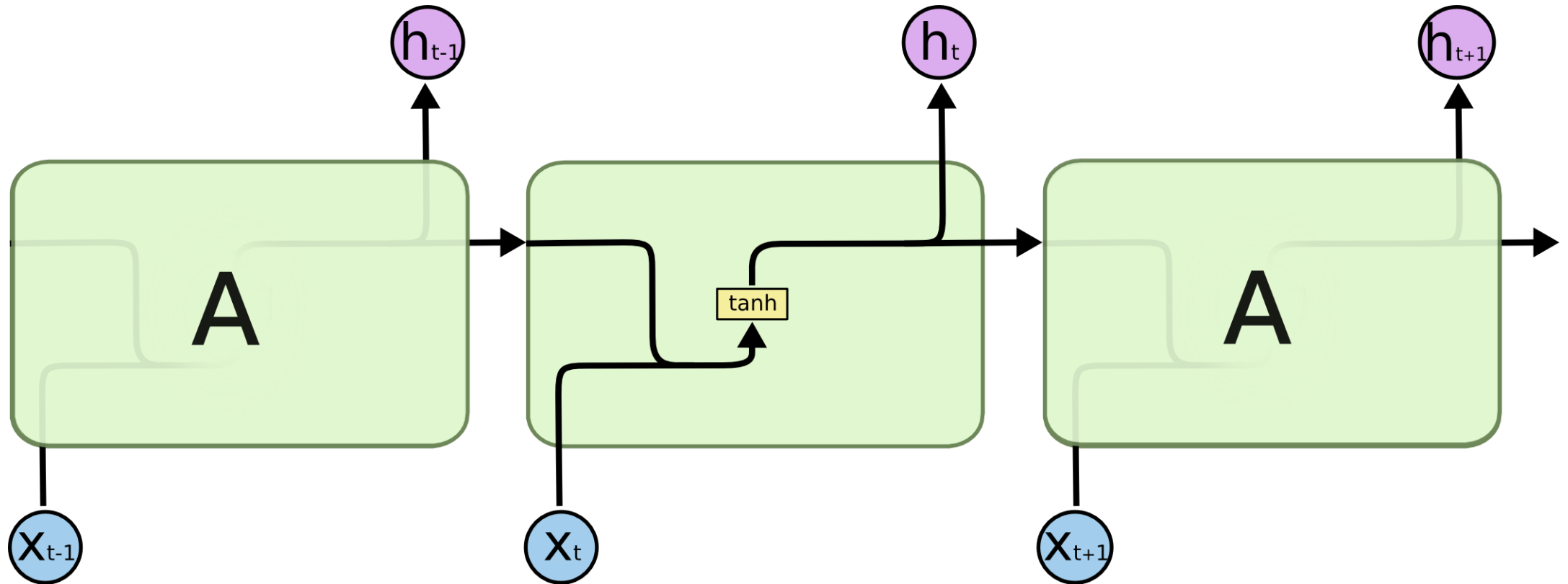


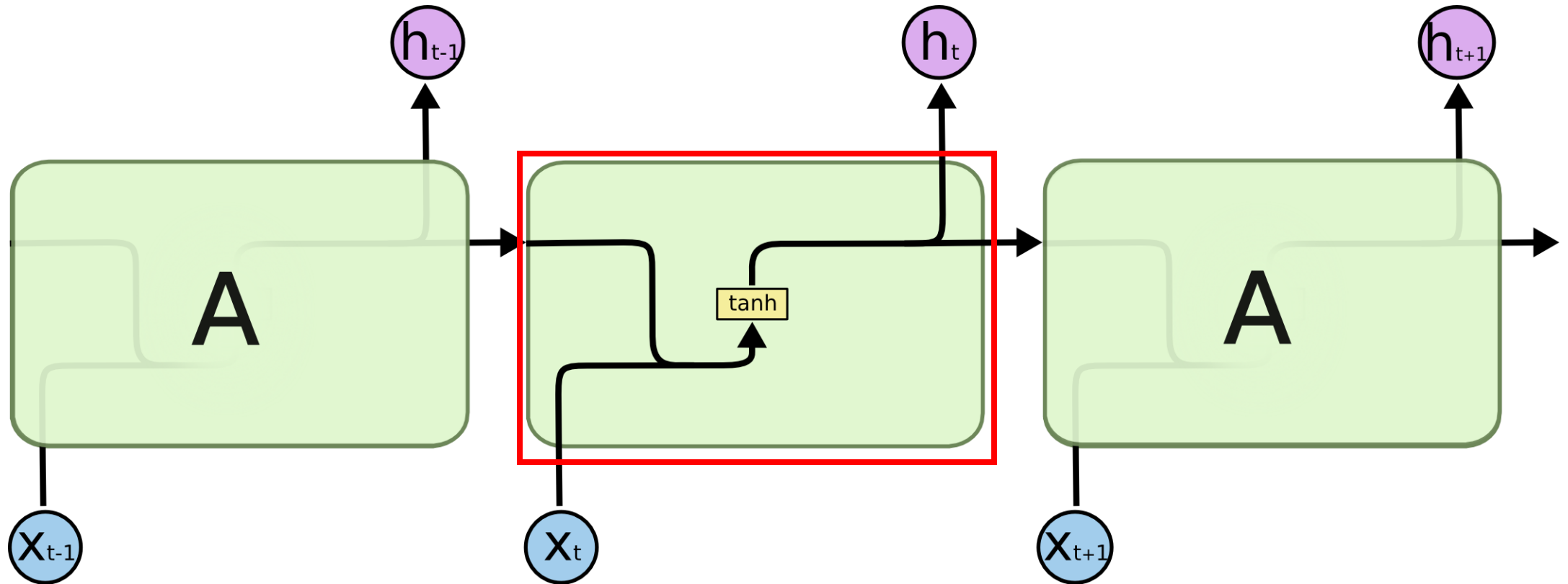
Long-Short Term Memory and RNN Architectures

Neural Networks Design And Application

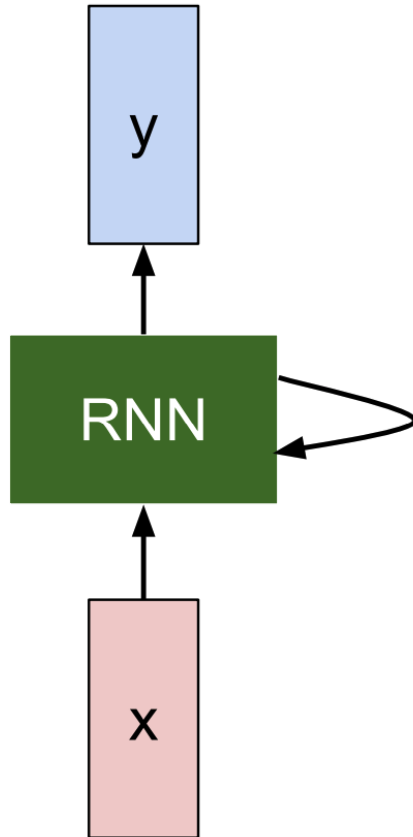
A conventional RNN



A conventional RNN



Recurrent neural networks



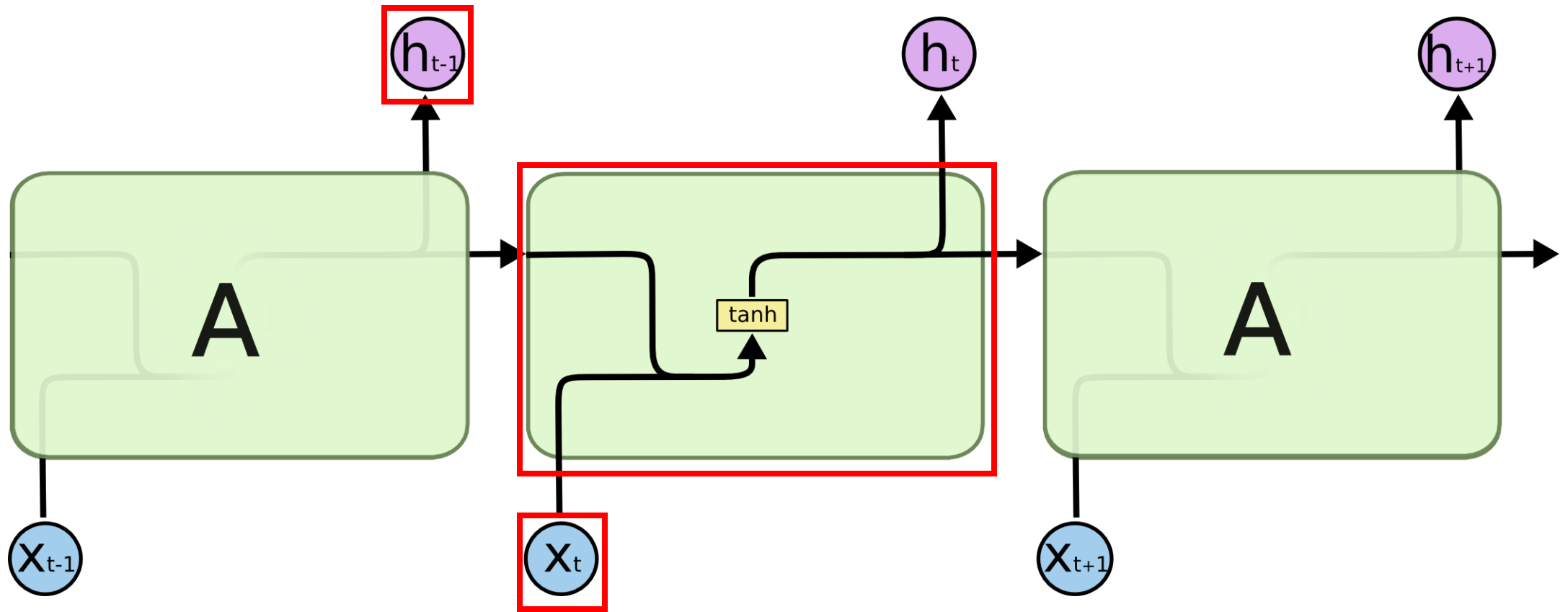
$$h_t = f \left[W * \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \right]$$

$$f = \tanh(\cdot)$$

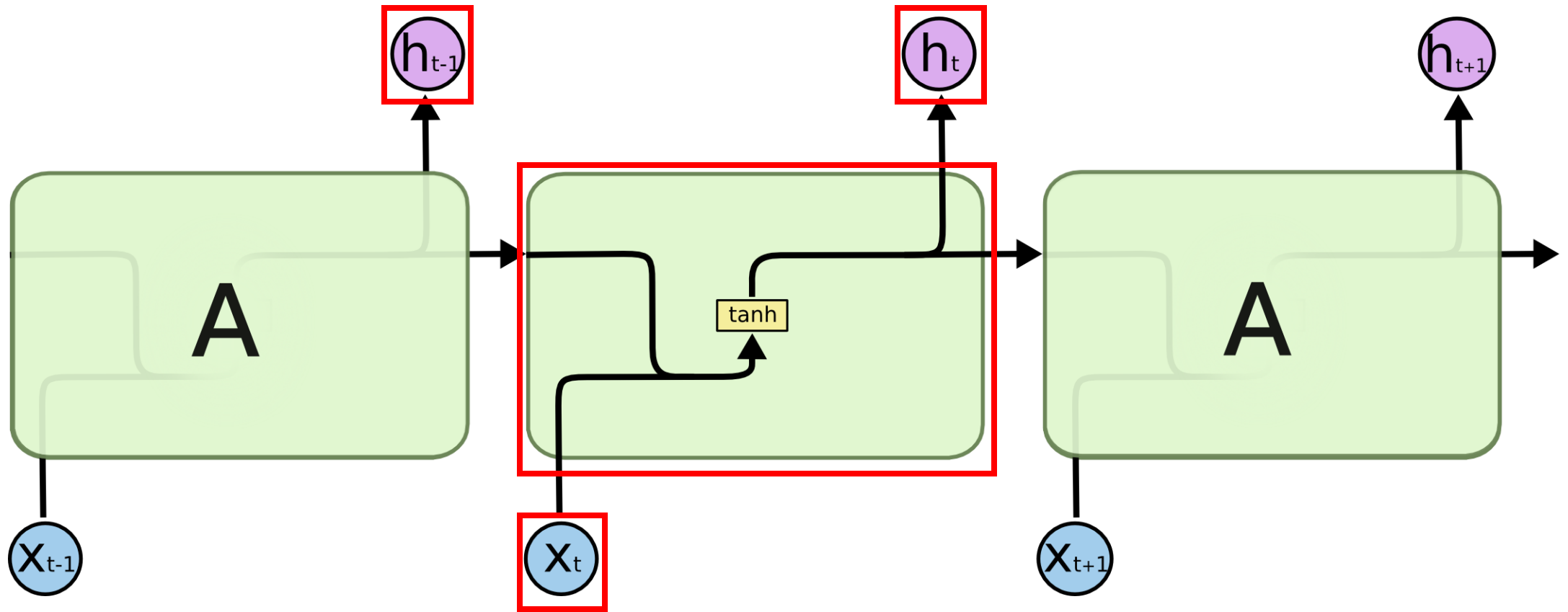
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state / some function with parameters W / old state / input vector at some time step

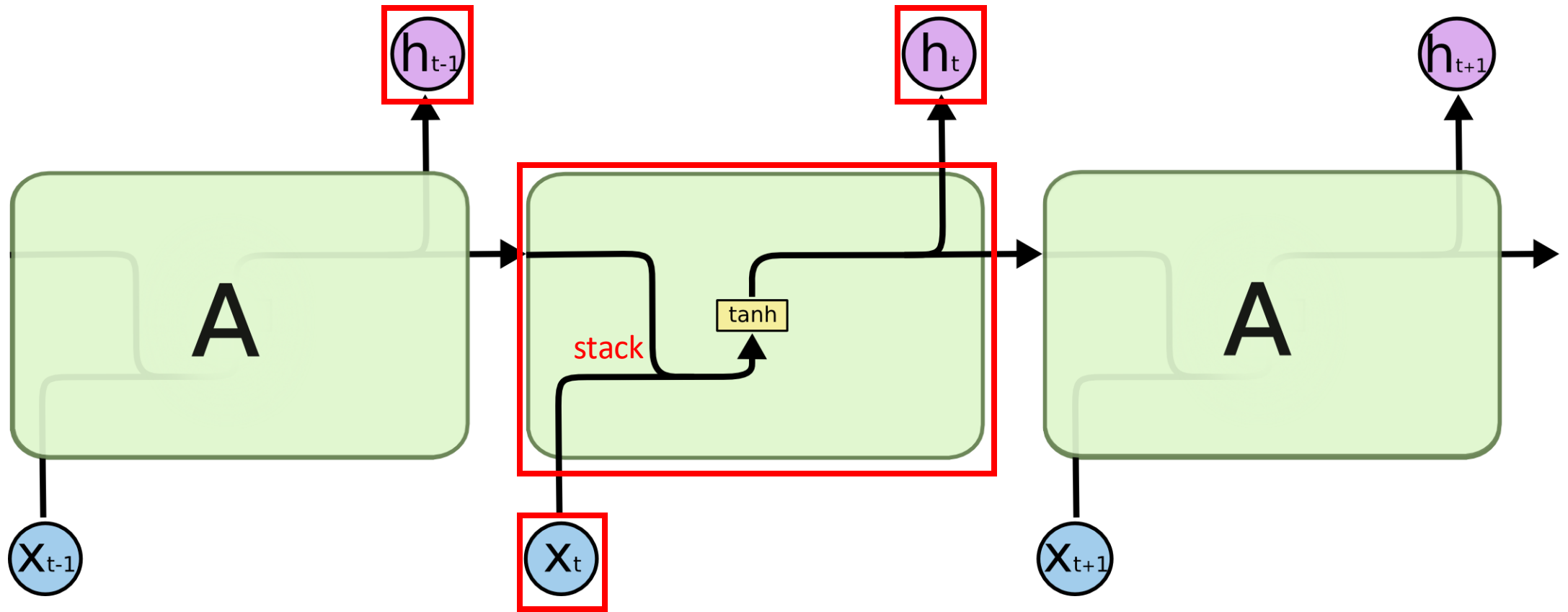
A conventional RNN



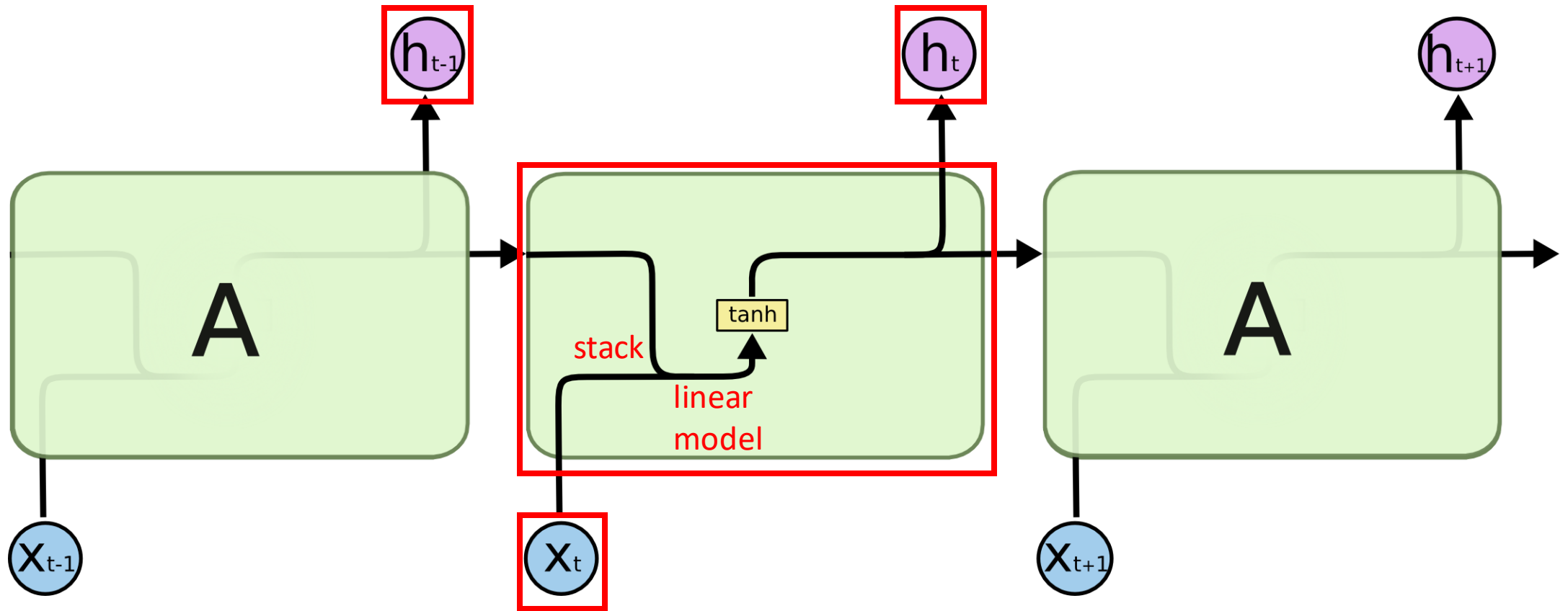
A conventional RNN



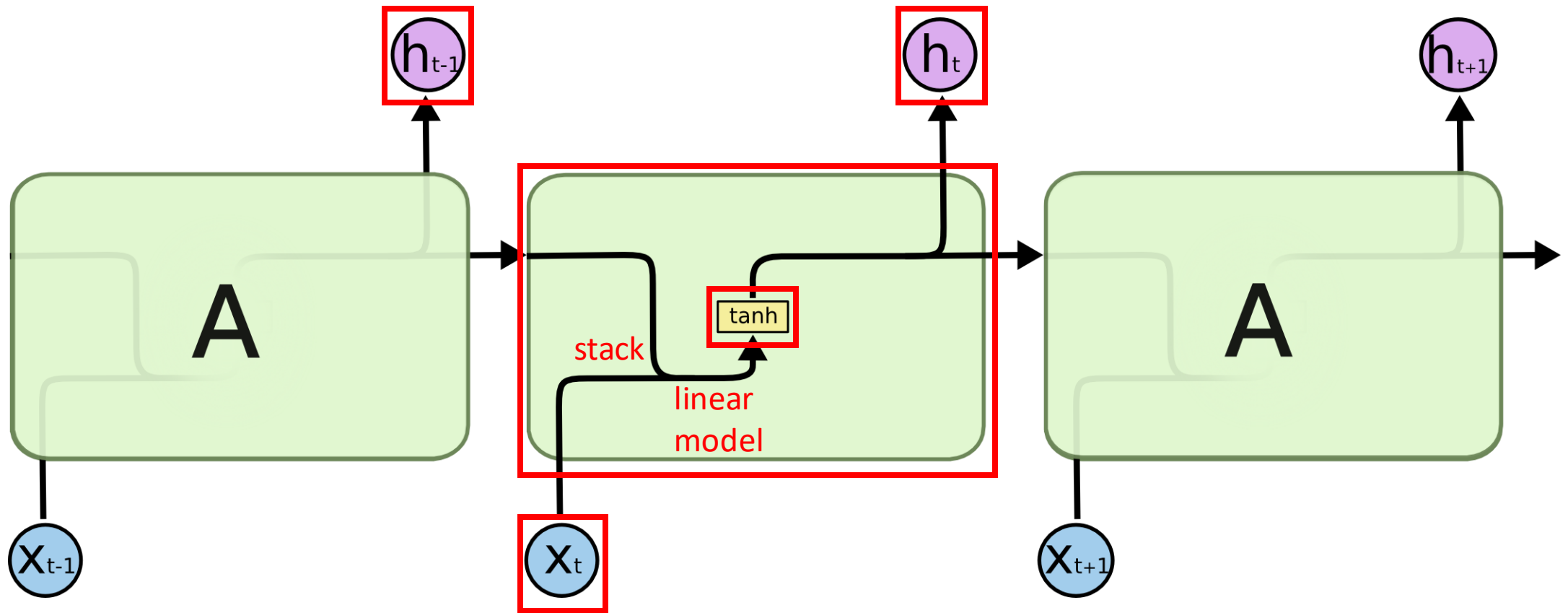
A conventional RNN



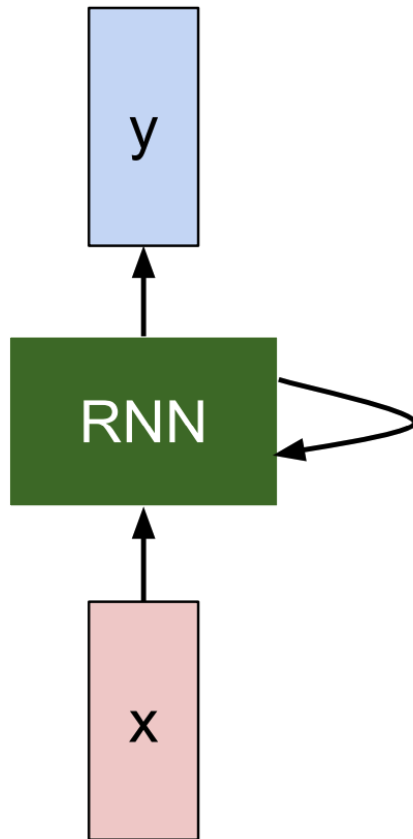
A conventional RNN



A conventional RNN



Recurrent neural networks



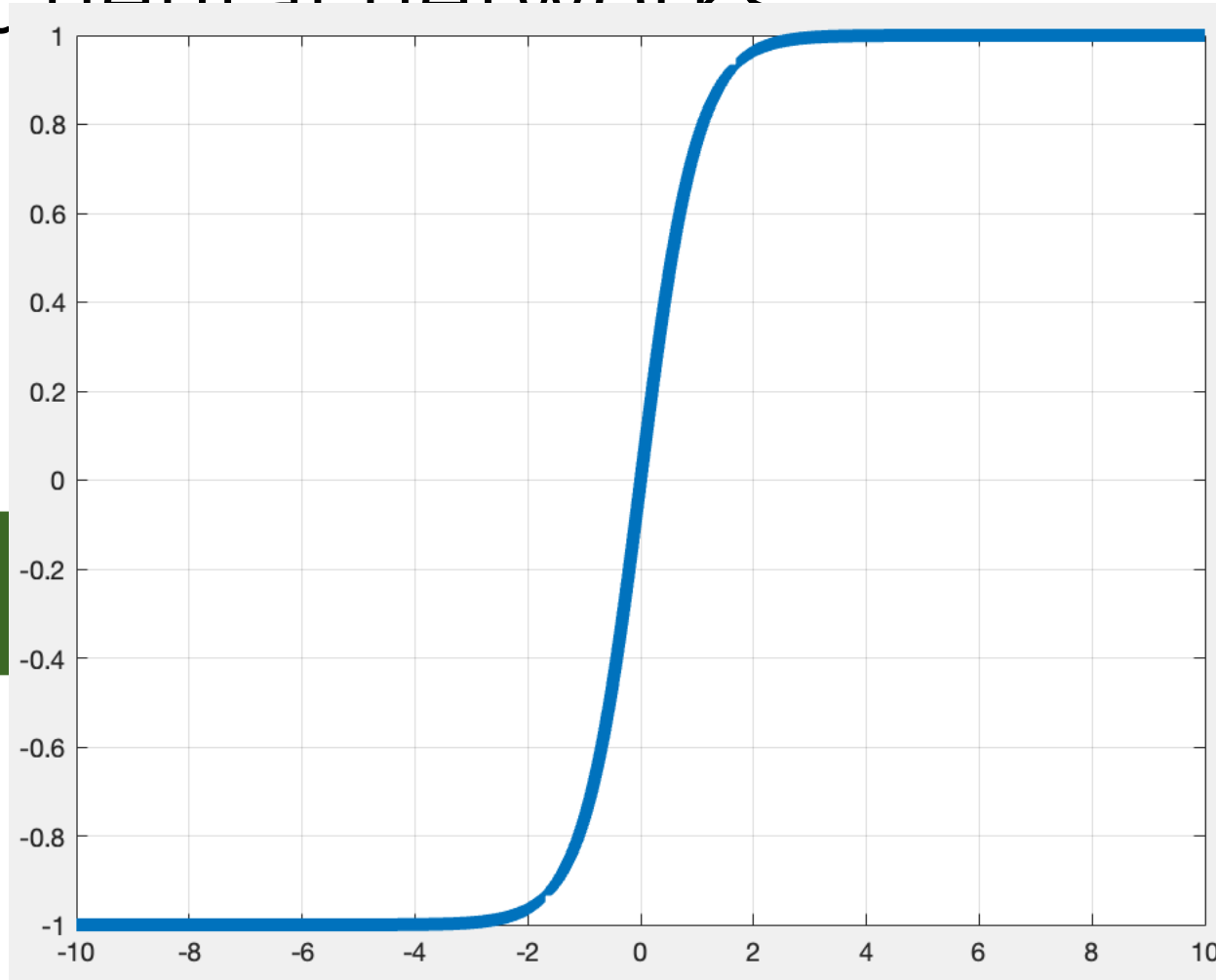
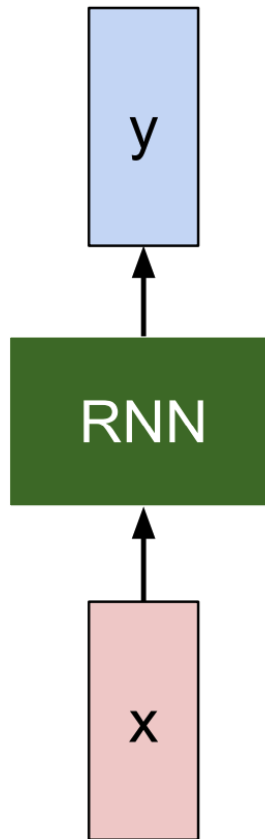
$$h_t = f \left[W * \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \right]$$

$$f = \tanh(\cdot)$$

$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state / some function with parameters W / old state / input vector at some time step

