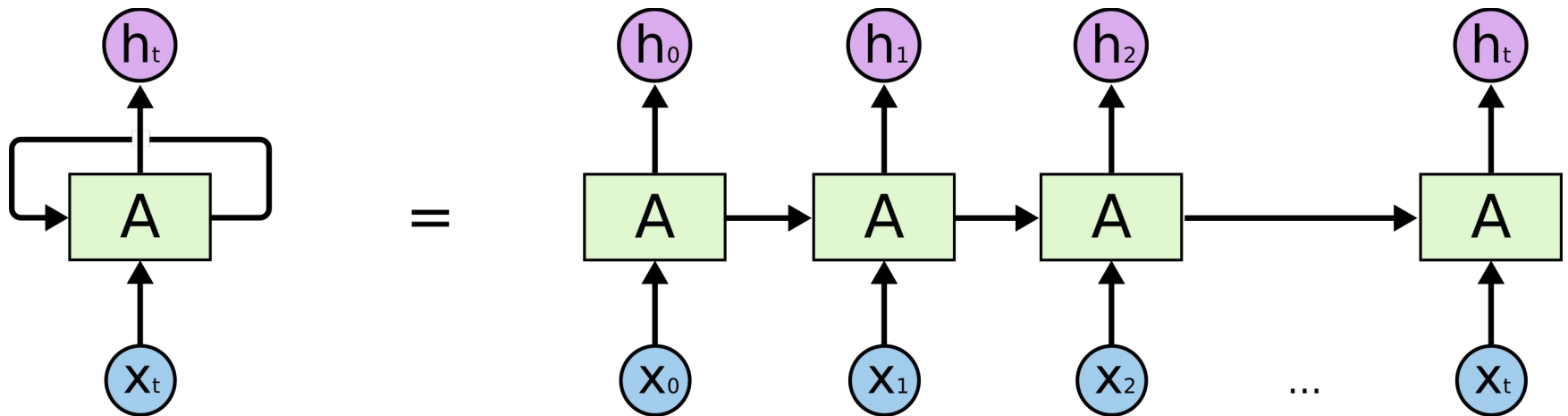
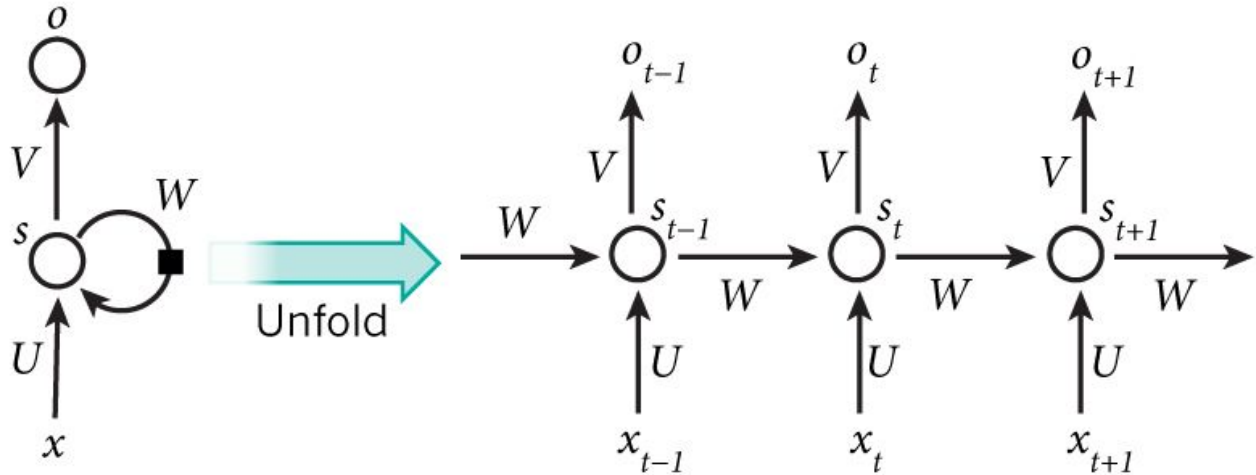


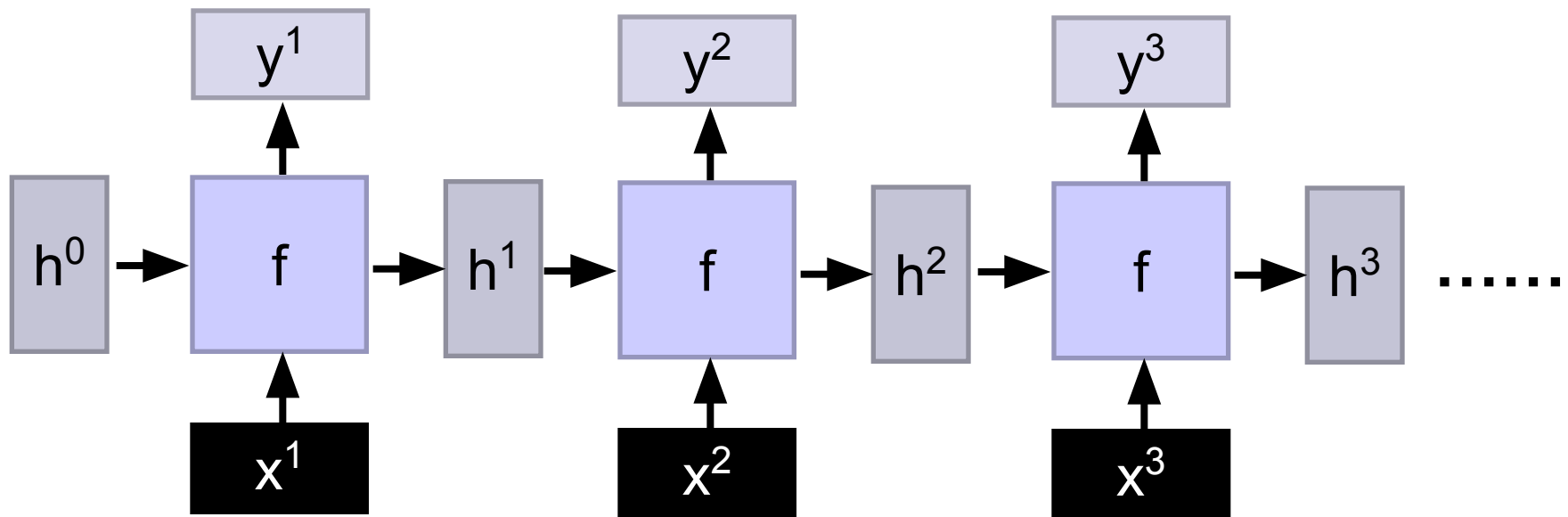
Recurrent Neural Network



How does RNN reduce complexity?

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

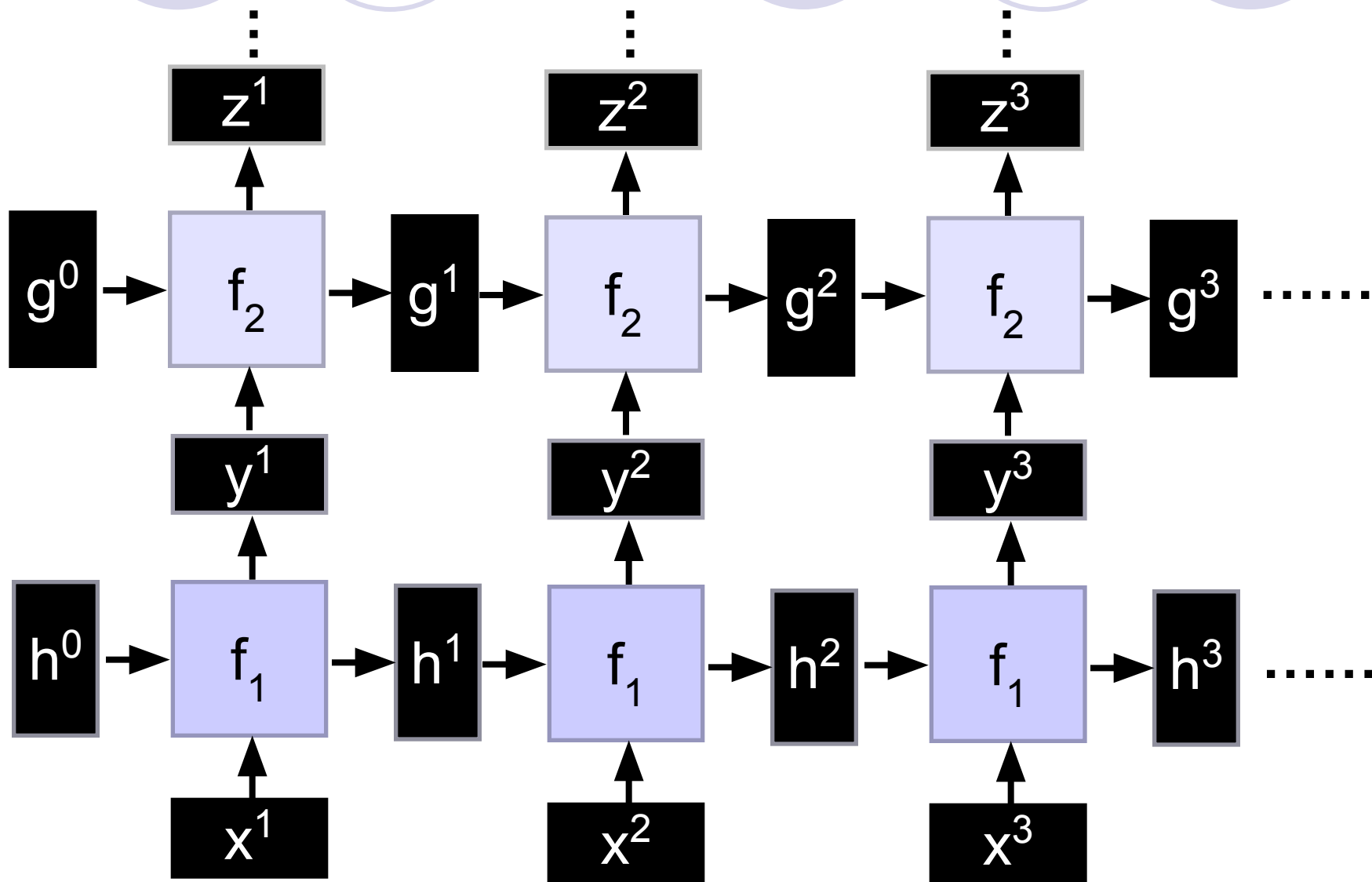
h and h' are vectors with the same dimension



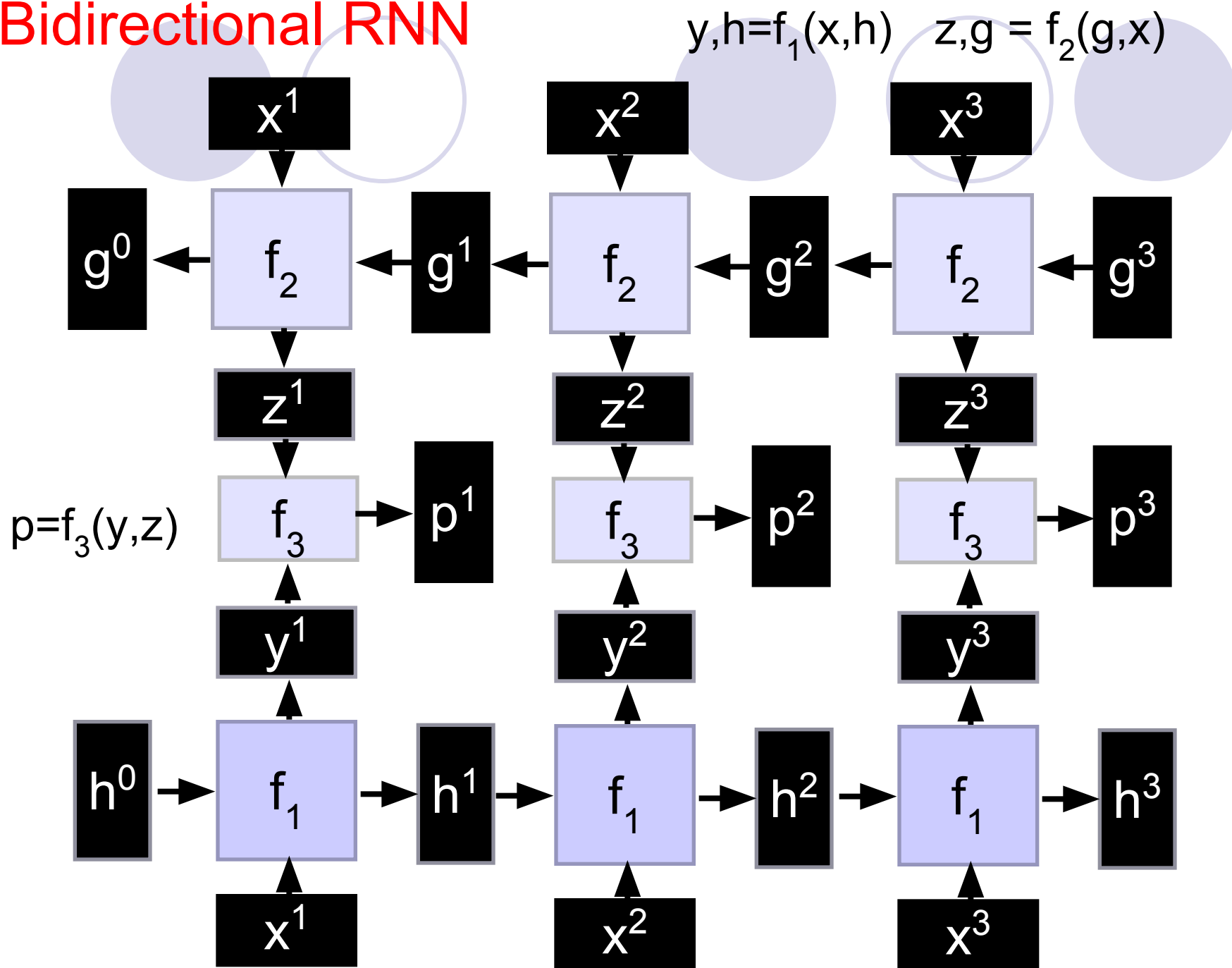
No matter how long the input/output sequence is, we only need one function f . If f 's are different, then it becomes a feedforward NN. This may be treated as another compression from fully connected network.

