


$$c^0 = \sum \hat{\alpha}_0^i h^i$$

$$= 0.5h^1 + 0.5h^2$$

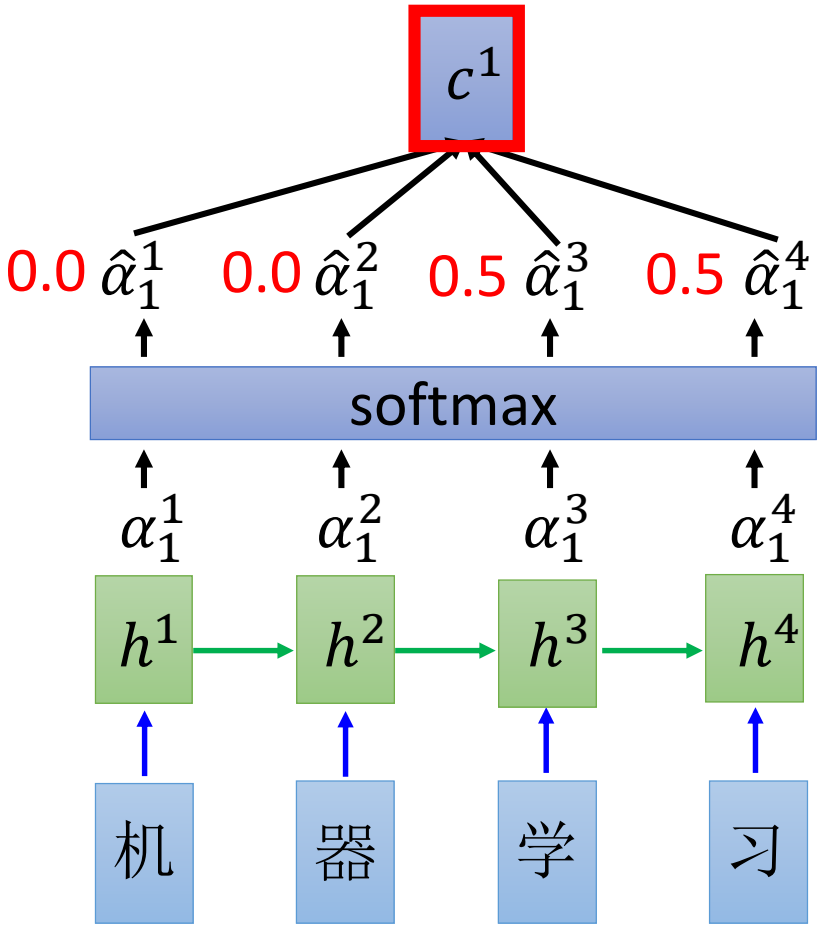
Machine Translation

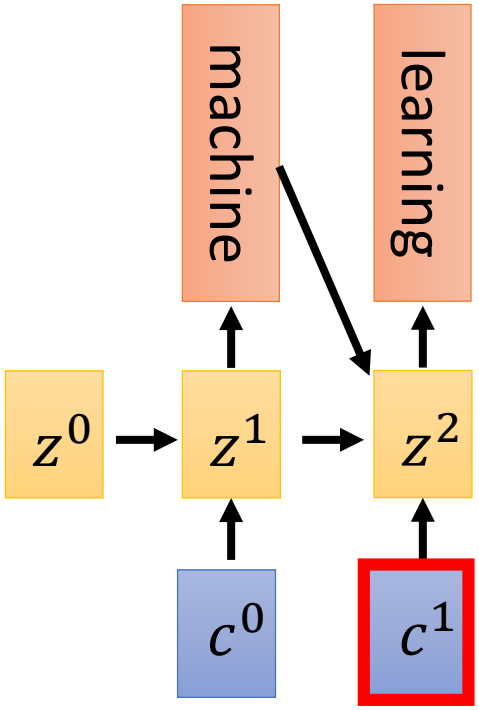
- Attention-based model



Machine Translation

- Attention-based model



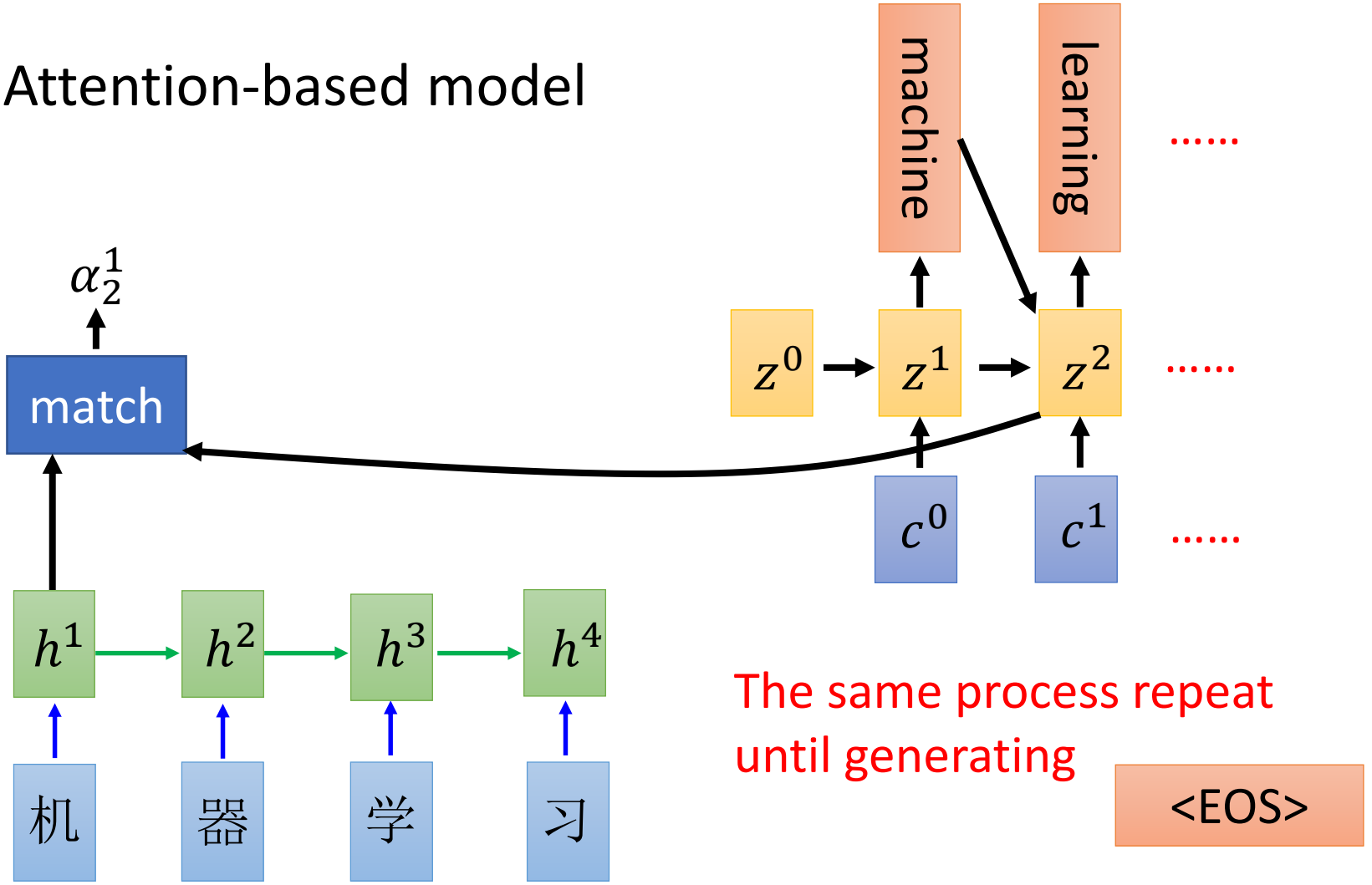