Recurrent neural networks



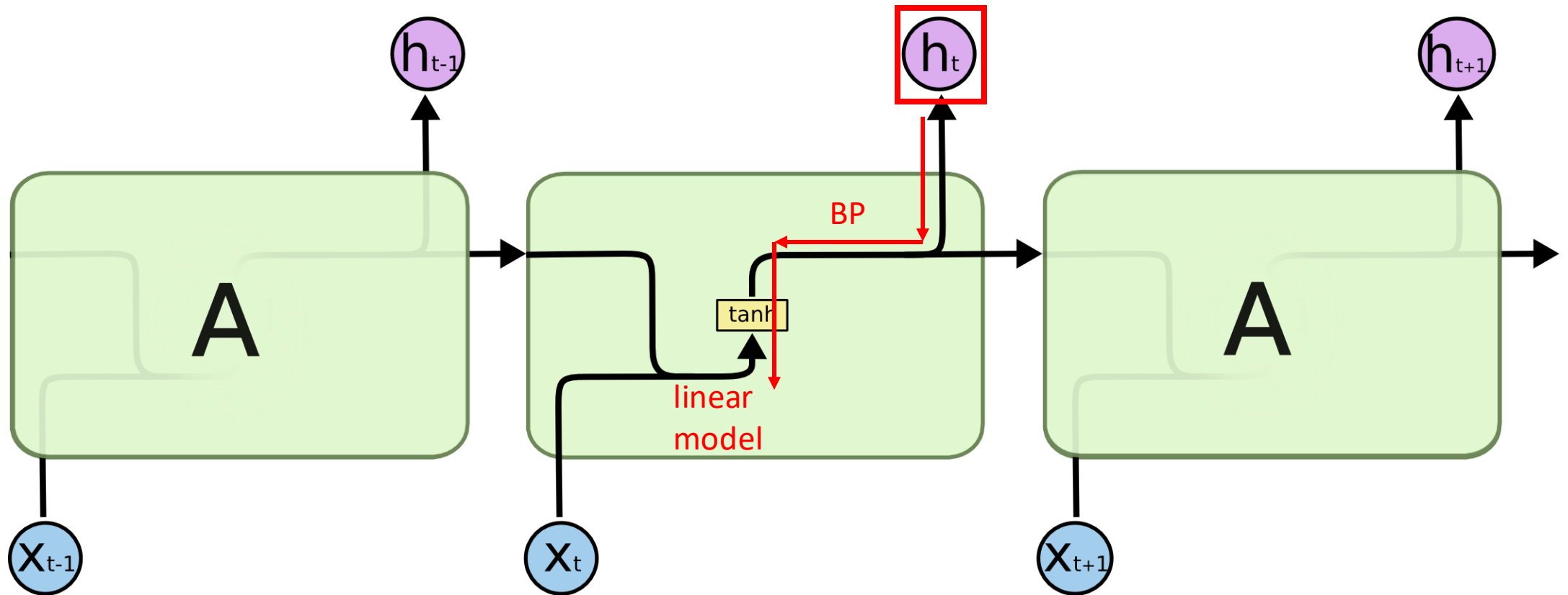
tanh function

$$f = \tanh(\cdot)$$

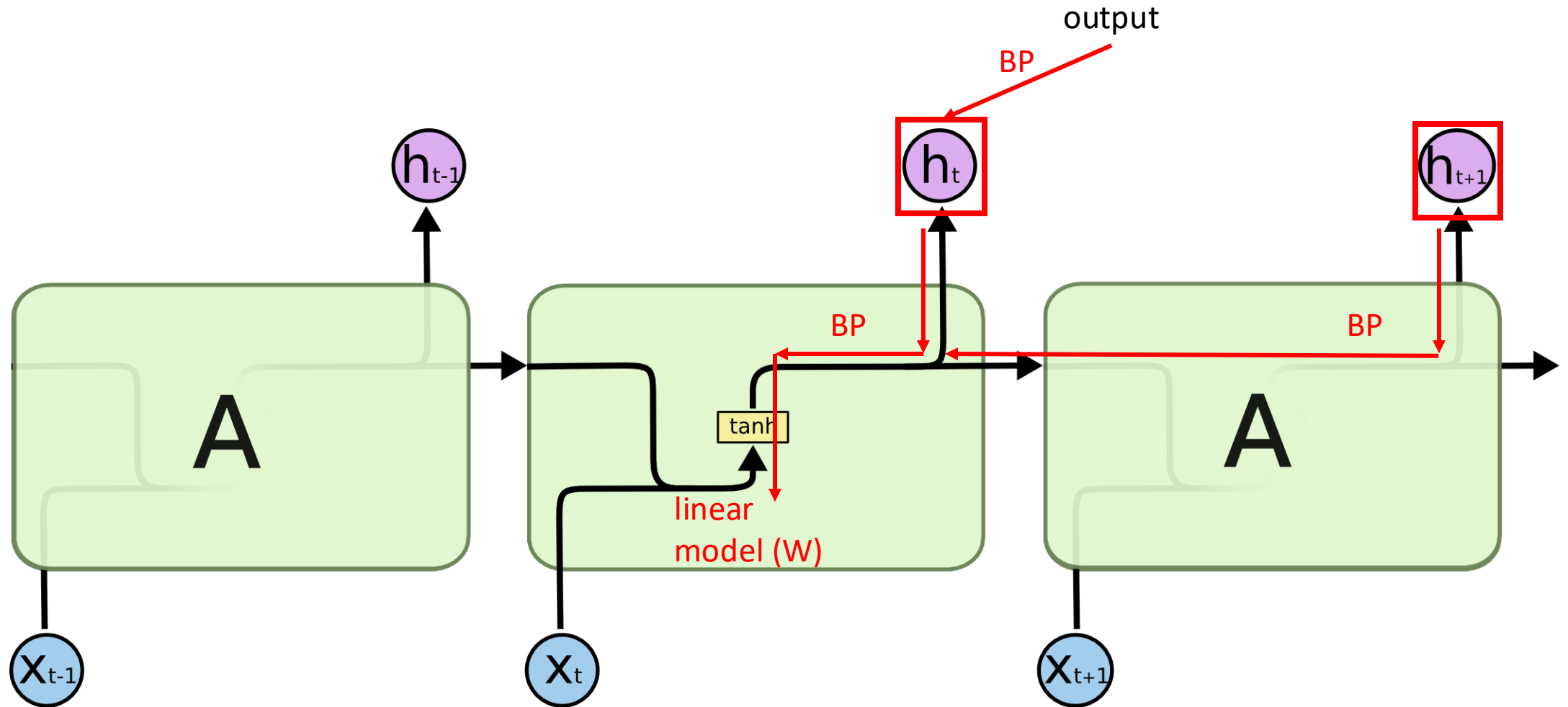
c_t

put vector at
same time step

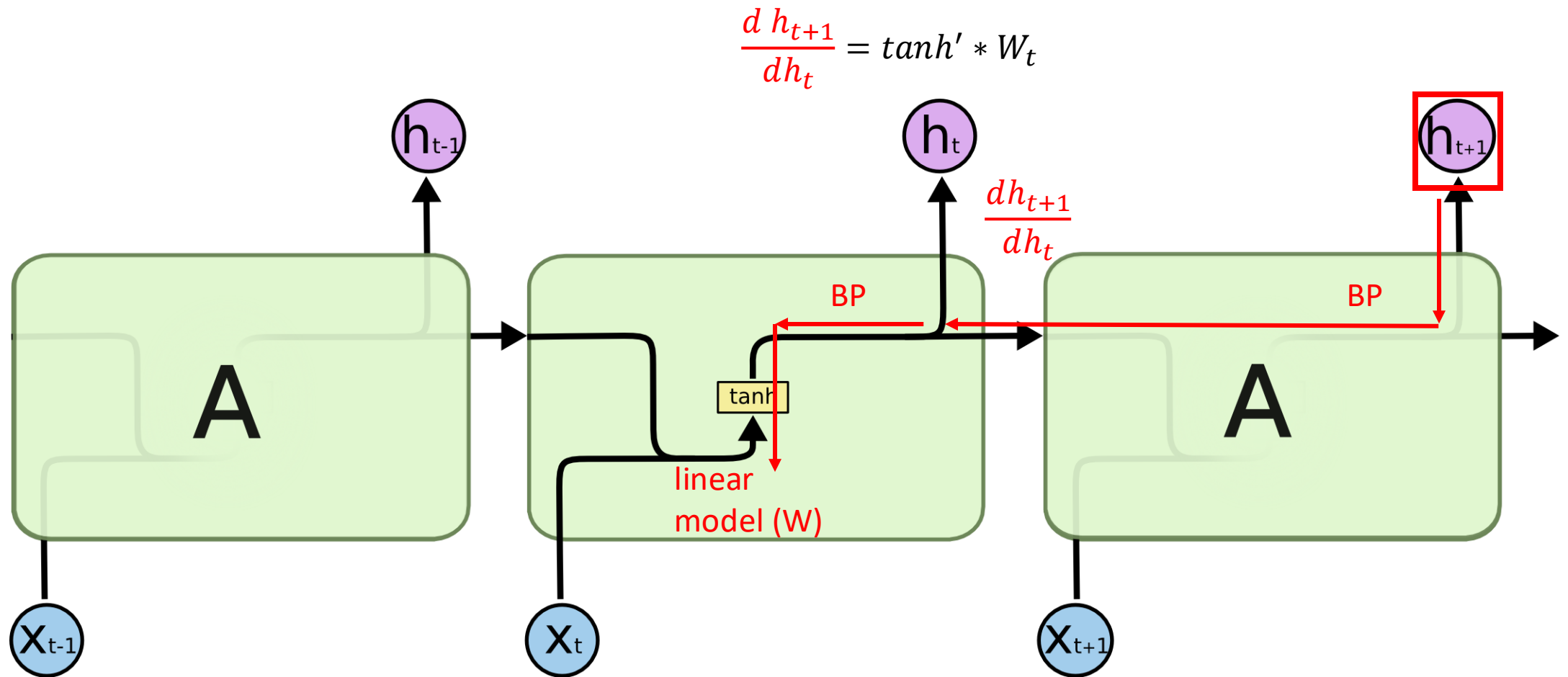
A conventional RNN



A conventional RNN



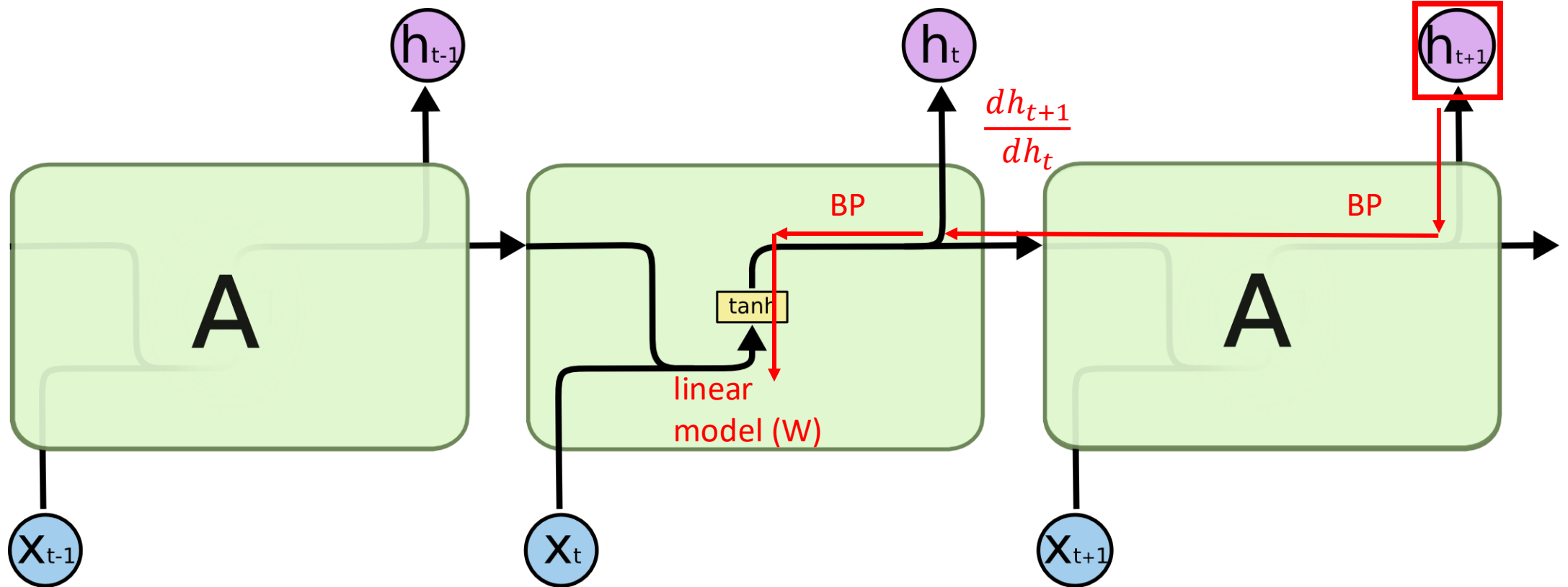
A conventional RNN



A conventional RNN

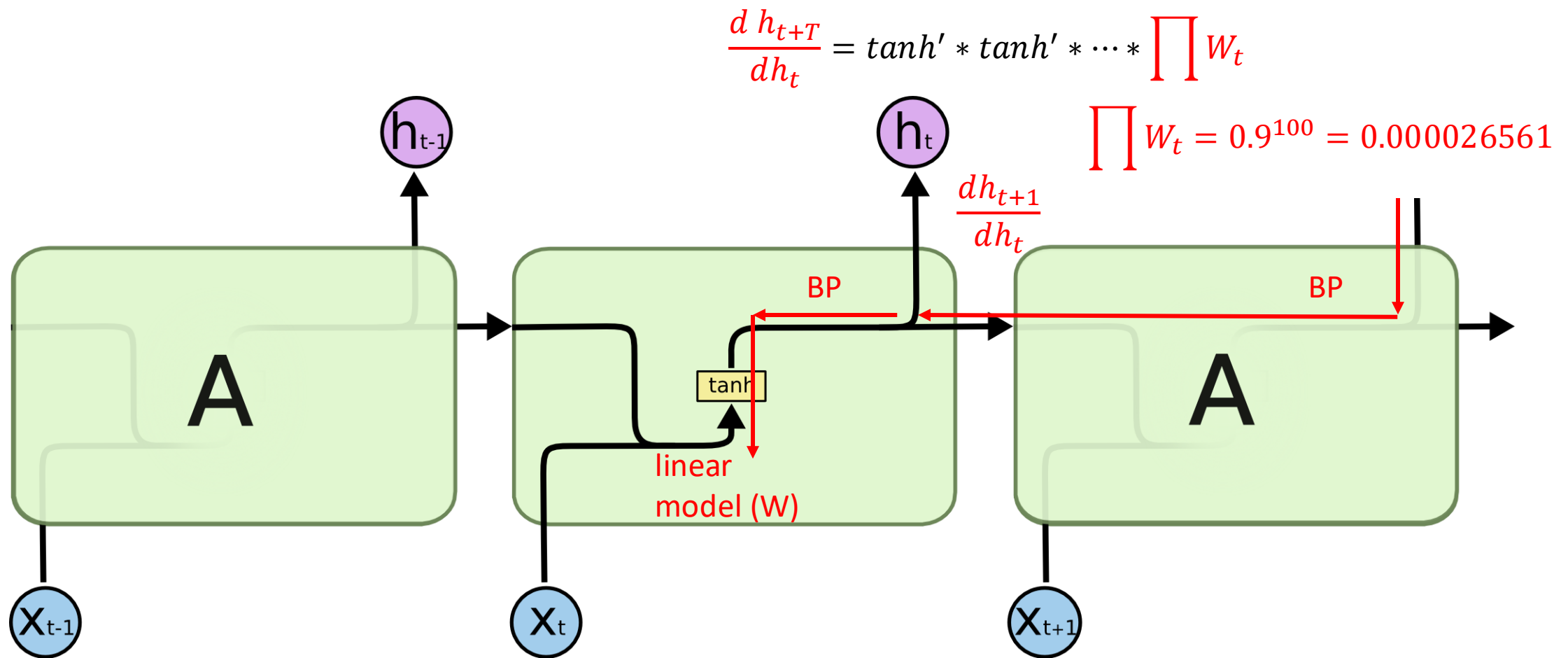
Q: what issue will we have?

$$\frac{d h_{t+T}}{d h_t} = \tanh' * \tanh' * \dots * \prod W_t$$



A conventional RNN

Q: what issue will we have?



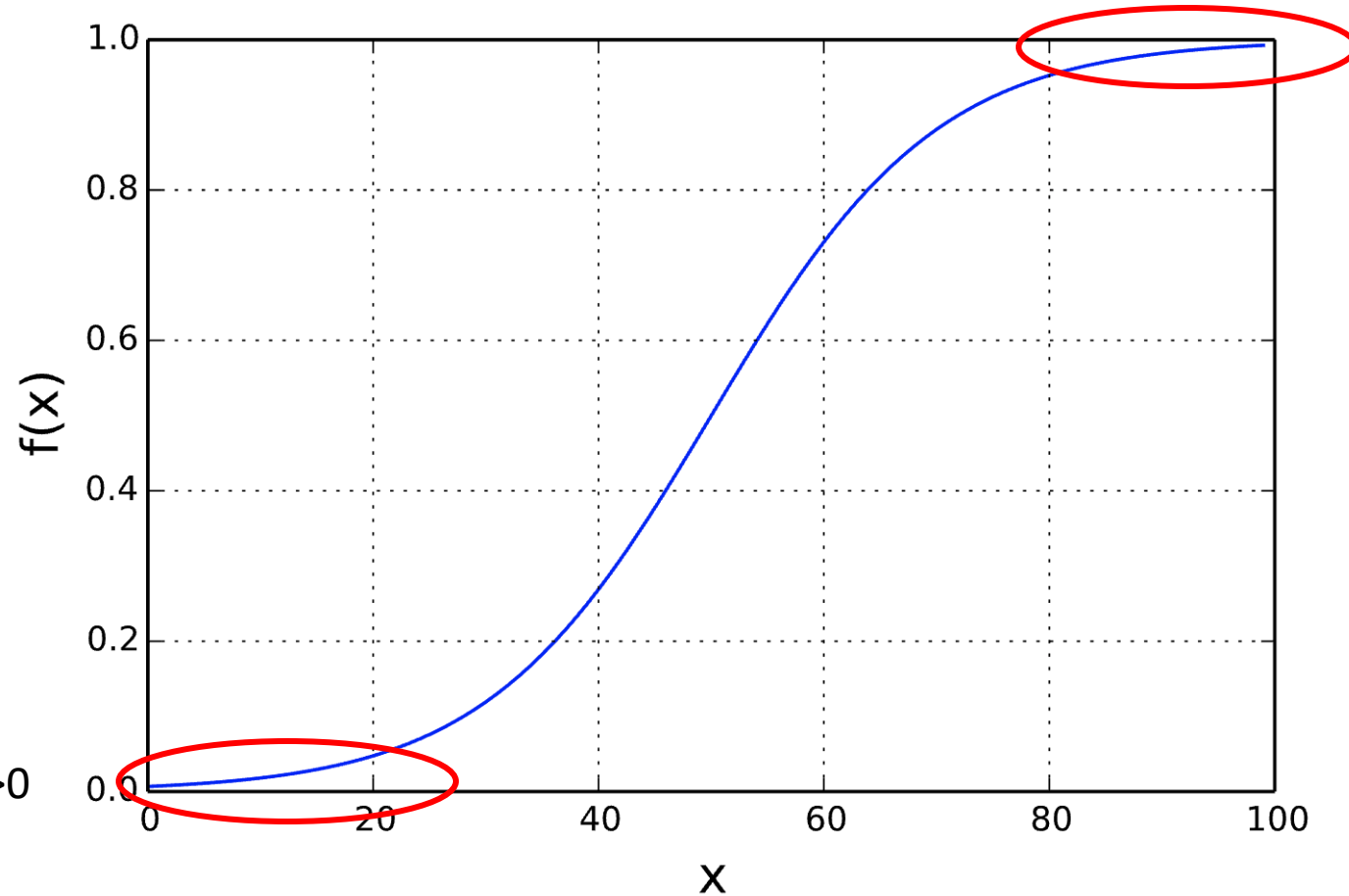
Gradient vanish

$$f_n \left(\dots \left(f_2 \left(f_1(x) \right) \right) \right) \rightarrow ?$$

$$f_i \rightarrow x_i$$

$$\frac{dx_n}{dx_1} = \frac{dx_n}{dx_{n-1}} \cdot \dots \cdot \frac{dx_2}{dx_1} \cdot \frac{dx_1}{dx}$$

gradients->0



gradients->0

Sigmoid function

ResNet: shortcut connection

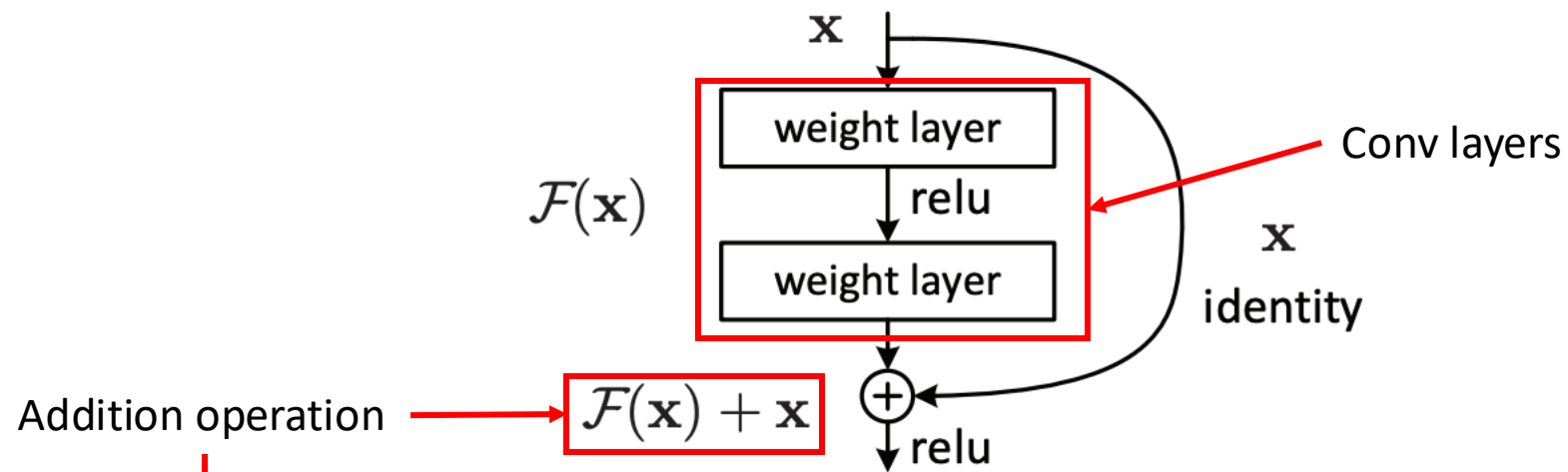
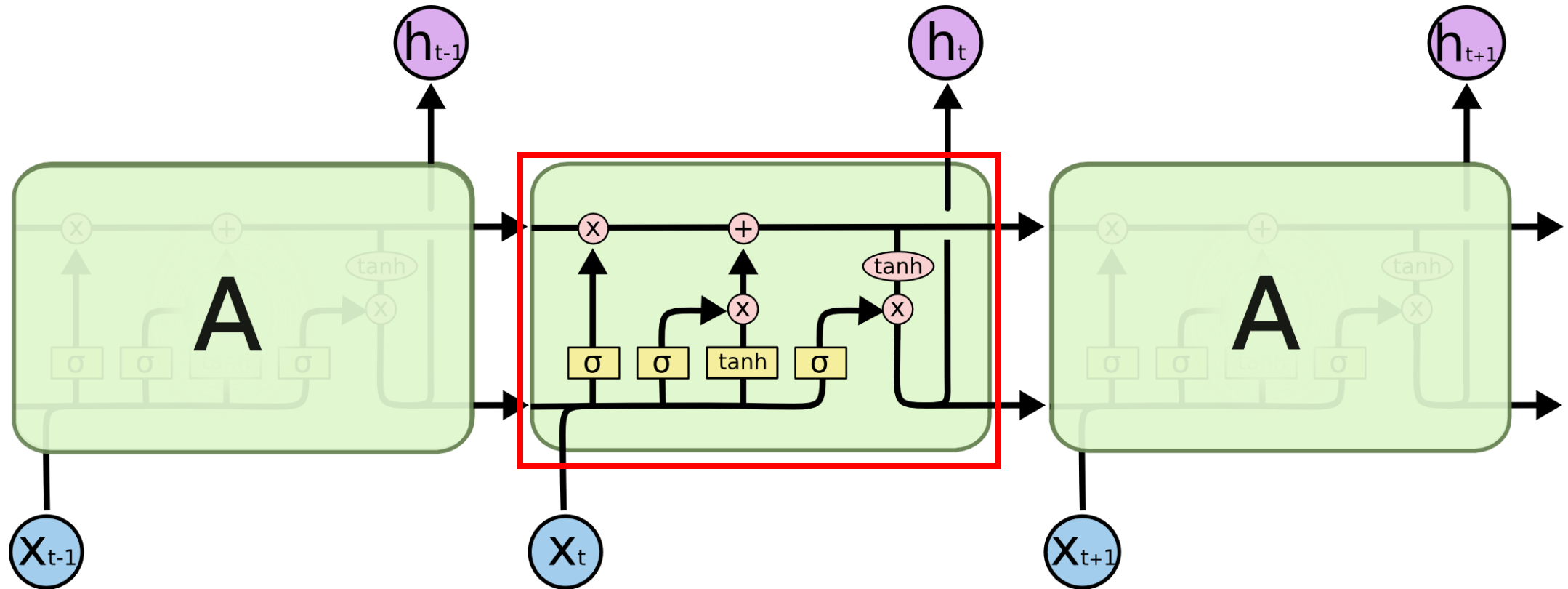


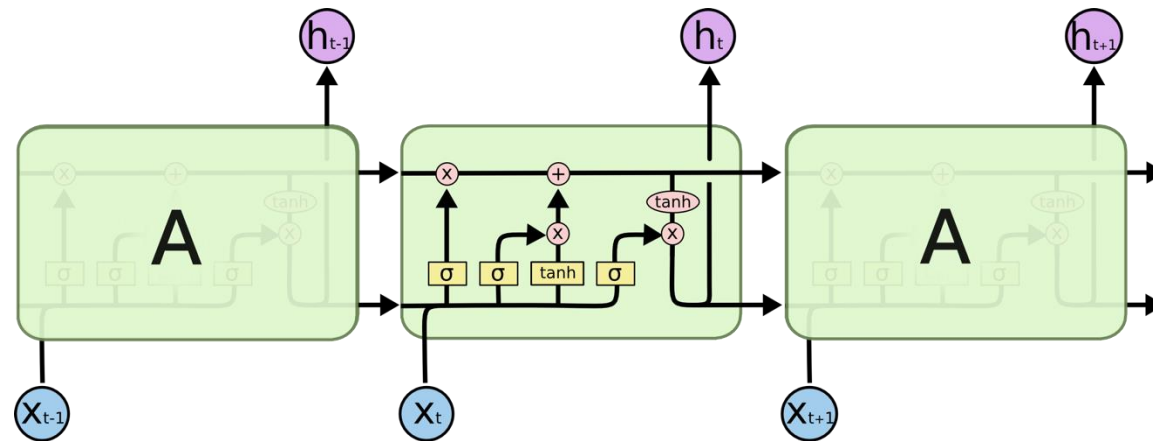
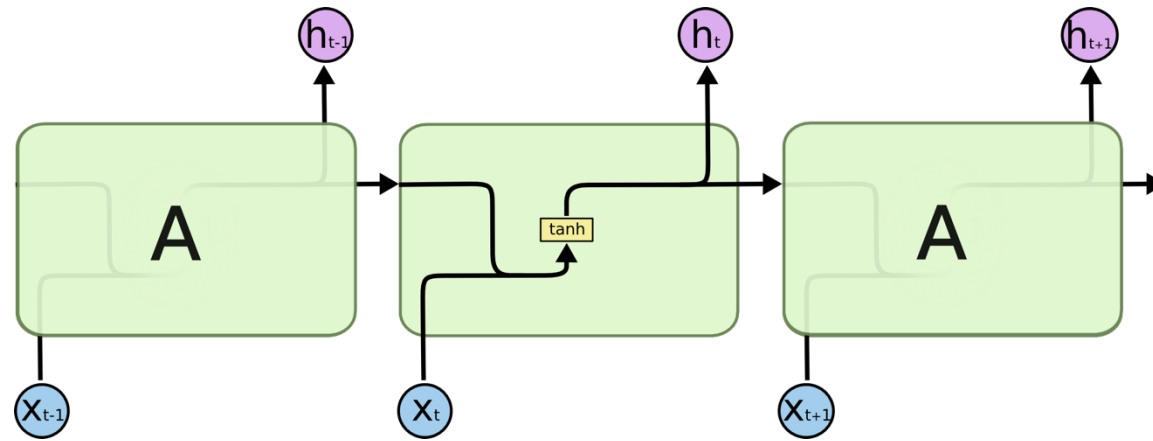
Figure 2. Residual learning: a building block.

implication: same dimension

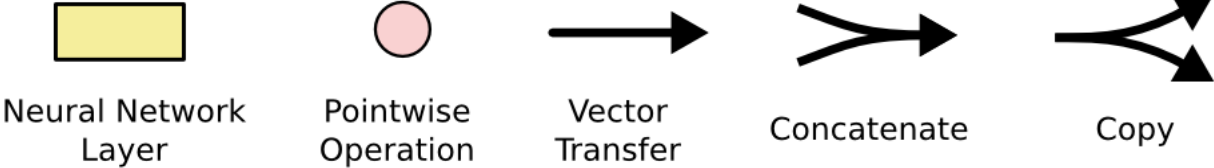
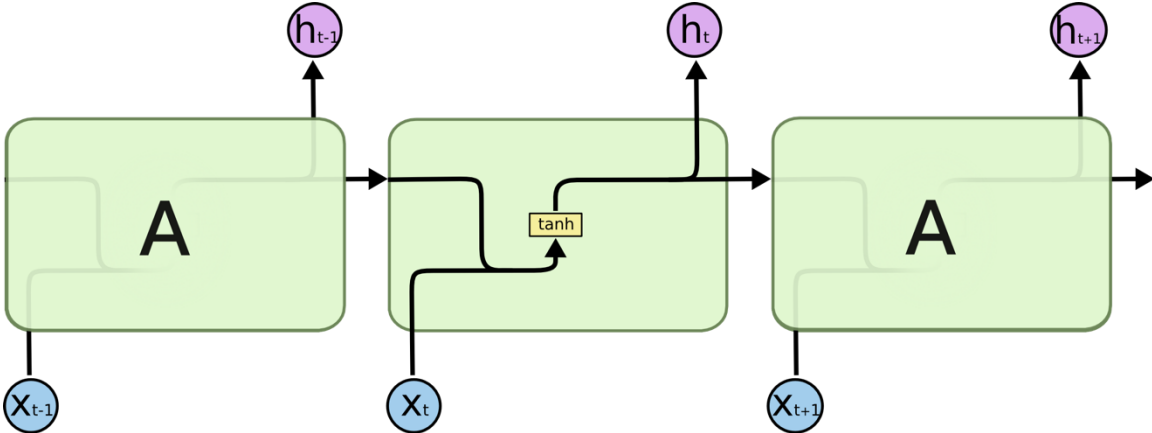
Long-short term memory (LSTM) networks



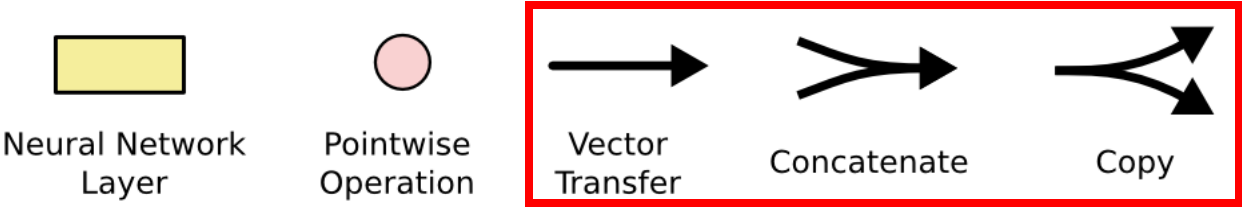
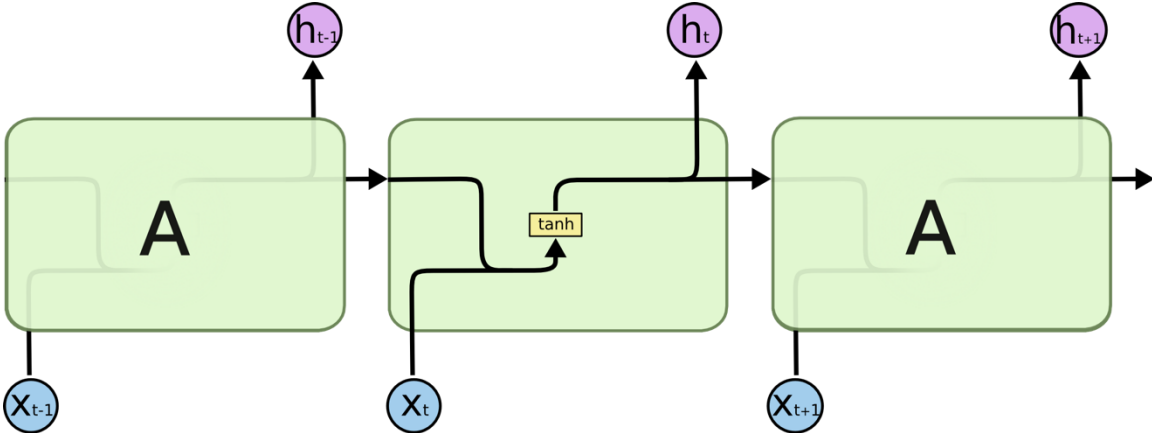
LSTM



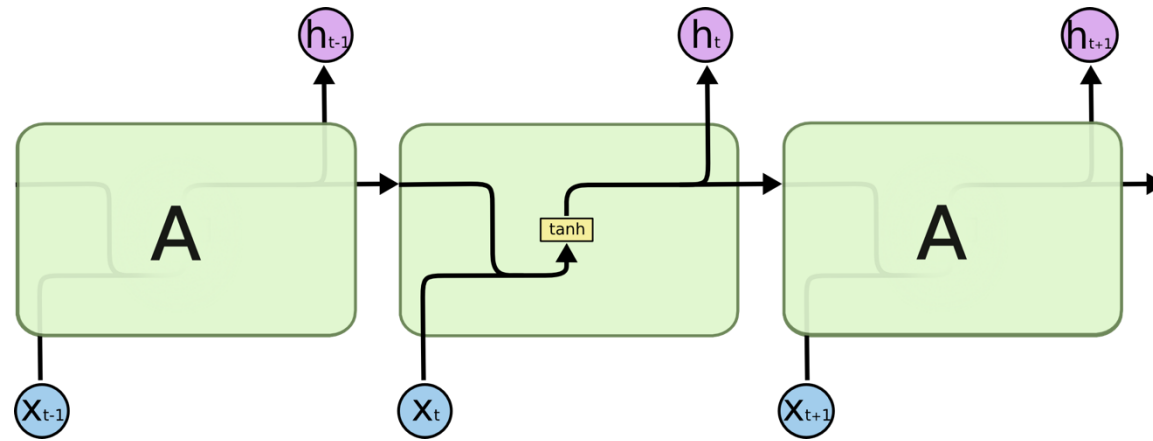
LSTM




LSTM



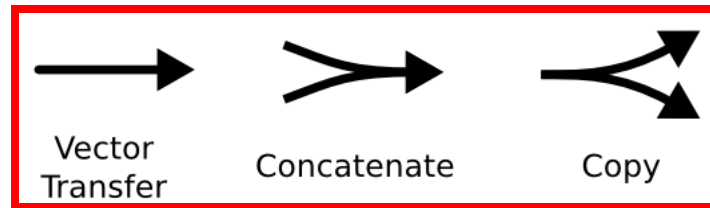
LSTM



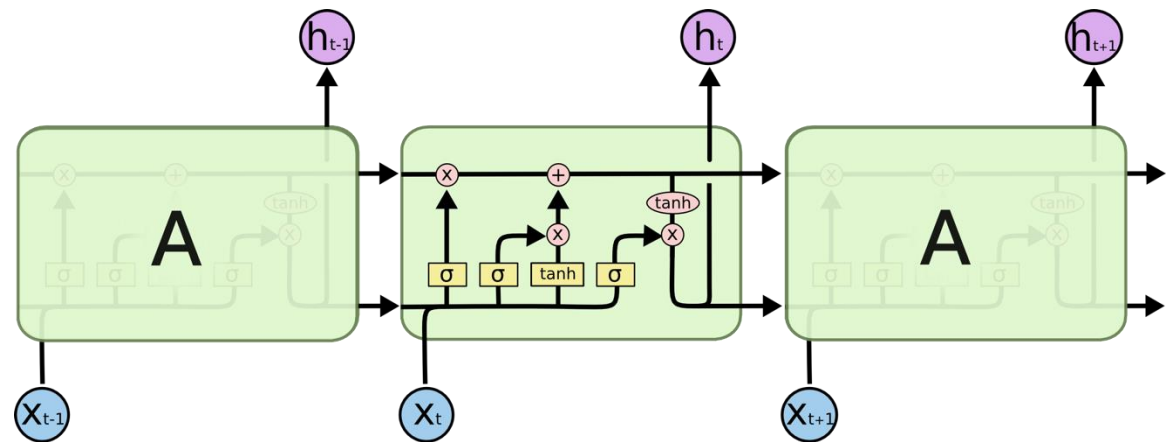
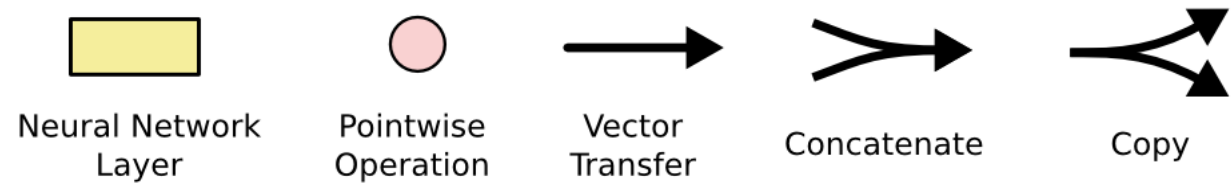
Composition of all $\{h_t\}$

 Neural Network Layer

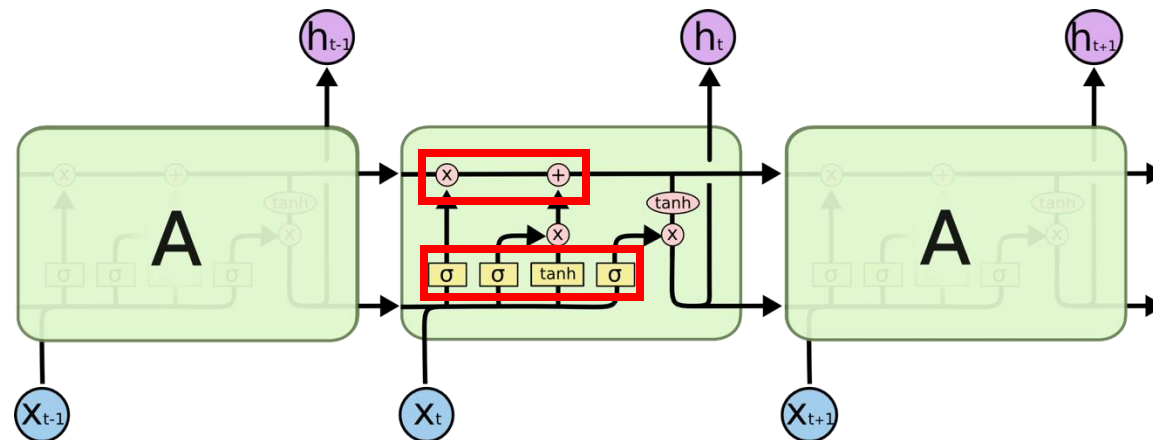
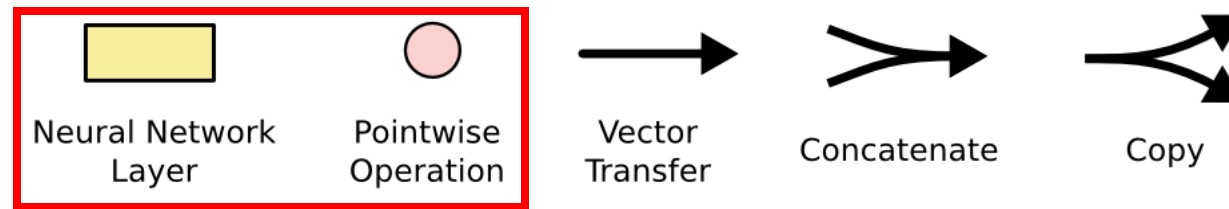
 Pointwise Operation