Deep RNN

$$h', y = f_1(h, x), \quad g', z = f_2(g, y) \quad \dots$$



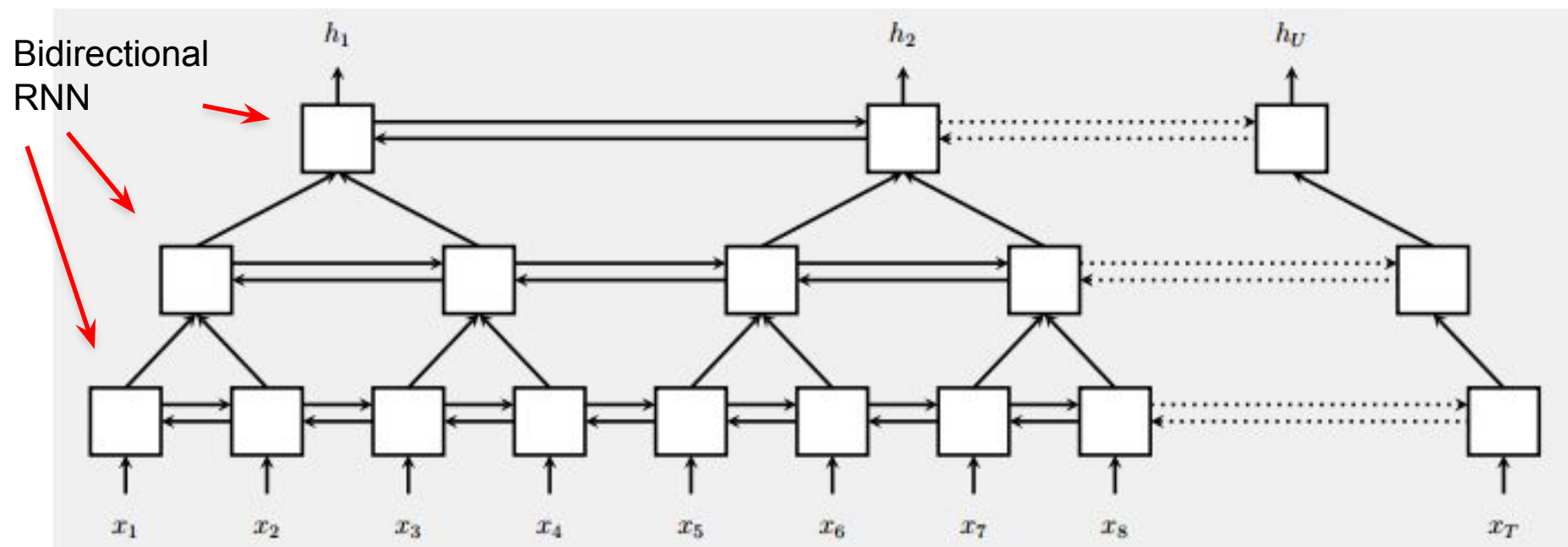
Bidirectional RNN



Pyramid RNN

Significantly speed up training

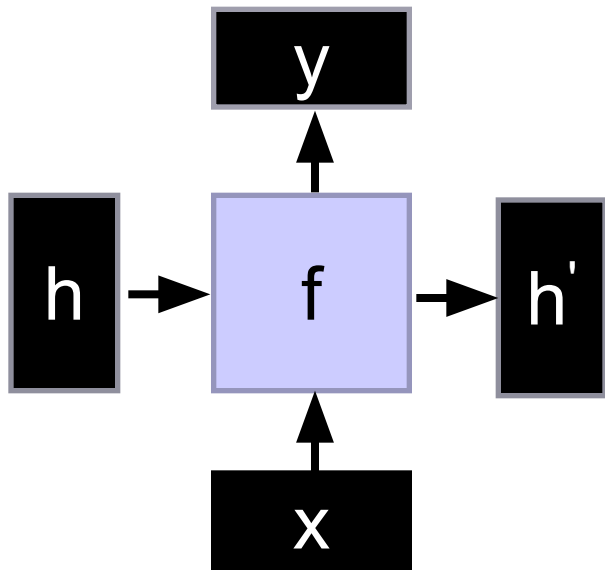
- Reducing the number of time steps



W. Chan, N. Jaitly, Q. Le and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” ICASSP, 2016

Naïve RNN

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



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

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

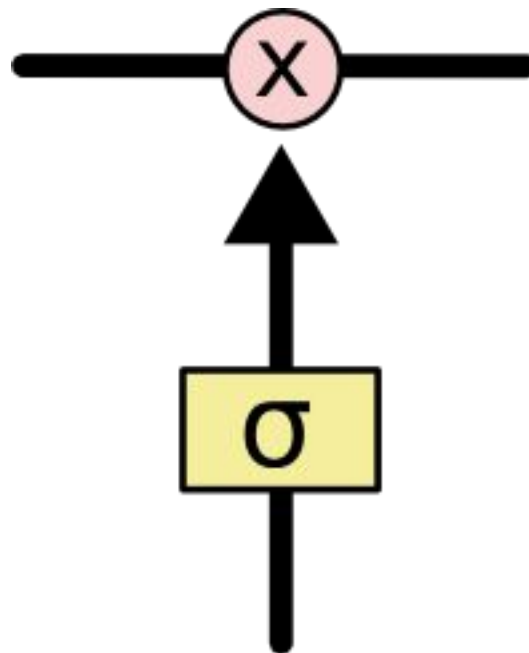
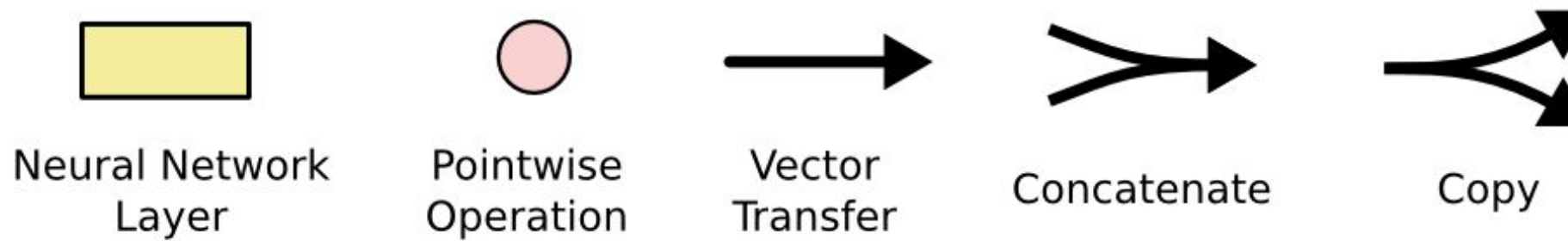
softmax

Note, y is computed from h'

We have ignored the bias

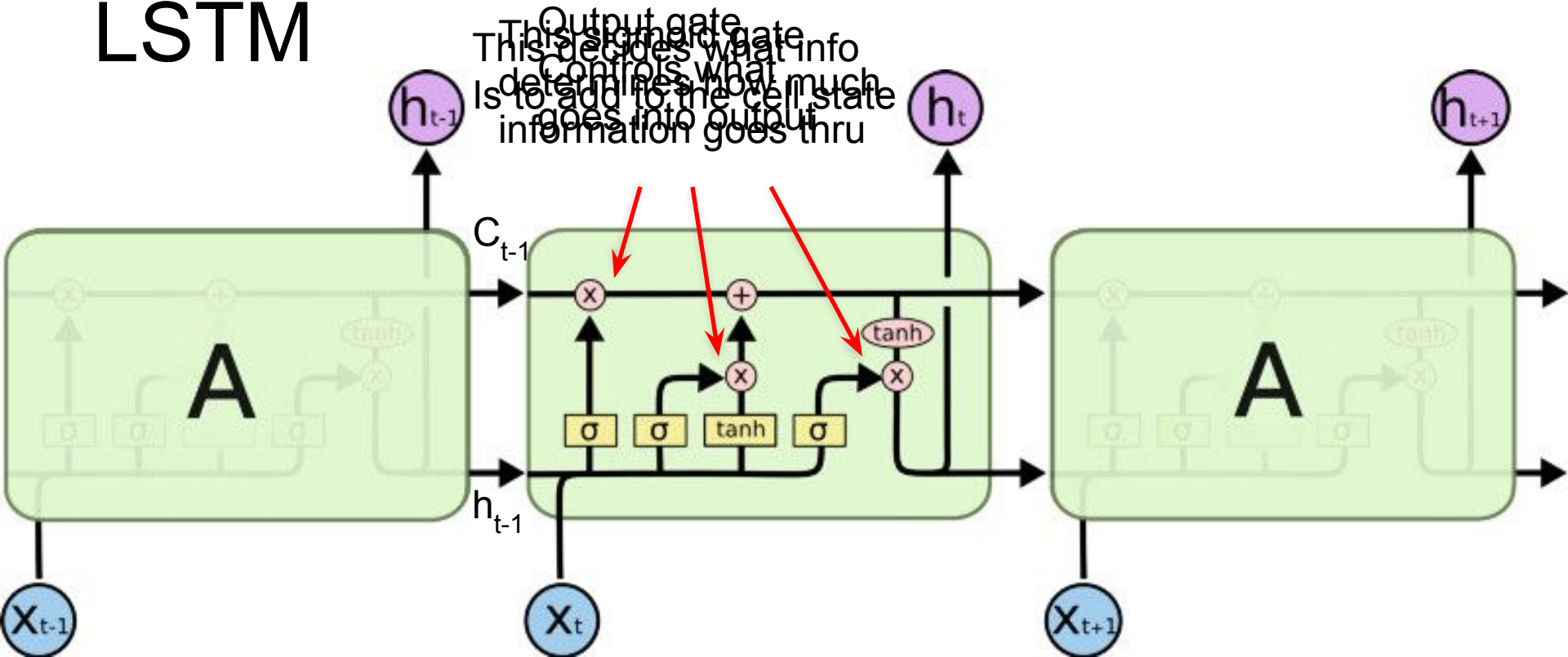
Problems with naive RNN

- When dealing with a time series, it tends to forget old information. When there is a distant relationship of unknown length, we wish to have a “memory” to it.
- Vanishing gradient problem.



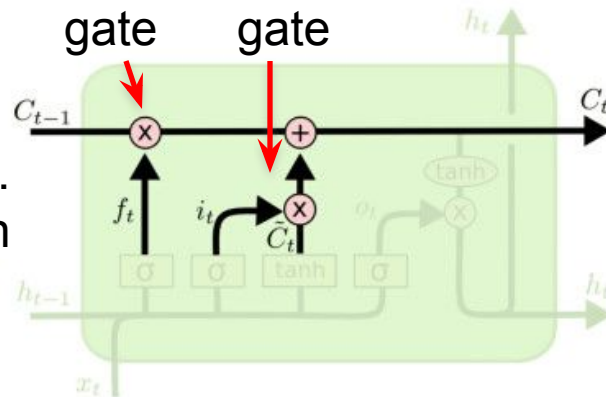
The sigmoid layer outputs numbers between 0-1 determine how much each component should be let through. Pink X gate is point-wise multiplication.

LSTM



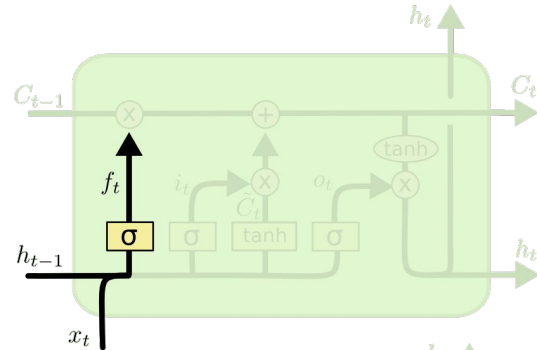
Output gate
This sigmoid gate
controls what info
is added to the cell
state
Is to add to the cell
information goes thru

Forget gate
input gate

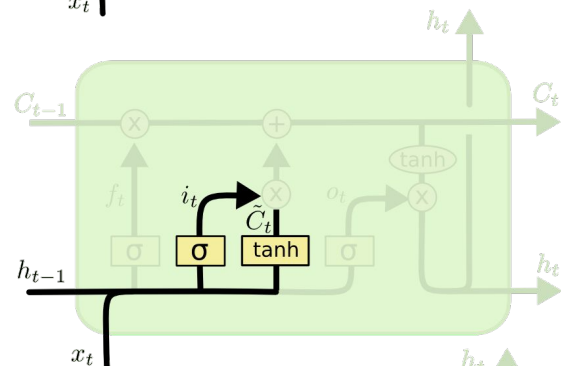


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

The core idea is this cell
state C_t it is changed
slowly, with only minor
Vanishing gradient problem in
linear interactions. It is very
easy for information to flow
along it unchanged.
ReLU replaces tanh ok?

