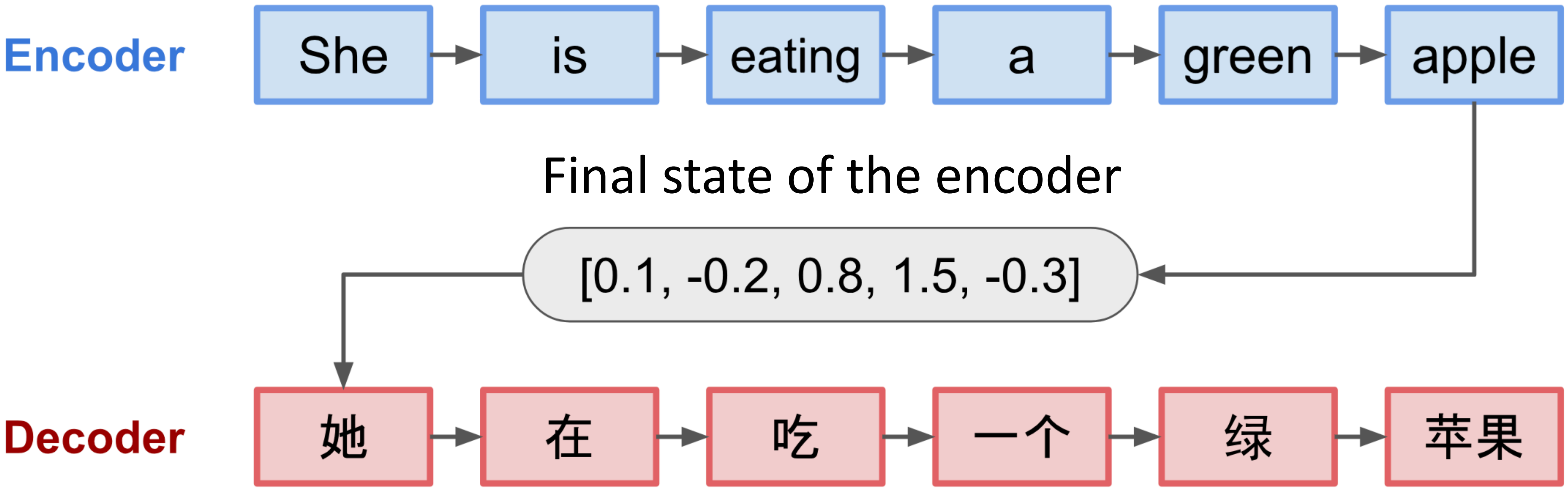
Attention

Shusen Wang

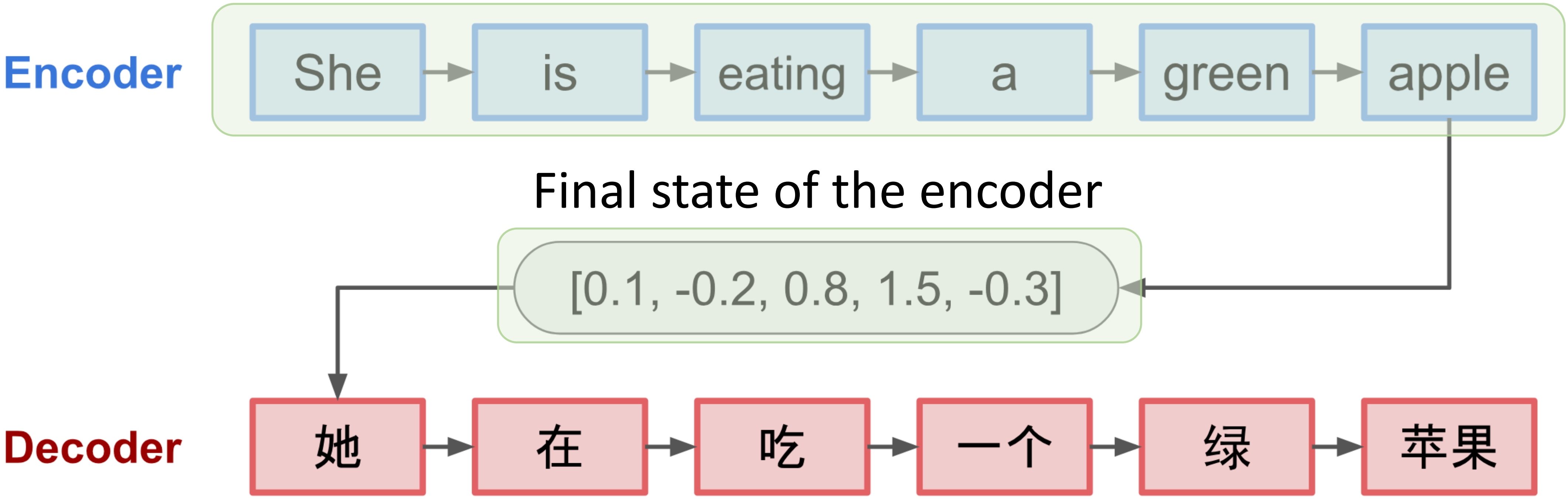
Seq2Seq Model



The figure is from blog lilianweng.github.io

Seq2Seq Model