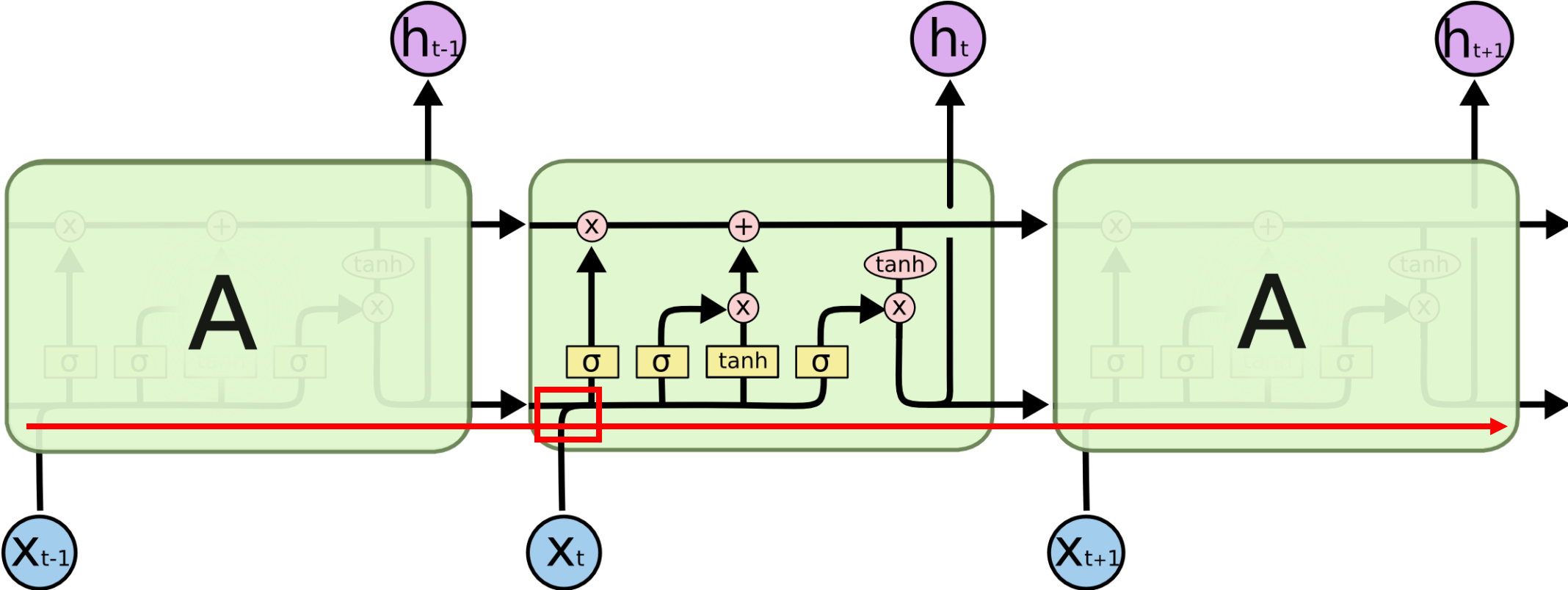
LSTM



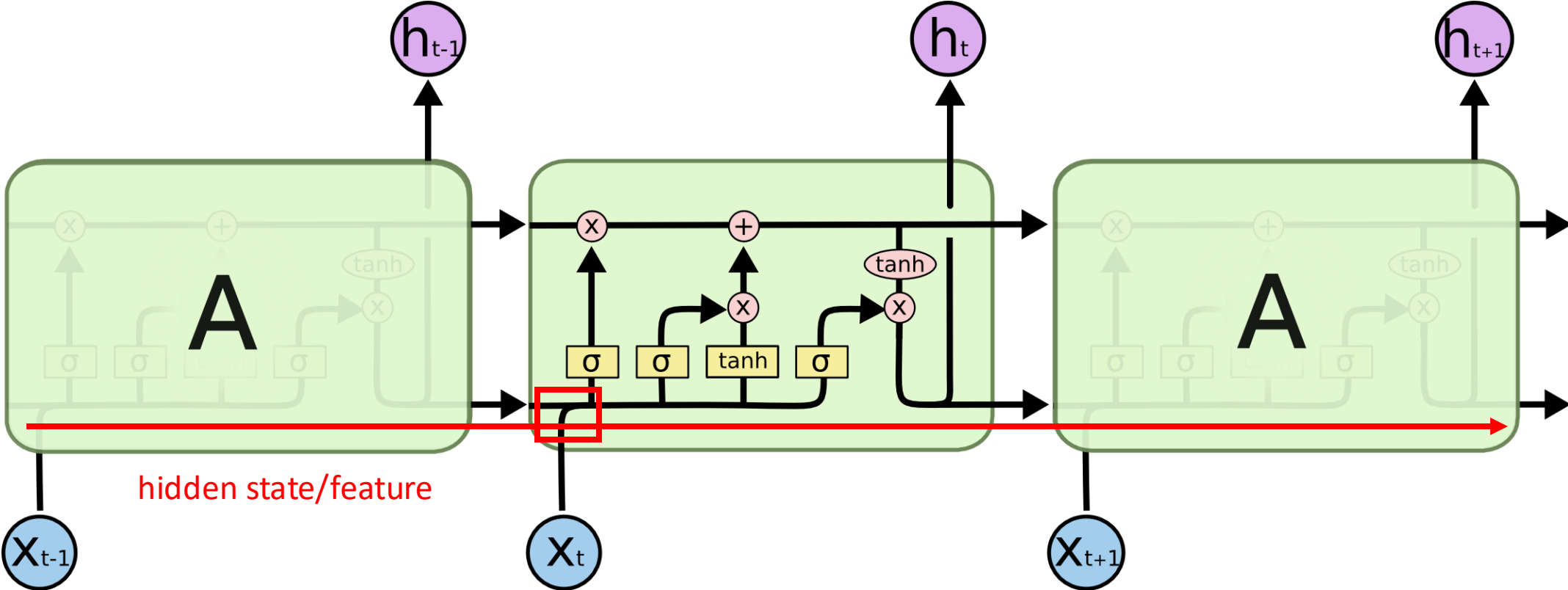
LSTM



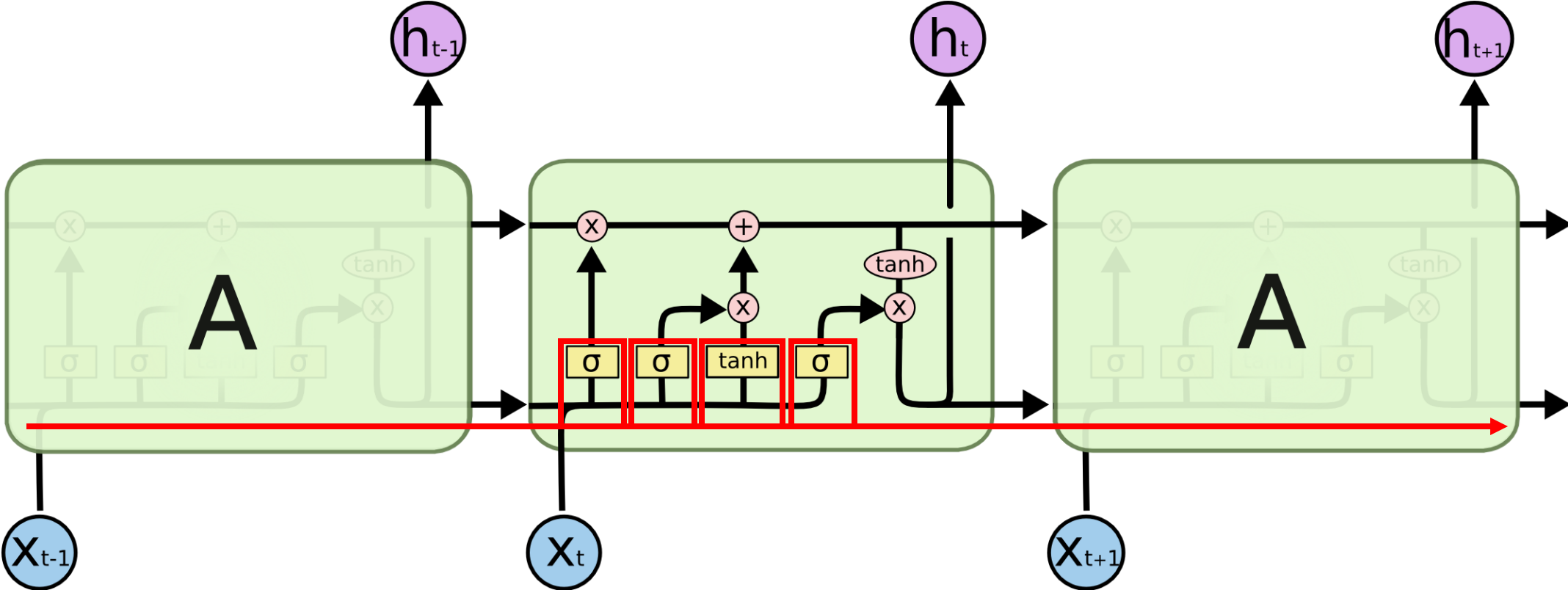
LSTM



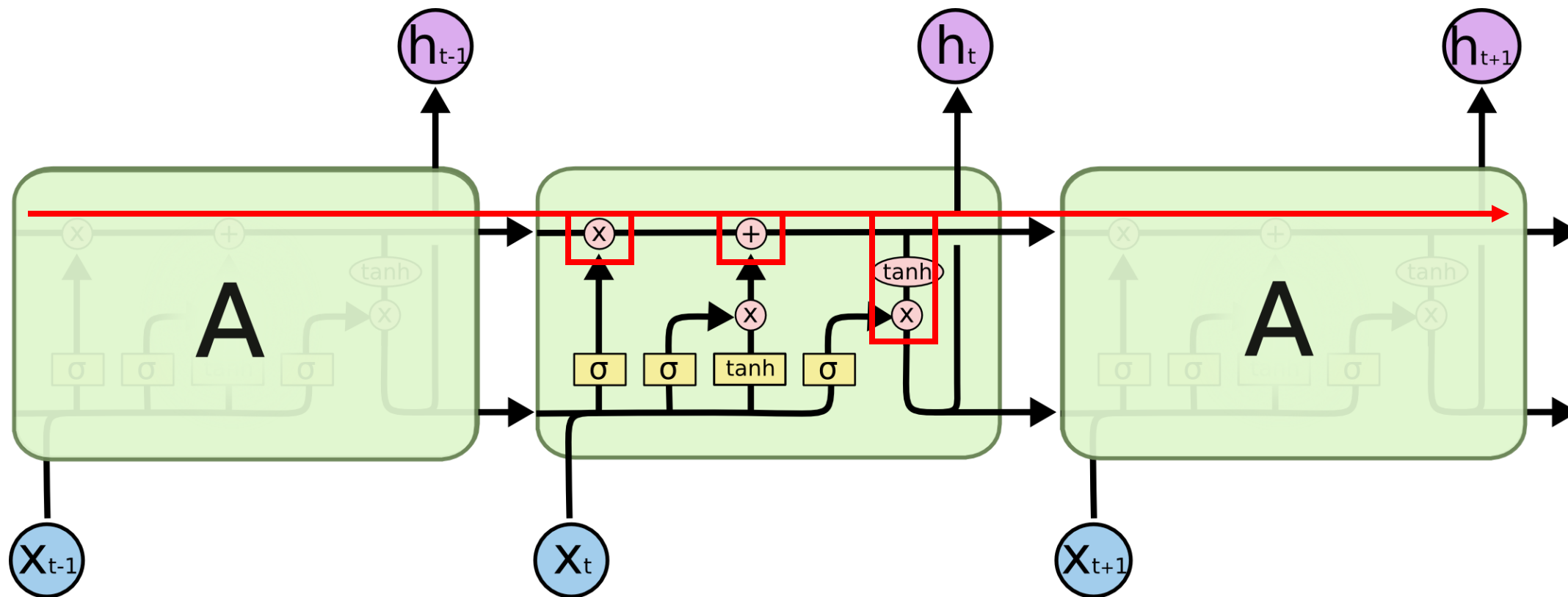
LSTM



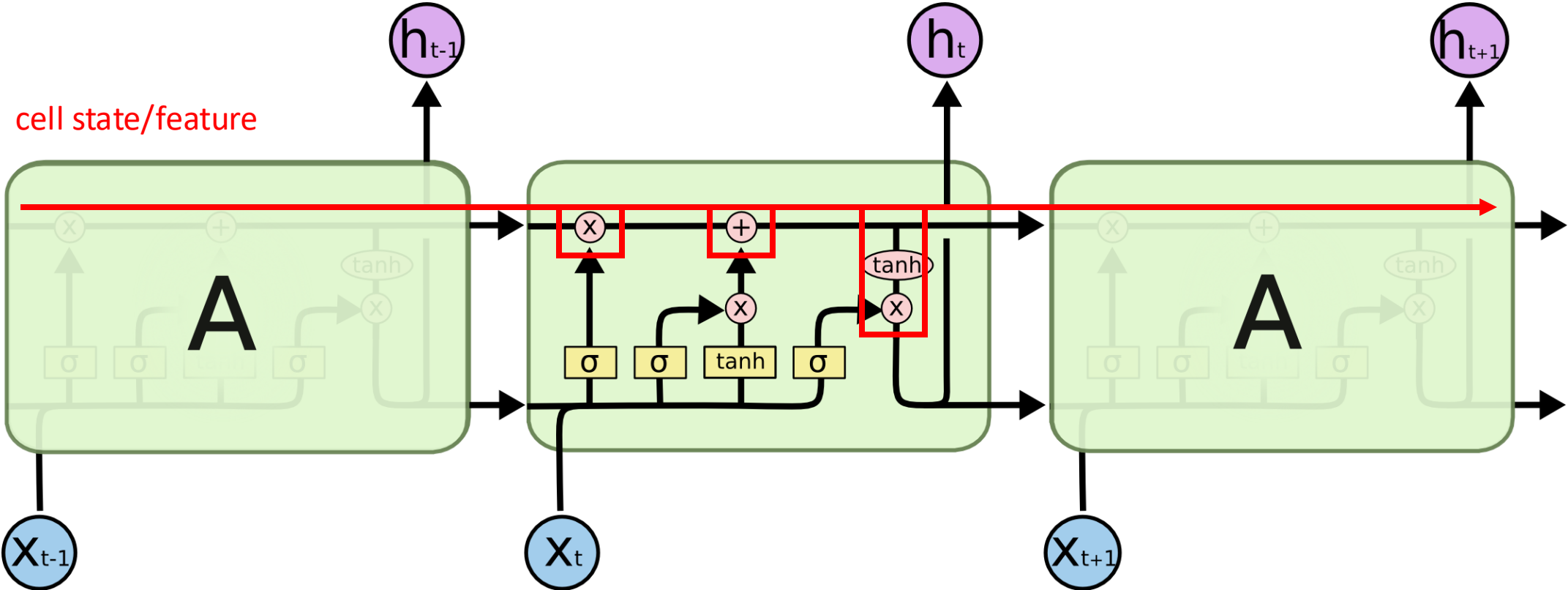
LSTM



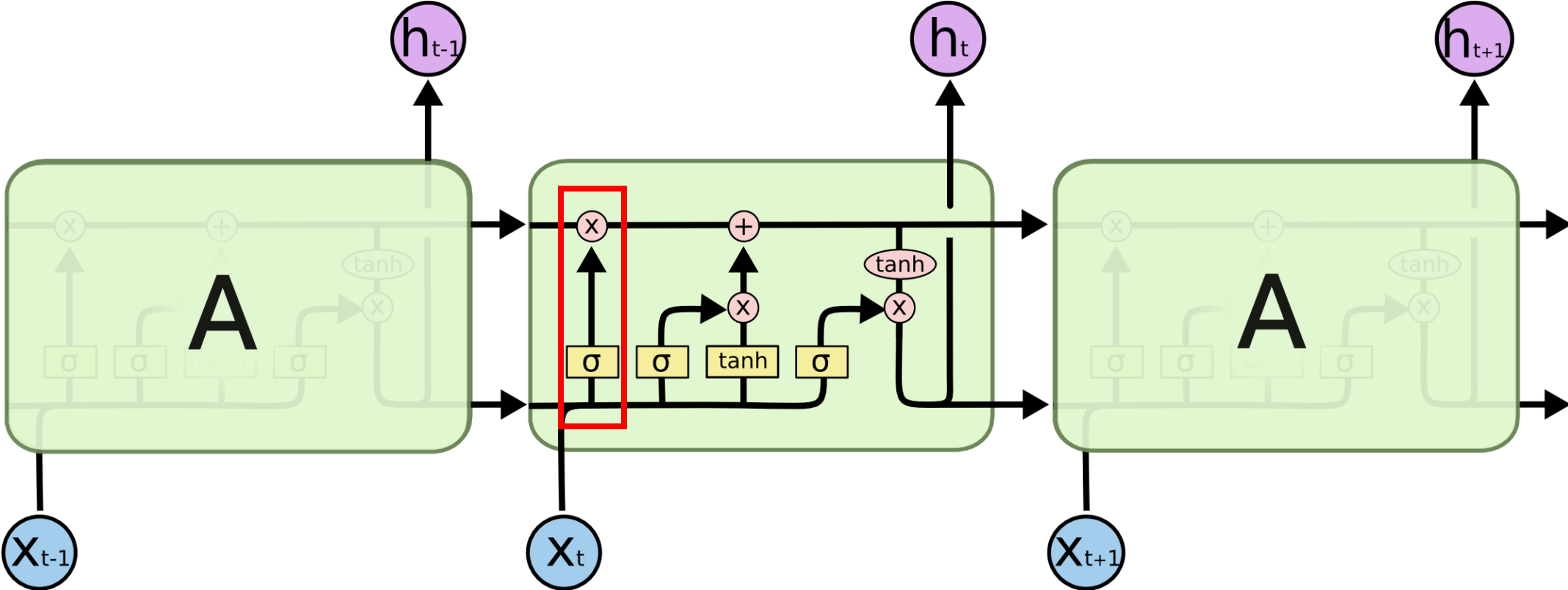
LSTM



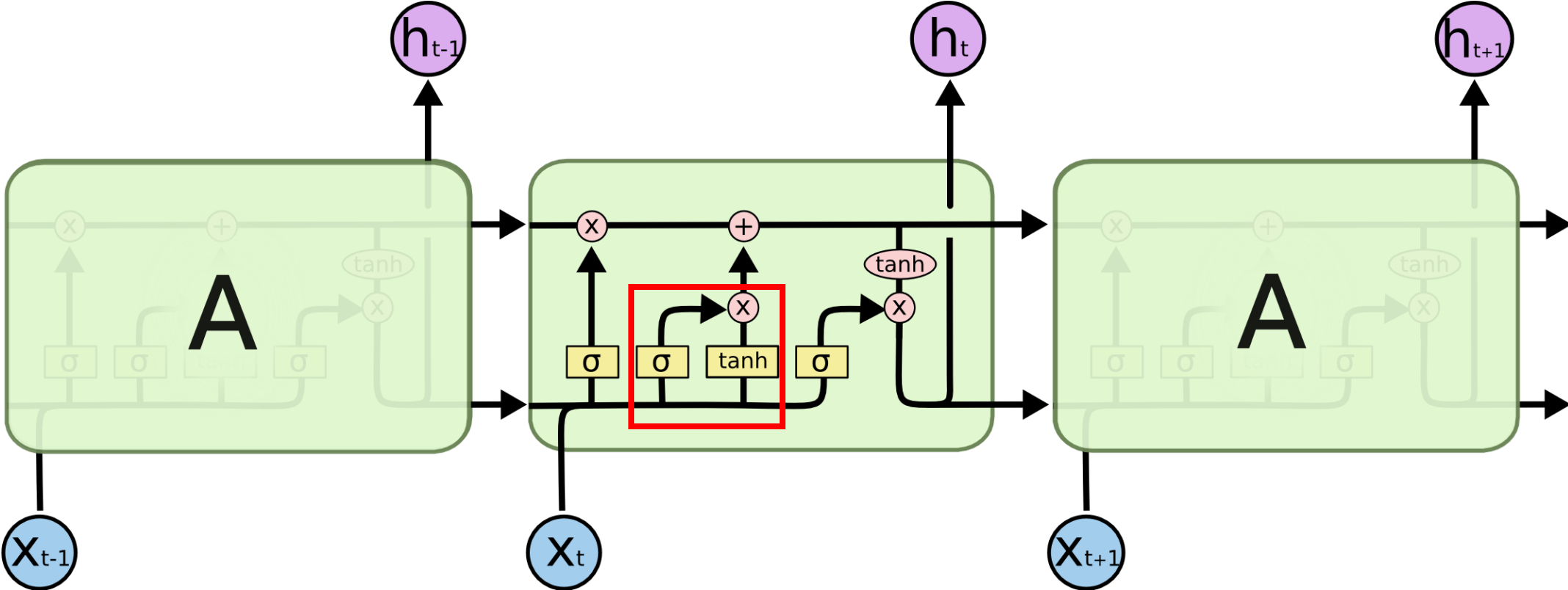
LSTM



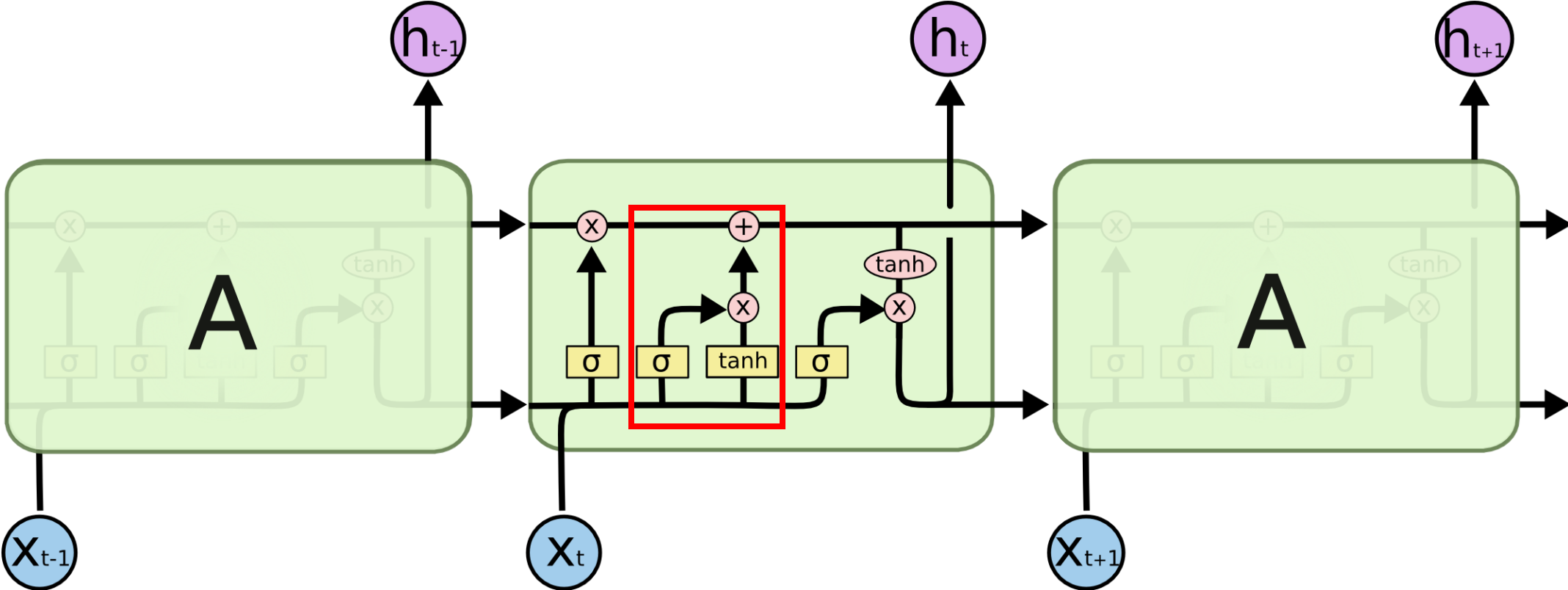
LSTM



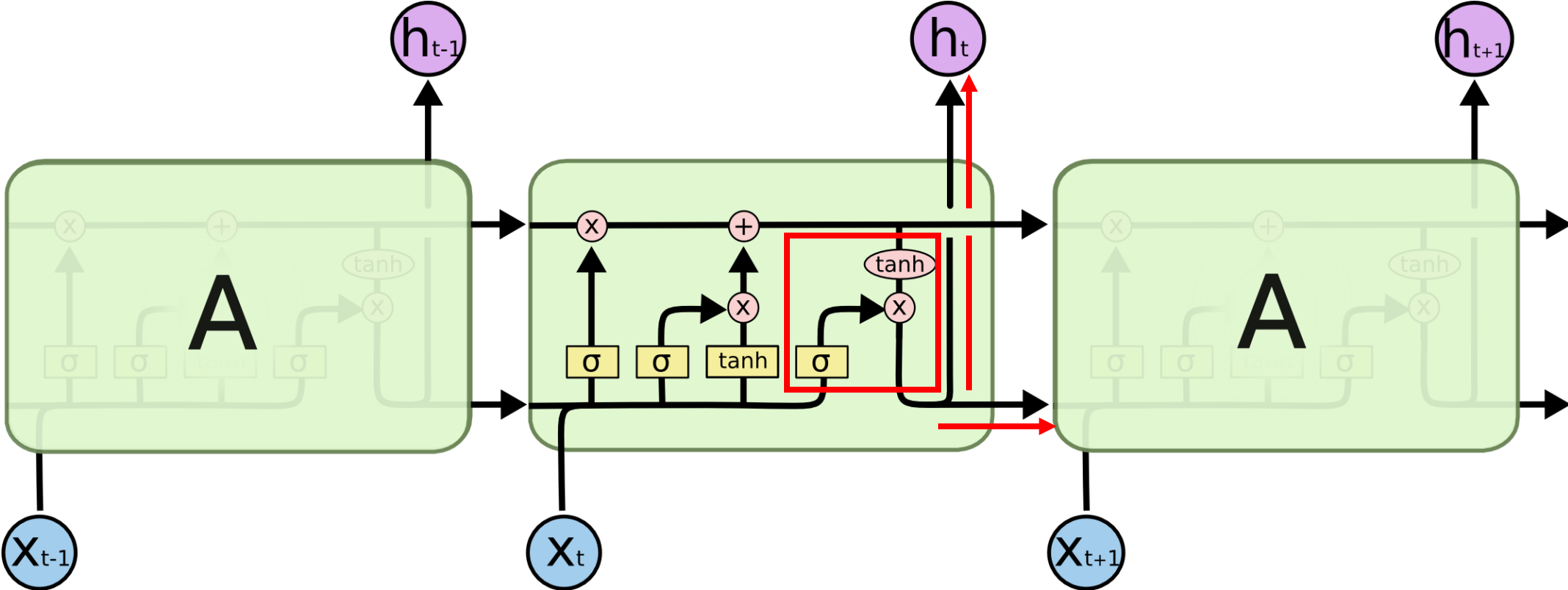
LSTM



LSTM



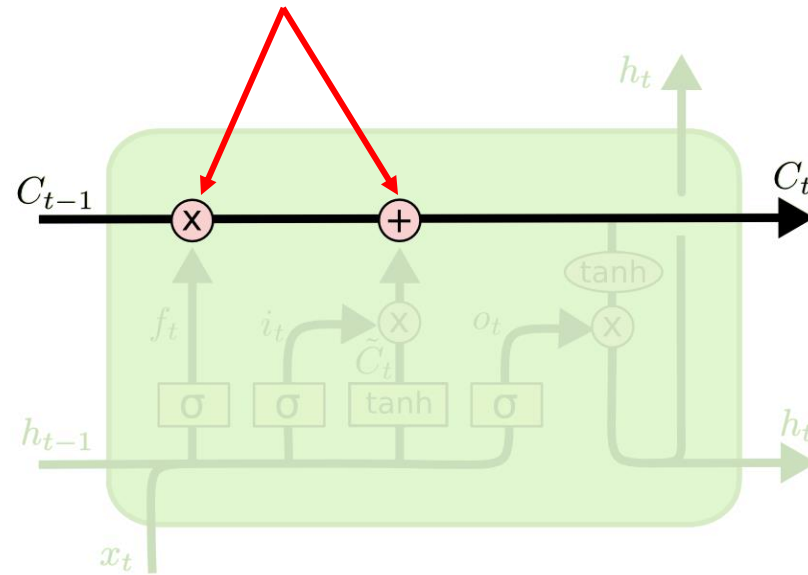
LSTM



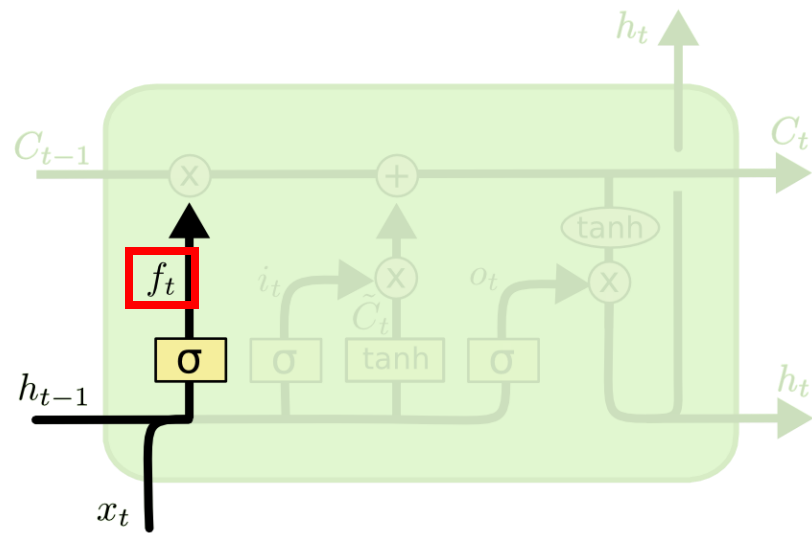
LSTM

Elementwise multiplication/addition

●
Pointwise
Operation

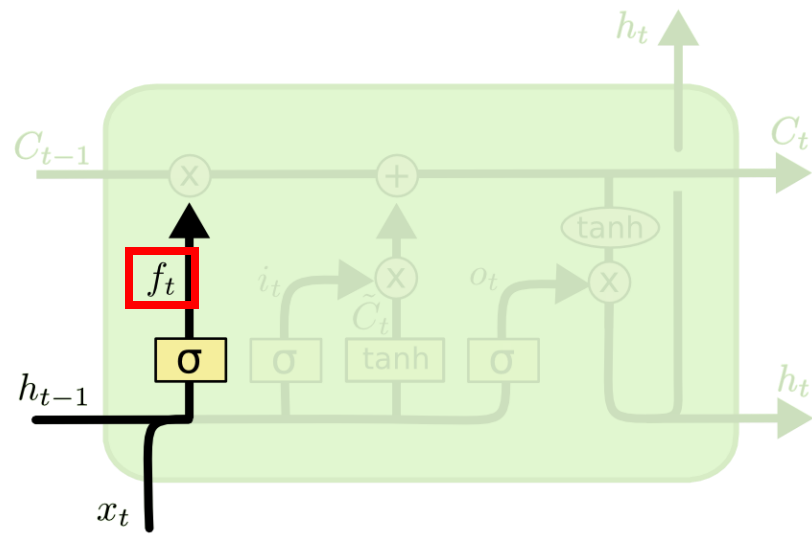


LSTM

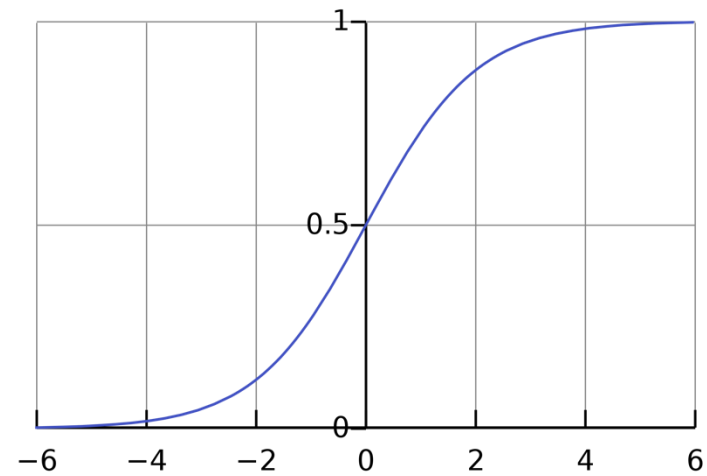


$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

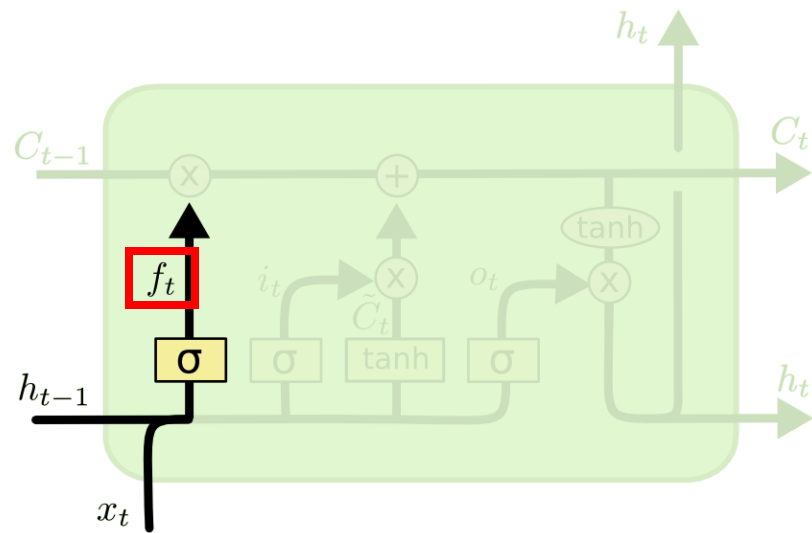
LSTM



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



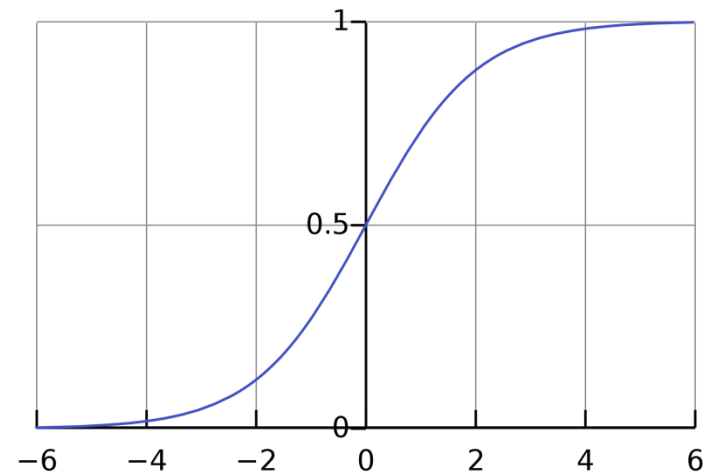
LSTM



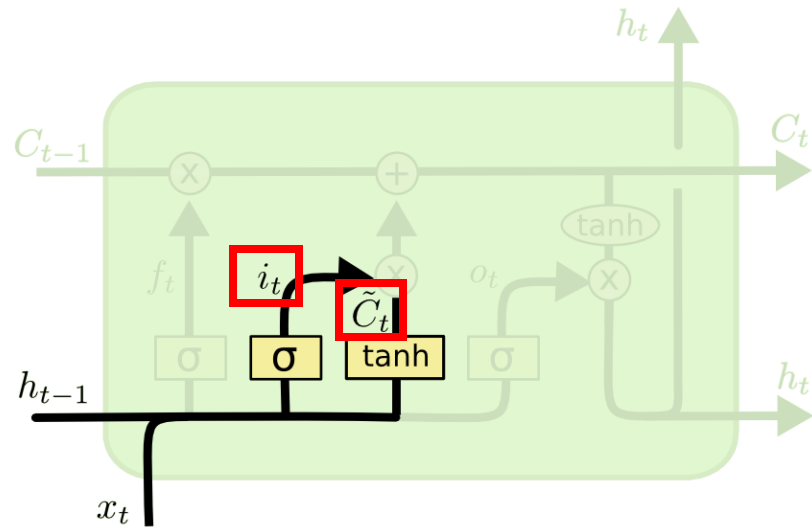
Forget gate:

Whether to erase cell

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



LSTM



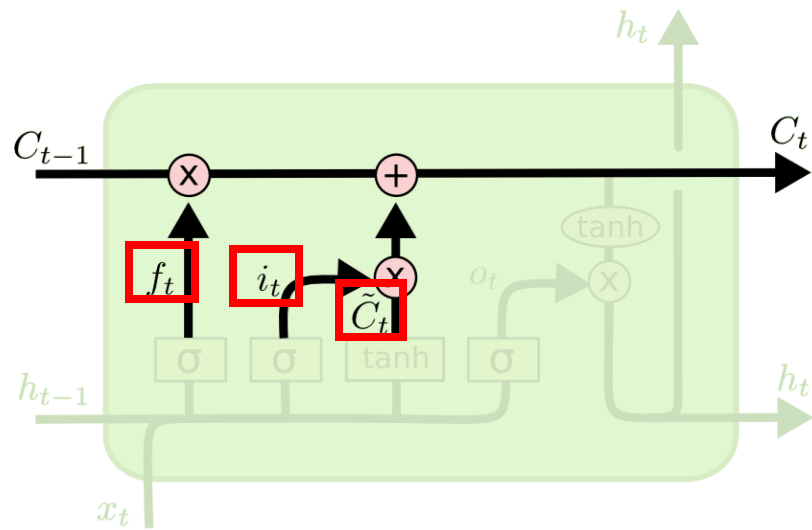
Input gate

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

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

cell input activation vector

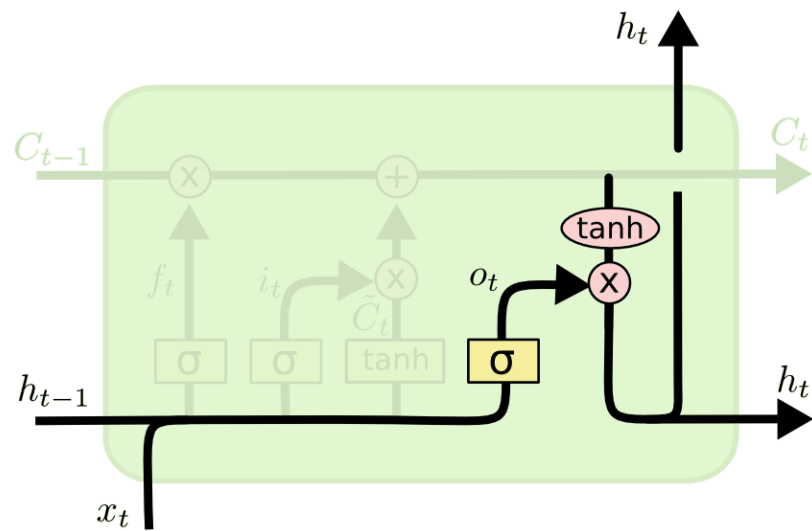
LSTM



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

Cell state

LSTM

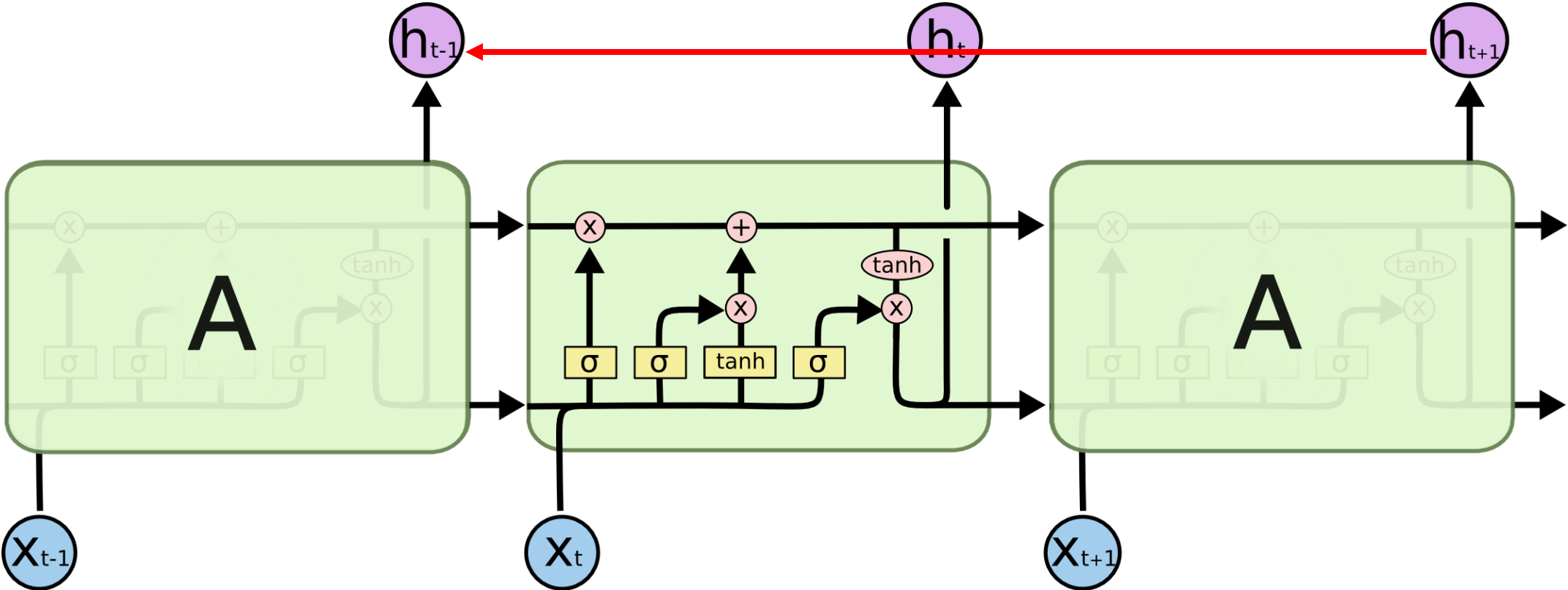


Output gate

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

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

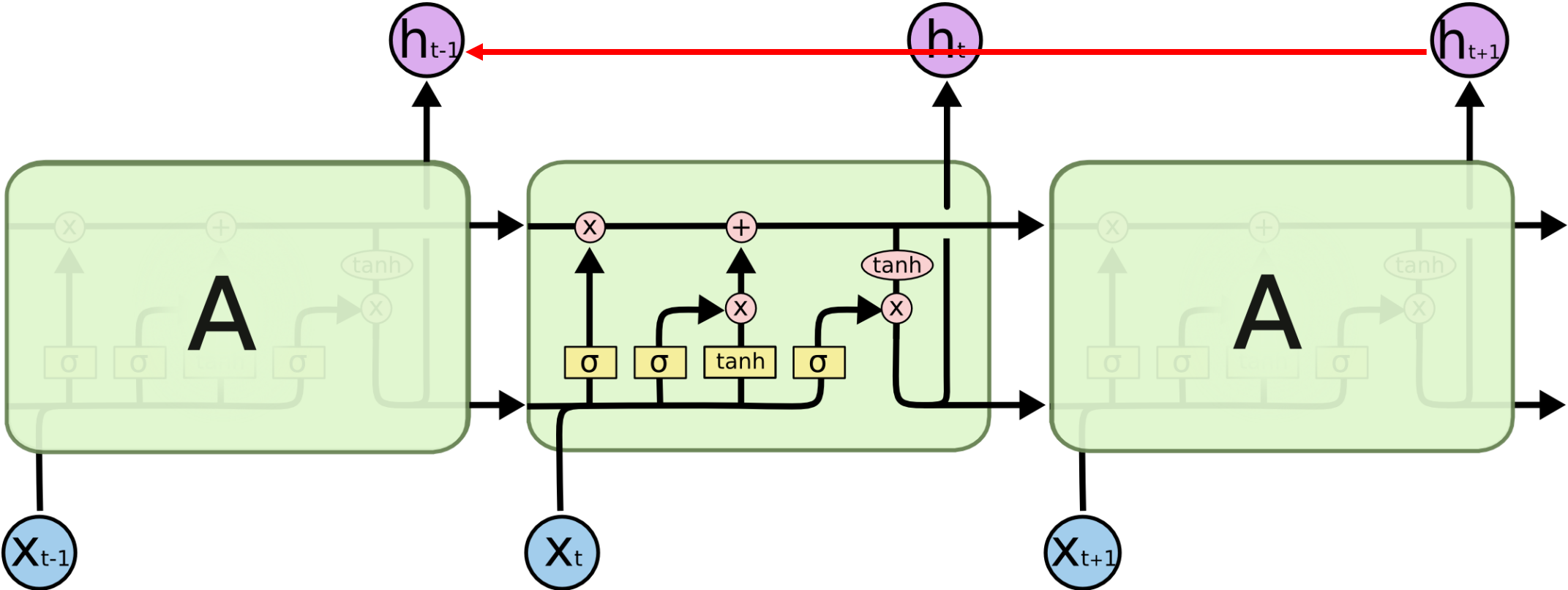
LSTM



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

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

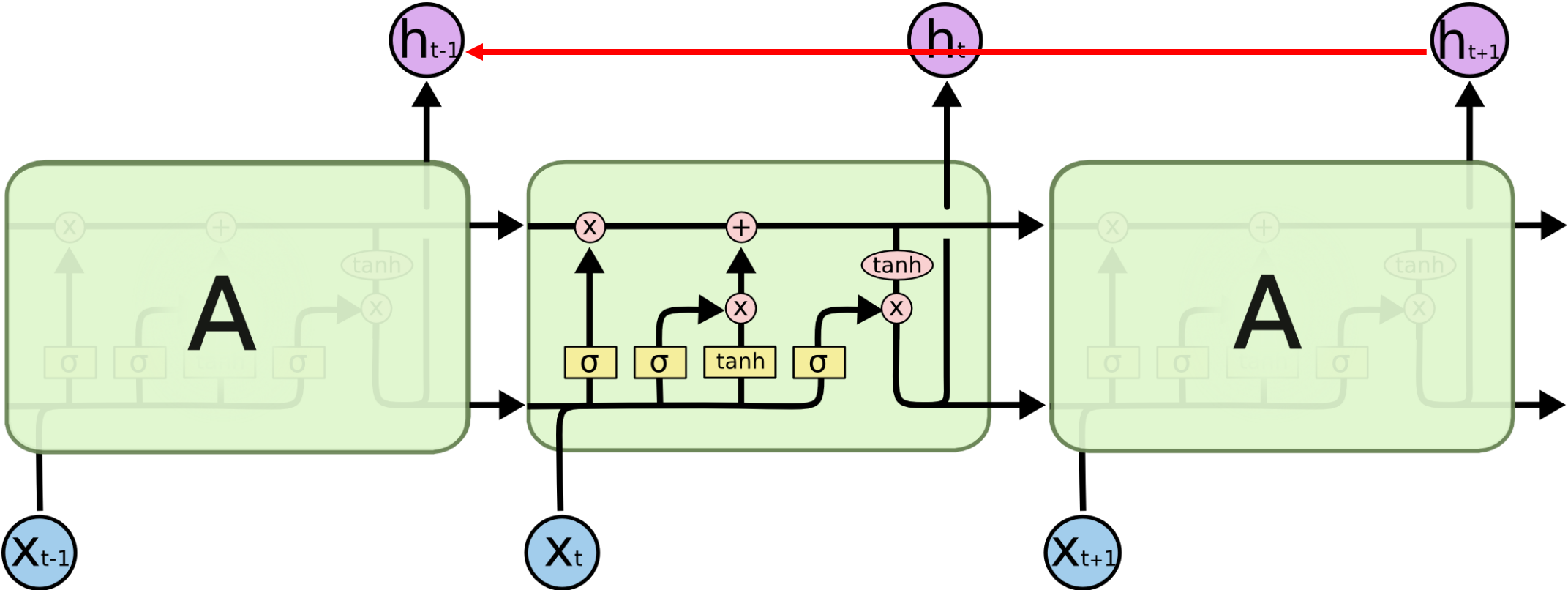
LSTM



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t = f_t * f_{t-1} * C_{t-2} + \text{others}$$

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

LSTM



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t = f_t * f_{t-1} * C_{t-2} + \text{others}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \text{ Not multiplication of weights } \prod W_t$$

Deep RNN

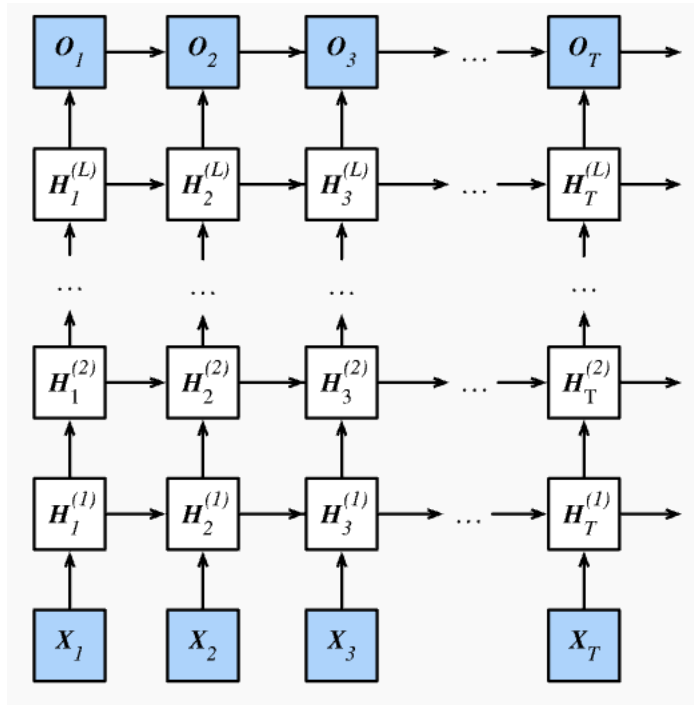


Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Deep RNN

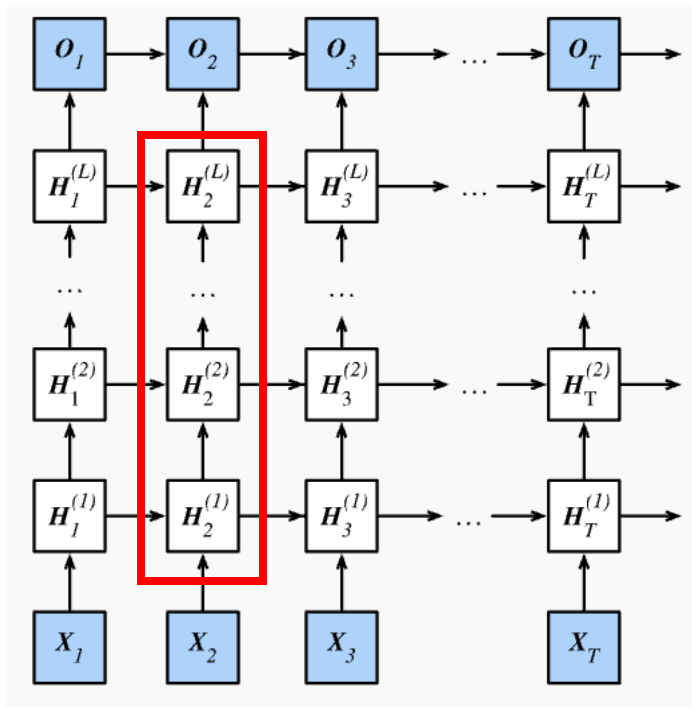
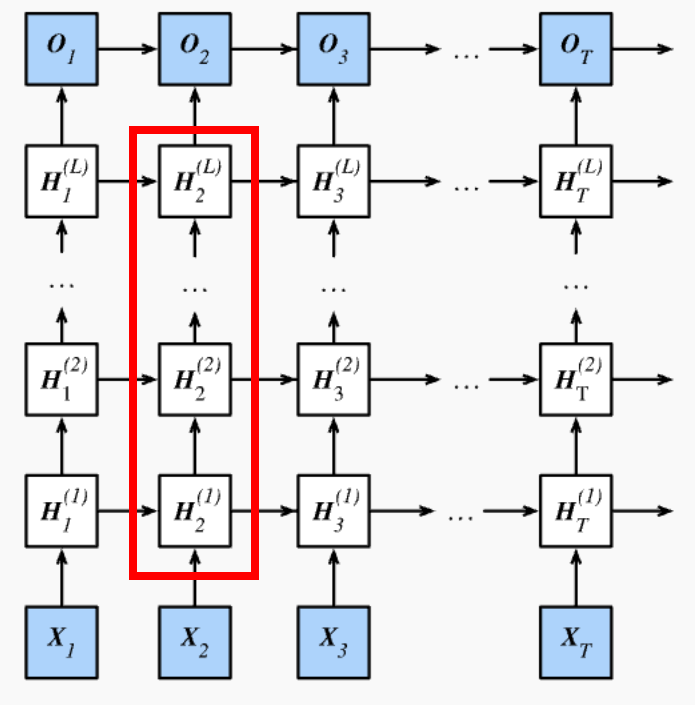


Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Deep RNN



For each unit: a mapping

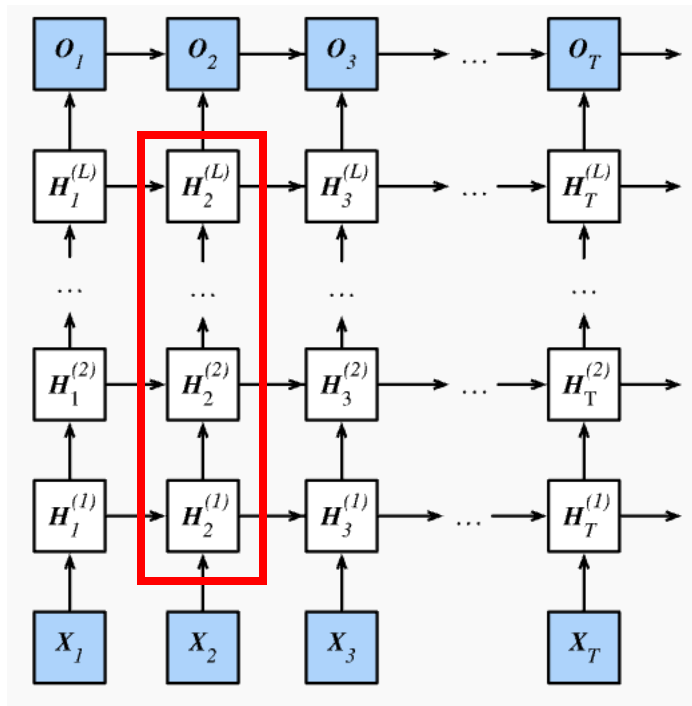
$$\text{RNN} : h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l$$

$$h_t^l \in \mathbb{R}^n$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep RNN



For each unit: a mapping

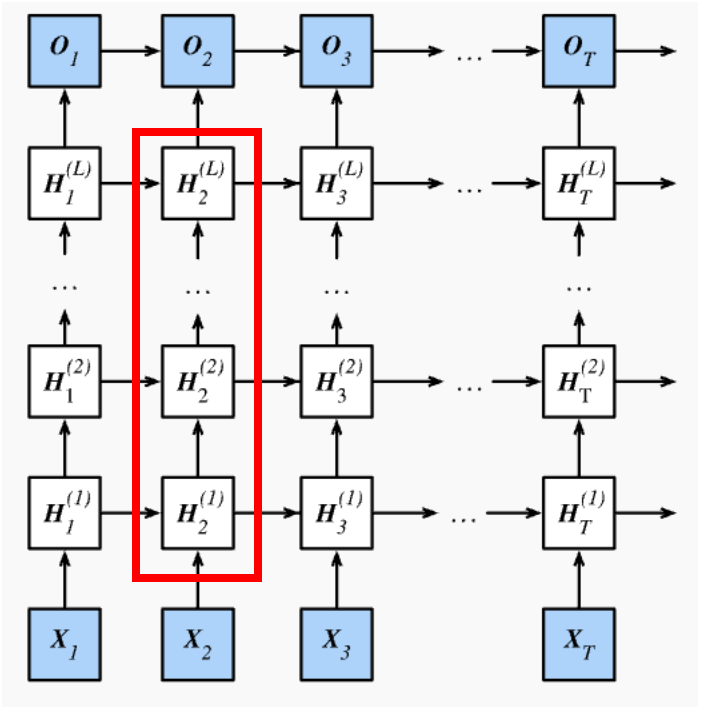
$$\text{RNN} : h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l \quad h_t^l \in \mathbb{R}^n$$

$$h_t^l = f(T_{n,n} h_t^{l-1} + T_{n,n} h_{t-1}^l), \text{ where } f \in \{\text{sigm}, \text{tanh}\}$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep RNN



For each unit: a mapping

$$\text{RNN} : h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l \quad h_t^l \in \mathbb{R}^n$$

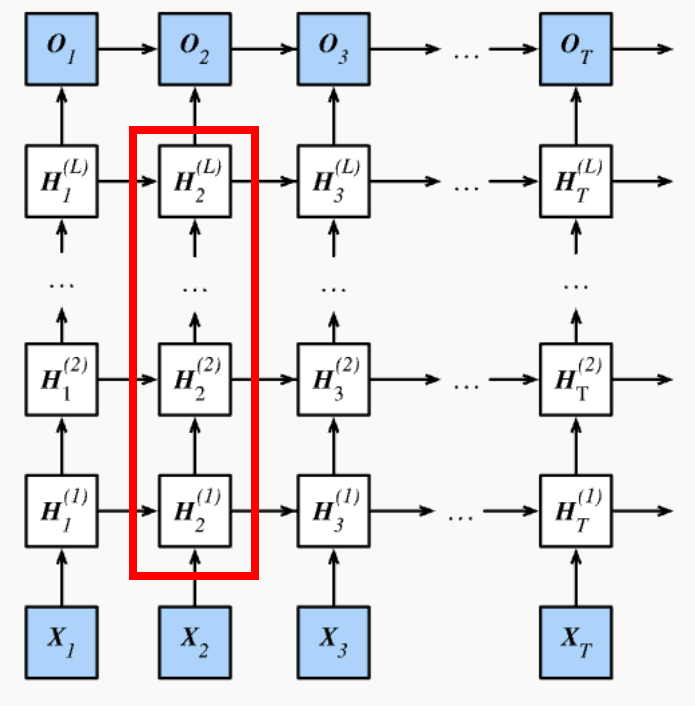
$$h_t^l = f(T_{n,n}h_t^{l-1} + T_{n,n}h_{t-1}^l), \text{ where } f \in \{\text{sigm}, \text{tanh}\}$$

$$T_{n,m} : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep RNN



For each unit: a mapping

$$\text{RNN} : h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l \quad h_t^l \in \mathbb{R}^n$$

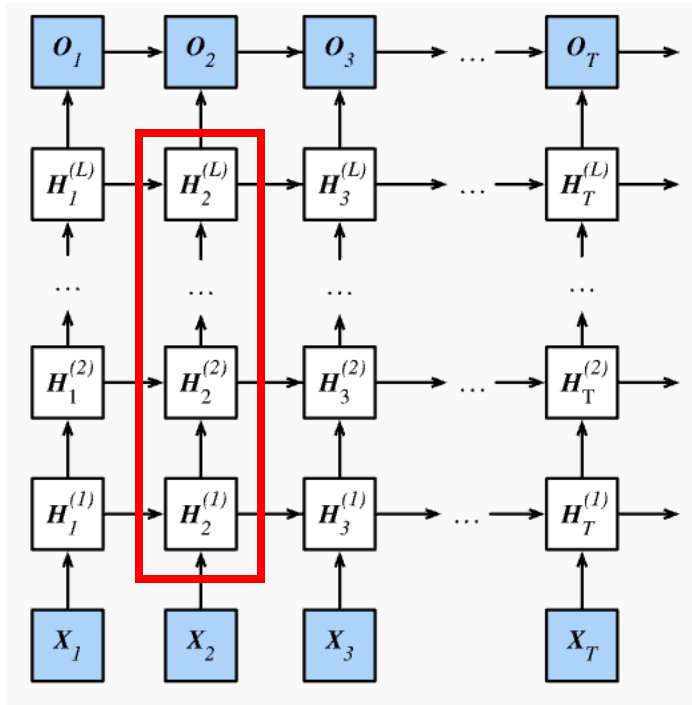
$$h_t^l = f(T_{n,n}h_t^{l-1} + T_{n,n}h_{t-1}^l), \text{ where } f \in \{\text{sigm}, \text{tanh}\}$$

$$T_{n,m} : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ (a mapping function)}$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep RNN



For each unit: a mapping

$$\text{RNN} : h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l \quad h_t^l \in \mathbb{R}^n$$

$$h_t^l = f(T_{n,n}h_t^{l-1} + T_{n,n}h_{t-1}^l), \text{ where } f \in \{\text{sigm}, \text{tanh}\}$$

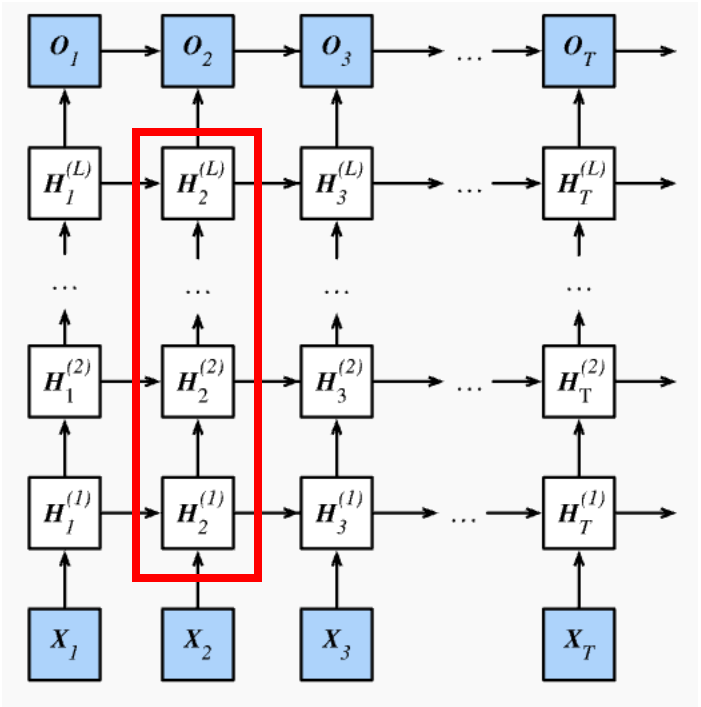
$$T_{n,m} : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ (a mapping function)}$$

$$Wx + b$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep RNN



For each unit: a mapping

$$\text{RNN} : h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l \quad h_t^l \in \mathbb{R}^n$$

$$h_t^l = f(T_{n,n} h_t^{l-1} + T_{n,n} h_{t-1}^l), \text{ where } f \in \{\text{sigm}, \text{tanh}\}$$

$$T_{n,m} : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ (a mapping function)}$$

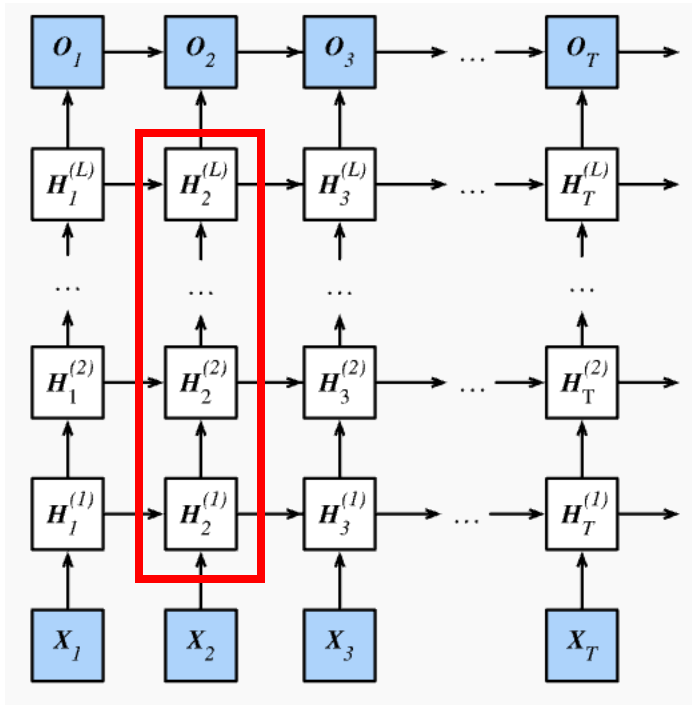
$$Wx + b$$

$$x \in \mathbb{R}^n, W \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep RNN



For each unit: a mapping

$$\text{RNN} : h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l \quad h_t^l \in \mathbb{R}^n$$

$$h_t^l = f(T_{n,n}h_t^{l-1} + T_{n,n}h_{t-1}^l), \text{ where } f \in \{\text{sigm}, \text{tanh}\}$$

$$T_{n,m} : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ (a mapping function)}$$

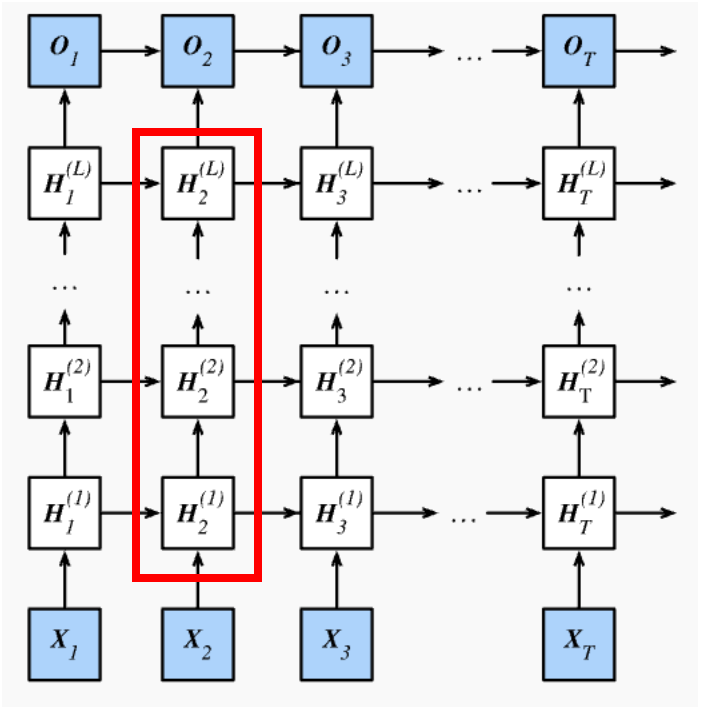
$$Wx + b$$

$$x \in \mathbb{R}^n, W \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep RNN



For each unit: a mapping

$$\text{RNN} : h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l \quad h_t^l \in \mathbb{R}^n$$

$$h_t^l = f(T_{n,n} h_t^{l-1} + T_{n,n} h_{t-1}^l), \text{ where } f \in \{\text{sigm}, \text{tanh}\}$$

$$T_{n,m} : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ (a mapping function)}$$

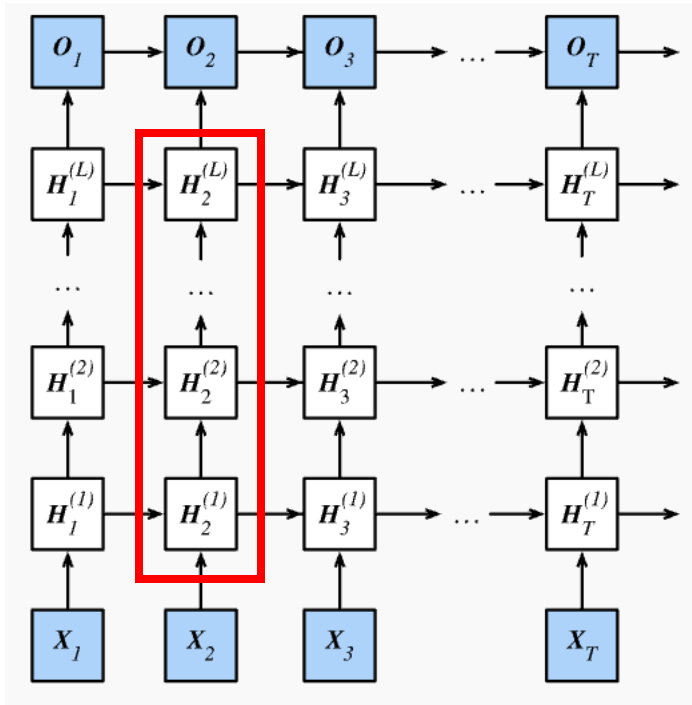
$$Wx + b$$

$$x \in \mathbb{R}^n, W \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep RNN



$$h_t = f \left[W * \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right]$$

For each unit: a mapping

$$\text{RNN} : h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l \quad h_t^l \in \mathbb{R}^n$$

$$h_t^l = f(T_{n,n} h_t^{l-1} + T_{n,n} h_{t-1}^l), \text{ where } f \in \{\text{sigm}, \text{tanh}\}$$

$$T_{n,m} : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ (a mapping function)}$$