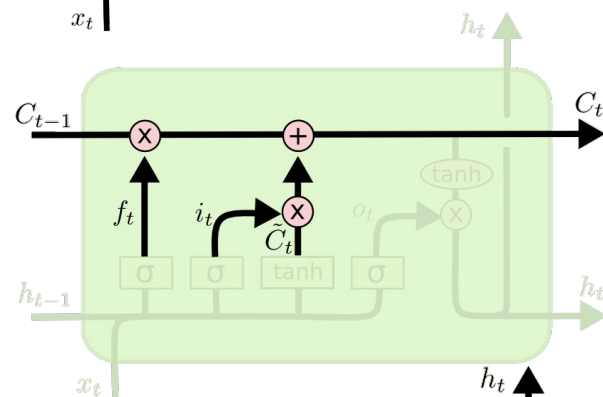


$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

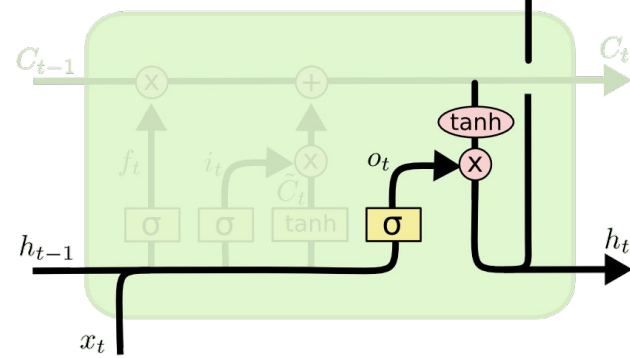


$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

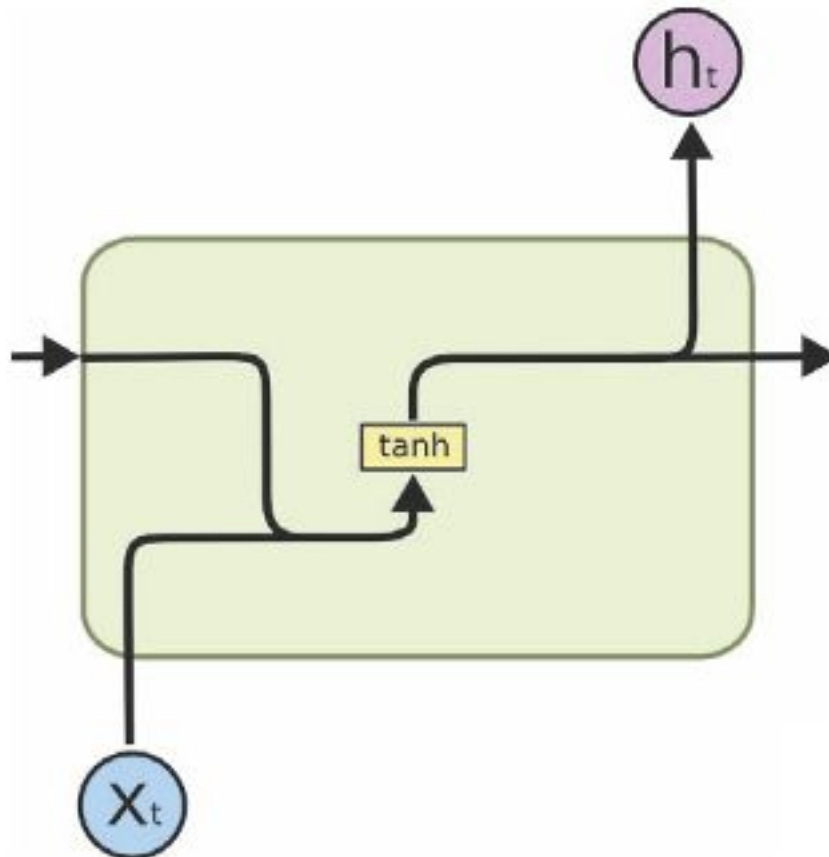
i_t decides what component is to be updated.

C'_t provides change contents

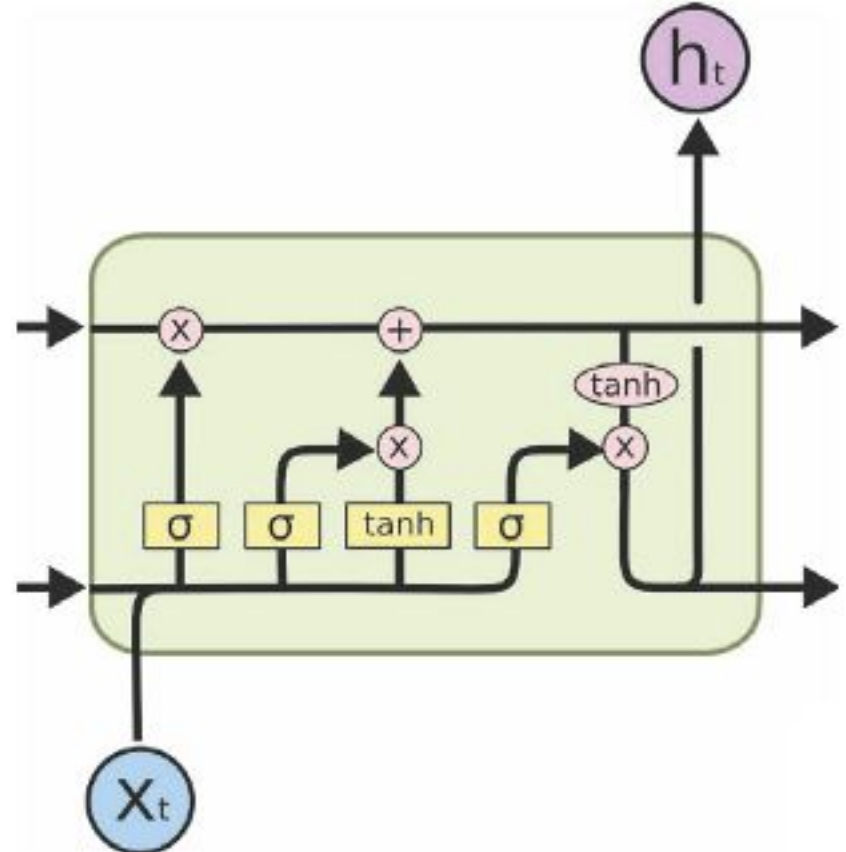
Updating the cell state

Decide what part of the cell state to output

RNN vs LSTM

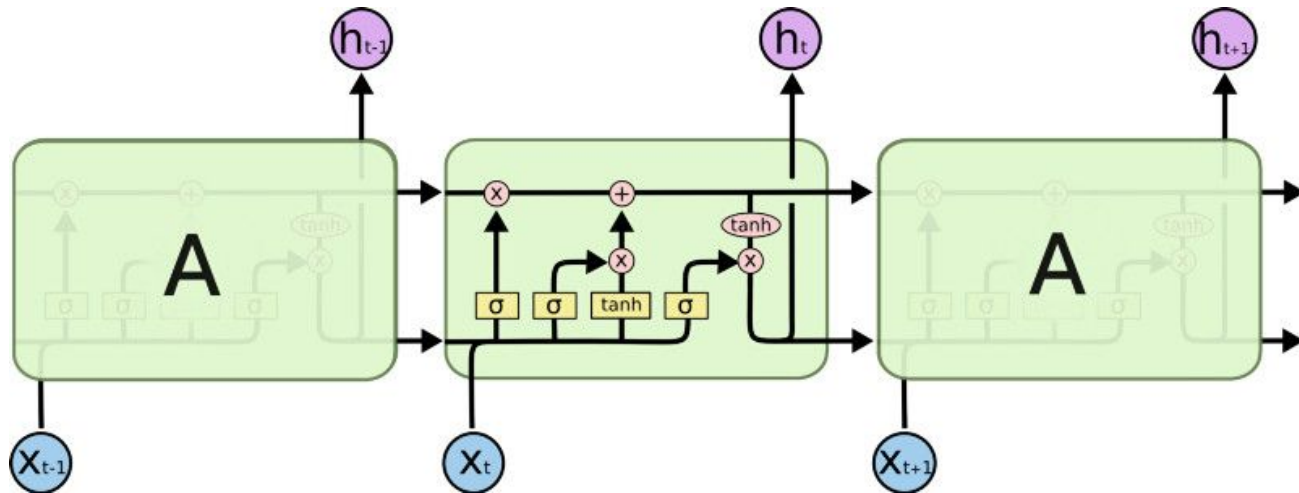


(a) RNN

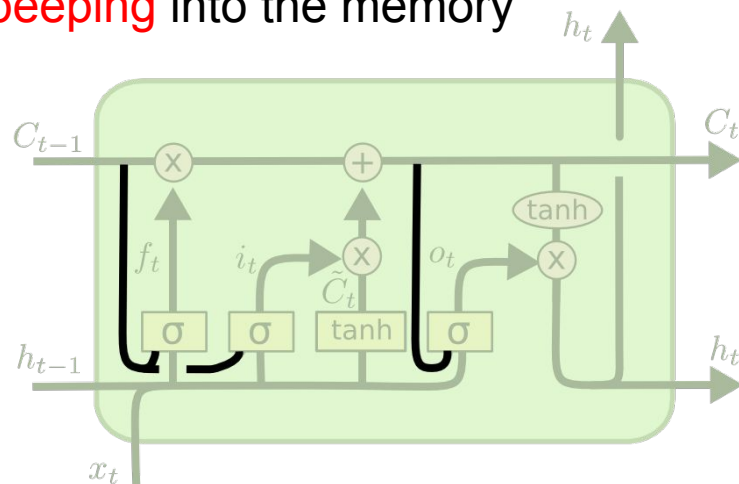


(b) LSTM

Peephole LSTM



Allows “**peeping** into the memory”

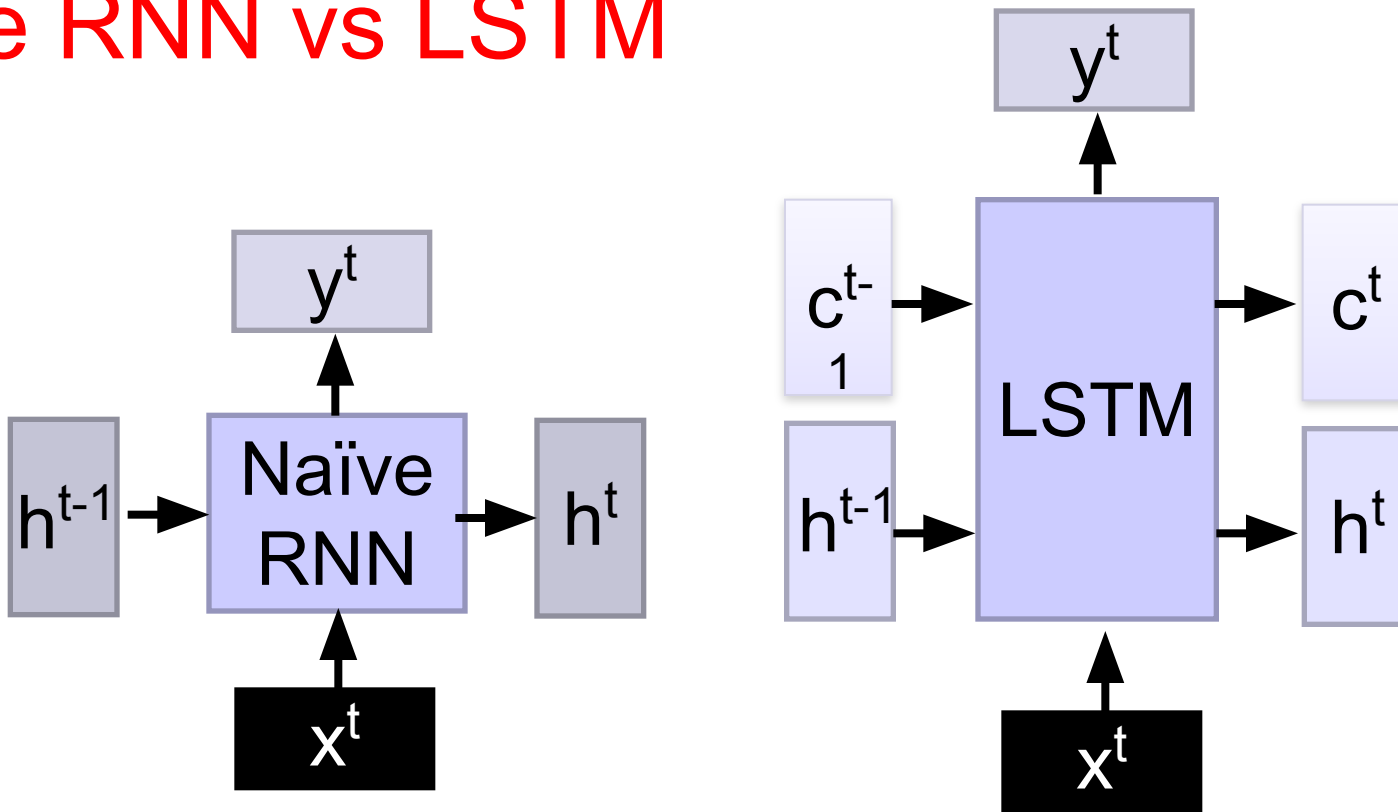


$$f_t = \sigma (W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma (W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i)$$

$$o_t = \sigma (W_o \cdot [C_t, h_{t-1}, x_t] + b_o)$$

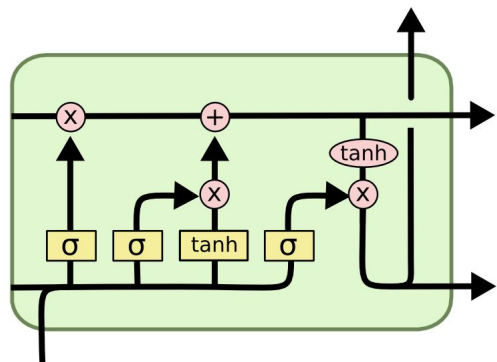
Naïve RNN vs LSTM



c changes slowly $\longrightarrow c^t$ is c^{t-1} added by something

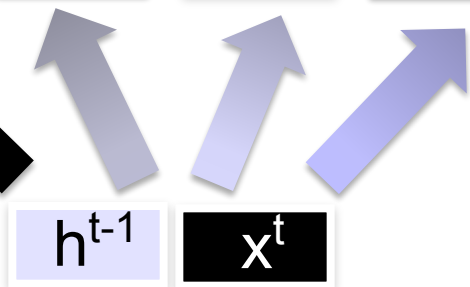
h changes faster $\longrightarrow h^t$ and h^{t-1} can be very different

These 4 matrix computation should be done concurrently.



c^{t-1}

Controls forget gate Controls input gate Updating information Controls Output gate



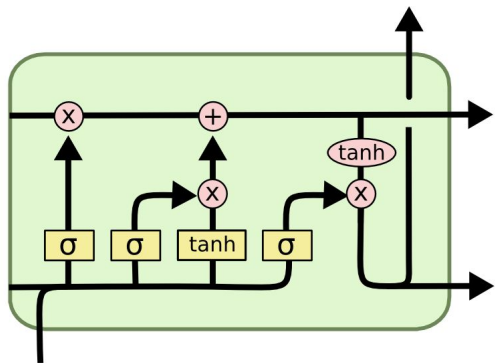
$$z = \tanh(W \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix})$$