$$c^1 = \sum \hat{\alpha}_1^i h^i$$

$$= 0.5h^3 + 0.5h^4$$

Machine Translation

- Attention-based model



The same process repeat until generating

<EOS>

Image Caption Generation

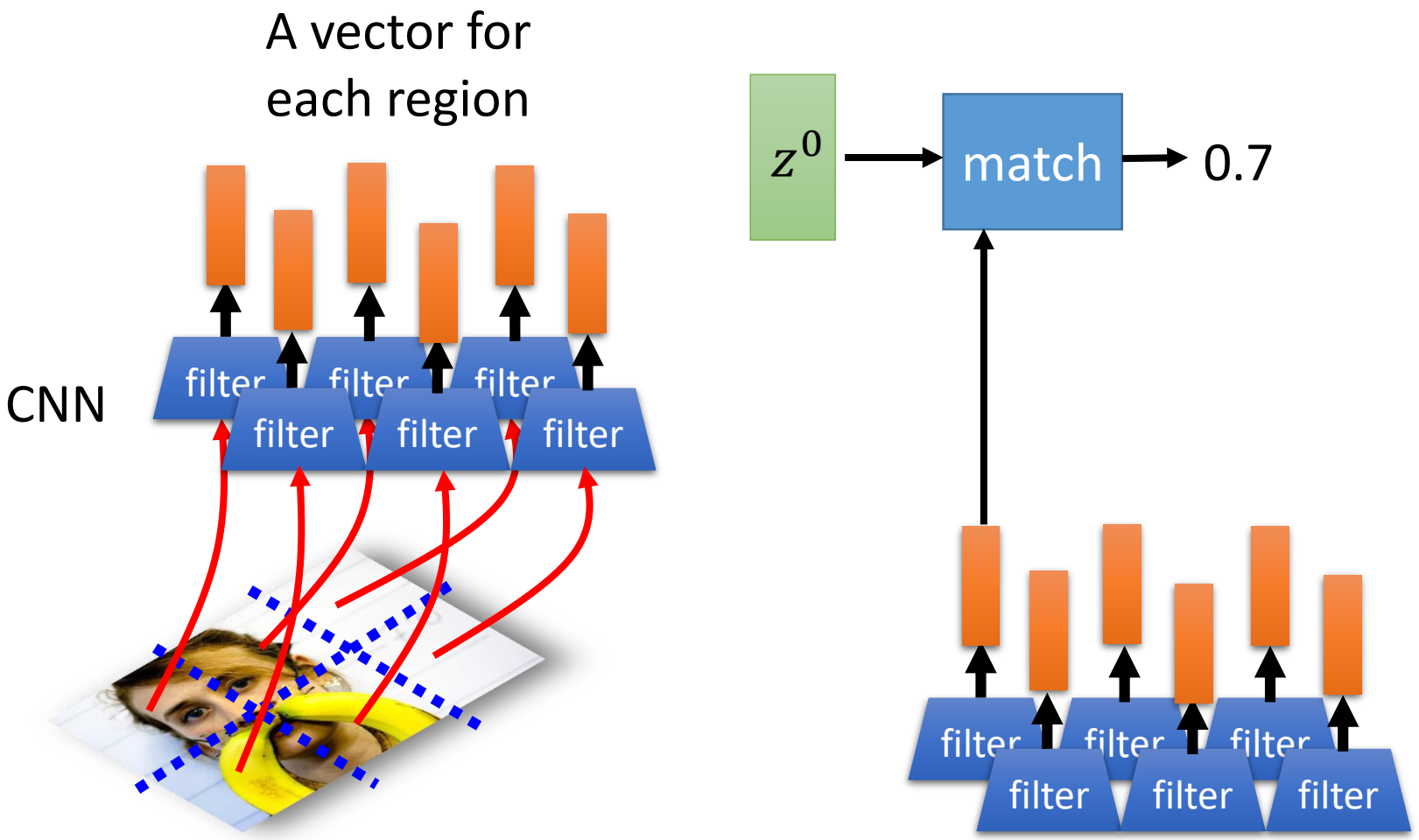


Image Caption Generation