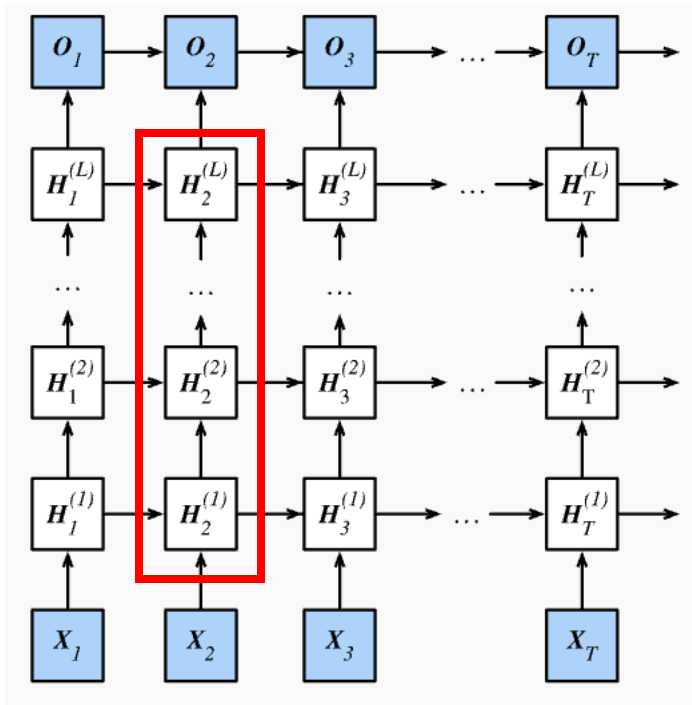
$$Wx + b$$

$$x \in \mathbb{R}^n, W \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep RNN



$$h_t = f \left[W * \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right] \quad f = \tanh(\cdot)$$

For each unit: a mapping

$$\text{RNN} : h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l \quad h_t^l \in \mathbb{R}^n$$

$$h_t^l = f(T_{n,n} h_t^{l-1} + T_{n,n} h_{t-1}^l), \text{ where } f \in \{\text{sigm}, \text{tanh}\}$$

$$T_{n,m} : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ (a mapping function)}$$

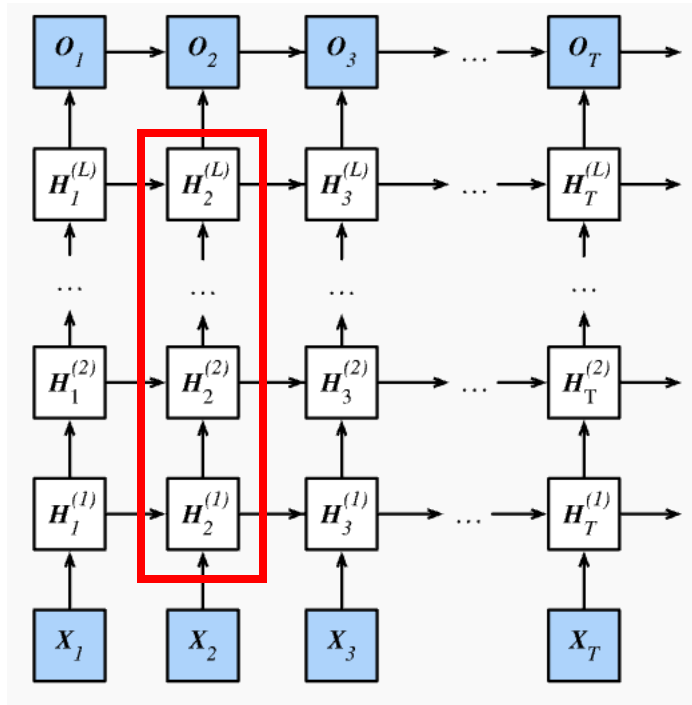
$$Wx + b$$

$$x \in \mathbb{R}^n, W \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep RNN



Vector feature

$$h_t = f \left[W * \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right] \quad f = \tanh(\cdot)$$

For each unit: a mapping

$$\text{RNN} : h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l \quad h_t^l \in \mathbb{R}^n$$

$$h_t^l = f(T_{n,n} h_t^{l-1} + T_{n,n} h_{t-1}^l), \text{ where } f \in \{\text{sigm}, \text{tanh}\}$$

$$T_{n,m} : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ (a mapping function)}$$

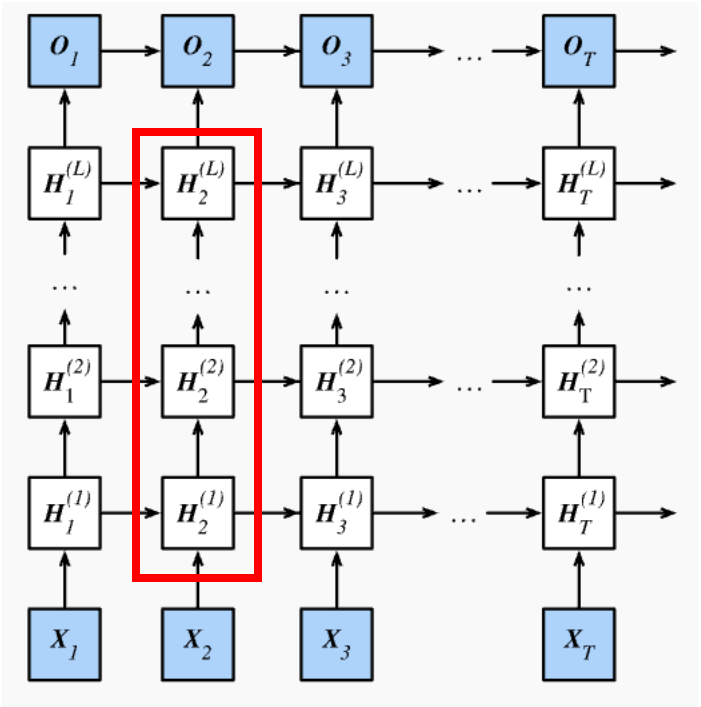
$$Wx + b$$

$$x \in \mathbb{R}^n, W \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep RNN



elementwise operation

Vector feature

$$h_t = f \left[W * \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right]$$

$f = \tanh(\cdot)$

For each unit: a mapping
 RNN : $h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l$ $h_t^l \in \mathbb{R}^n$

$$h_t^l = f(T_{n,n} h_t^{l-1} + T_{n,n} h_{t-1}^l), \text{ where } f \in \{\text{sigm}, \text{tanh}\}$$

$$T_{n,m} : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ (a mapping function)}$$

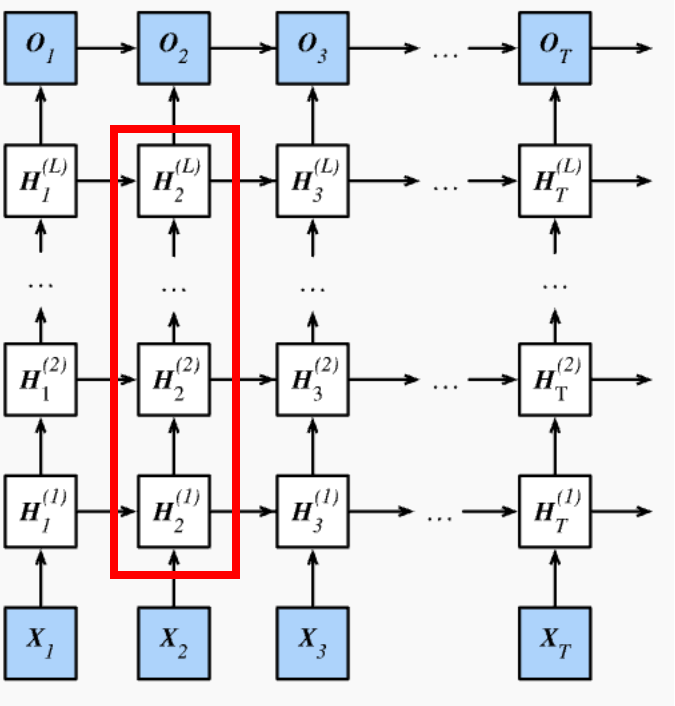
$$Wx + b$$

$$x \in \mathbb{R}^n, W \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep LSTM

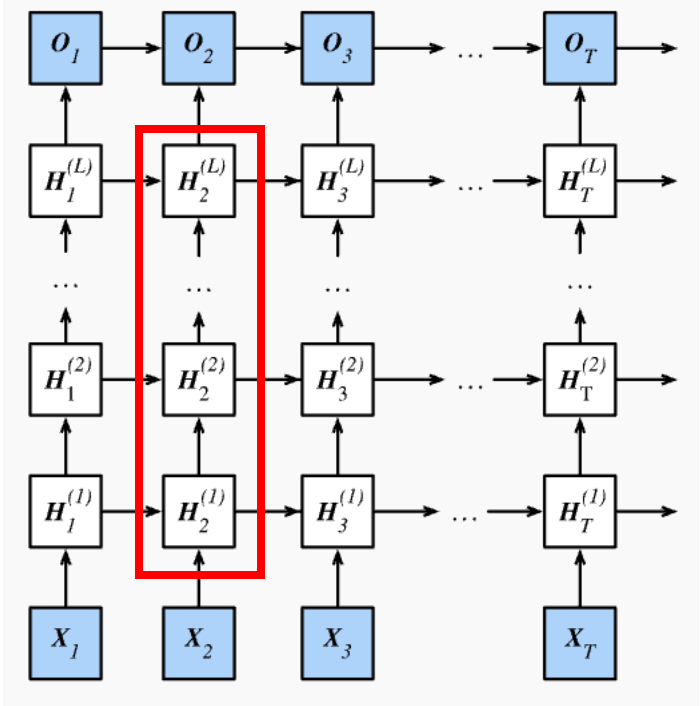


$$\text{LSTM} : h_t^{l-1}, h_{t-1}^l, c_{t-1}^l \rightarrow h_t^l, c_t^l$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep LSTM



$$\text{LSTM} : h_t^{l-1}, h_{t-1}^l, c_{t-1}^l \rightarrow h_t^l, c_t^l$$

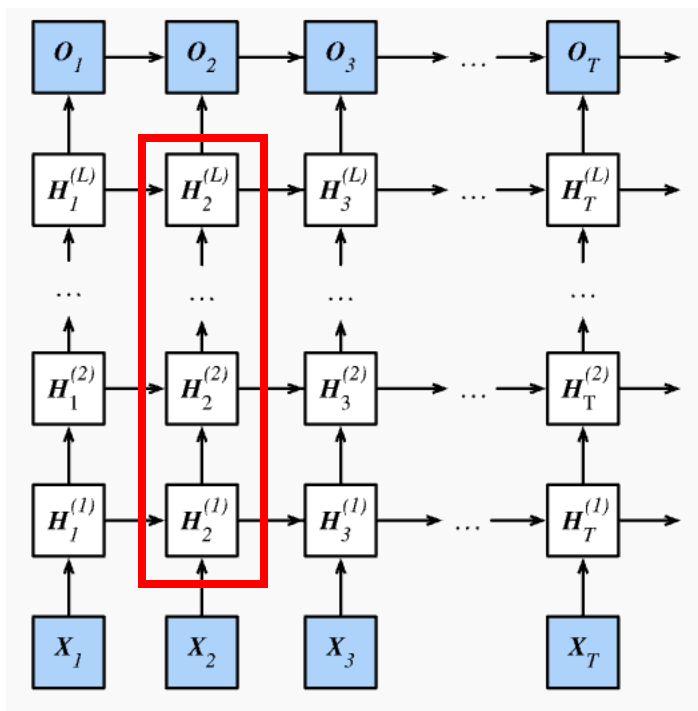
$$c_t^l \in \mathbb{R}^n$$

$$h_t^l \in \mathbb{R}^n$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep LSTM



$$\text{LSTM} : h_t^{l-1}, h_{t-1}^l, c_{t-1}^l \rightarrow h_t^l, c_t^l$$

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \text{tanh} \end{pmatrix} T_{2n,4n} \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$h_t^l = o \odot \tanh(c_t^l)$$

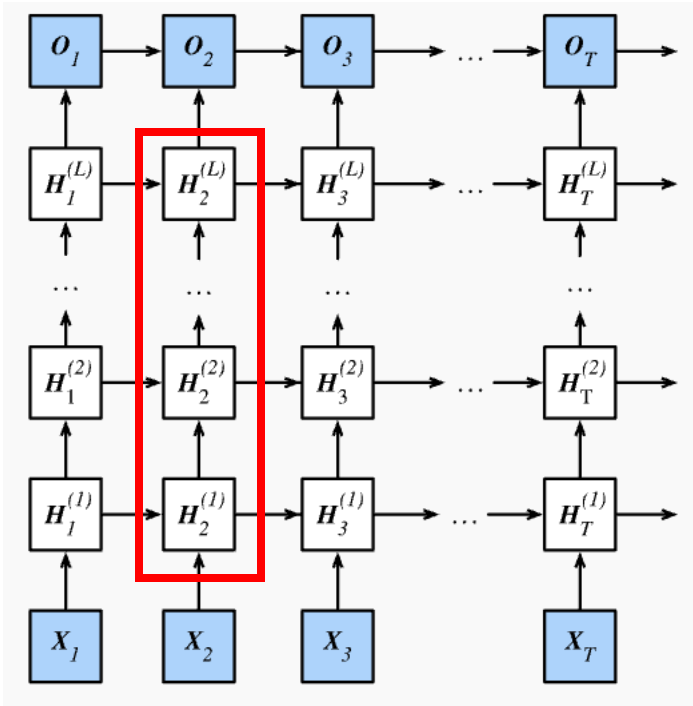
$$c_t^l \in \mathbb{R}^n$$

$$h_t^l \in \mathbb{R}^n$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep LSTM



$$\text{LSTM} : h_t^{l-1}, h_{t-1}^l, c_{t-1}^l \rightarrow \boxed{h_t^l, c_t^l}$$

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \text{tanh} \end{pmatrix} T_{2n,4n} \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

$$\boxed{c_t^l} = f \odot c_{t-1}^l + i \odot g$$

$$\boxed{h_t^l} = o \odot \tanh(c_t^l)$$

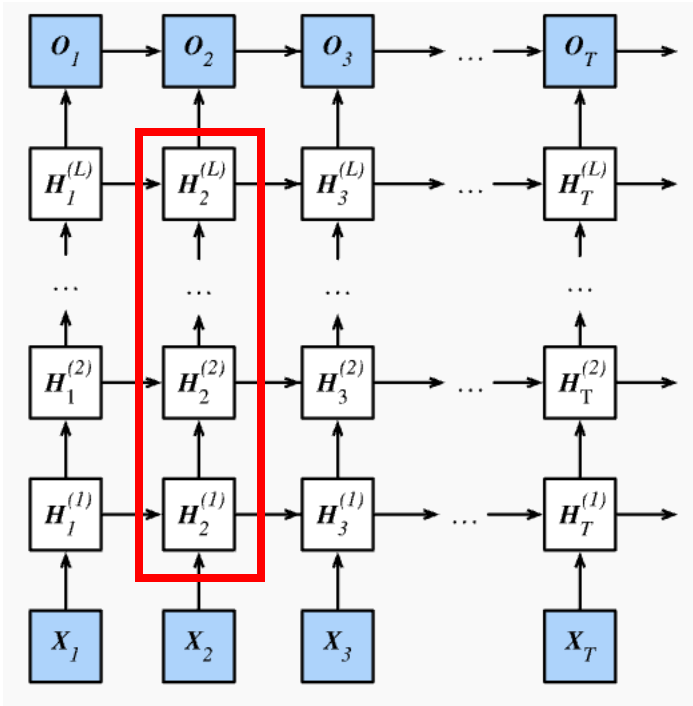
$$c_t^l \in \mathbb{R}^n$$

$$h_t^l \in \mathbb{R}^n$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Deep LSTM



$$\text{LSTM} : h_t^{l-1}, h_{t-1}^l, c_{t-1}^l \rightarrow \boxed{h_t^l, c_t^l}$$

$$\begin{pmatrix} \boxed{i} \\ \boxed{f} \\ \boxed{o} \\ \boxed{g} \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \text{tanh} \end{pmatrix} T_{2n,4n} \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

$$c_t^l = \boxed{f} \odot c_{t-1}^l + \boxed{i} \odot g$$

$$h_t^l = \boxed{o} \odot \tanh(c_t^l)$$

weights

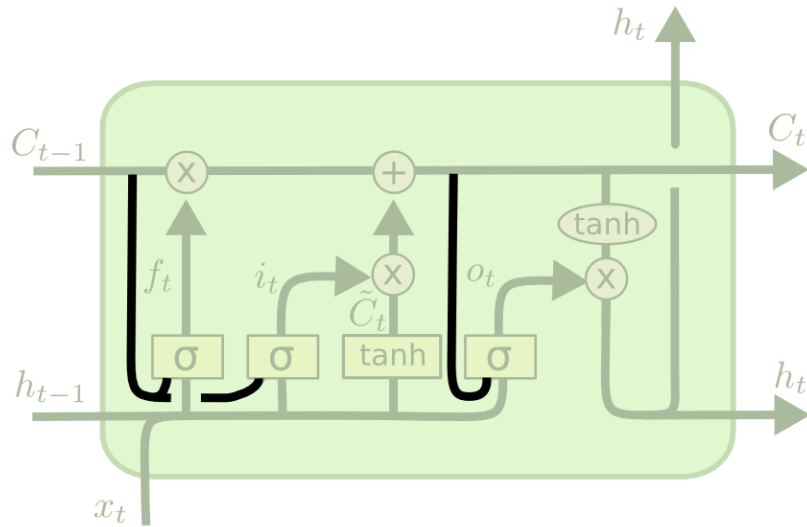
$$c_t^l \in \mathbb{R}^n$$

$$h_t^l \in \mathbb{R}^n$$

Image from Fig 8.10.1 of *Dive into Deep Learning* at https://classic.d2l.ai/chapter_recurrent-neural-networks/deep-rnn.html

Formulations from: Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).

Variant: peephole connections

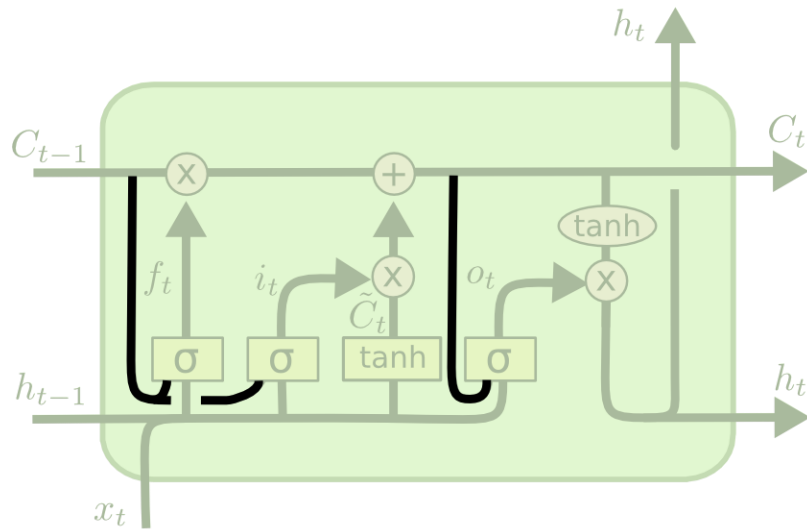


$$f_t = \sigma (W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f)$$

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

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

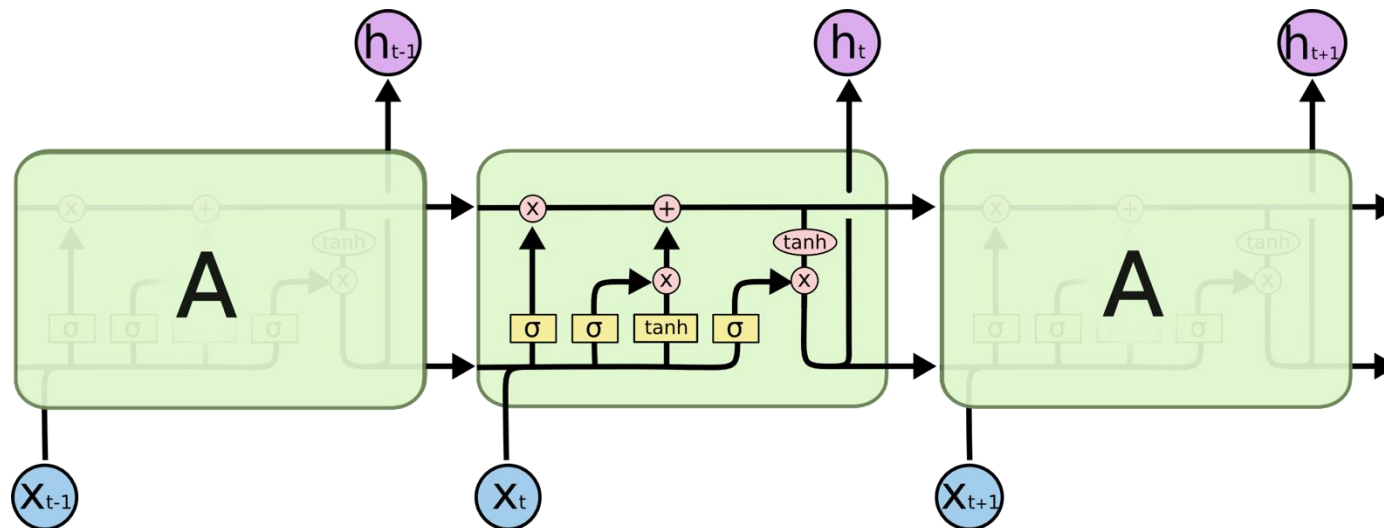
Variant: peephole connections



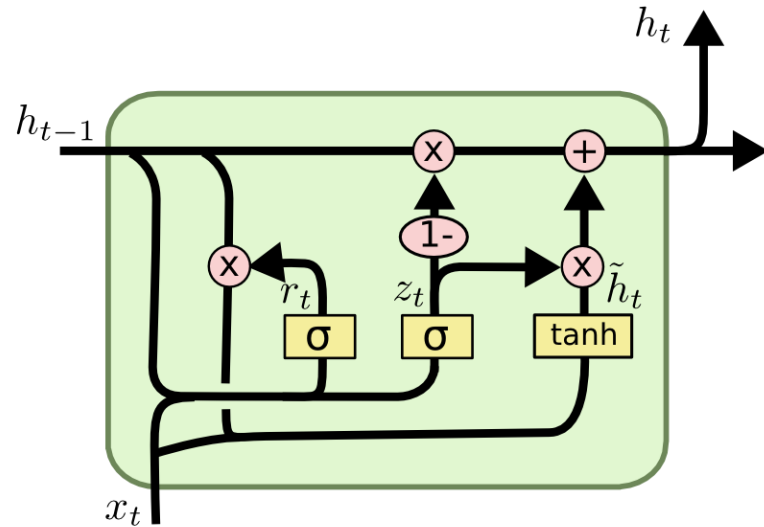
$$f_t = \sigma(W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f)$$

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

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



Variant: gated recurrent unit (GRU)



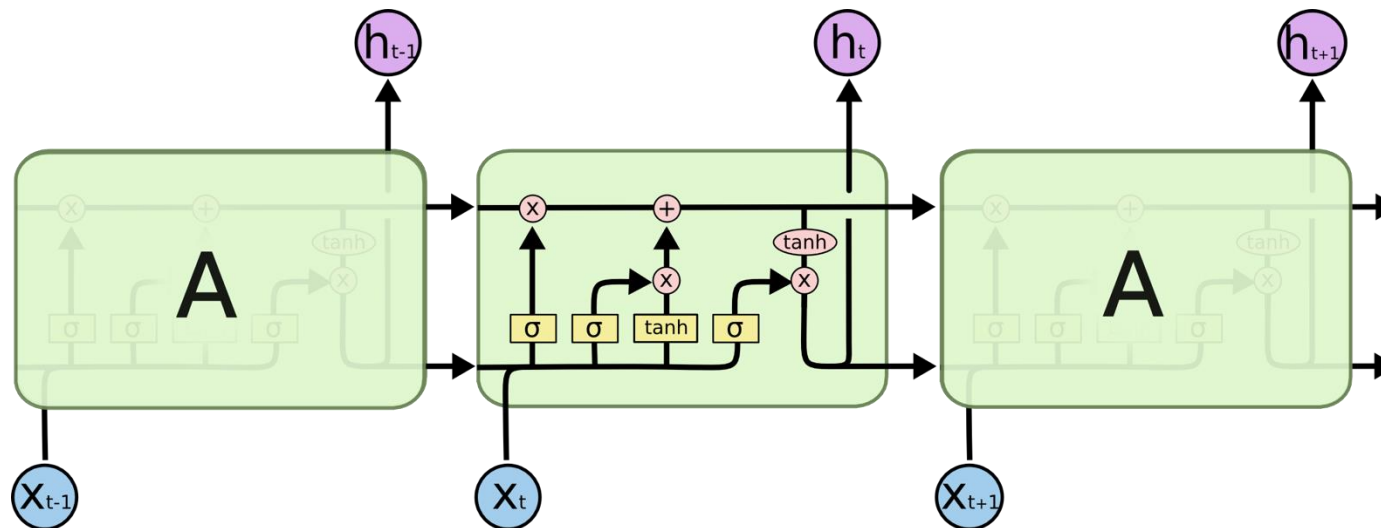
$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

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

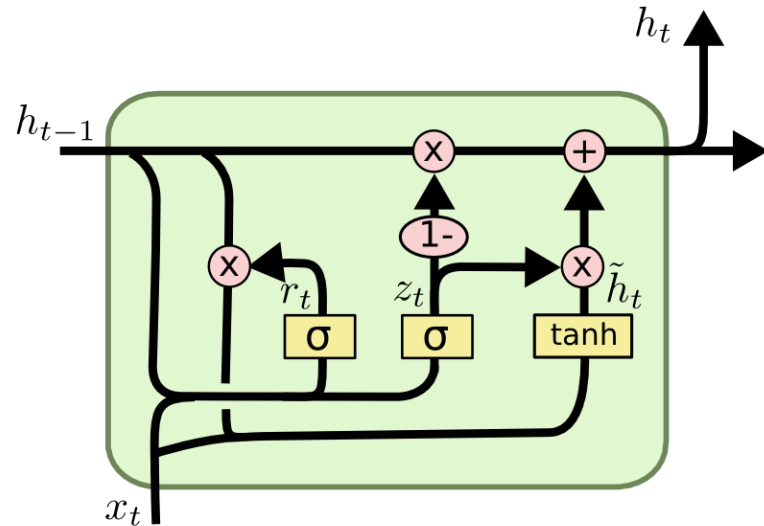
$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

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

Weighted average



Variant: gated recurrent unit (GRU)



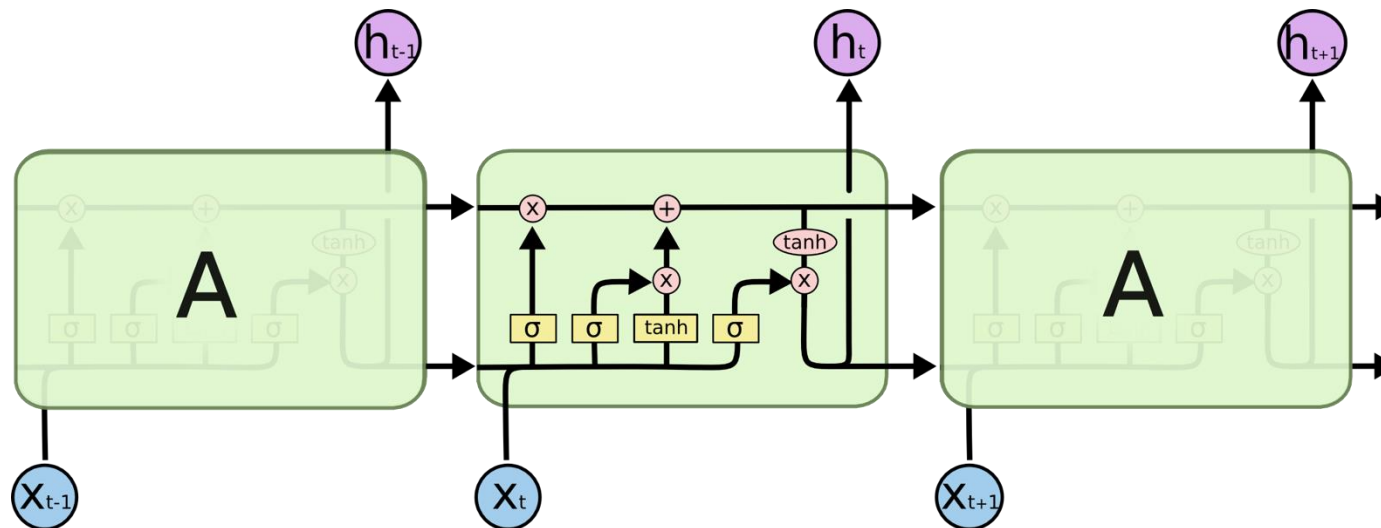
$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

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

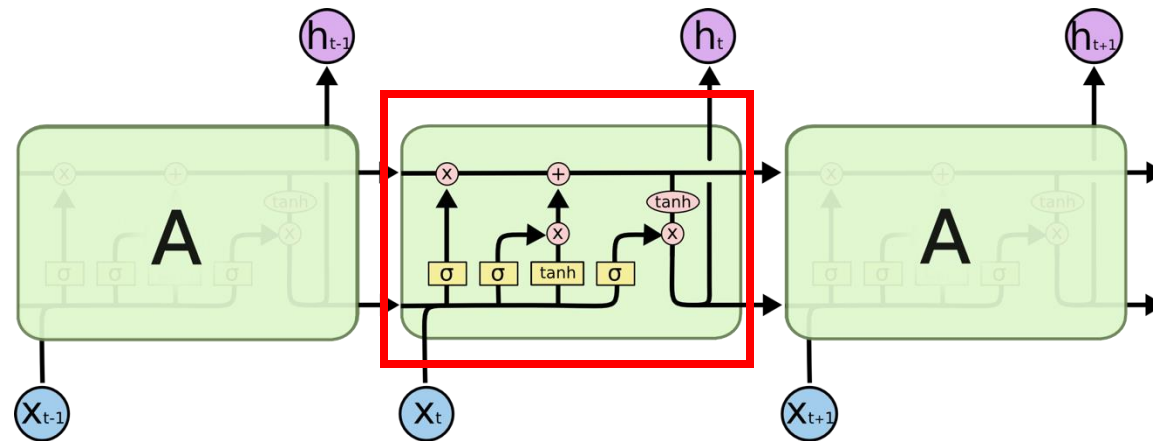
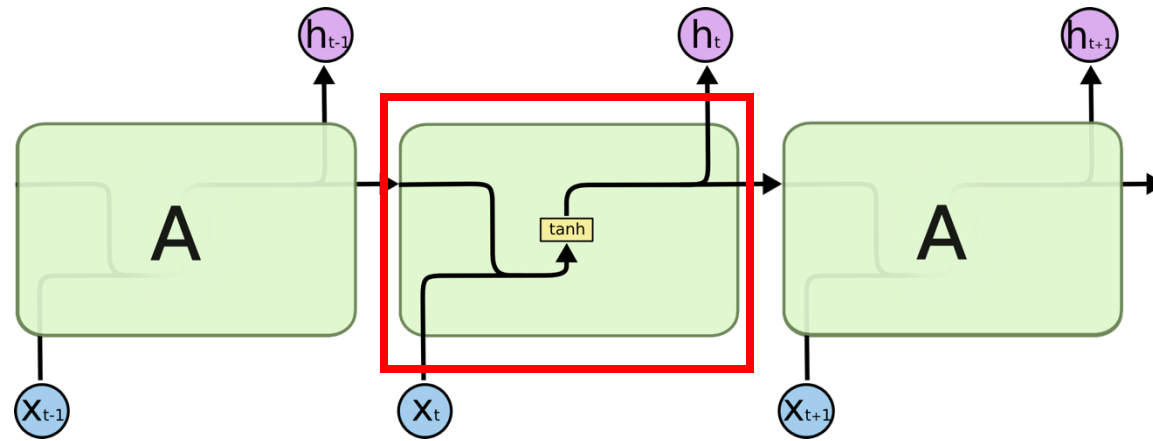
$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

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

Weighted average



Vanilla RNN and LSTM



Generating baby names

*Rudi Levette Berice Lussa Hany Mareanne Chrestina Carissy Marylen
Hammine Janye Marlise Jacacrie Hendred Romand Charienna Nenotto Ette
Dorane Wallen Marly Darine Salina Elvyn Ersia Maralena Minoria Ellia
Charmin Antley Nerille Chelon Walmor Evena Jeryly Stachon Charisa Allisa
Anatha Cathanie Geetra Alexie Jerin Cassen Herbett Cossie Velen Daurenge
Robester Shermond Terisa Licia Roselen Ferine Jayn Lusine Charyanne
Sales Sanny Resa Wallon Martine Merus Jelen Candica Wallin Tel Rachene
Tarine Ozila Ketia Shanne Arnande Karella Roselina Alessia Chasty Deland
Berther Geamar Jackein Mellisand Sagdy Nenc Lessie Rasemy Guen Gavi
Milea Anneda Margoris Janin Rodelin Zeanna Elyne Janah Ferzina Susta
Pey Castina*

Generating C code

```
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
    int error;
    if (fd == MARN_EPT) {
        /*
         * The kernel blank will coeld it to userspace.
         */
        if (ss->segment < mem_total)
            unblock_graph_and_set_blocked();
        else
            ret = 1;
        goto bail;
    }
    segaddr = in_SB(in.addr);
    selector = seg / 16;
    setup_works = true;
    for (i = 0; i < blocks; i++) {
        seq = buf[i++];
        bpf = bd->bd.next + i * search;
        if (fd) {
            current = blocked;
        }
    }
    rw->name = "Getjbbregs";
    bprm_self_clearl(&iv->version);
    regs->new = blocks[(BPF_STATS << info->historidac)] | PFMR_CLOBATHINC_SECONDS << 12;
    return segtable;
}
```

Generating academic articles

For $\bigoplus_{n=1, \dots, m} \mathcal{L}_{m, n} = 0$, hence we can find a closed subset \mathcal{H} in \mathcal{H} and any sets \mathcal{F} on X , U is a closed immersion of S , then $U \rightarrow T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \text{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \rightarrow V$. Consider the maps M along the set of points Sch_{fppf} and $U \rightarrow U$ is the fibre category of S in U in Section, ?? and the fact that any U affine, see Morphisms, Lemma ???. Hence we obtain a scheme S and any open subset $W \subset U$ in $\text{Sh}(G)$ such that $\text{Spec}(R') \rightarrow S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S . We claim that $\mathcal{O}_{X, x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X, x'} \rightarrow \mathcal{O}_{X', x'}$ is separated. By Algebra, Lemma ?? we can define a map of complexes $\text{GL}_{S'}(x'/S'')$ and we win. \square

To prove study we see that $\mathcal{F}|_U$ is a covering of \mathcal{X}' , and \mathcal{T}_i is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and \mathcal{F}_p exists and let \mathcal{F}_i be a presheaf of \mathcal{O}_X -modules on \mathcal{C} as a \mathcal{F} -module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\text{Spec}(k)} \mathcal{O}_{S, s} - i_X^{-1} \mathcal{F}$$

is a unique morphism of algebraic stacks. Note that

$$\text{Arrows} = (\text{Sch}/S)_{fppf}^{opp}, (\text{Sch}/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \mapsto (U, \text{Spec}(A))$$

is an open subset of X . Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S .