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

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

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

Information flow of LSTM

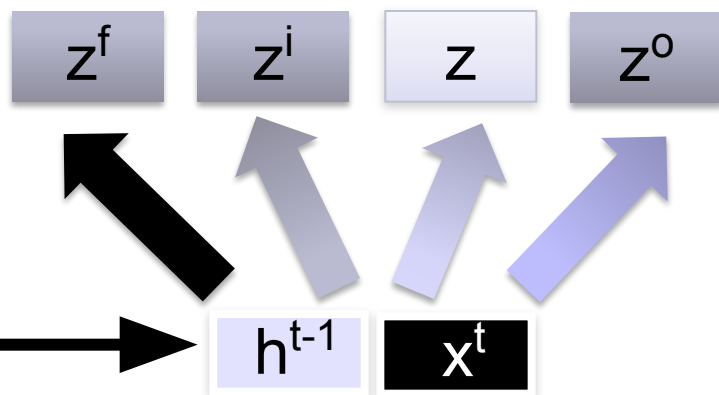


$$z = \tanh(W \begin{bmatrix} x^t \\ h^{t-1} \\ c^{t-1} \end{bmatrix})$$

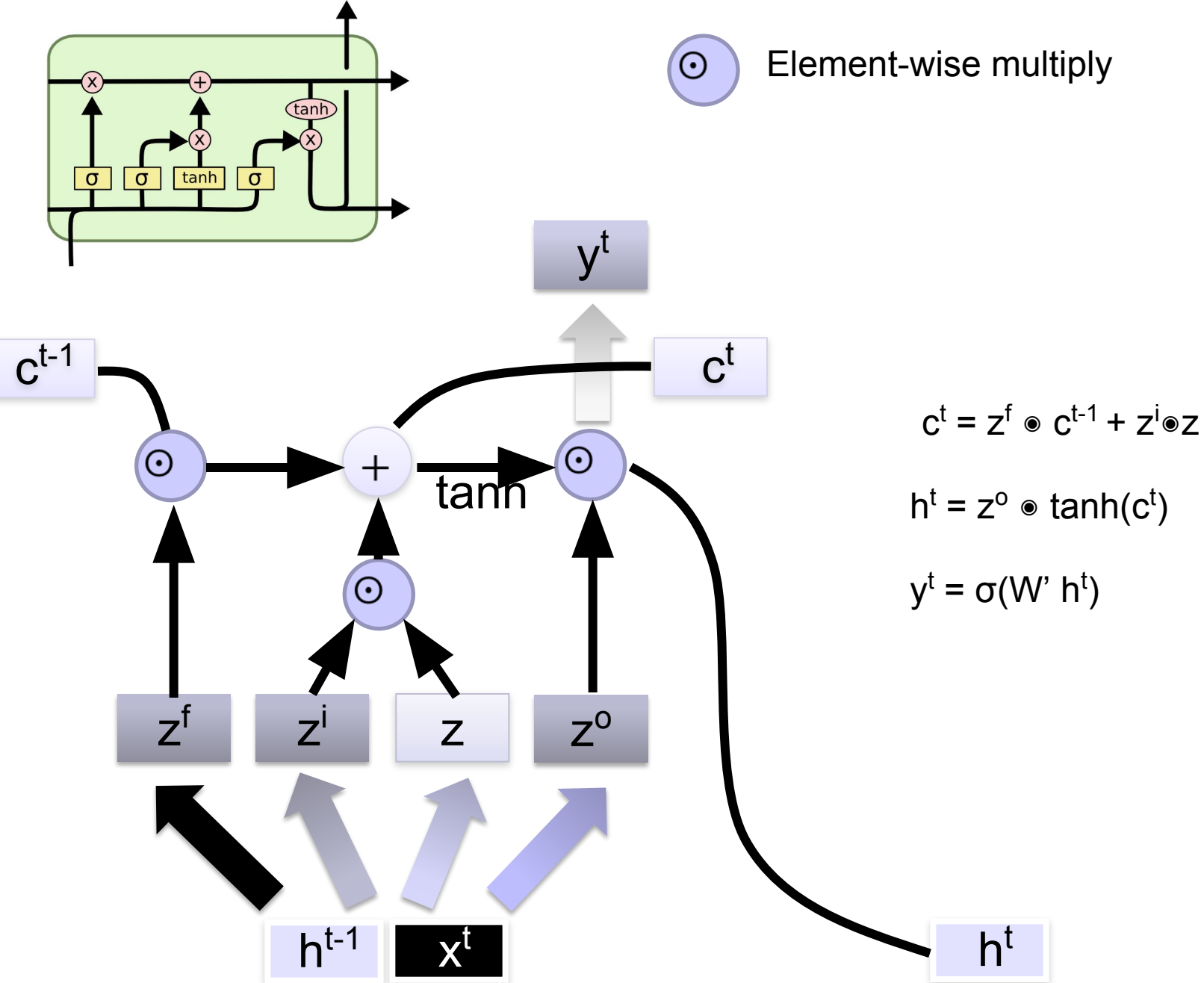
↑
diagonal

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

“peephole”

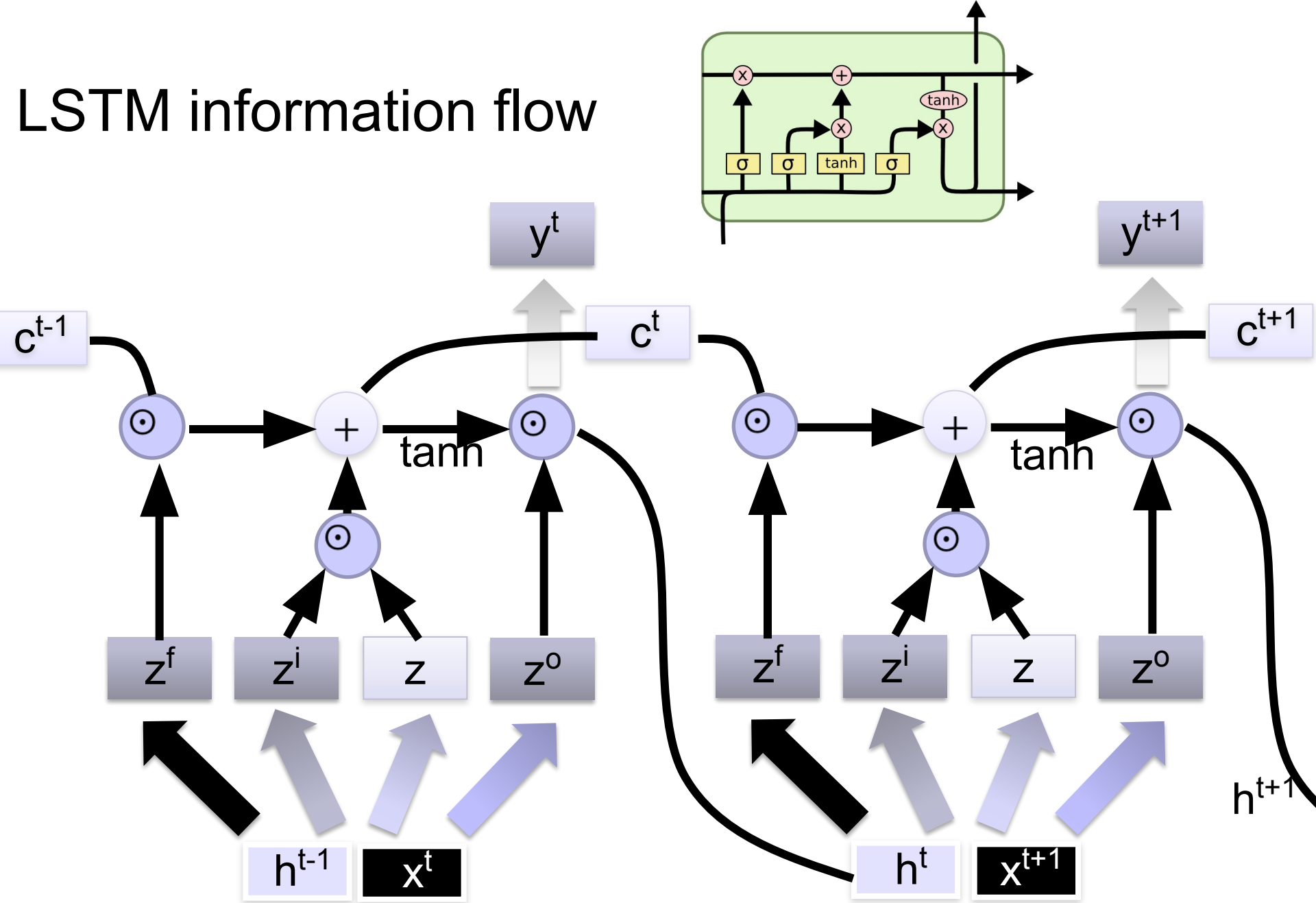


Information flow of LSTM



Information flow of LSTM

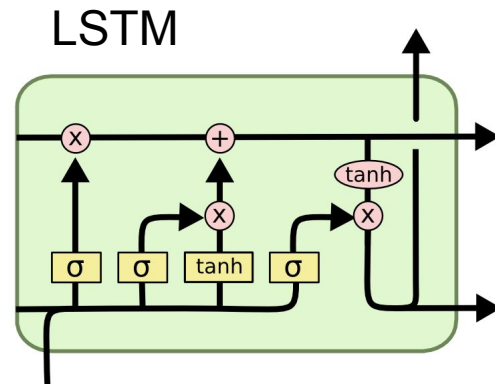
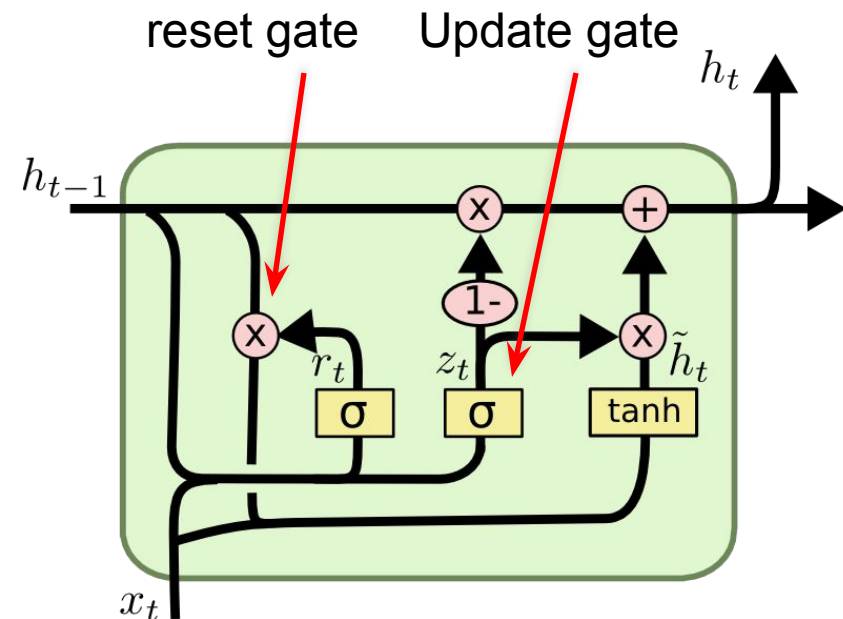
LSTM information flow



Information flow of LSTM

GRU – gated recurrent unit

(more compression)



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

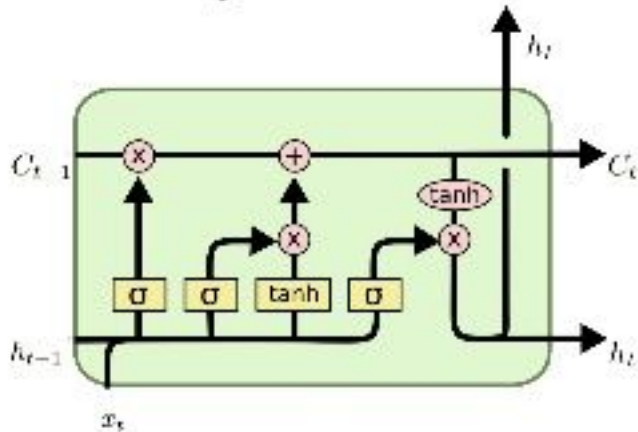
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

It combines the **forget** and **input** into a single **update gate**.
It also merges the cell state and hidden state. This is simpler than LSTM. There are many other variants too.

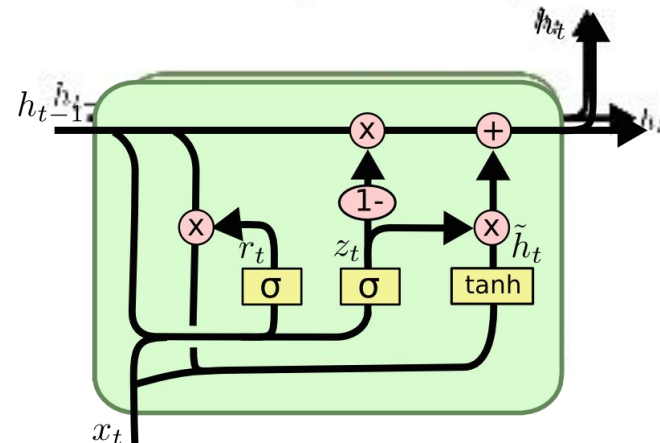
$X, *$: element-wise multiply

LSTM and GRU

- LSTM [Hochreiter&Schmidhuber97]



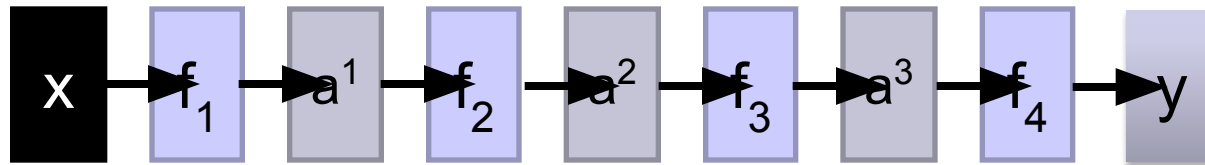
- GRU [Cho+14]



GRUs also take x_t and h_{t-1} as inputs. They perform some calculations and then pass along h_t . What makes them different from LSTMs is that GRUs don't need the cell layer to pass values along. The calculations within each iteration ensure that the h_t values being passed along either retain a high amount of old information or are jump-started with a high amount of new information.

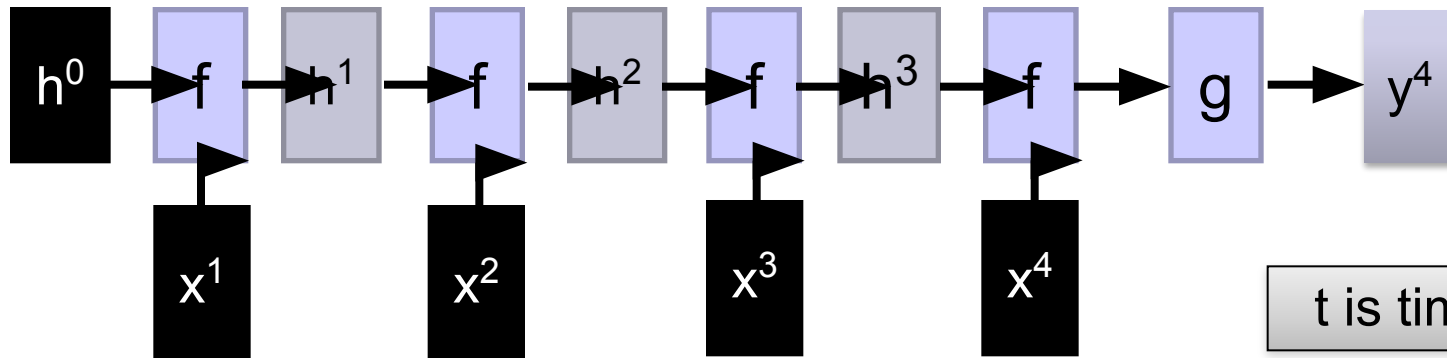
Feed-forward vs Recurrent Network

1. Feedforward network does not have input at each step
2. Feedforward network has different parameters for each layer



$$a^t = f_t(a^{t-1}) = \sigma(W^t a^{t-1} + b^t)$$

t is layer



$$a^t = f(a^{t-1}, x^t) = \sigma(W^h a^{t-1} + W^i x^t + b^i)$$

t is time step

We will turn the recurrent network 90 degrees.