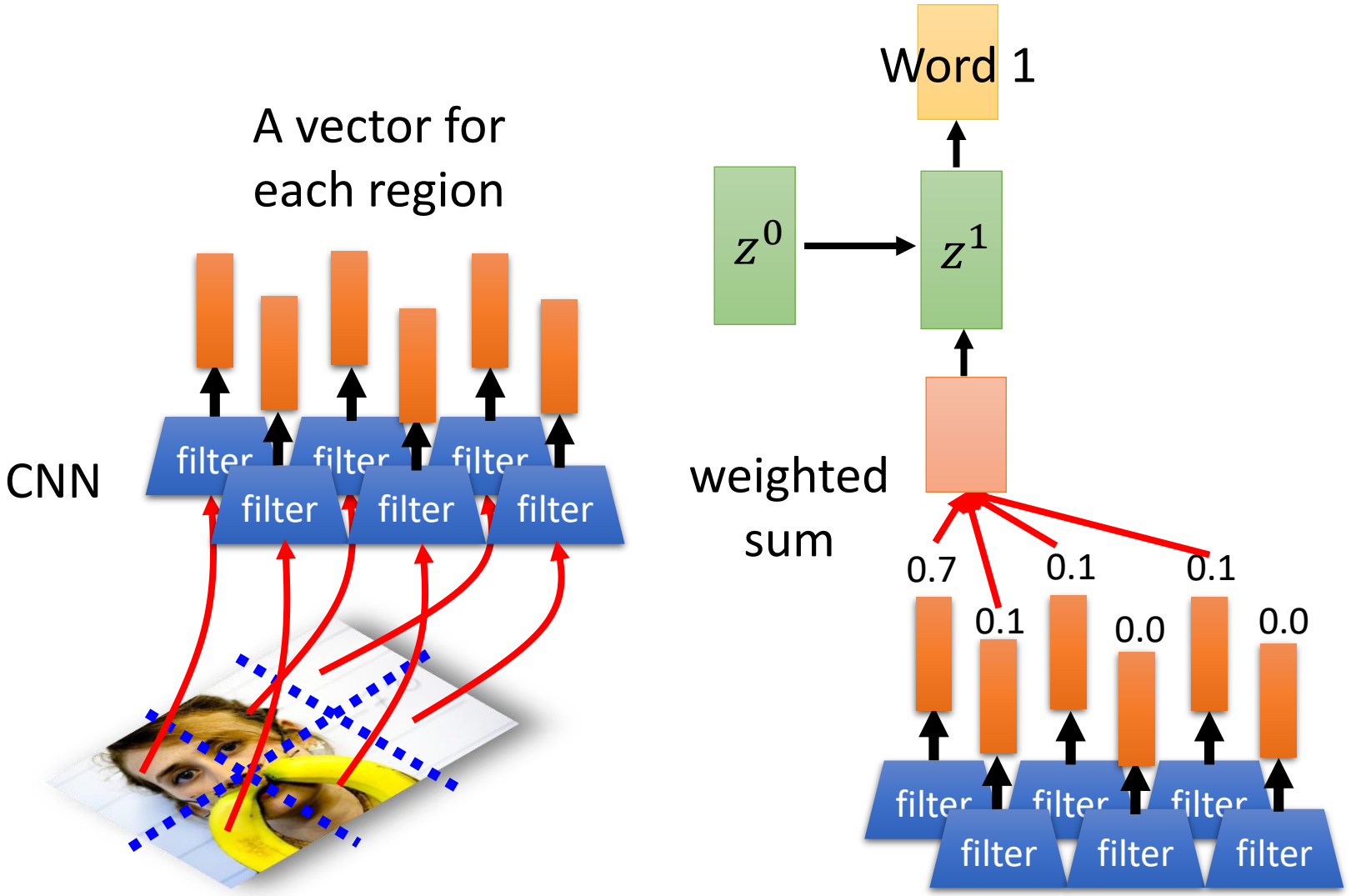


Image Caption Generation

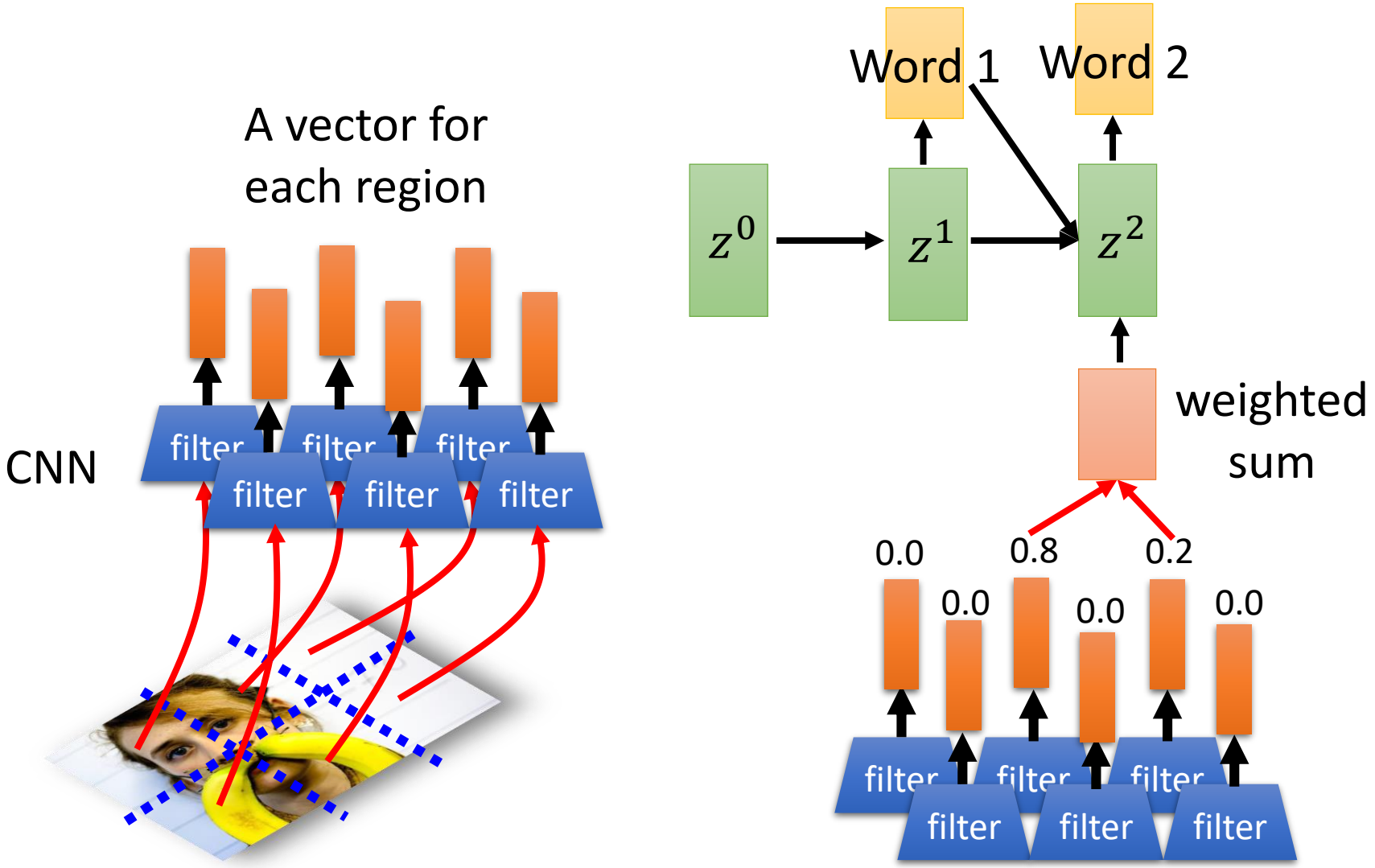


Image Caption Generation

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio, “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”, ICML, 2015

Image Caption Generation

A large white bird standing in a forest.

A woman holding a clock in her hand.

A man wearing a hat and
a hat on a skateboard.

A person is standing on a beach
with a surfboard.

A woman is sitting at a table
with a large pizza.