Proof. See discussion of sheaves of sets. \square

The result for prove any open covering follows from the less of Example ???. It may replace S by $X_{spaces, \text{etale}}$ which gives an open subspace of X and T equal to S_{Zar} , see Descent, Lemma ???. Namely, by Lemma ?? we see that R is geometrically regular over S .

Lemma 0.1. Assume (3) and (3) by the construction in the description. Suppose $X = \lim |X|$ (by the formal open covering X and a single map $\text{Proj}_X(\mathcal{A}) = \text{Spec}(B)$ over U compatible with the complex

$$\text{Set}(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X, \mathcal{O}_X}).$$

When in this case of to show that $\mathcal{Q} \rightarrow \mathcal{C}_{Z/X}$ is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition ?? (without element is when the closed subschemes are catenary. If T is surjective we may assume that T is connected with residue fields of S . Moreover there exists a closed subspace $Z \subset X$ of X where U in X' is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem

(1) f is locally of finite type. Since $S = \text{Spec}(R)$ and $Y = \text{Spec}(R)$.

Proof. This is form all sheaves of sheaves on X . But given a scheme U and a surjective étale morphism $U \rightarrow X$. Let $U \cap U = \coprod_{i=1, \dots, n} U_i$ be the scheme X over S at the schemes $X_i \rightarrow X$ and $U = \lim_i X_i$. \square

The following lemma surjective restrocomposes of this implies that $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{X, \dots, 0}$.

Lemma 0.2. Let X be a locally Noetherian scheme over S , $E = \mathcal{F}_{X/S}$. Set $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}_n$. Since $\mathcal{I}^n \subset \mathcal{I}^n$ are nonzero over $i_0 \leq \mathfrak{p}$ is a subset of $\mathcal{J}_{n, 0} \circ \overline{A}_2$ works.

Lemma 0.3. In Situation ???. Hence we may assume $\mathfrak{q}' = 0$.

Proof. We will use the property we see that \mathfrak{p} is the next functor (??). On the other hand, by Lemma ?? we see that

$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

where K is an F -algebra where δ_{n+1} is a scheme over S . \square

In practice

Machine translation

English - detected ↔ French

European economic area ×

Espace Economique Européen

🔊 🎤 📄 🔊 🛡️ Verified

[Open in Google Translate](#) • [Feedback](#)

In practice

Machine translation

The screenshot shows the Google Translate interface. At the top, it says "Machine translation". Below that, there are two dropdown menus for language selection. The first is set to "English - detected" and the second is set to "French". A double-headed arrow indicates the translation direction. The input text is "European economic area" with a close button (X) to its right. The output text is "Espace Economique Européen". Below the input text are icons for a speaker and a microphone. Below the output text are icons for a document, a speaker, and a shield with the word "Verified". At the bottom right, there are links for "Open in Google Translate" and "Feedback".

Q: can vanilla RNN handle machine translation?

In practice

Machine translation

The screenshot shows the Google Translate interface. At the top, it says "Machine translation". Below that, there are two dropdown menus for language selection. The left one is set to "English - detected" and the right one is set to "French". A double-headed arrow is between them. Below the left dropdown, the text "European economic area" is entered, with a close button (X) to its right. Below the right dropdown, the translated text "Espace Economique Européen" is displayed. At the bottom of the interface, there are icons for speaker and microphone on the left, and copy, speaker, and a "Verified" badge on the right. At the very bottom, there are links for "Open in Google Translate" and "Feedback".

Q: can we find some correlation?

In practice

The image shows a Google Translate interface. On the left, the source language is 'English - detected' and the text 'European economic area' is entered. The word 'European' is highlighted with a red box. On the right, the target language is 'French' and the translation 'Espace Economique Européen' is shown. The word 'Européen' is also highlighted with a red box. A red arrow points from the red box around 'European' to the red box around 'Européen'. Below the text, there are icons for audio playback and microphone input. At the bottom right, there is a 'Verified' badge and a link to 'Open in Google Translate'.

English - detected ↔ French

European ×
economic area

Espace Economique
Européen

Open in Google Translate • Feedback

In practice

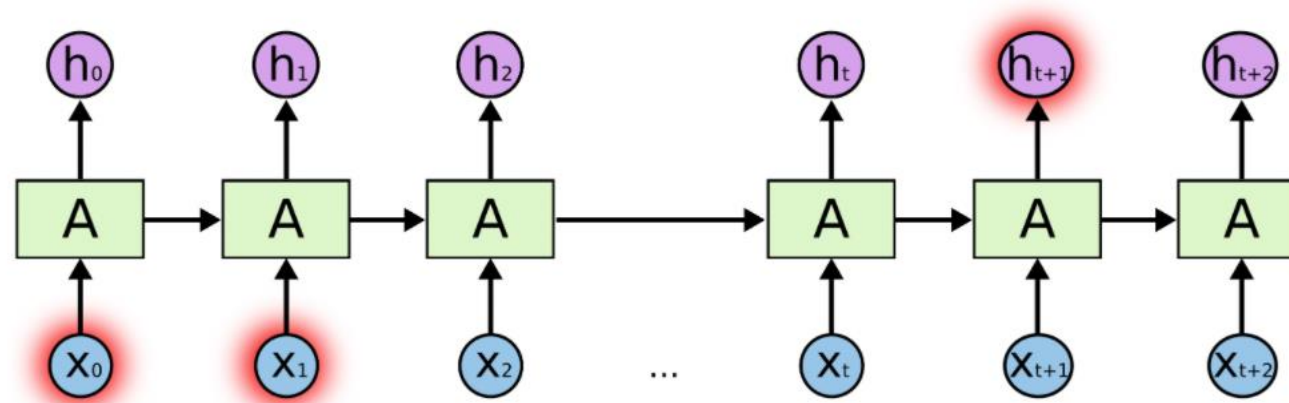
The image shows a Google Translate interface. On the left, the source language is 'English - detected' and the text 'European economic area' is entered. The word 'economic' is highlighted with a red box. On the right, the target language is 'French' and the translation 'Espace Economique Européen' is shown. The word 'Economique' is highlighted with a red box. A red line connects the 'economic' box to the 'Economique' box. Below the text are icons for audio playback and microphone input. At the bottom right, there is a 'Verified' badge and a 'Feedback' link. At the bottom center, there is a link to 'Open in Google Translate'.

English - detected ↔ French

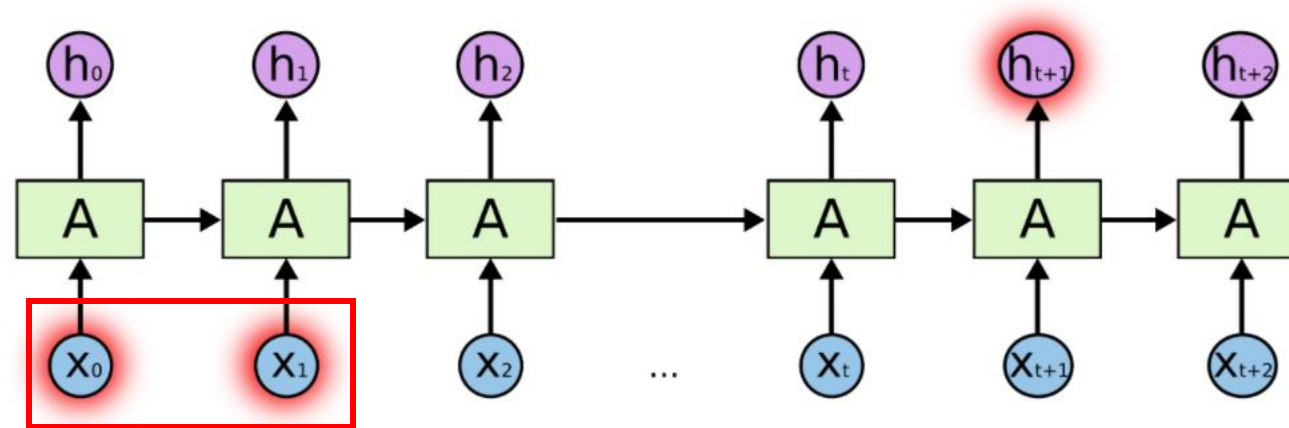
European economic area × Espace Economique Européen

Open in Google Translate • Feedback

Vanilla RNN information flow

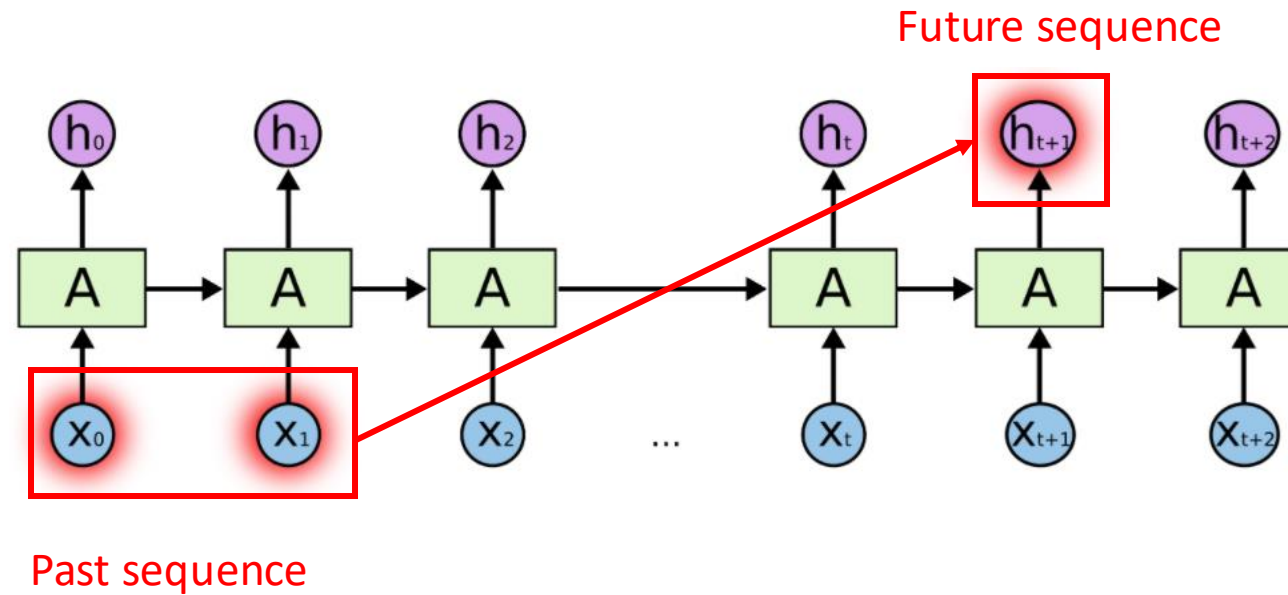


Vanilla RNN information flow

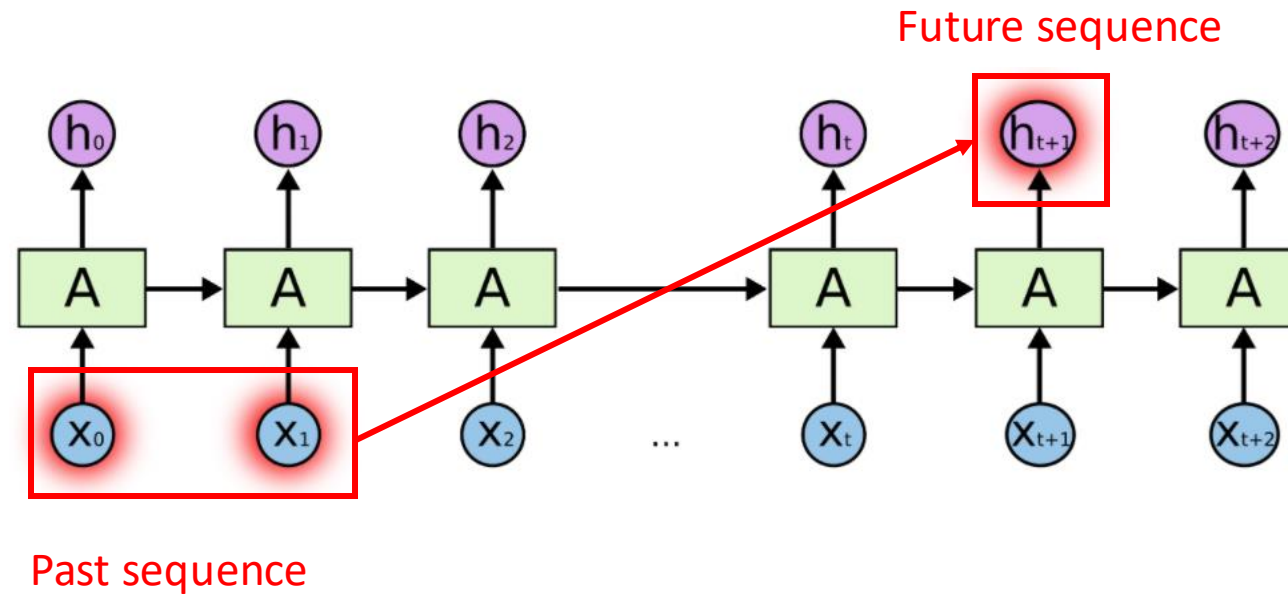


Past sequence

Vanilla RNN information flow

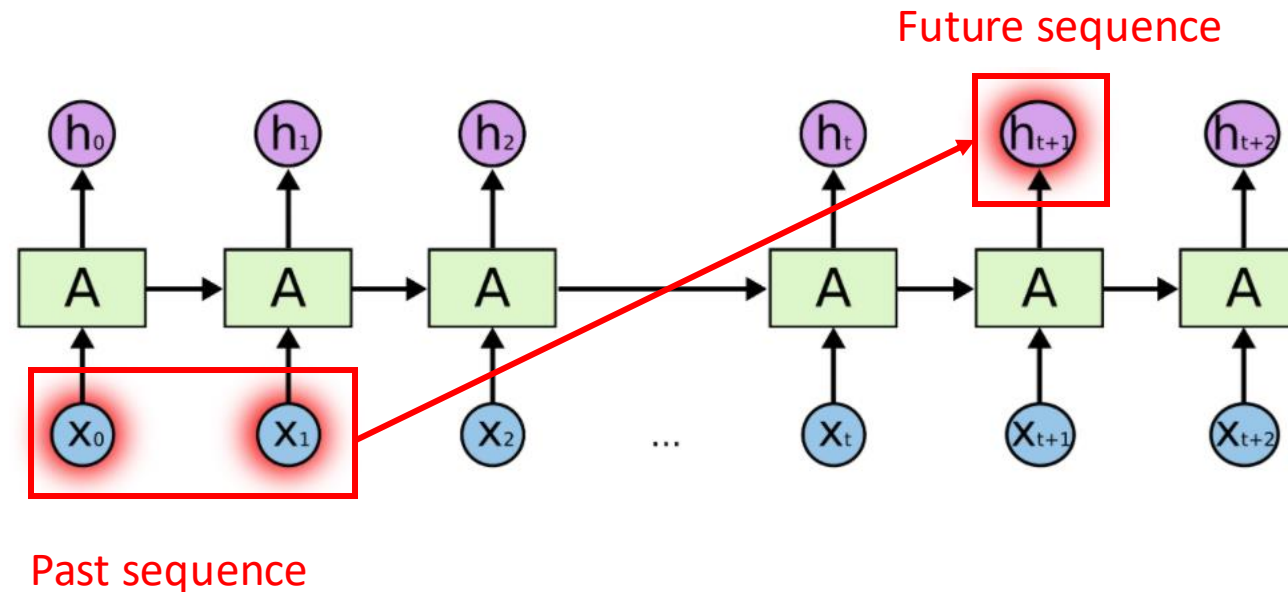


Vanilla RNN information flow



the clouds are in the sky

Vanilla RNN information flow



I like this town very much. I started my undergraduate study in 2020 and my major is computer science. I like programming and reading. I usually get up at 7AM and do some exercise. I also go fishing at weekend. I grew up in **France**. I spent my childhood outdoors. Whether it was riding my bicycle around my neighborhood pretending it was a motorcycle, making mud cakes, going on treasure hunts, making and selling perfume out of strong smelling flowers, or simply laying on the grass underneath the sun with a soccer ball waiting for someone to come out and play with me, the outdoors was where I spent my childhood and I cannot be more appreciative of it. I speak fluent **French**.

In practice

English - detected ↔ French

1st
European
economic area

Espace Economique
Européen
3rd

Open in Google Translate • Feedback

Q: is vanilla RNN able to use information flow to generate **Européen**?

In practice

The image shows a Google Translate interface. On the left, the source language is 'English - detected' and the text is 'European economic area'. The word 'area' is highlighted with a red box and labeled '3rd'. On the right, the target language is 'French' and the text is 'Espace Economique Européen'. The word 'Espace' is highlighted with a red box and labeled '1st'. Below the text are icons for audio playback and microphone input. At the bottom right, there is a 'Verified' badge and a 'Feedback' link. At the bottom center, there is a link to 'Open in Google Translate'.

English - detected ↔ French

European economic **area** × Espace Economique Européen

3rd 1st

Open in Google Translate • Feedback

Q: is vanilla RNN able to use information flow to generate **Espace**?

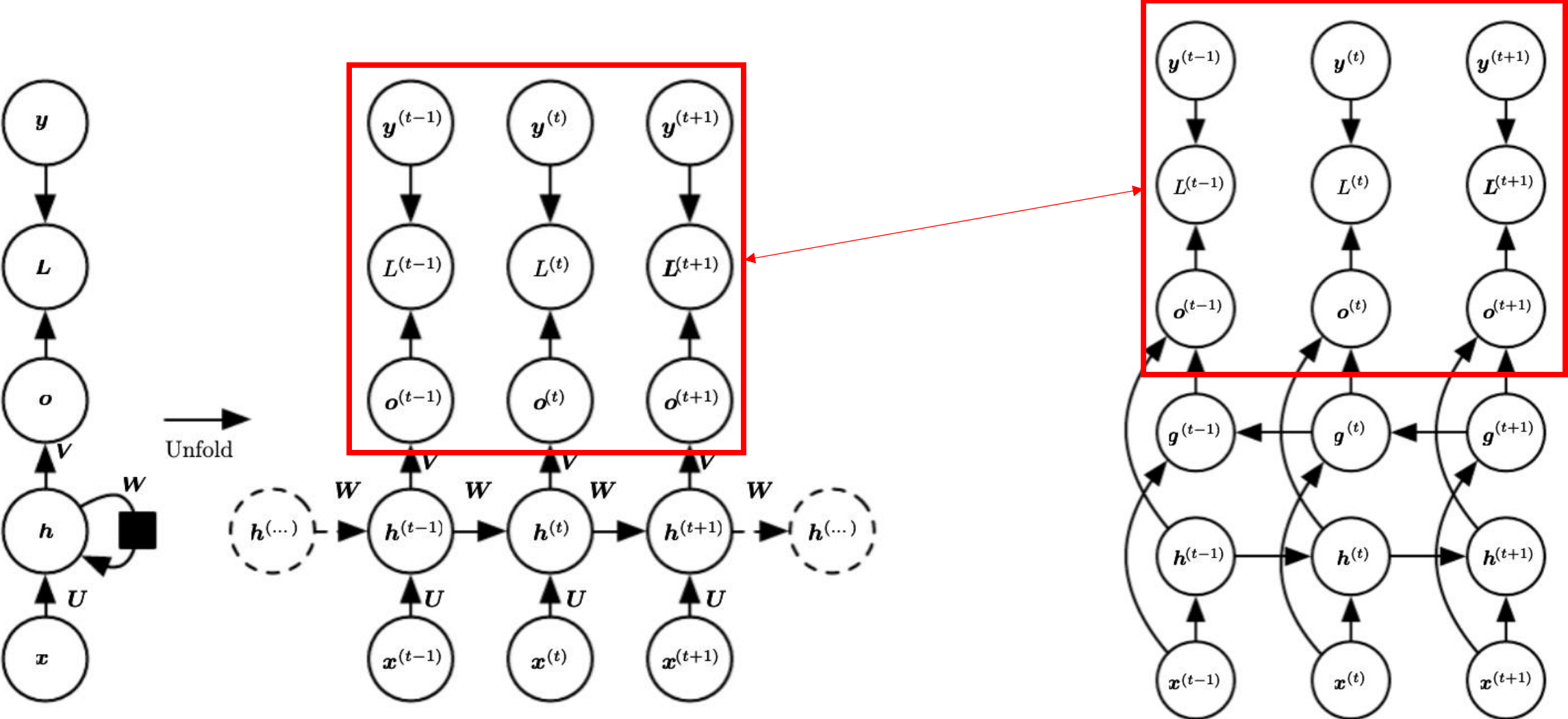
In practice

The image shows a Google Translate interface. On the left, the source language is 'English - detected' and the text is 'European economic area'. The word 'area' is highlighted with a red box and labeled '3rd'. On the right, the target language is 'French' and the translation is 'Espace Economique Européen'. The word 'Espace' is highlighted with a red box and labeled '1st'. Below the text are icons for speaker, microphone, copy, and a 'Verified' badge. At the bottom right, there are links for 'Open in Google Translate' and 'Feedback'.

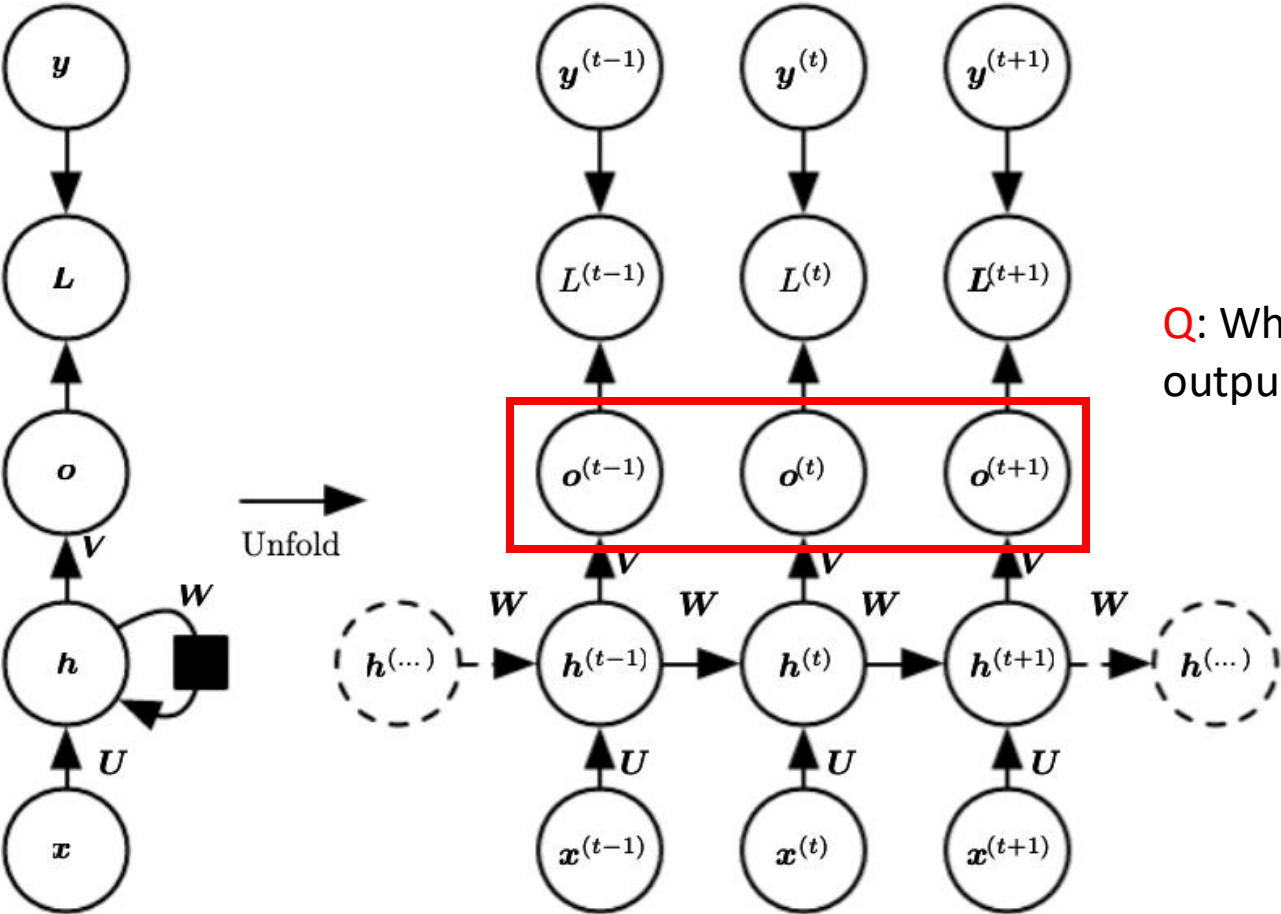
Q: is vanilla RNN able to use information flow to generate **Espace**?

Q: what if we need some **future sequence** to determine the output sequence?

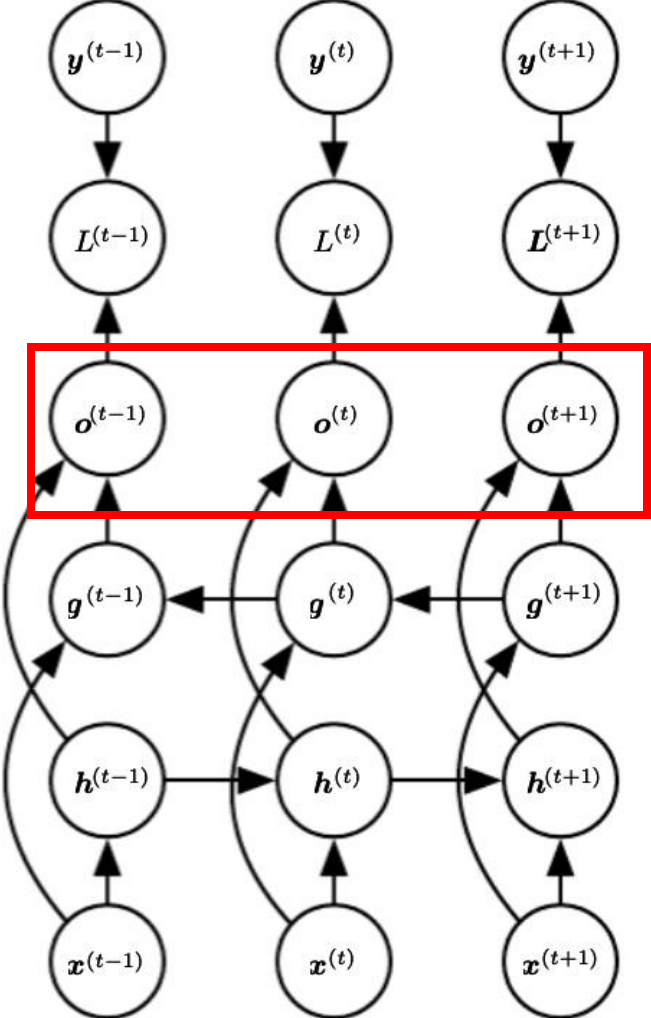
Bi-directional RNN



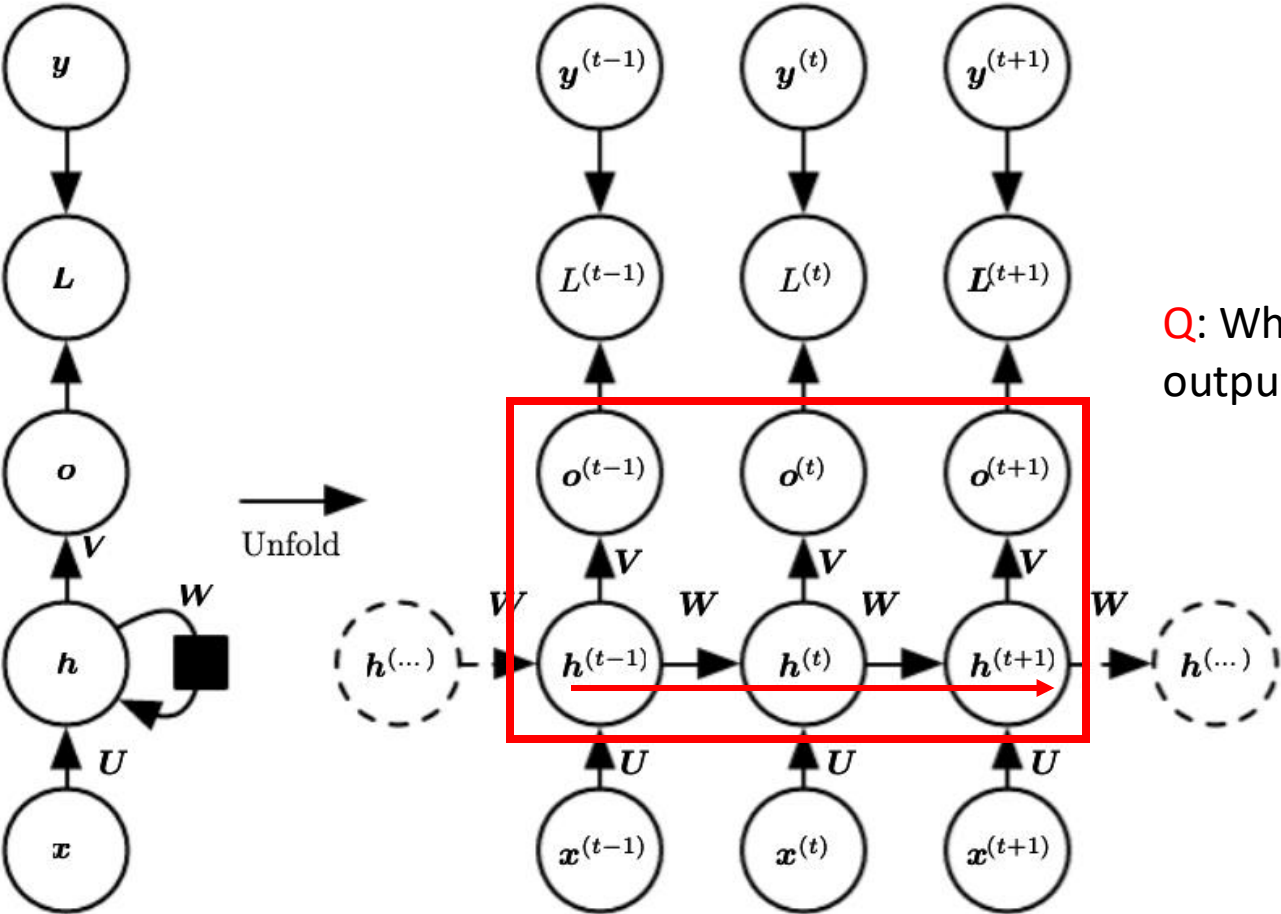
Bi-directional RNN



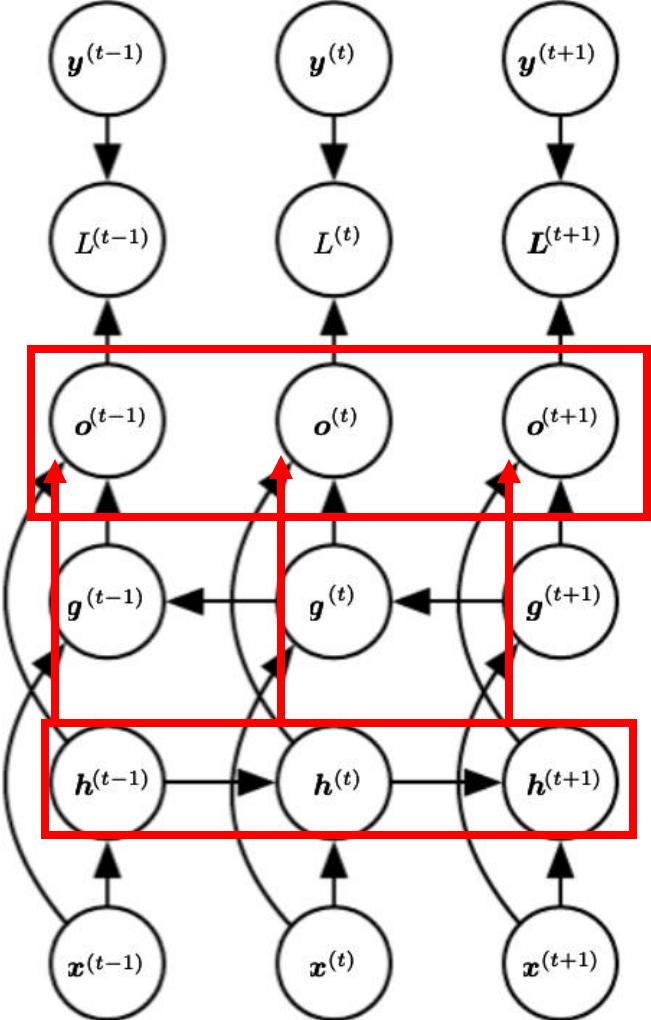
Q: What determine output sequence?