GRU \rightarrow Highway Network

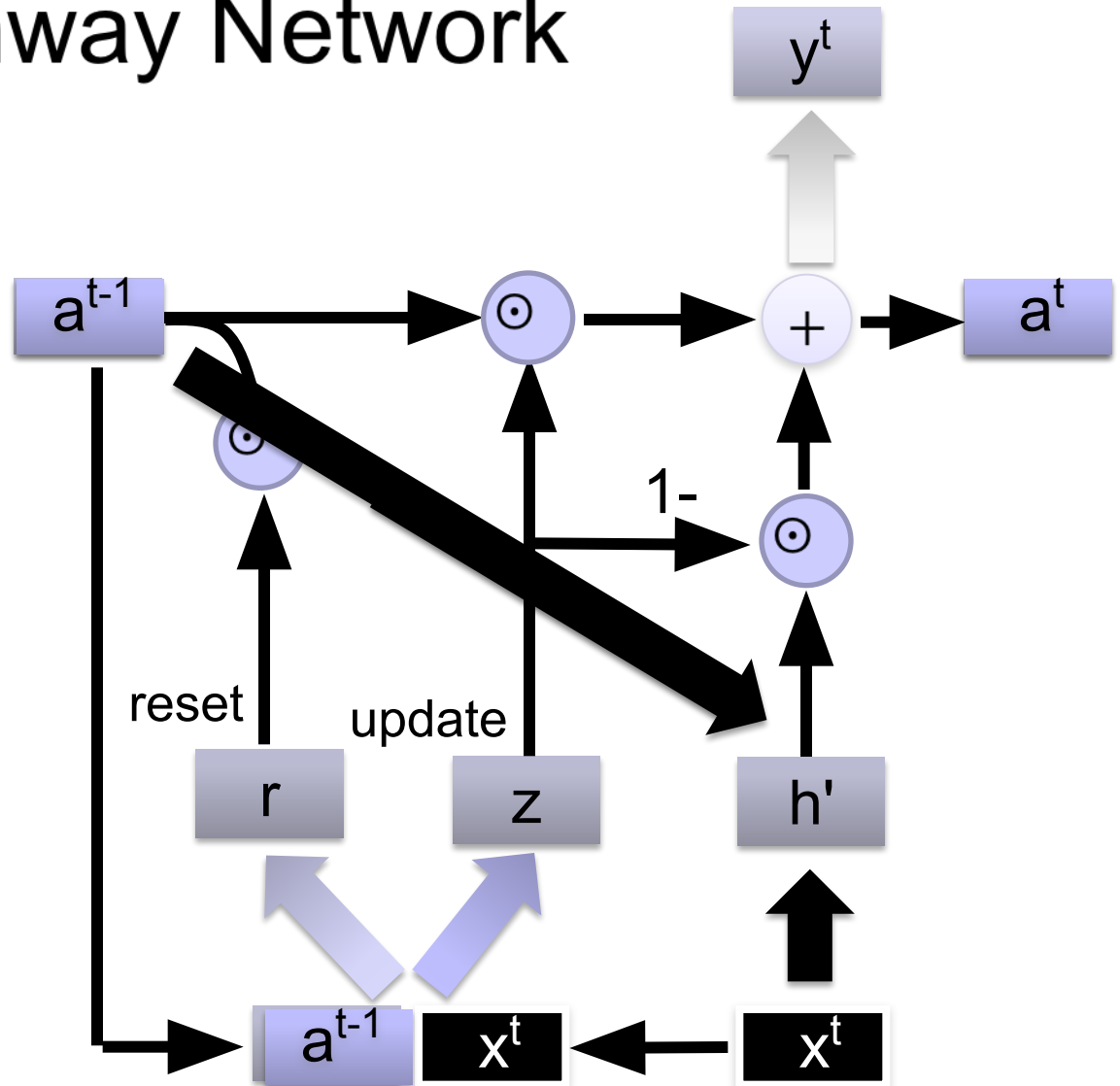
No input x^t at each step

No output y^t at each step

a^{t-1} is the output of the $(t-1)$ -th layer

\mathbf{a}^t is the output of the t -th layer

No reset gate



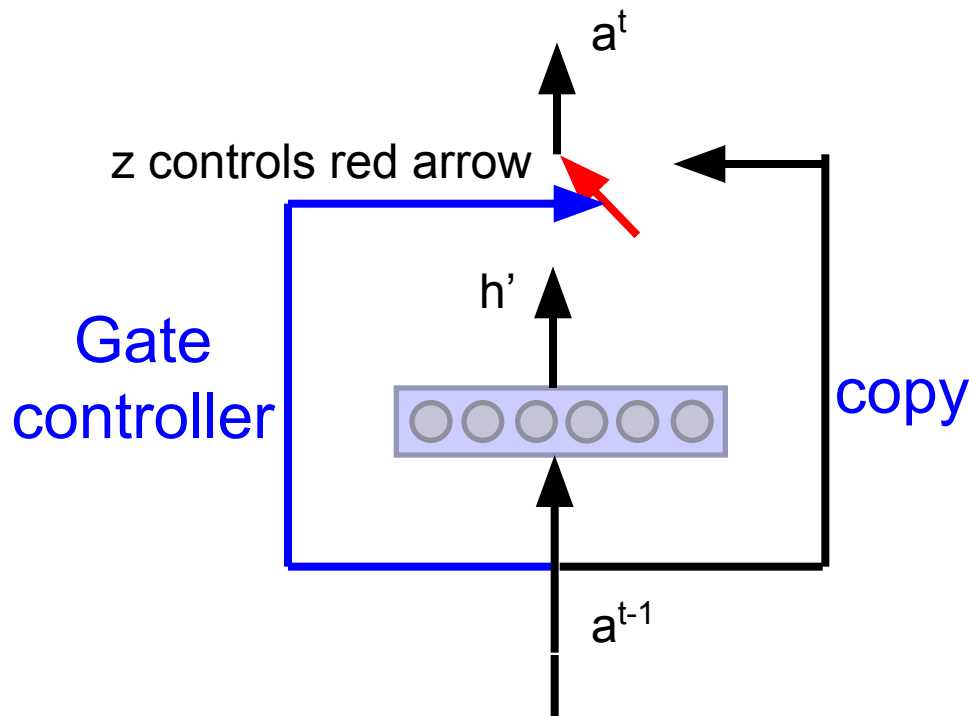
Highway Network

$$h' = \sigma(Wa^{t-1})$$

$$z = \sigma(W'a^{t-1})$$

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

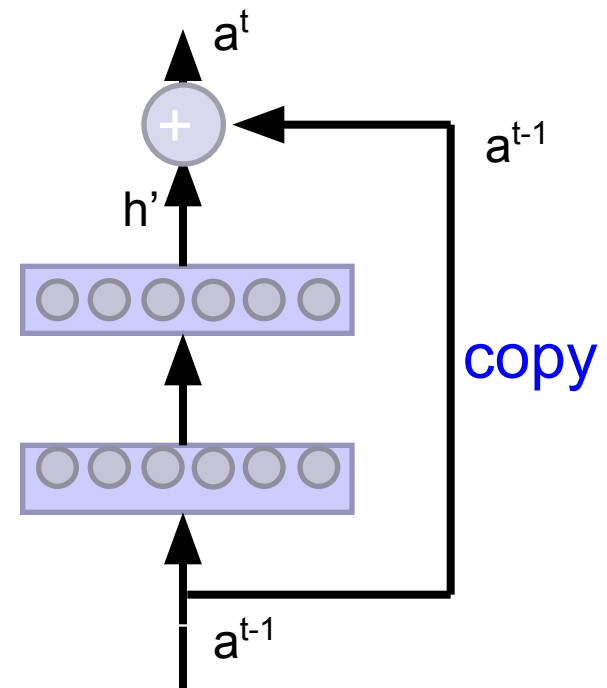
• Highway Network



Training Very Deep Networks

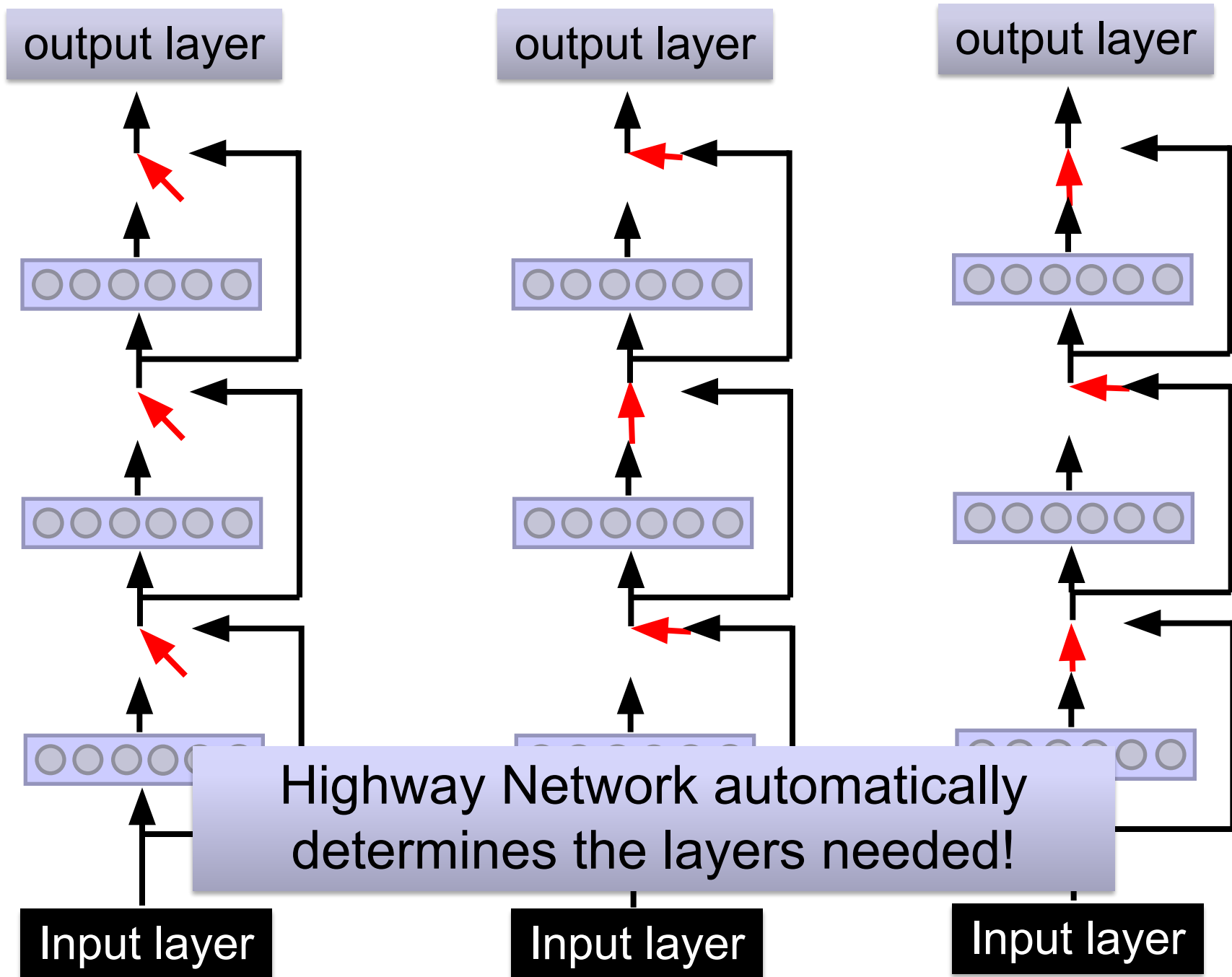
<https://arxiv.org/pdf/1507.06228v2.pdf>