A man is talking on his cell phone
while another man watches.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio, “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”, ICML, 2015

Ref: A man and a woman ride a motorcycle

A **man** and a **woman** are **talking** on the **road**

Ref: A woman is frying food

Someone is **frying** a **fish** in a **pot**

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, Aaron Courville, "Describing Videos by Exploiting Temporal Structure", ICCV, 2015

Mismatch between Train and Test

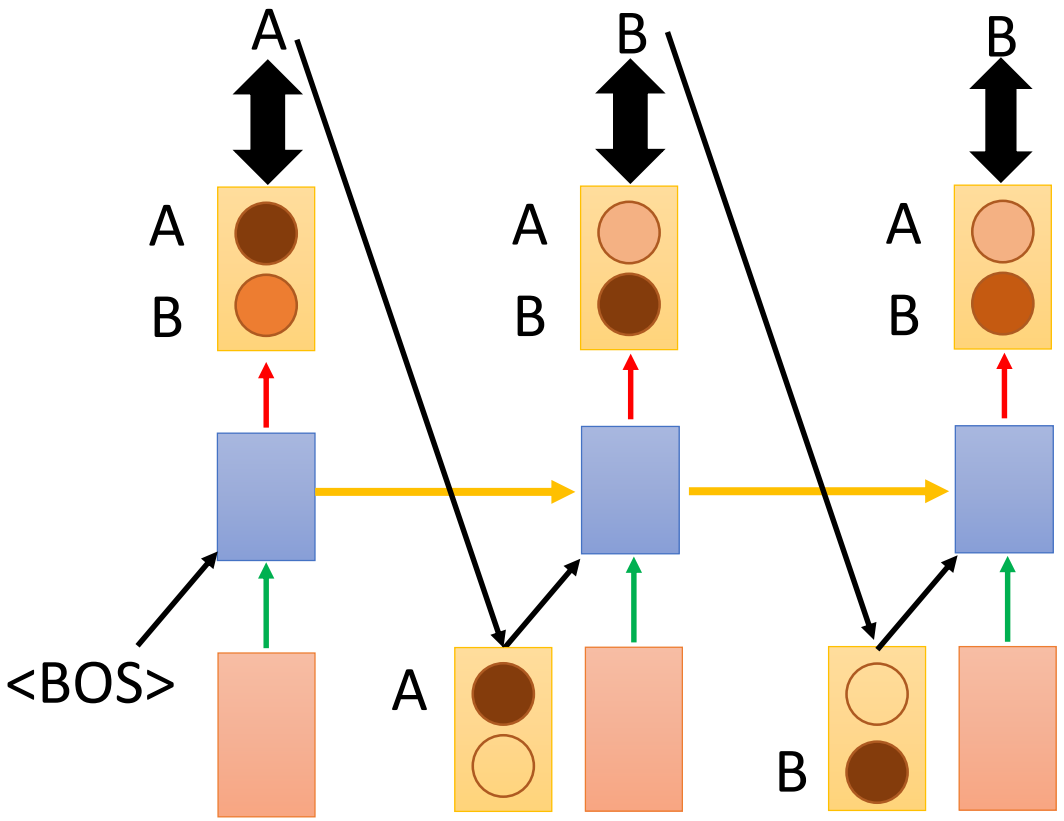
- ***Training***

$$C = \sum_t C_t$$

Minimizing cross-entropy of each component

: condition

Reference:



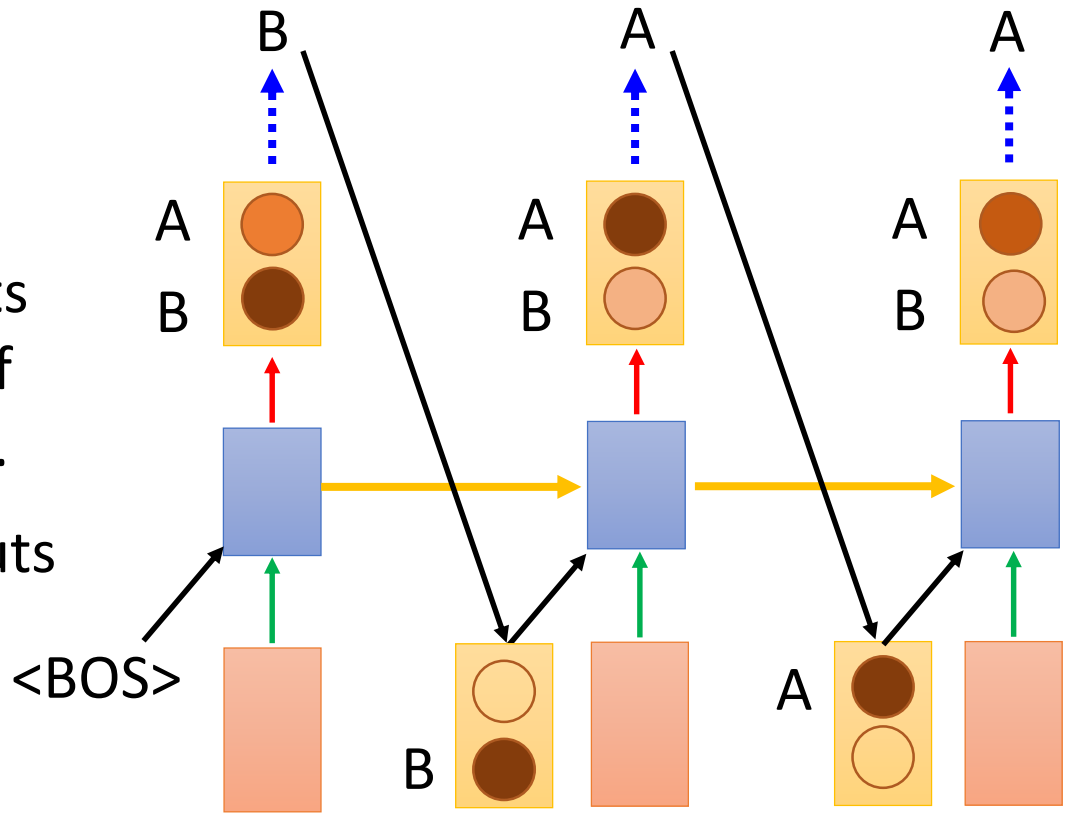
Mismatch between Train and Test

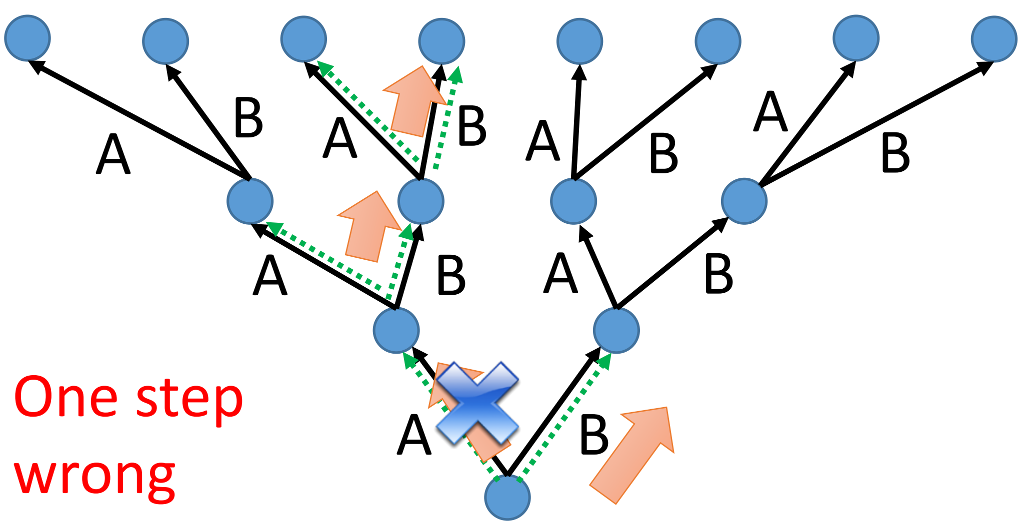
- ***Generation***

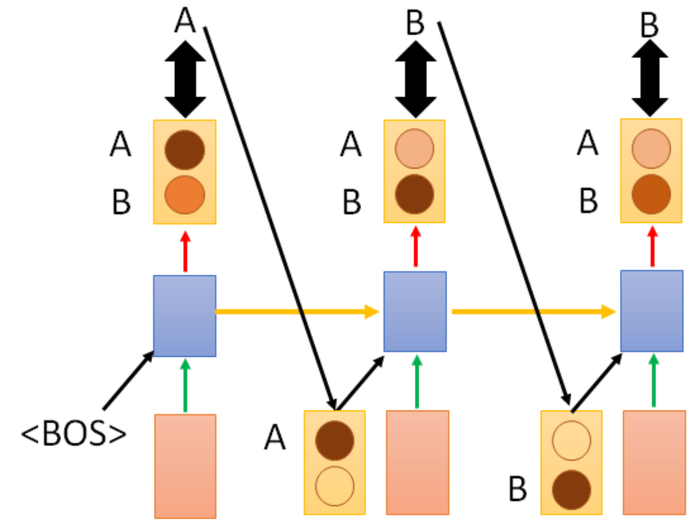
We do not know the reference

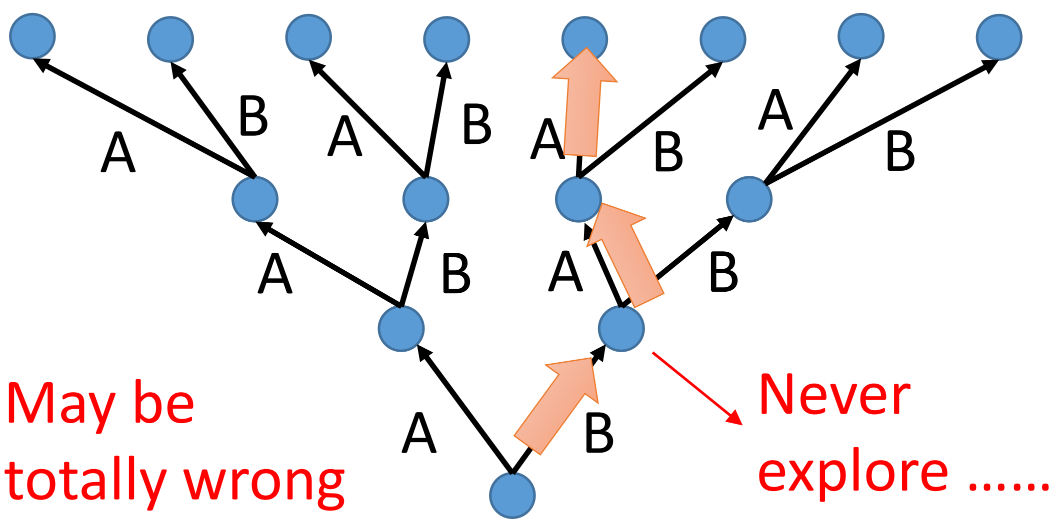
Testing: The inputs are the outputs of the last time step.