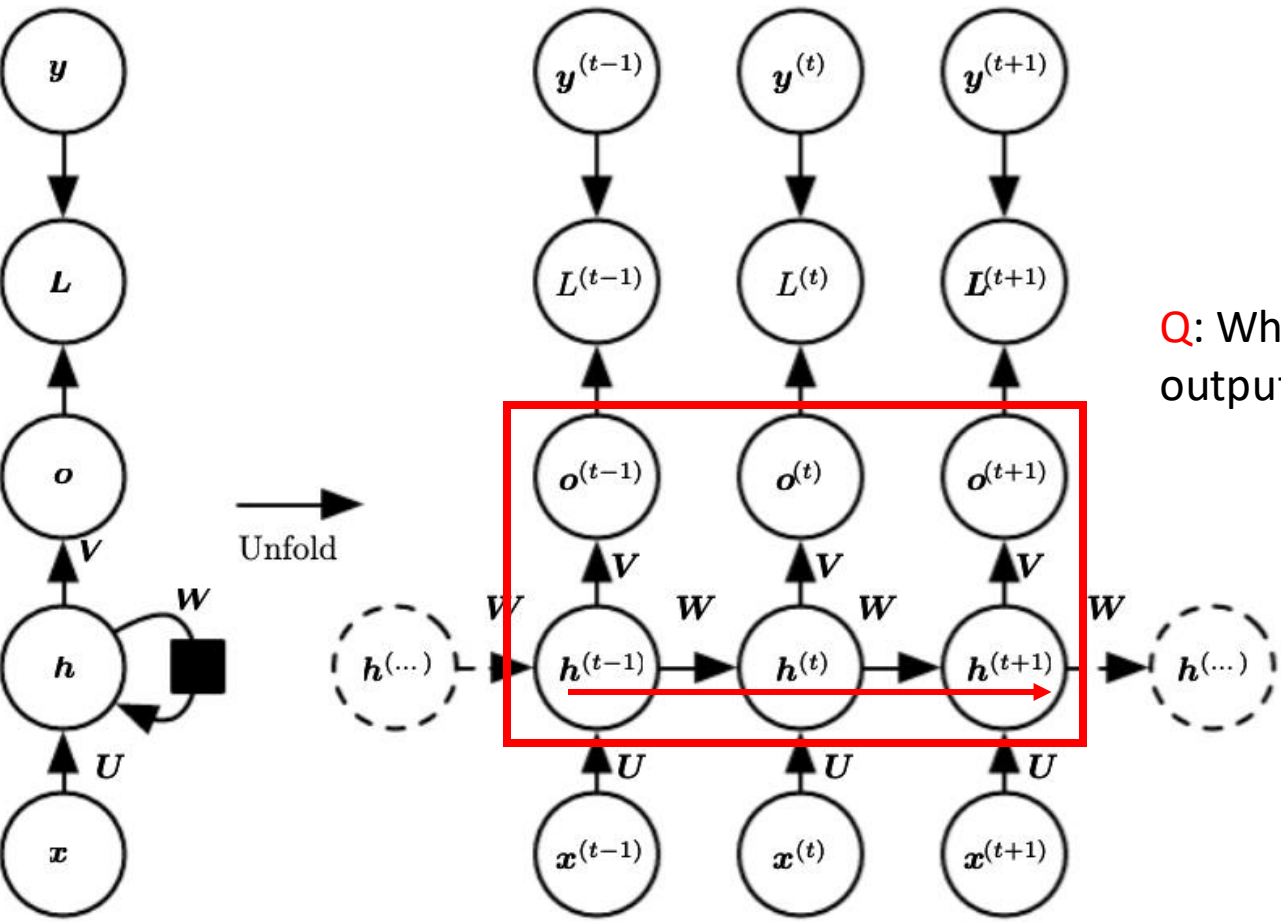
Bi-directional RNN



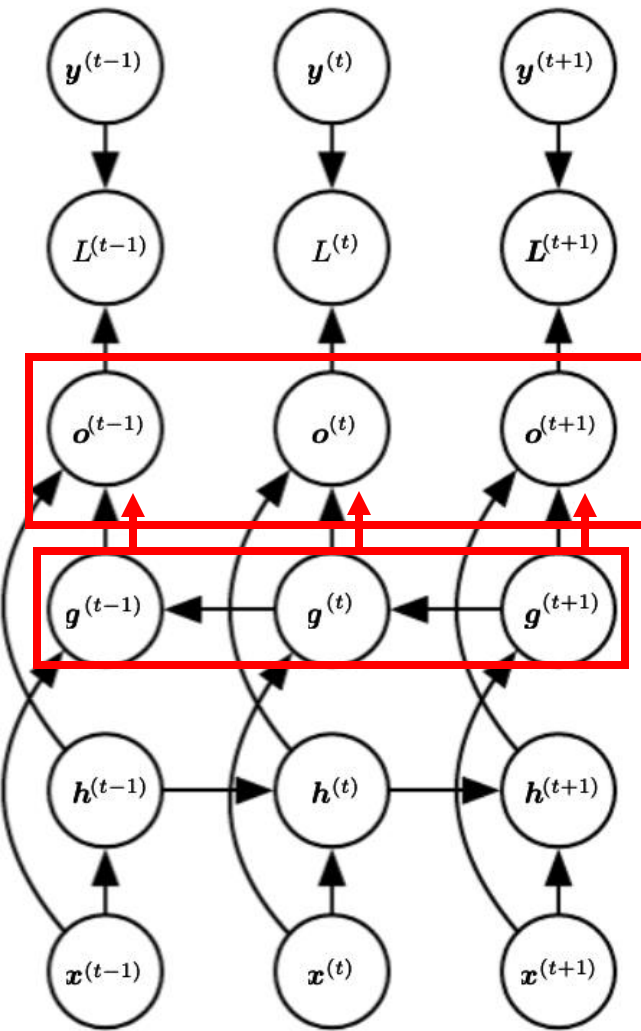
Q: What determine output sequence?



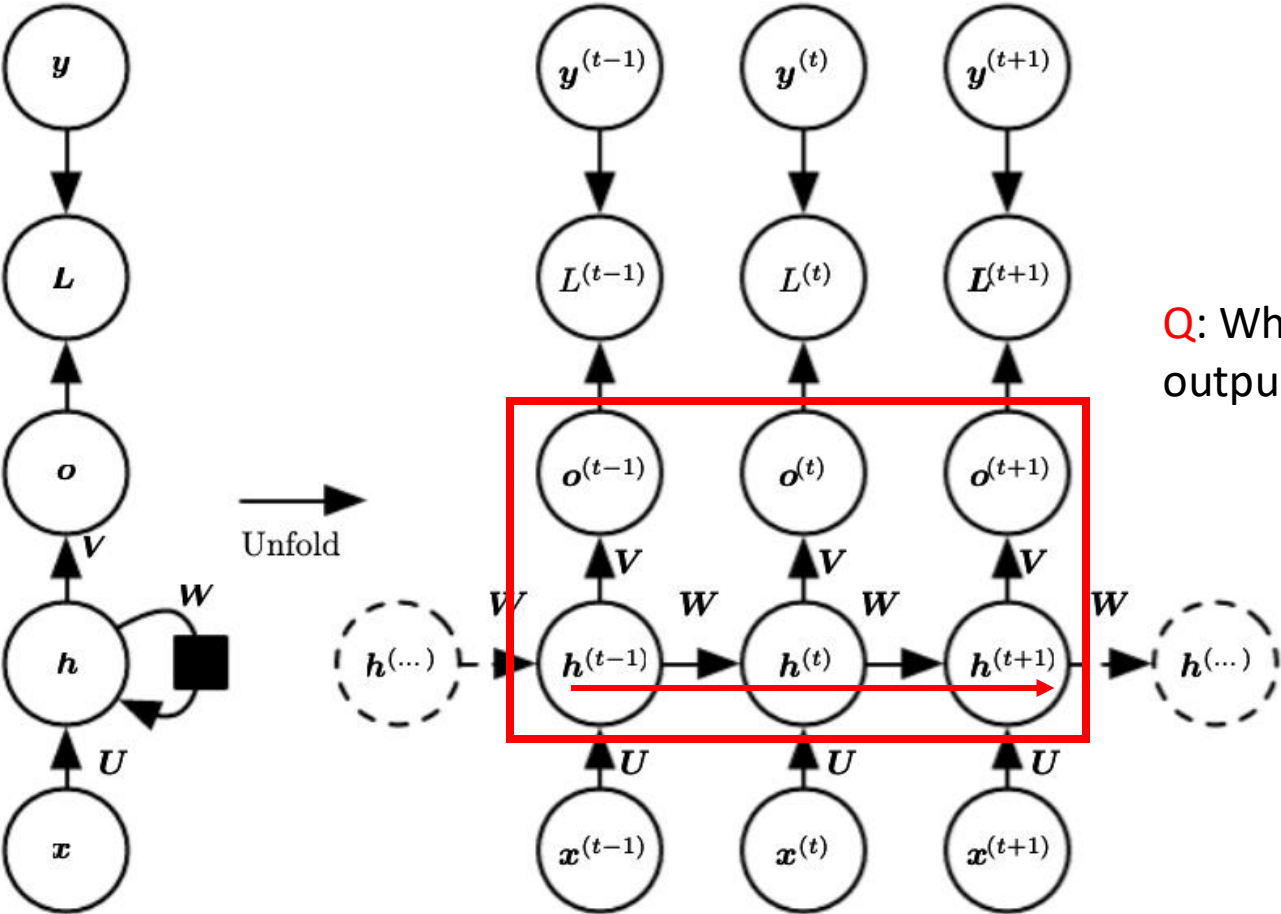
Bi-directional RNN



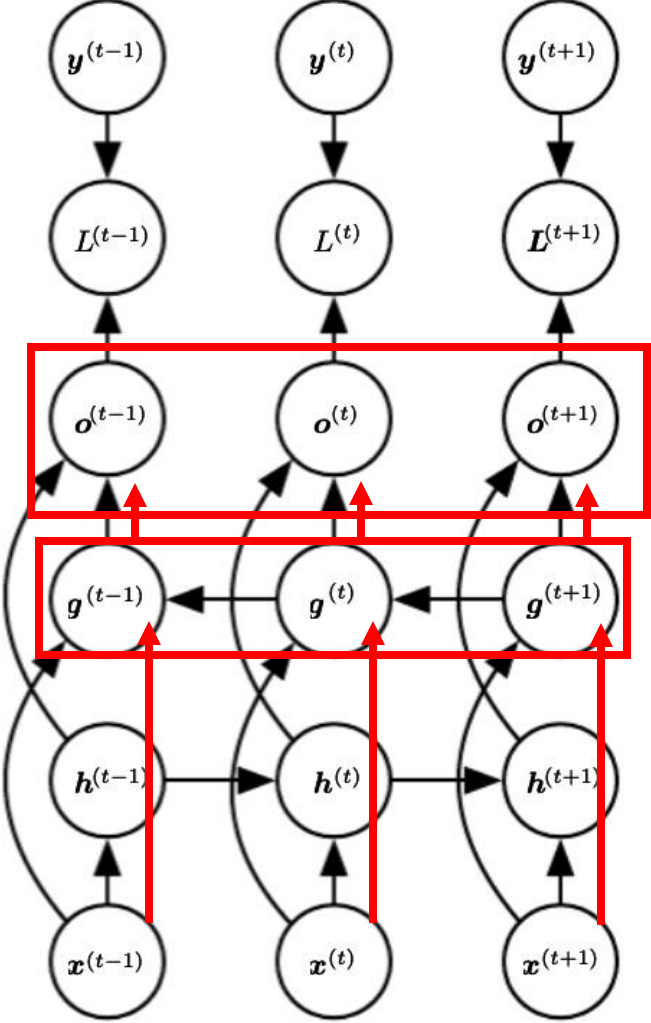
Q: What determine output sequence?



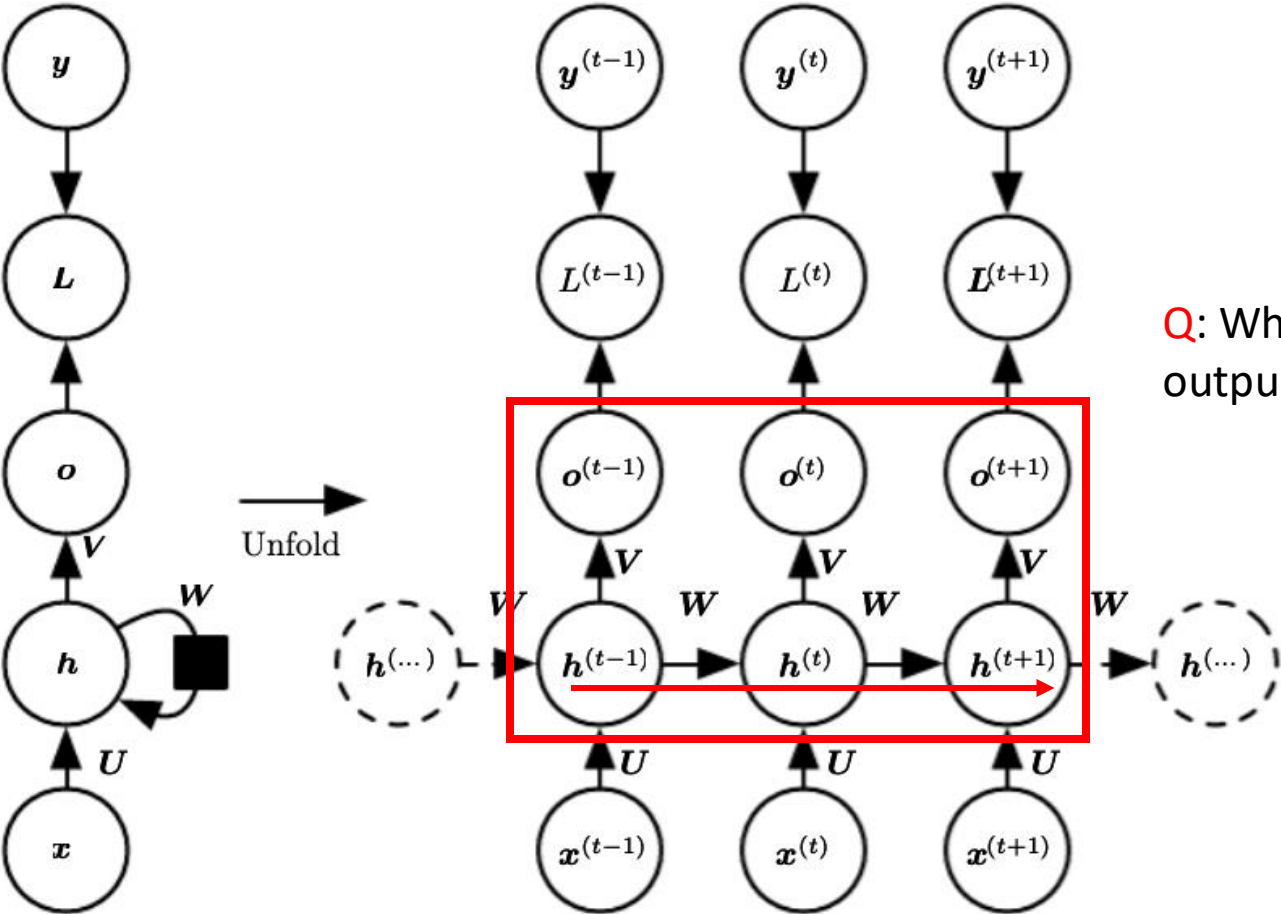
Bi-directional RNN



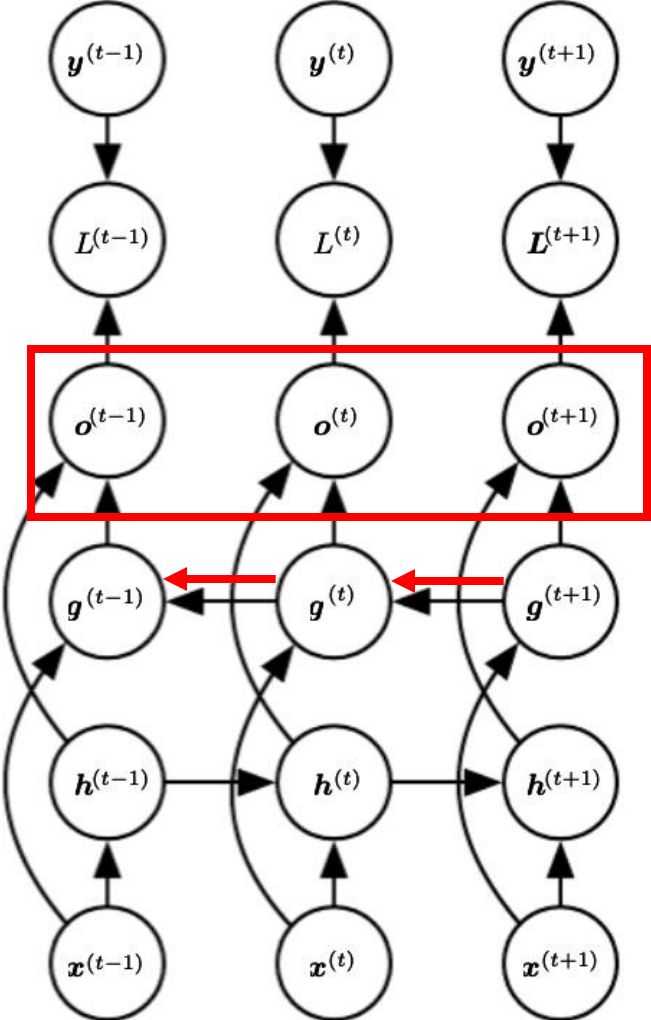
Q: What determine output sequence?



Bi-directional RNN



Q: What determine output sequence?



Seq2seq architecture

The image shows a Google Translate interface. On the left, the source language is set to 'English' and the text 'european economic area' is entered. On the right, the target language is set to 'Japanese' and the translated text '欧州経済領域' (Ōshūkeizairyōiki) is displayed. A double-headed arrow between the language boxes indicates the bidirectional nature of the Seq2Seq model. Below the text boxes are icons for audio playback and microphone input. At the bottom right, there are links for 'Open in Google Translate' and 'Feedback'.

English ↔ Japanese

european economic area ×

欧州経済領域
Ōshūkeizairyōiki

Open in Google Translate • Feedback

Seq2seq architecture

The image shows a Google Translate interface illustrating a Seq2Seq architecture. On the left, the source language is set to English, and the target language is Japanese. The source text "european economic area" is displayed, with "european" and "economic area" highlighted in red boxes. Below the source text are icons for a speaker and a microphone. On the right, the translated text "欧州経済領域" (Ōshūkeizairyōiki) is shown in a light gray box, with a copy icon and a speaker icon below it. A double-headed arrow between the language dropdowns indicates the bidirectional nature of the architecture. At the bottom right, there are links for "Open in Google Translate" and "Feedback".

English ↔ Japanese

european ×
economic area

欧州経済領域
Ōshūkeizairyōiki

Open in Google Translate • Feedback

Seq2seq architecture

The image shows a Google Translate interface. On the left, the source language is set to 'English'. The input text is 'European Economic Area', with 'European', 'Economic', and 'Area' each enclosed in a red rectangular box. A grey 'X' icon is positioned to the right of the input text. Below the input are icons for a speaker and a microphone. On the right, the target language is set to 'Japanese'. The output text is '欧州経済領域' (Ōshūkeizairyōiki), with each character ('欧', '州', '経', '済', '領', '域') enclosed in a red rectangular box. Below the output are icons for a copy button and a speaker. A double-headed arrow between the language boxes indicates the translation direction.

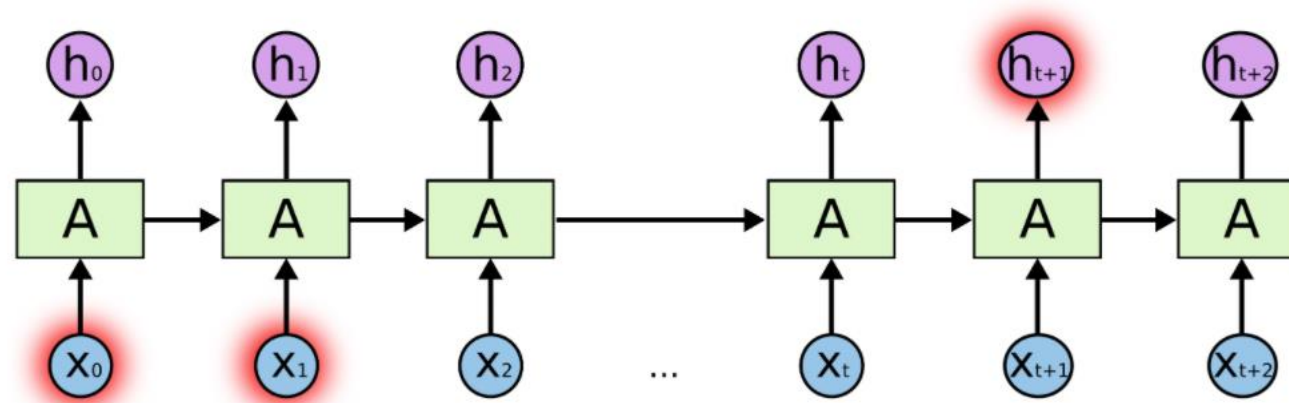
[Open in Google Translate](#) • [Feedback](#)

Seq2seq architecture

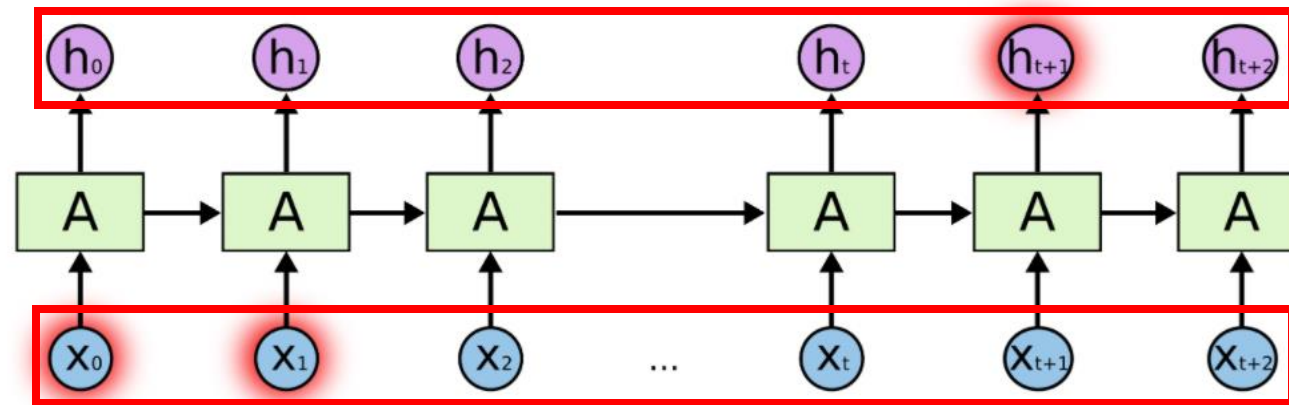
The image shows a screenshot of the Google Translate interface. On the left, the source language is set to 'English' and the text 'european economic area' is entered. The words 'european', 'economic', and 'area' are each enclosed in a red rectangular box. On the right, the target language is set to 'Japanese' and the translation '欧州経済領域' (Ōshūkeizairyōiki) is displayed. Each character in the Japanese text is enclosed in a red rectangular box. Below the text input fields are icons for speaker and microphone. At the bottom right, there are links for 'Open in Google Translate' and 'Feedback'.

Q: is vanilla RNN able to generate an output with **different length** of input?

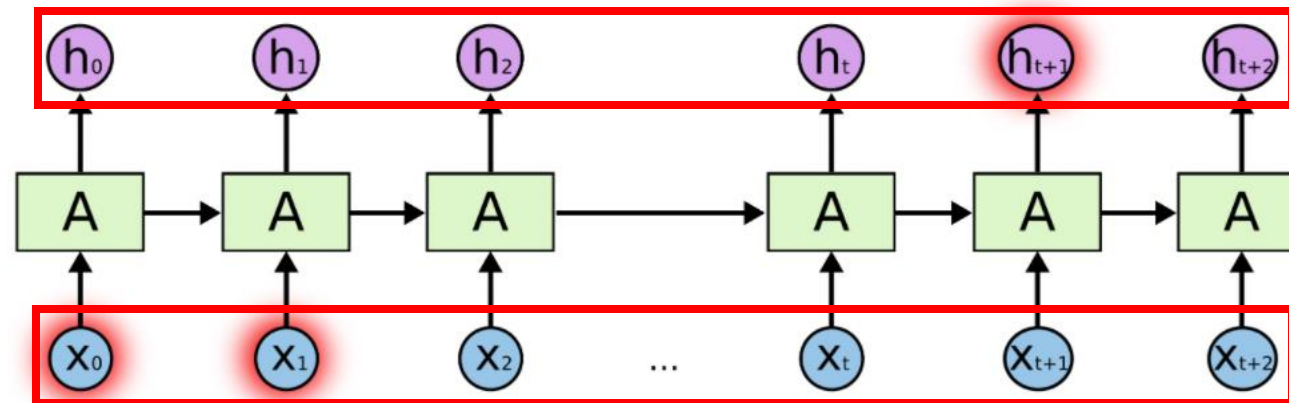
Vanilla RNN information flow



Vanilla RNN information flow

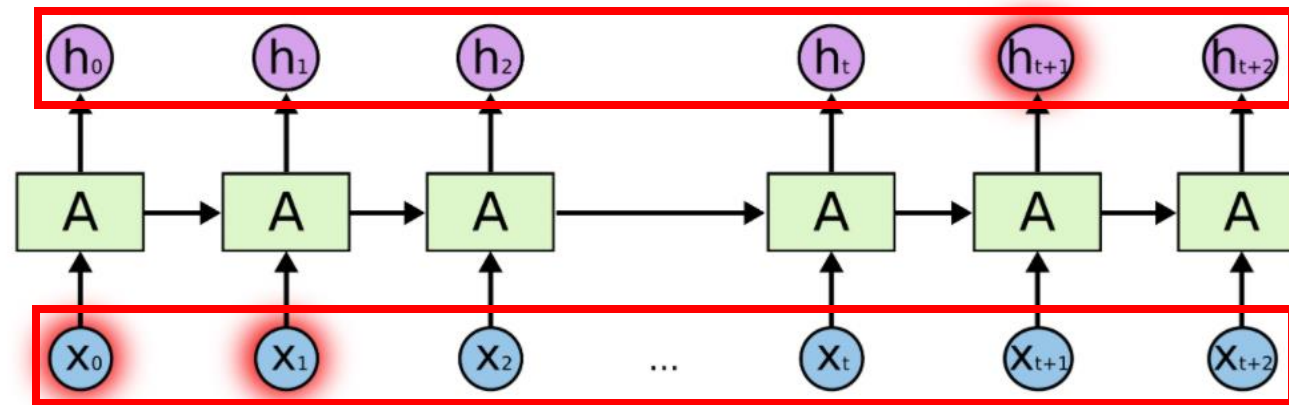


Vanilla RNN information flow



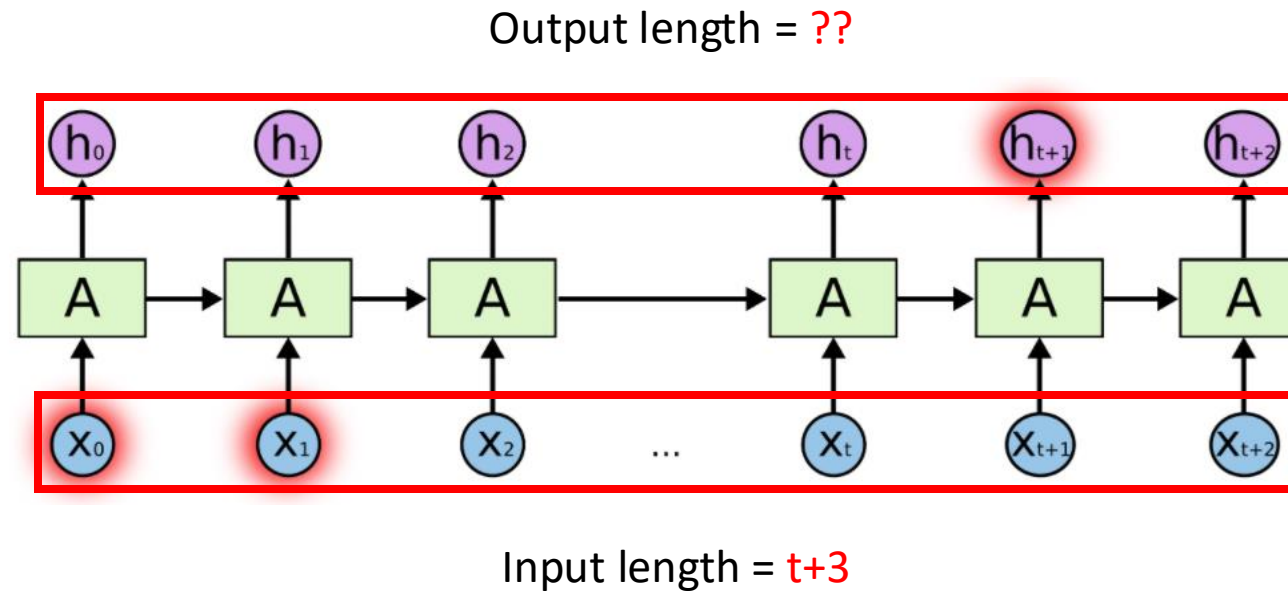
Input length = ??

Vanilla RNN information flow

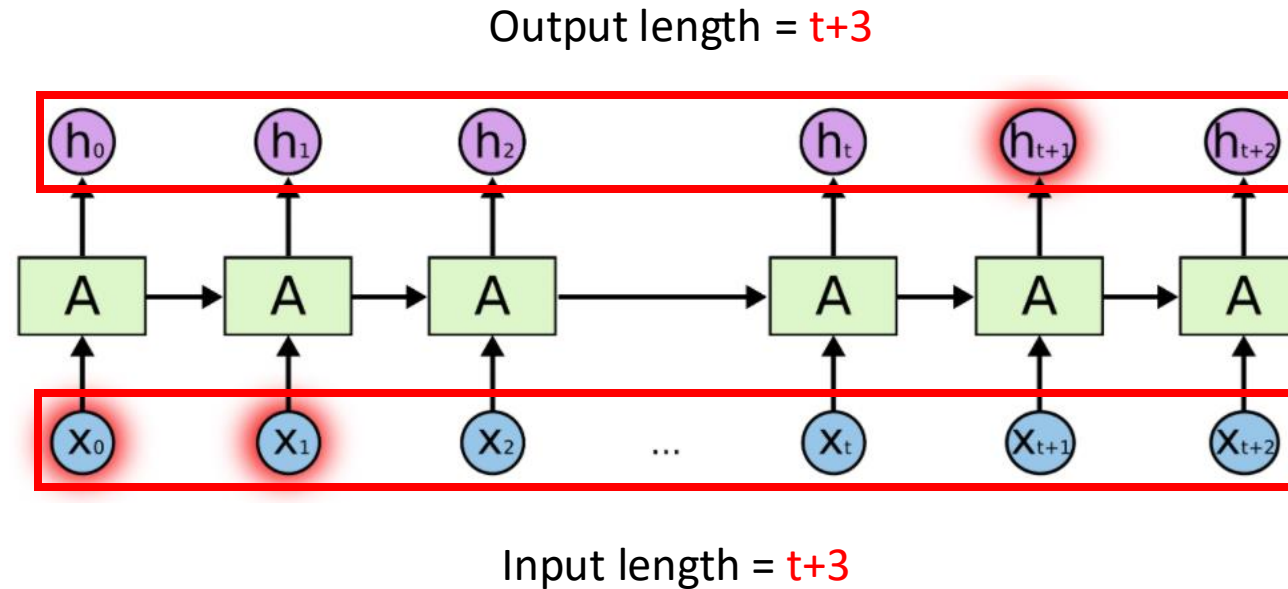


Input length = $t+3$

Vanilla RNN information flow



Vanilla RNN information flow



Q: what if the input and output sequences are of different length?