• Residual Network

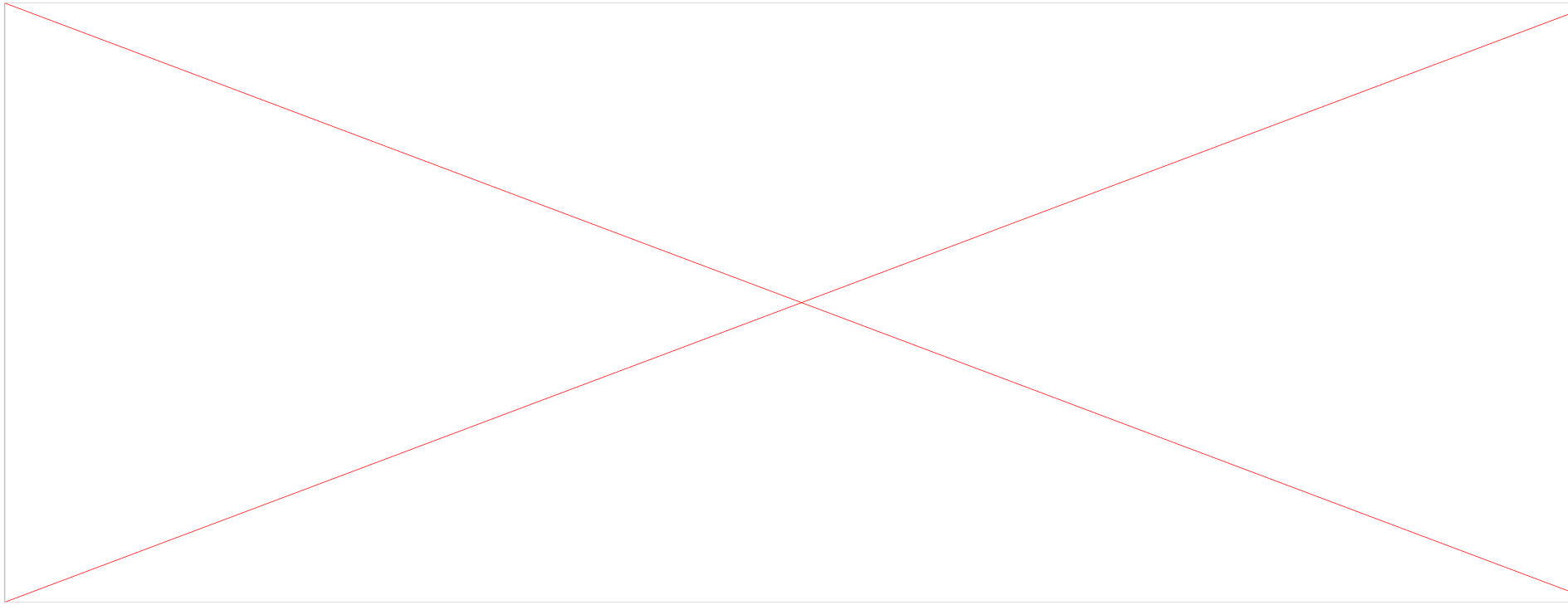
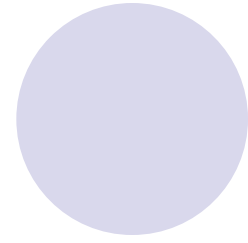
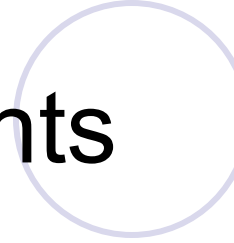
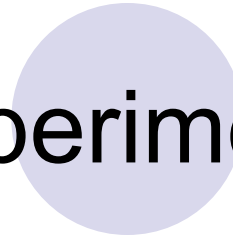
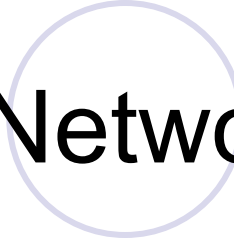


Deep Residual Learning for Image Recognition

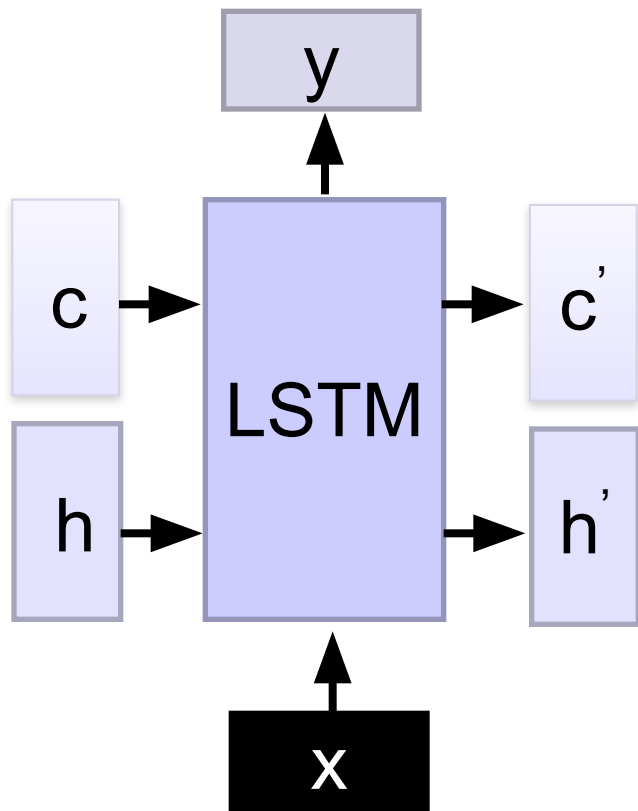
<http://arxiv.org/abs/1512.03385>



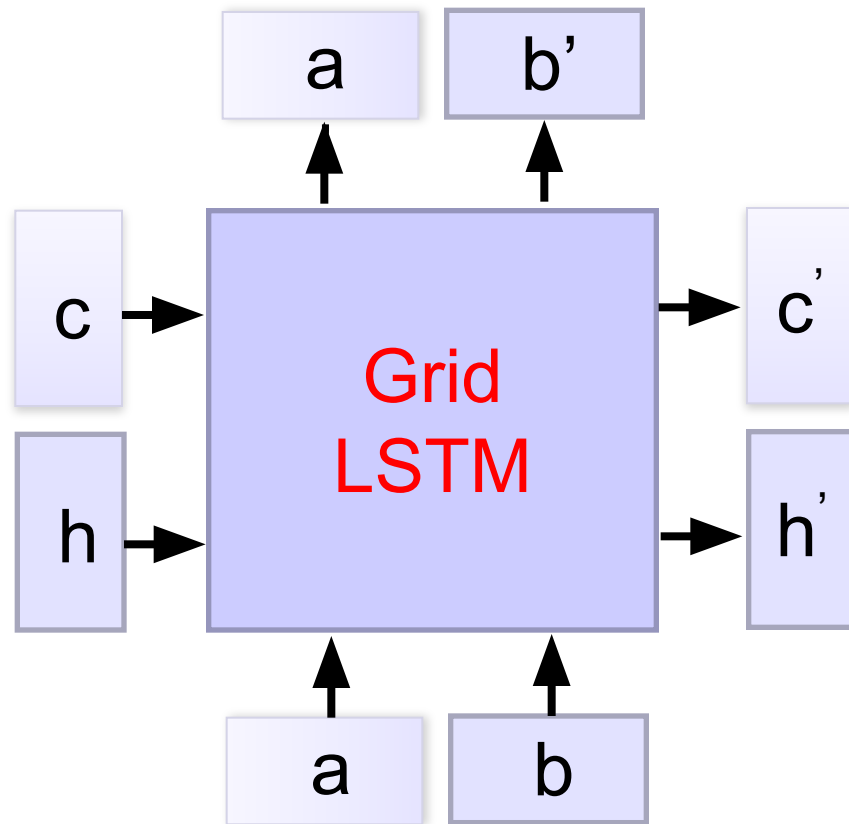
Highway Network Experiments



Grid LSTM



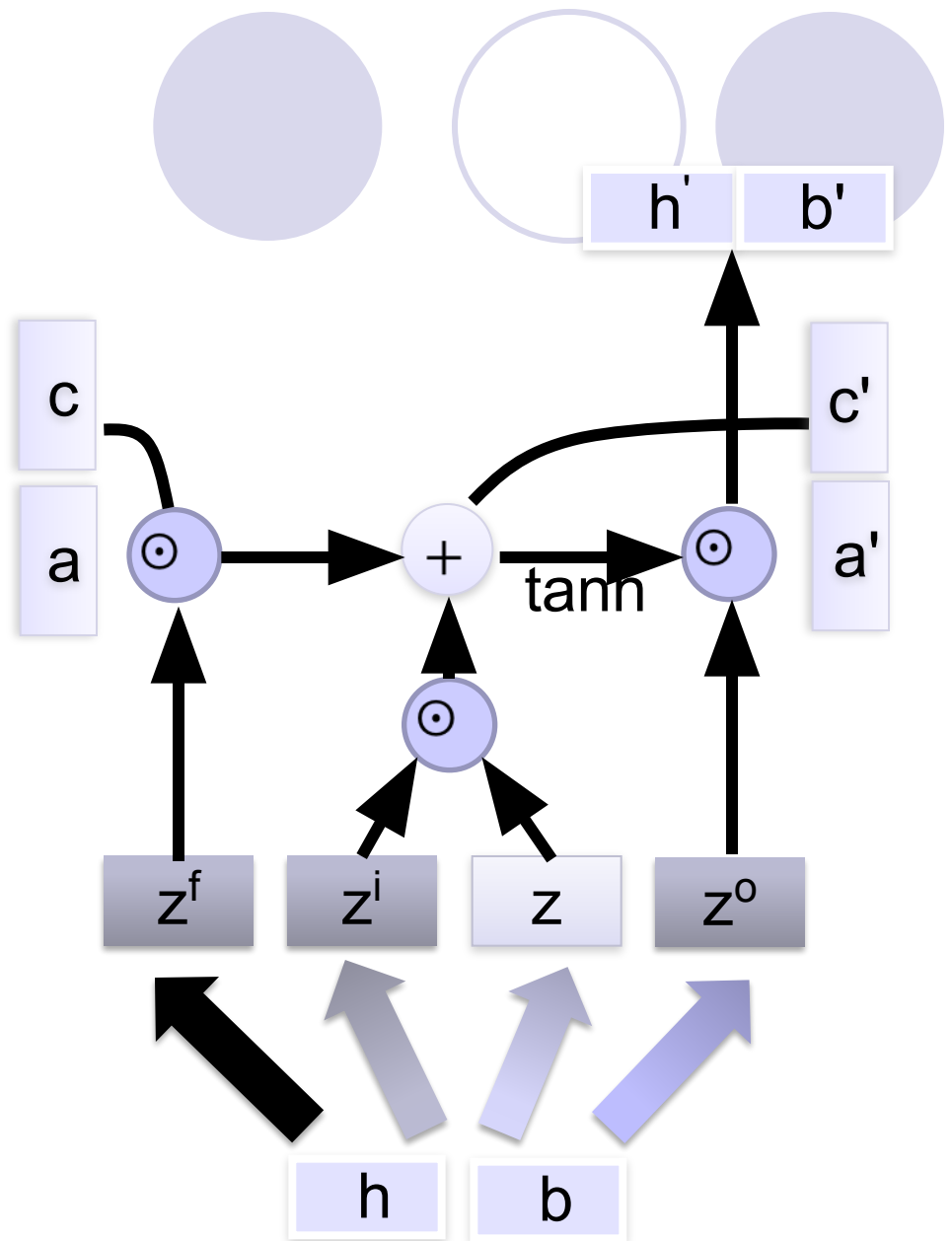
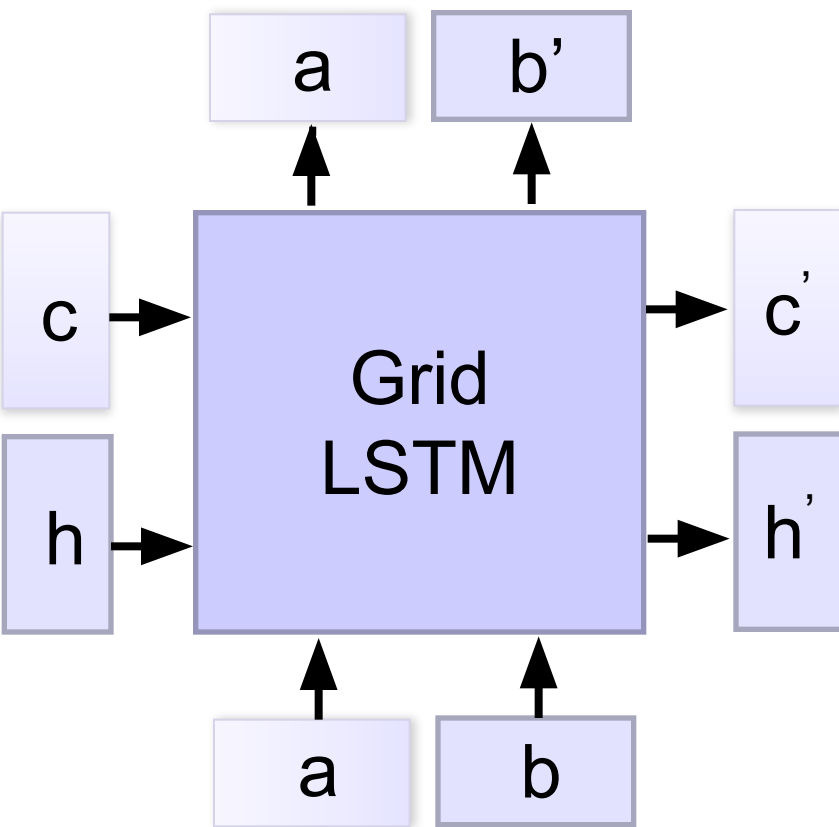
depth



Memory for both
time and **depth**

time

Grid LSTM

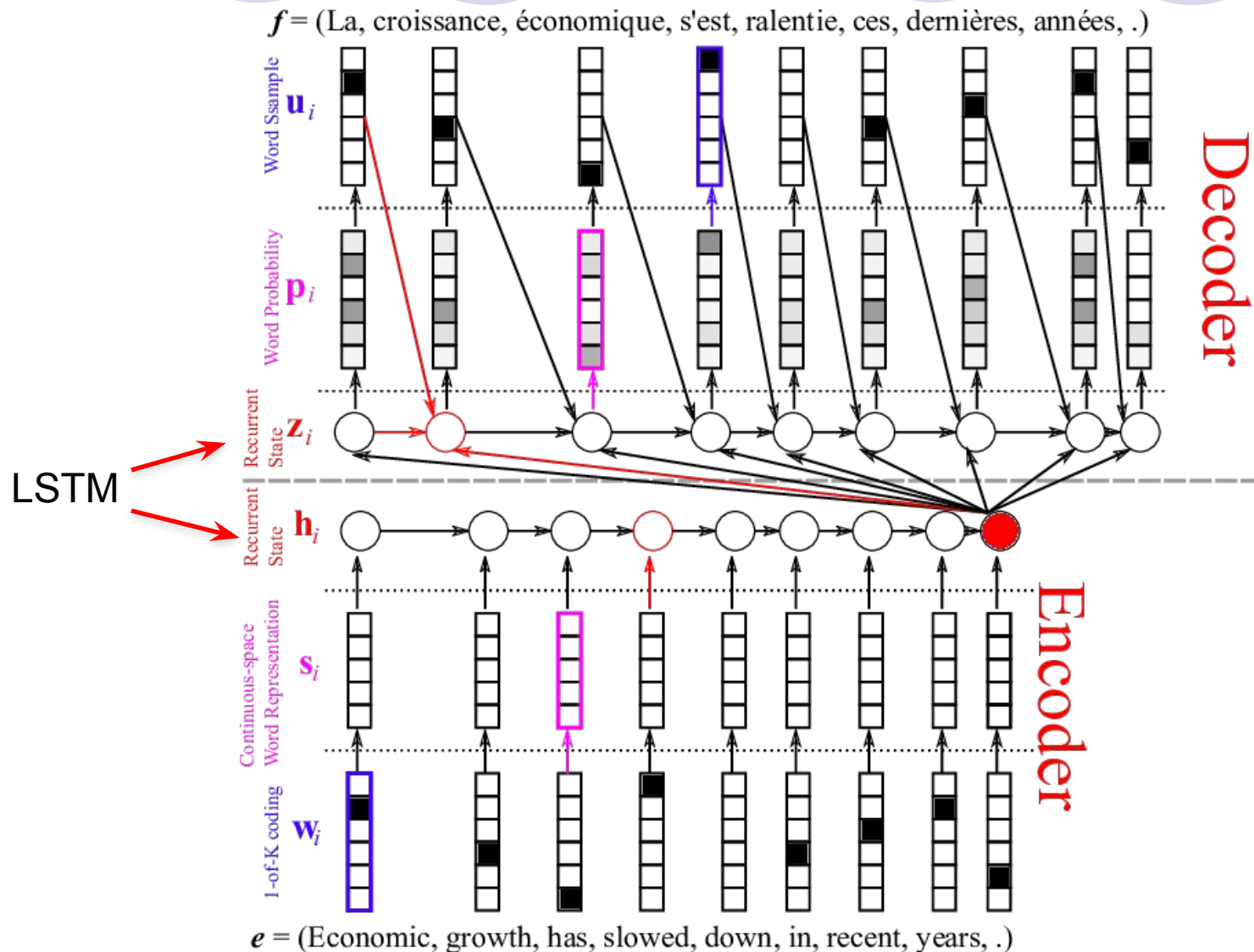


You can generalize this to 3D, and more.