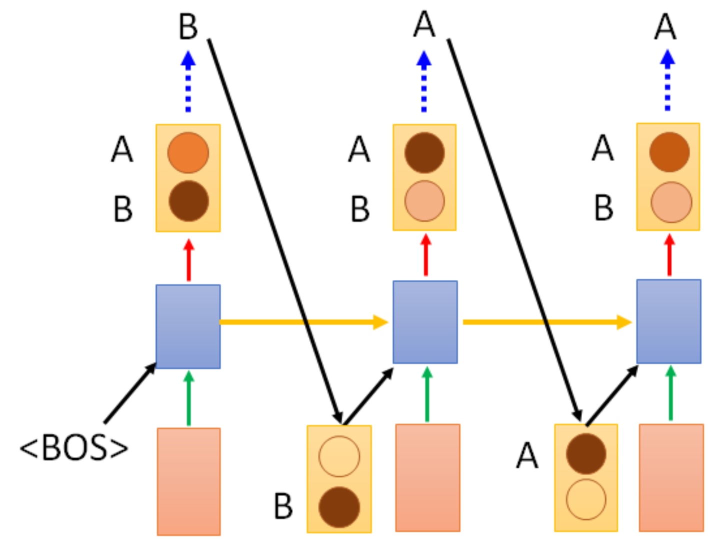
Training: The inputs are reference.