Seq2seq RNN architecture

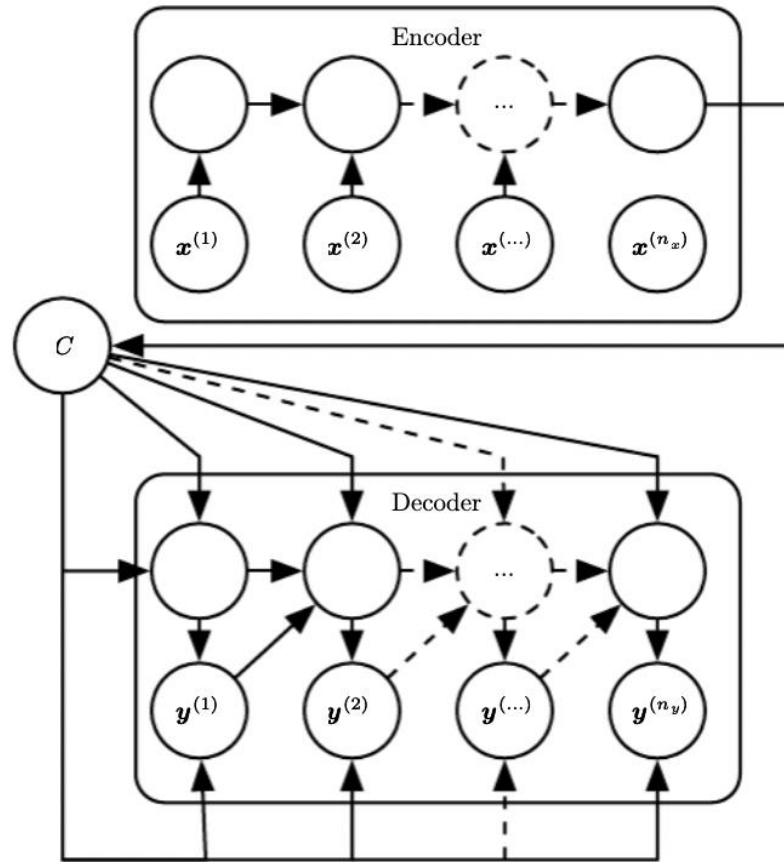
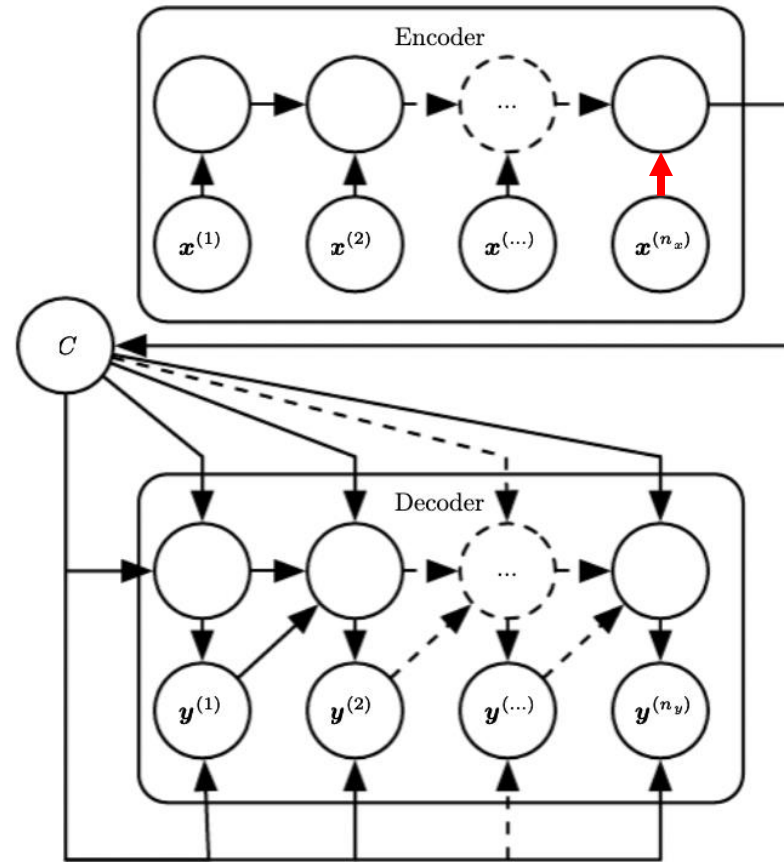


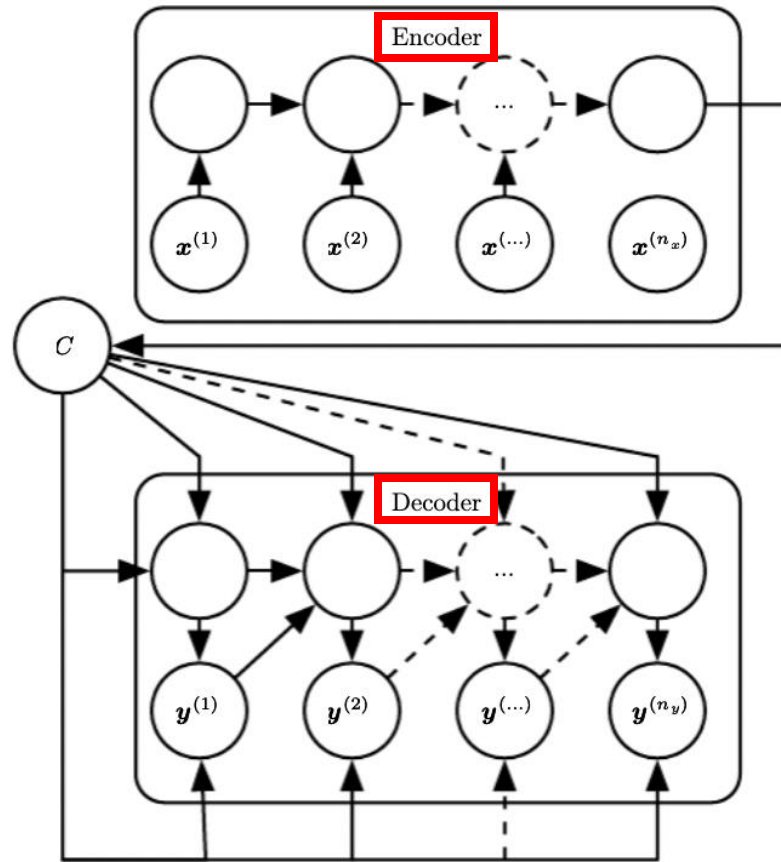
Figure 10.12 in deep learning book

Seq2seq RNN architecture

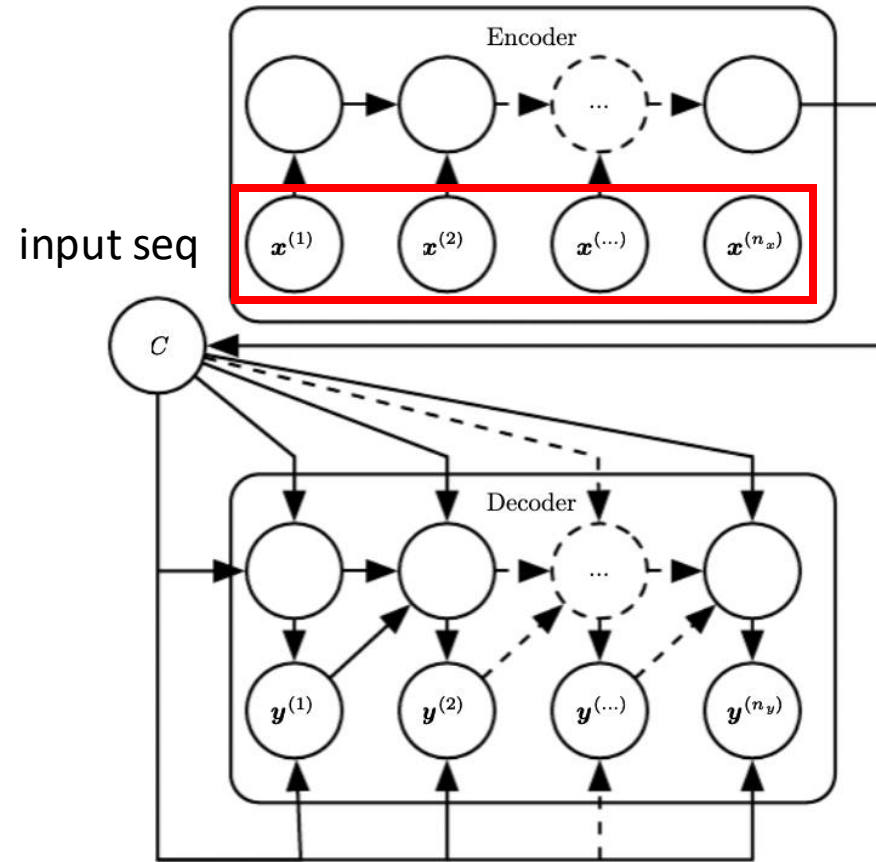


A typo in Figure 10.12 in deep learning book

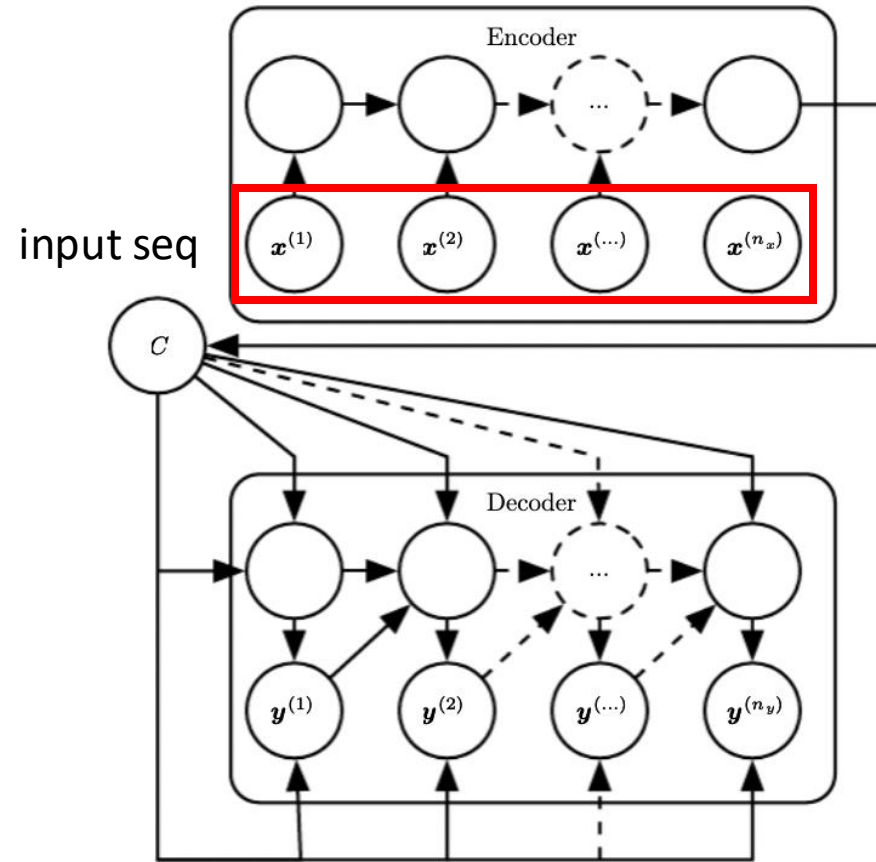
Seq2seq RNN architecture



Seq2seq RNN architecture

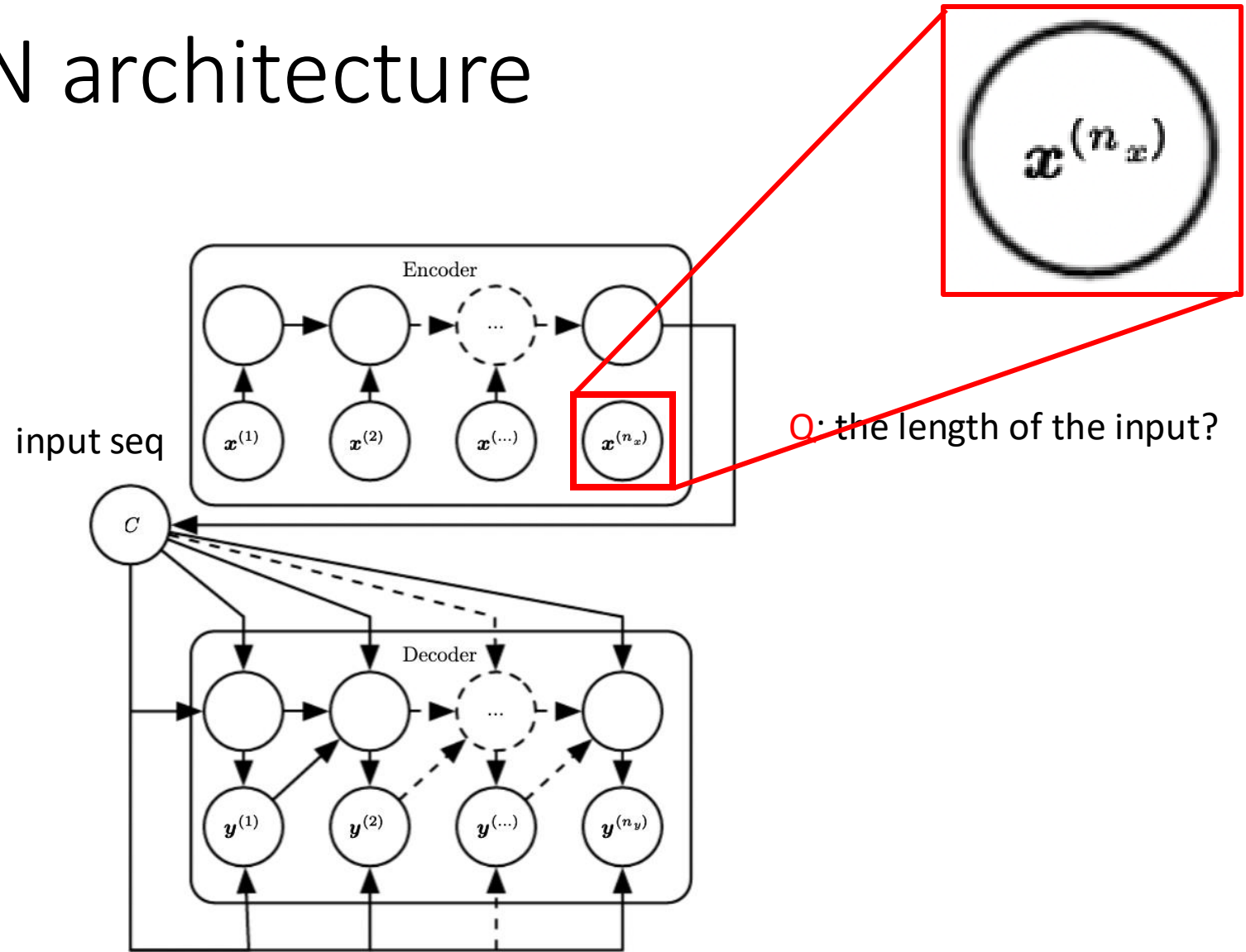


Seq2seq RNN architecture

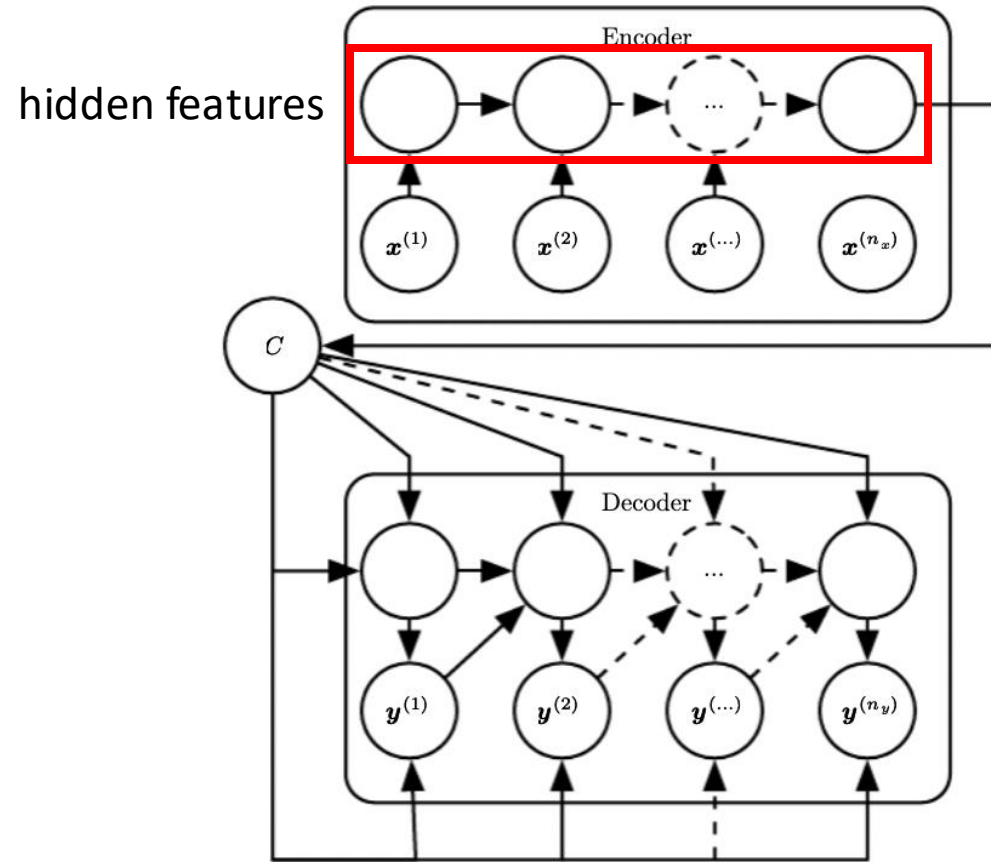


Q: the length of the input?

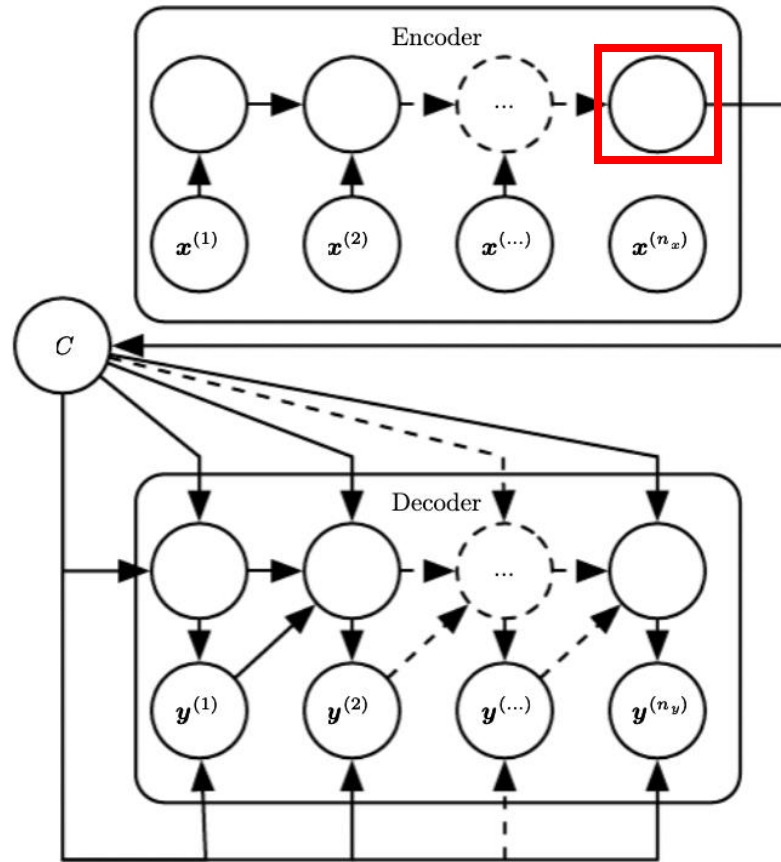
Seq2seq RNN architecture



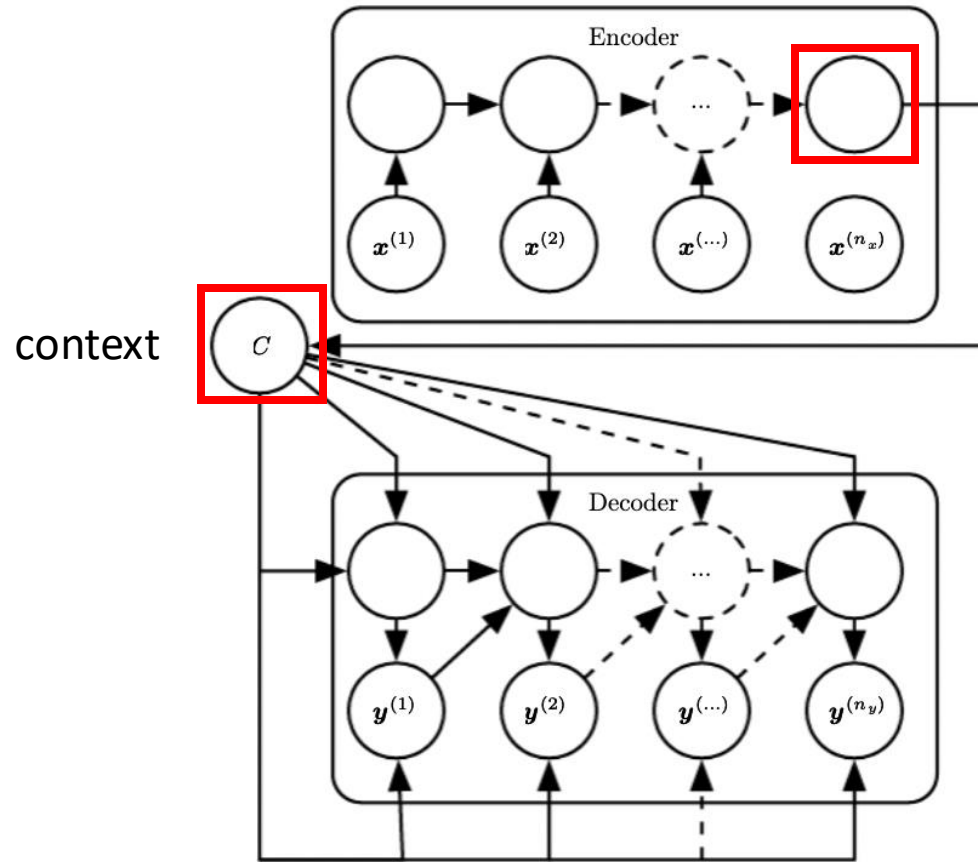
Seq2seq RNN architecture



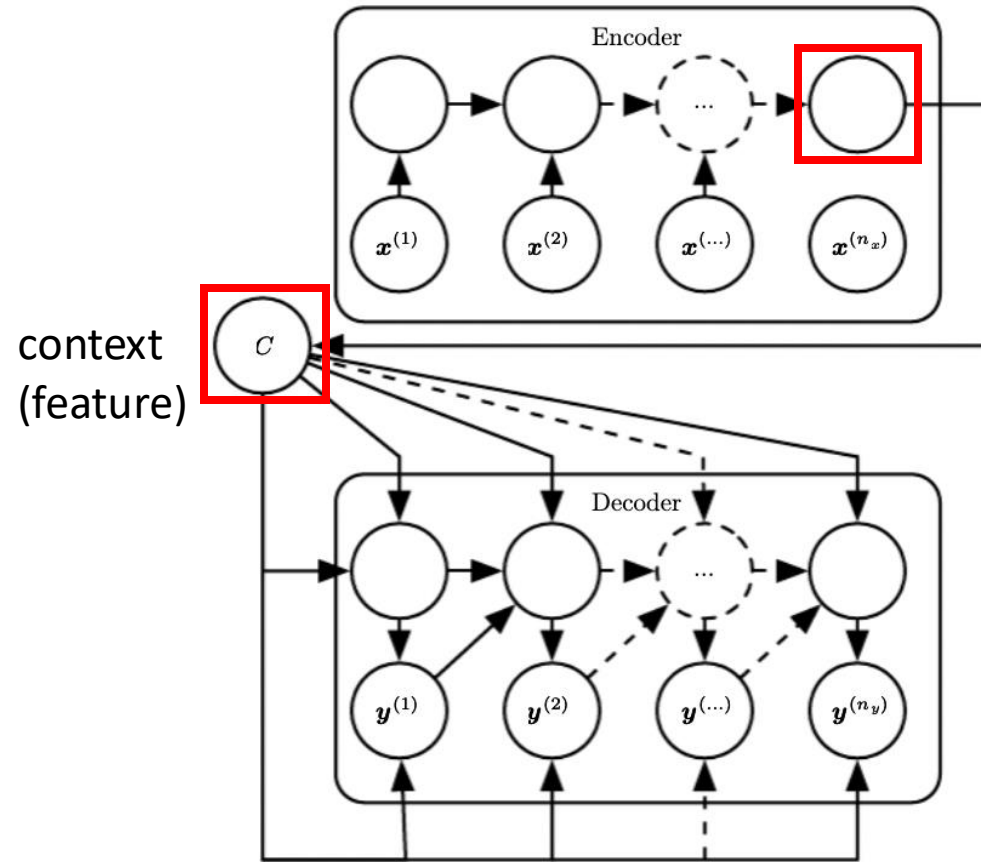
Seq2seq RNN architecture



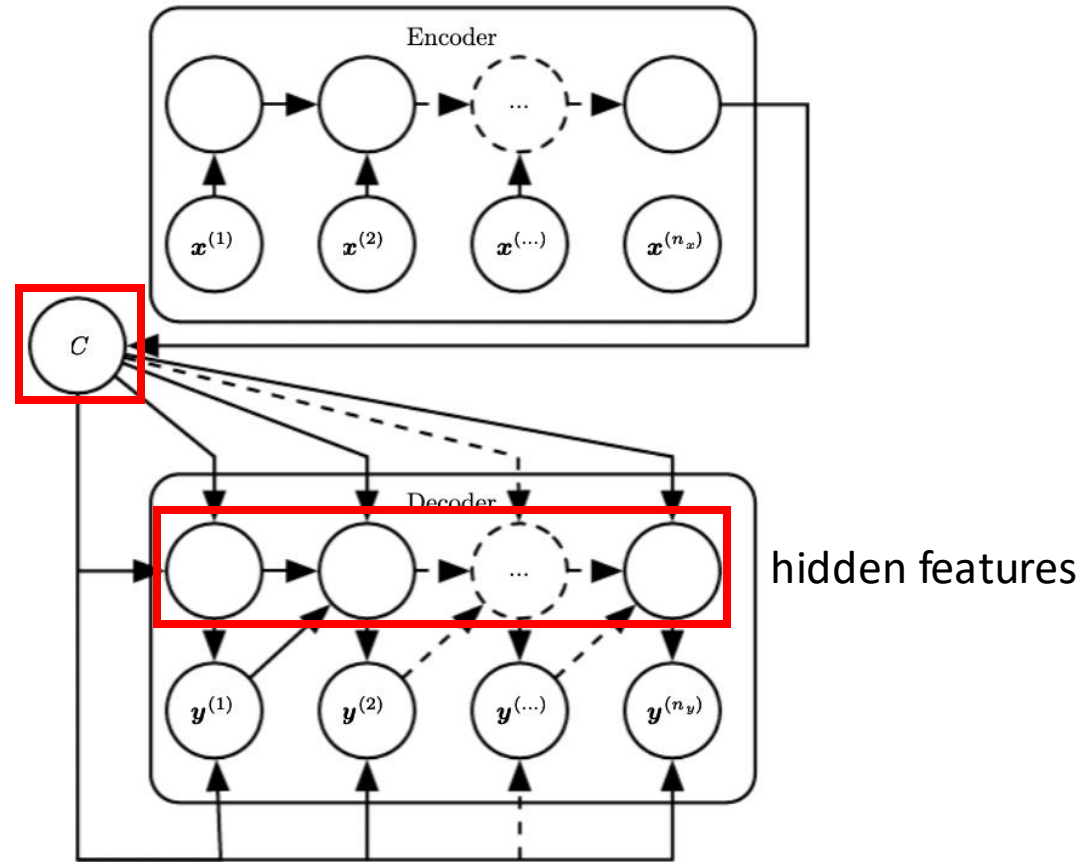
Seq2seq RNN architecture



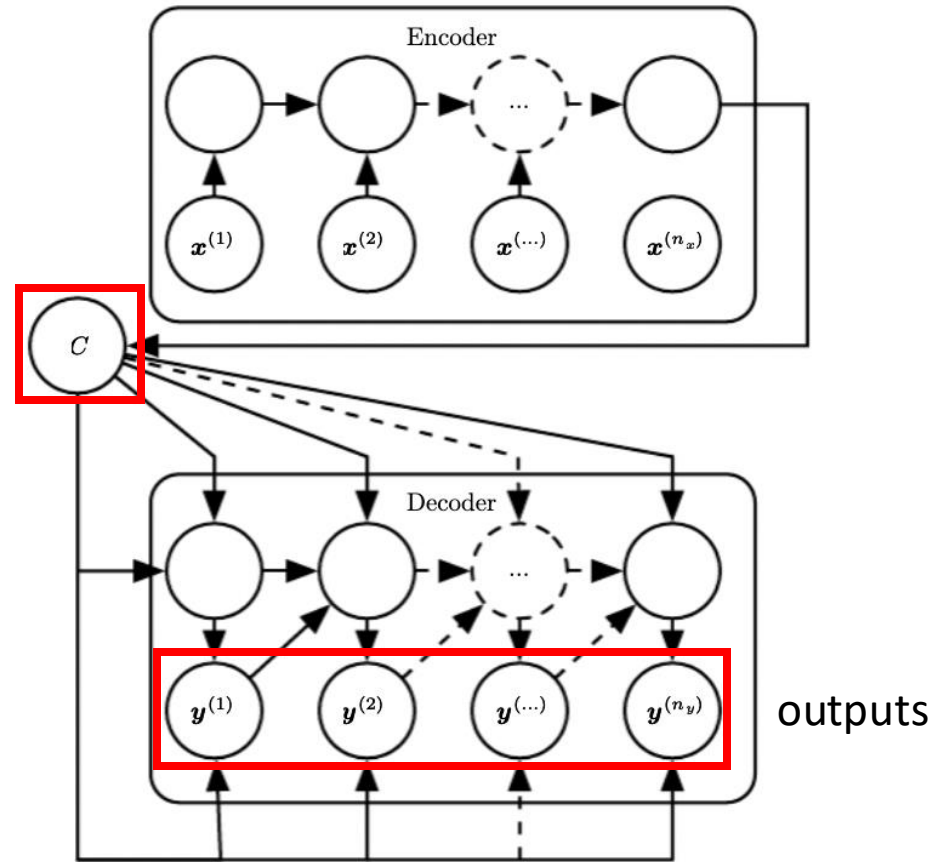
Seq2seq RNN architecture



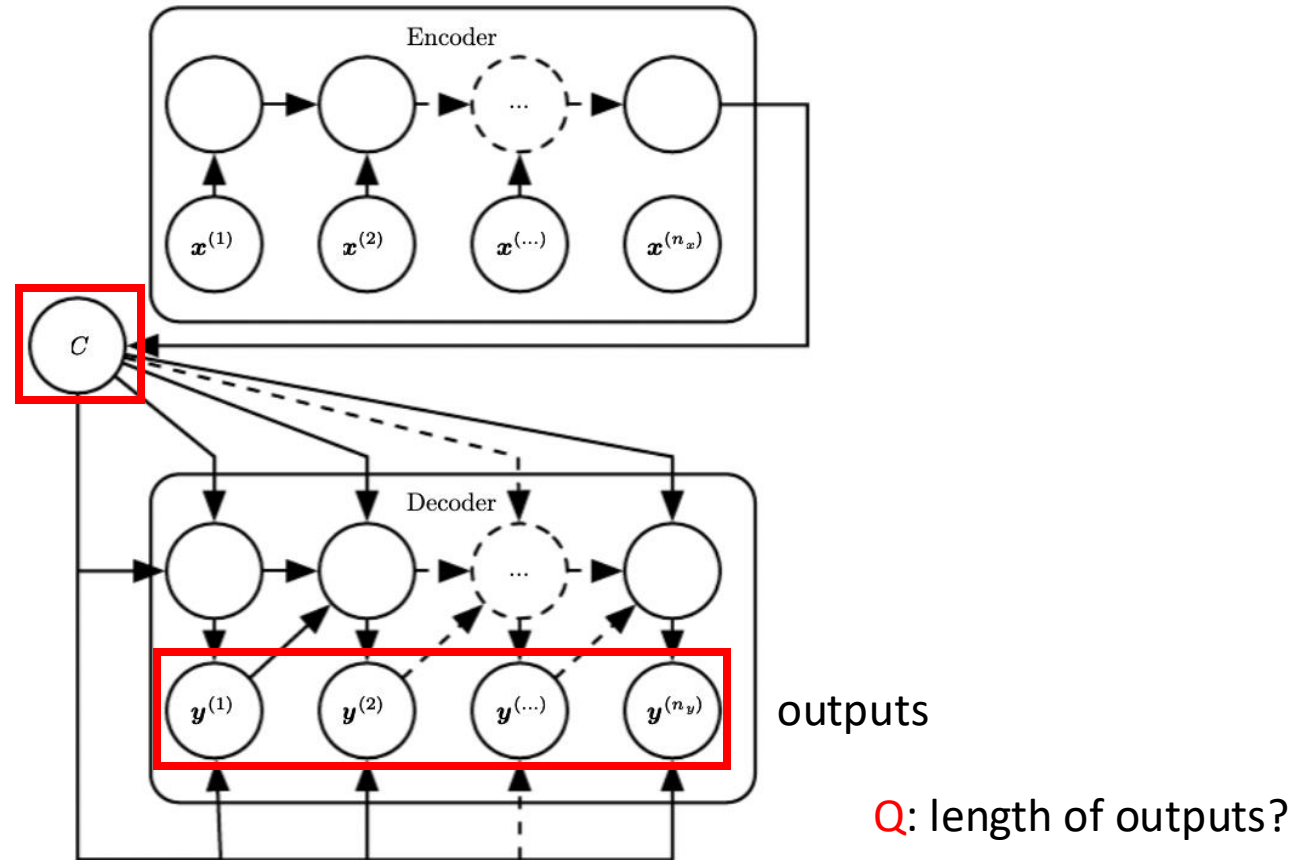
Seq2seq RNN architecture



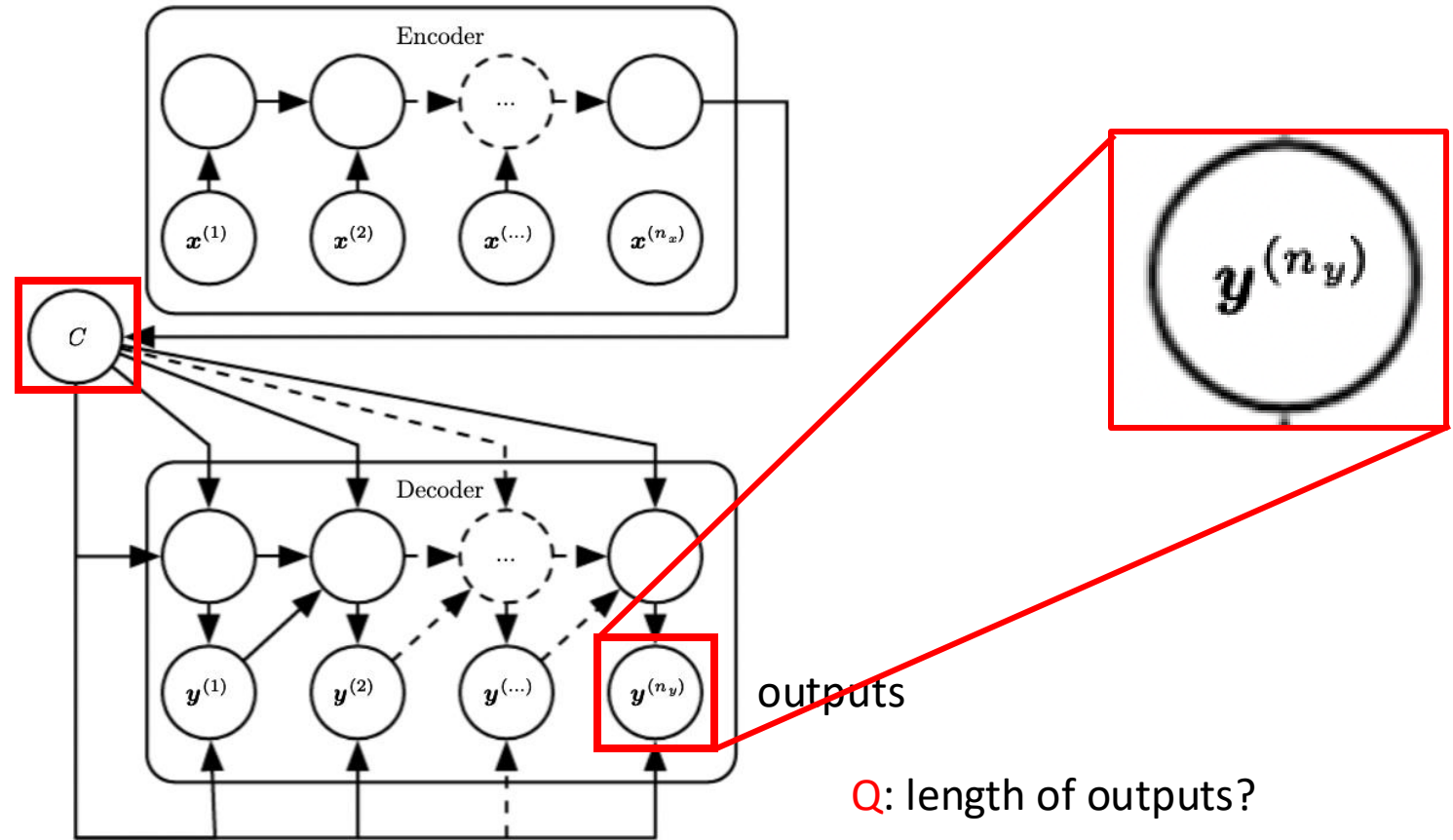
Seq2seq RNN architecture



Seq2seq RNN architecture



Seq2seq RNN architecture



Seq2seq RNN architecture