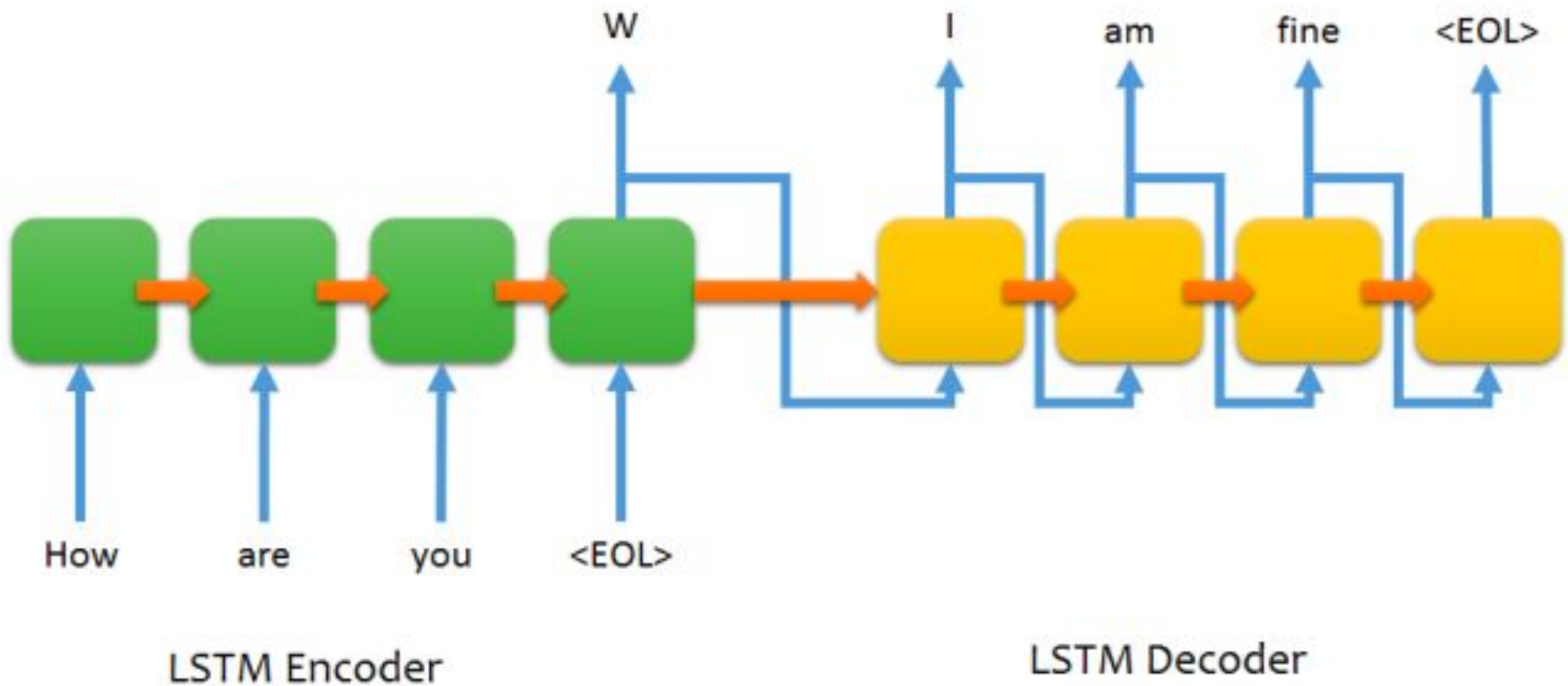
Applications of LSTM / RNN



Neural machine translation



Sequence to sequence chat model

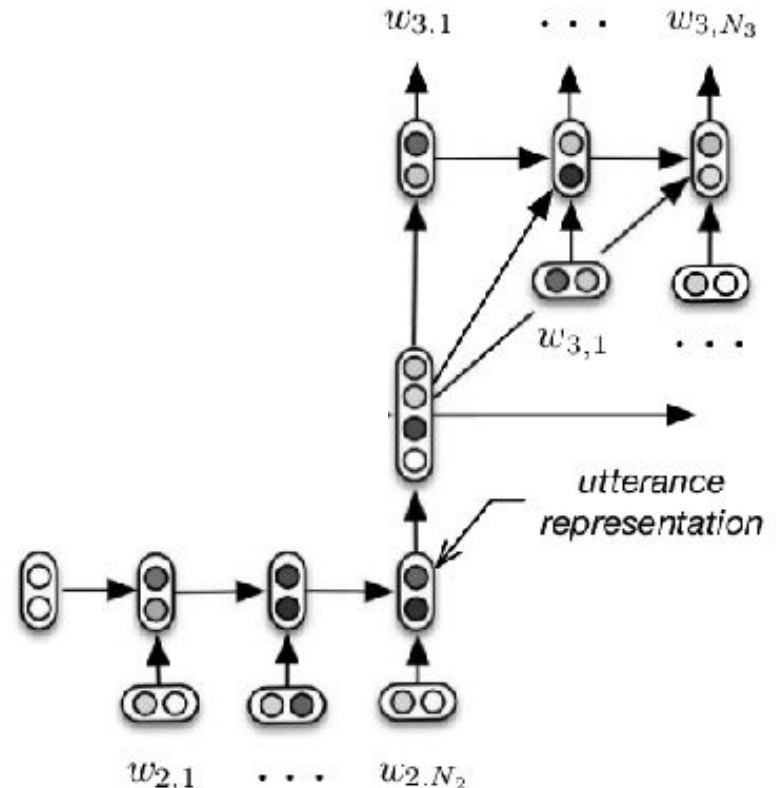
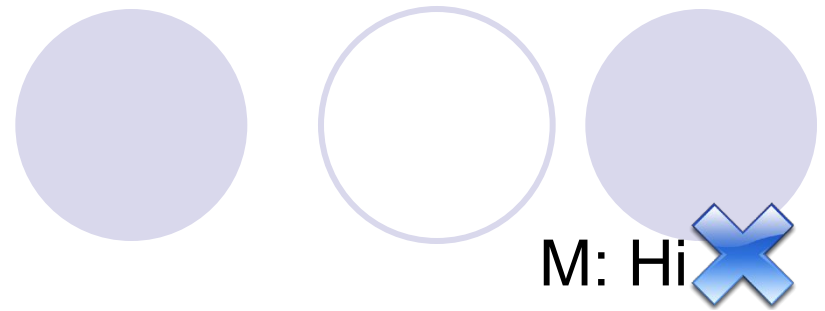


Chat with context

M: Hello

U: Hi

M: Hi



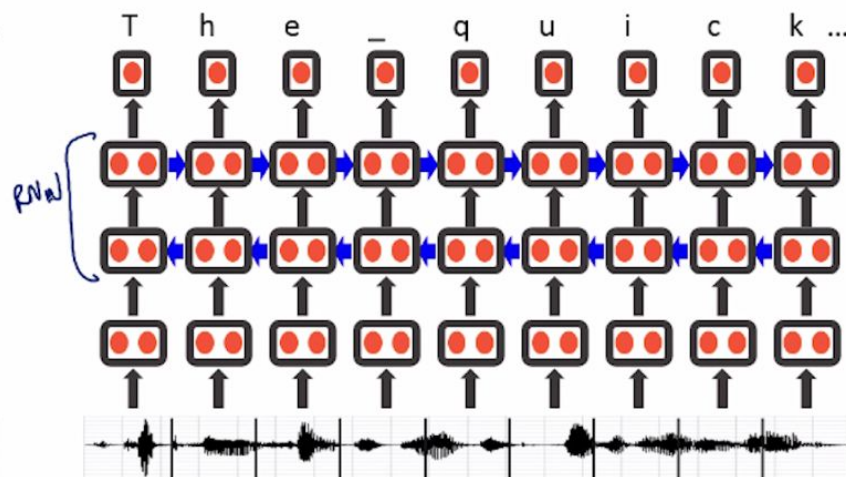
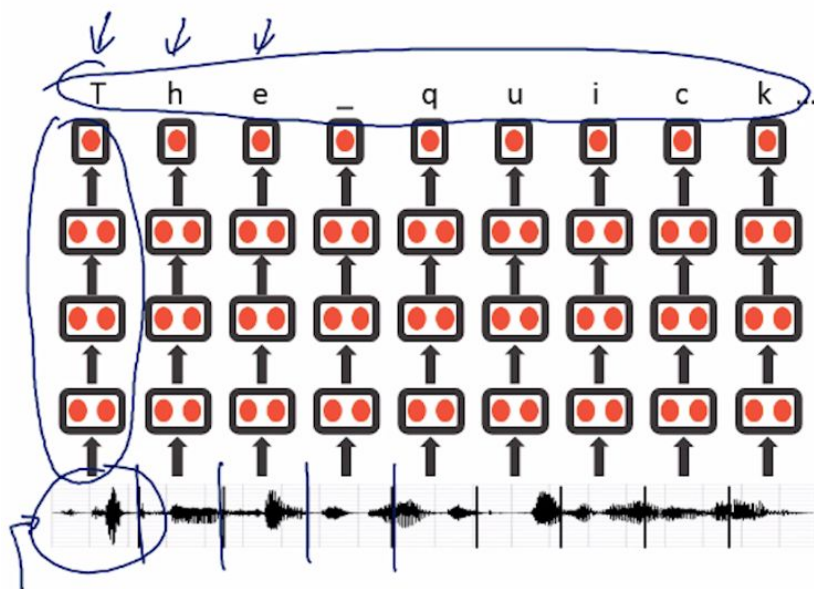
M: Hello

U: Hi

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"

Baidu's speech recognition using RNN

Speech recognition example (Deep Speech)



Attention

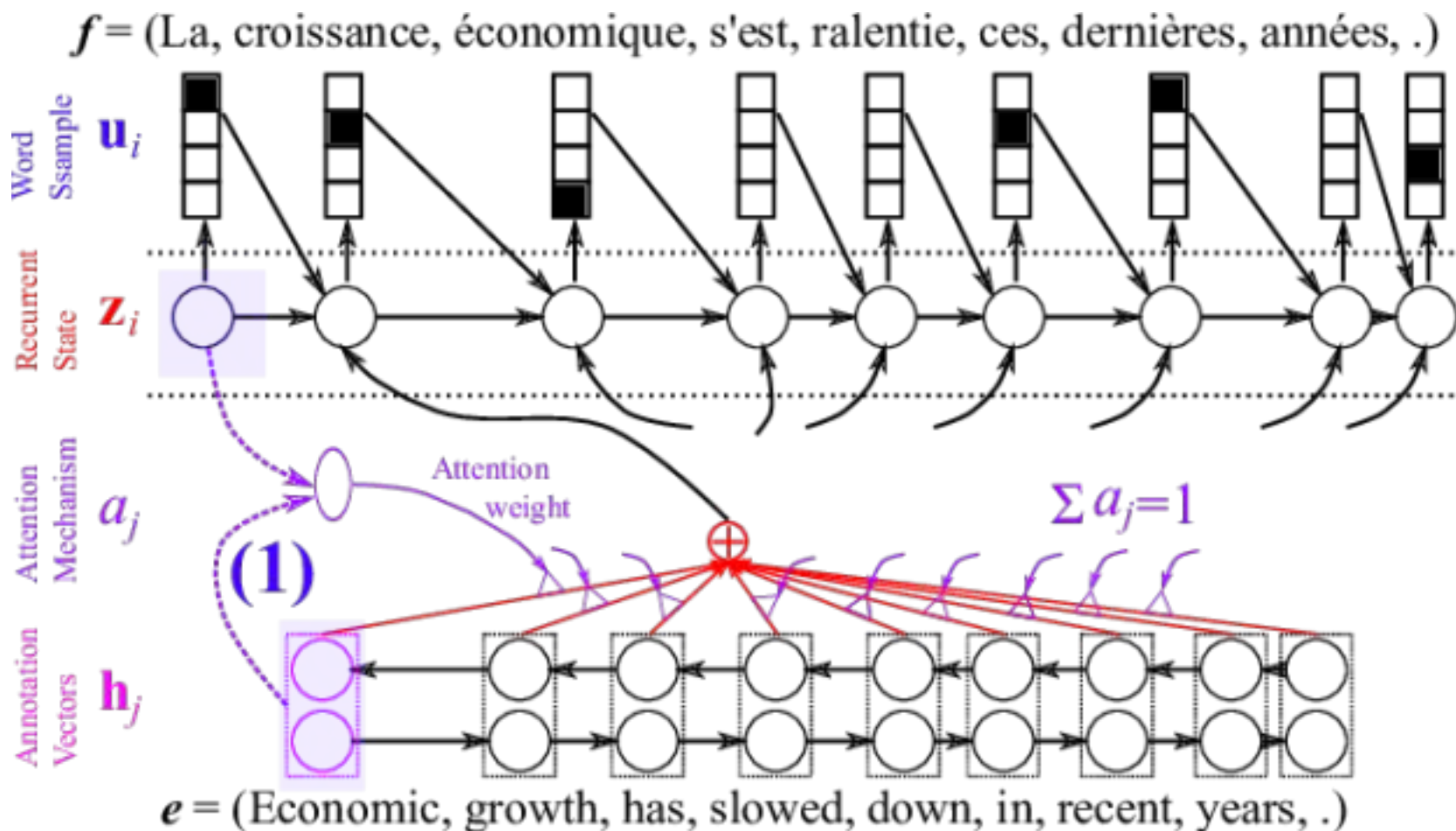


Image caption generation using attention

(From CY Lee lecture)