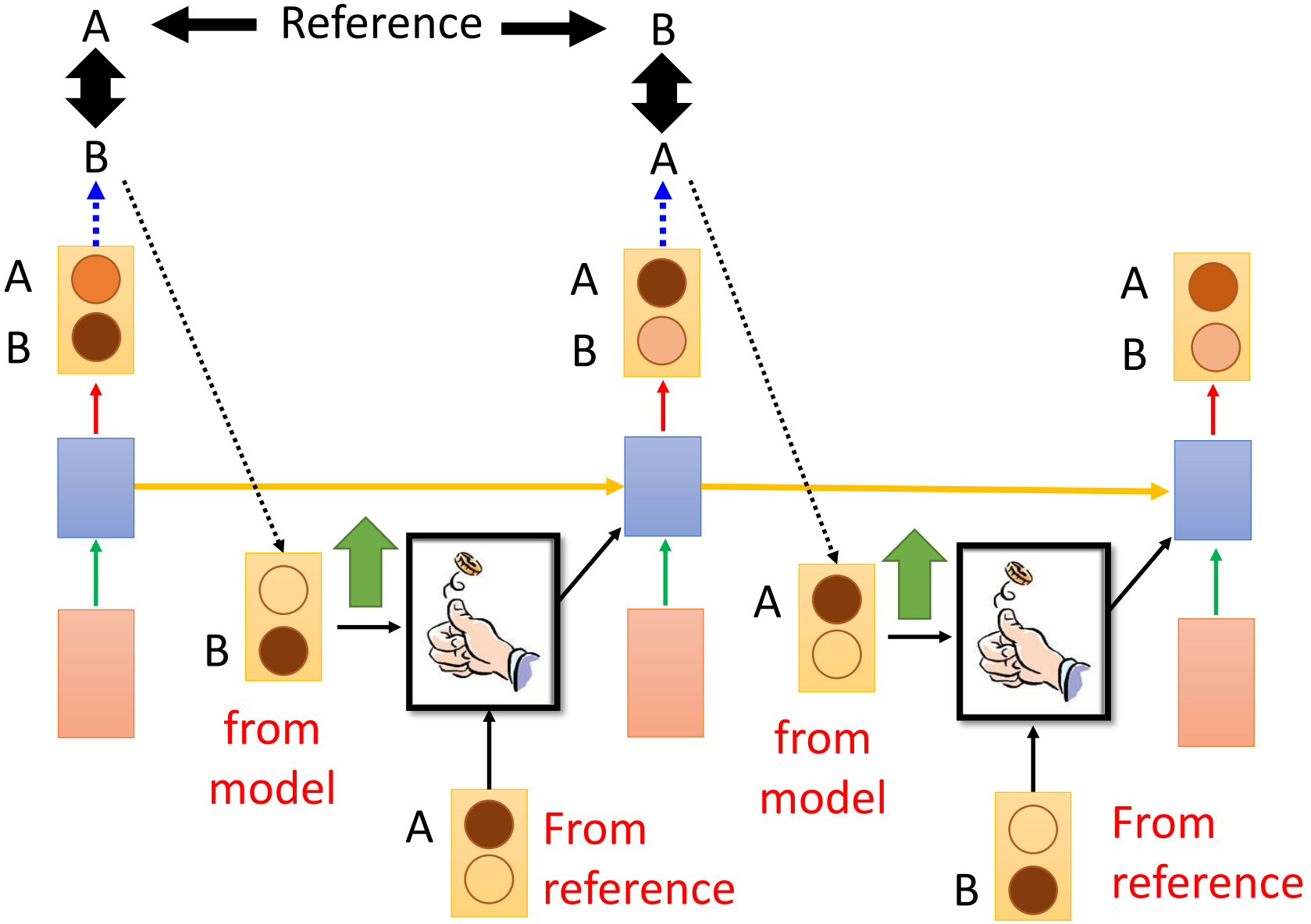






一步错，步步错

Scheduled Sampling

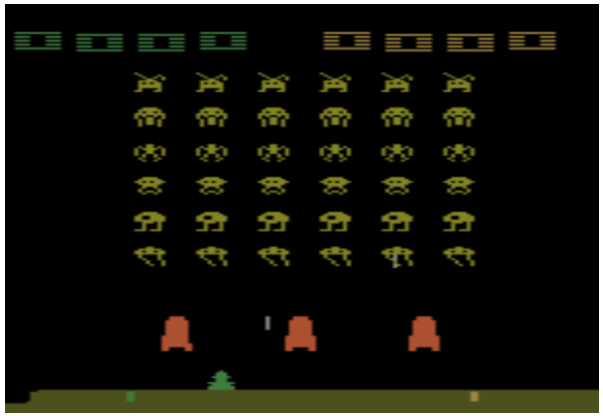


Reinforcement learning?

Start with observation s_1

Observation s_2

Observation s_3



Obtain reward $r_1 = 0$

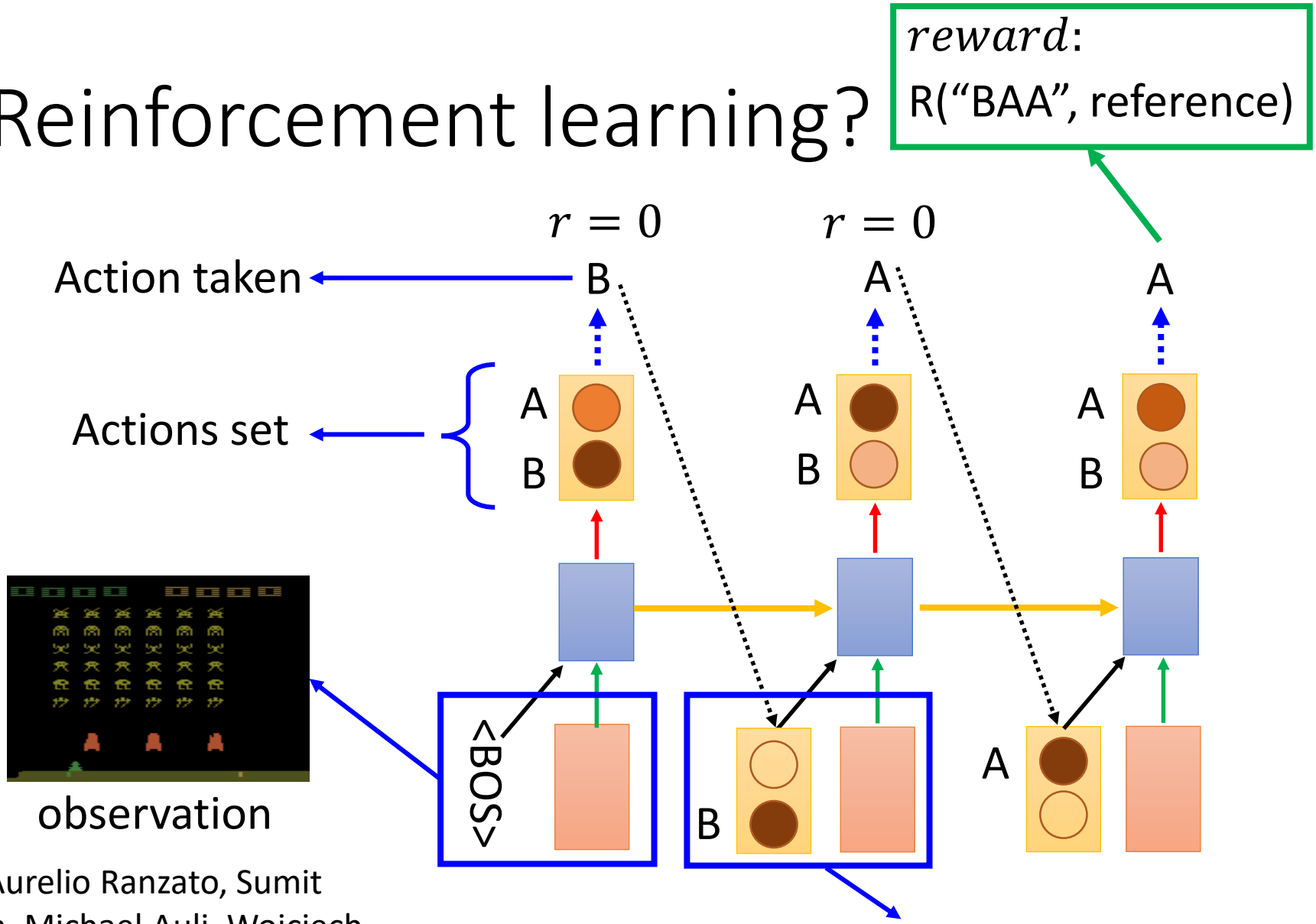
Action a_1: "right"

Obtain reward $r_2 = 5$

Action a_2 : "fire"

(kill an alien)

Reinforcement learning?



Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, Wojciech Zaremba, "Sequence Level Training with Recurrent Neural Networks", ICLR, 2016

The End!