z^0 is initial parameter, it is also learned

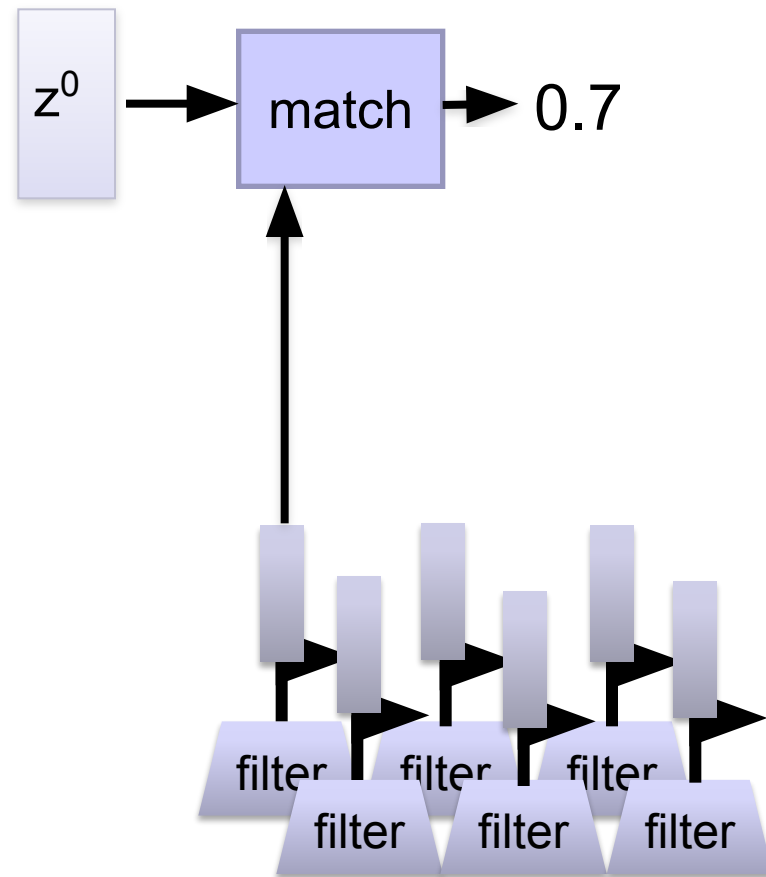
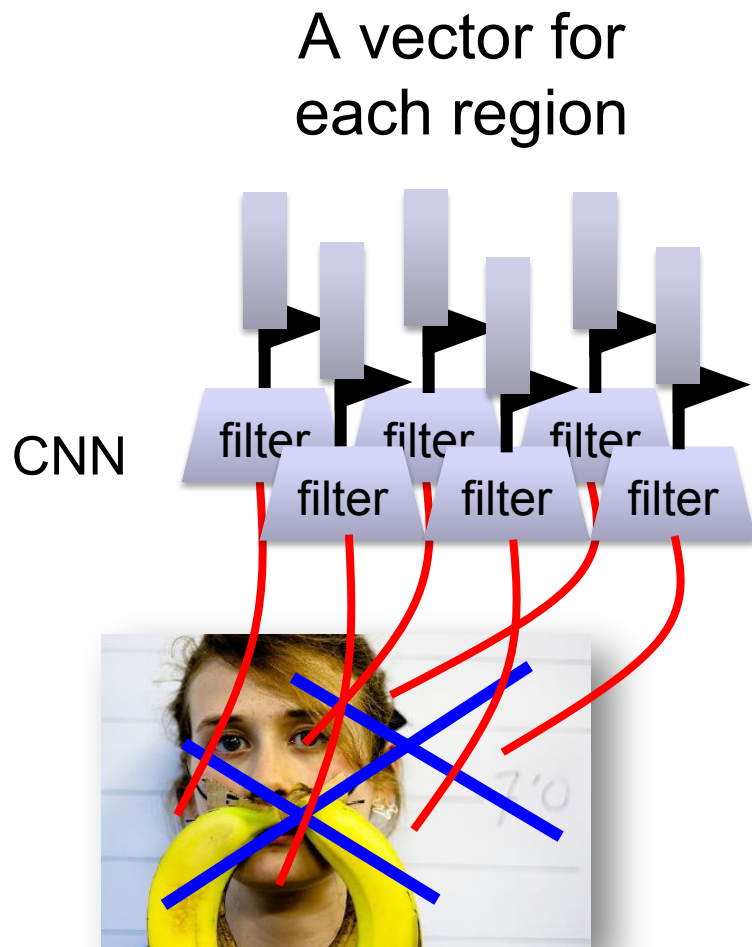


Image Caption Generation

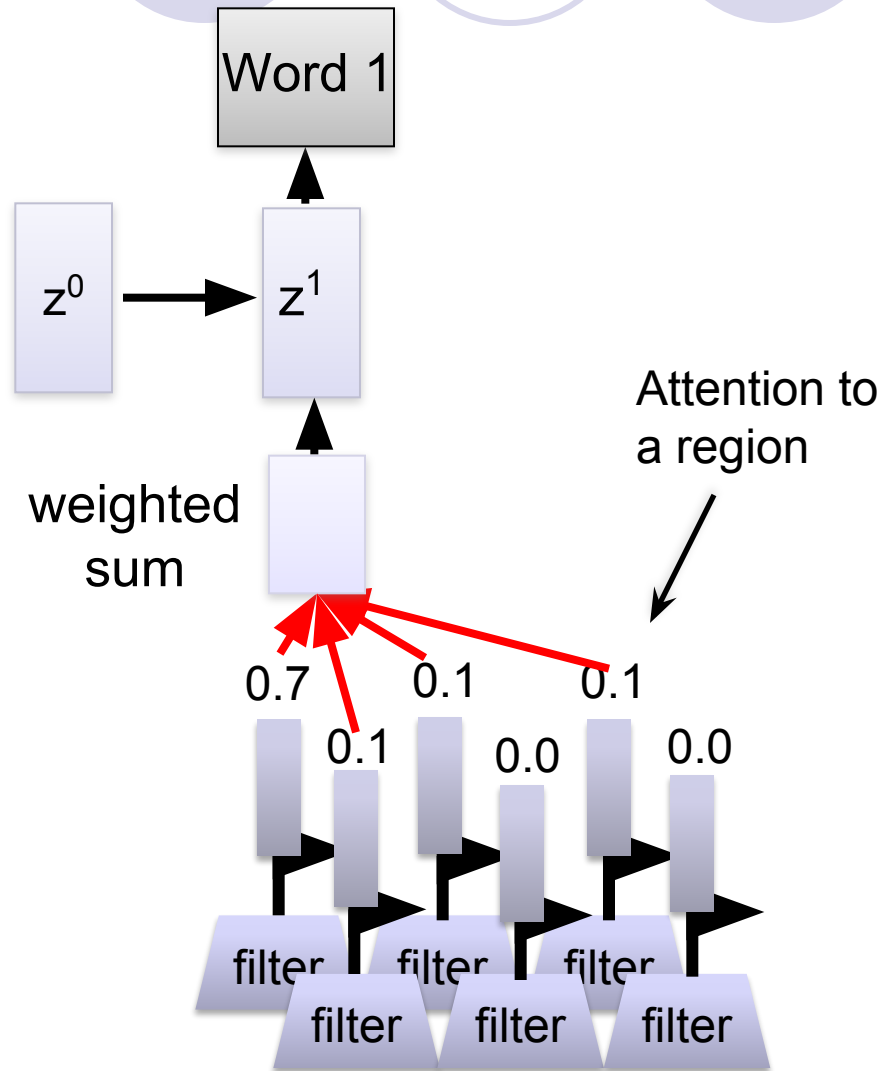
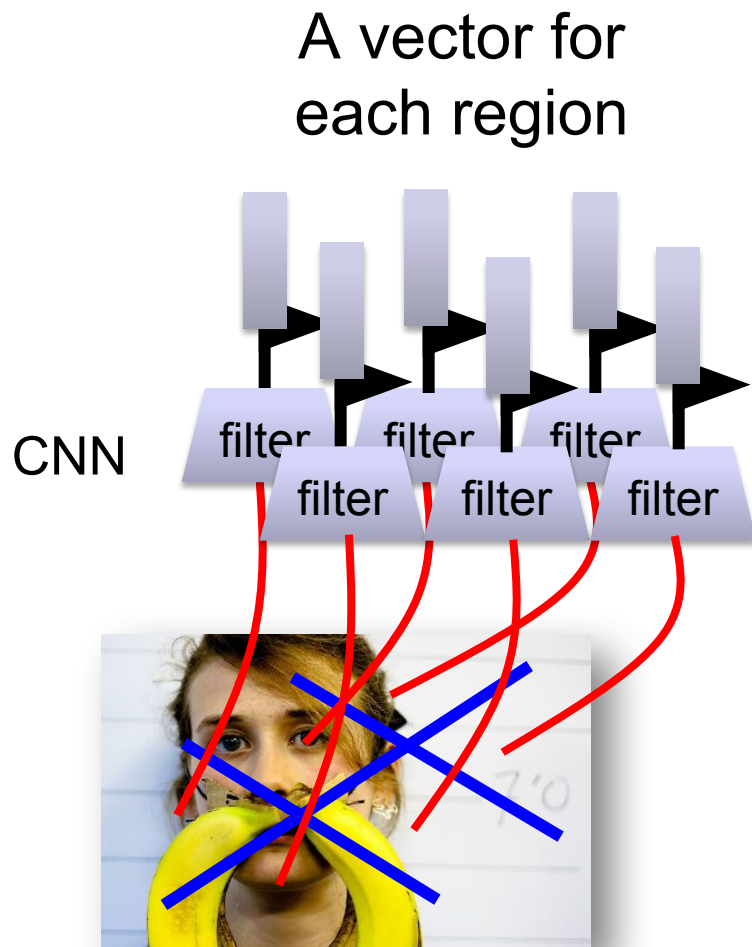


Image Caption Generation

A vector for
each region

CNN

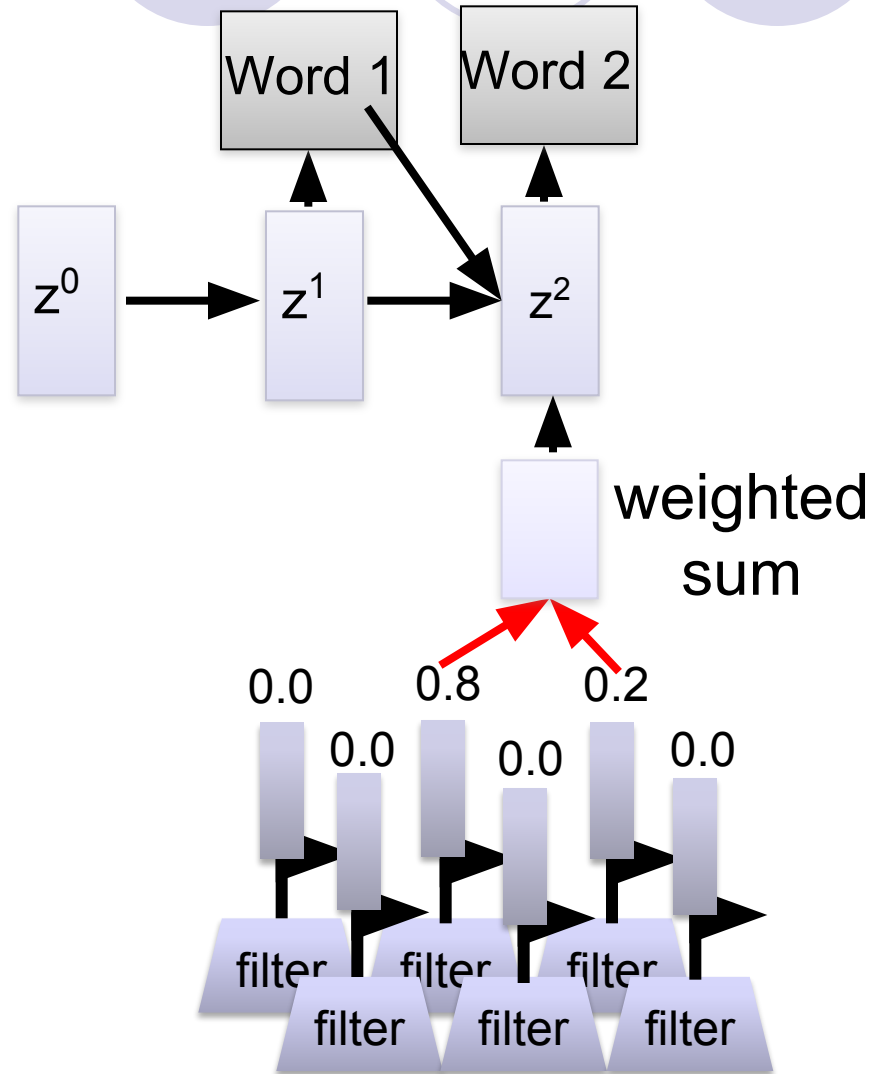
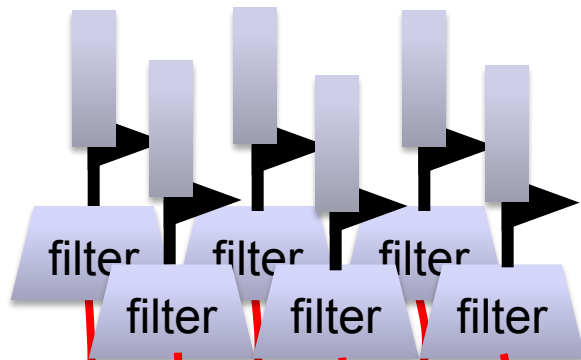
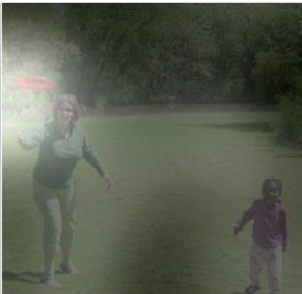


Image Caption Generation



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.



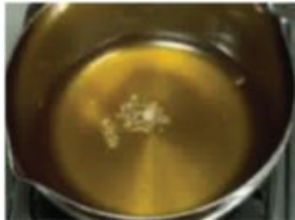
A man is talking on his cell phone while another man watches.

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**



* Possible project?

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