Shortcoming: The final state is incapable of remembering a **long** sequence.



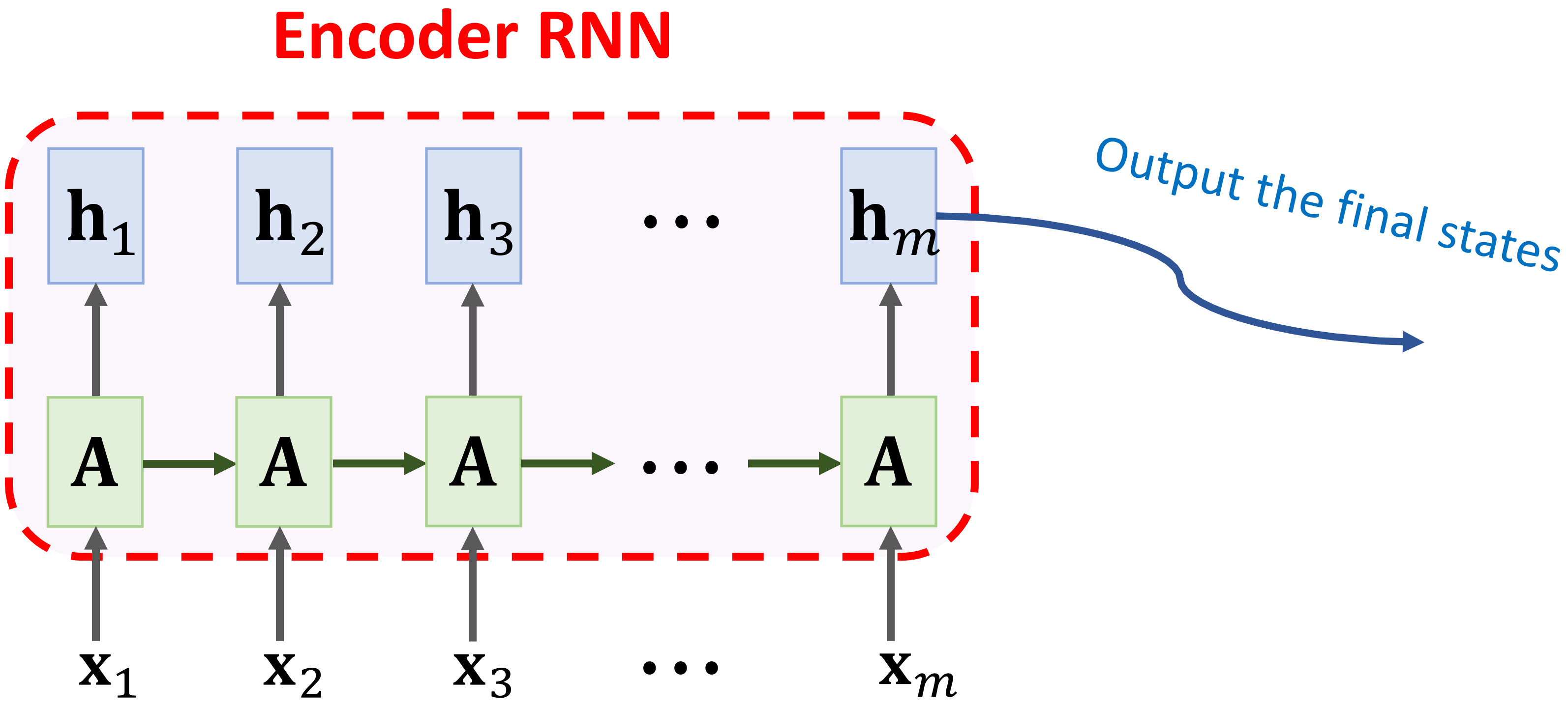
The figure is from blog lilianweng.github.io

Seq2Seq Model with Attention

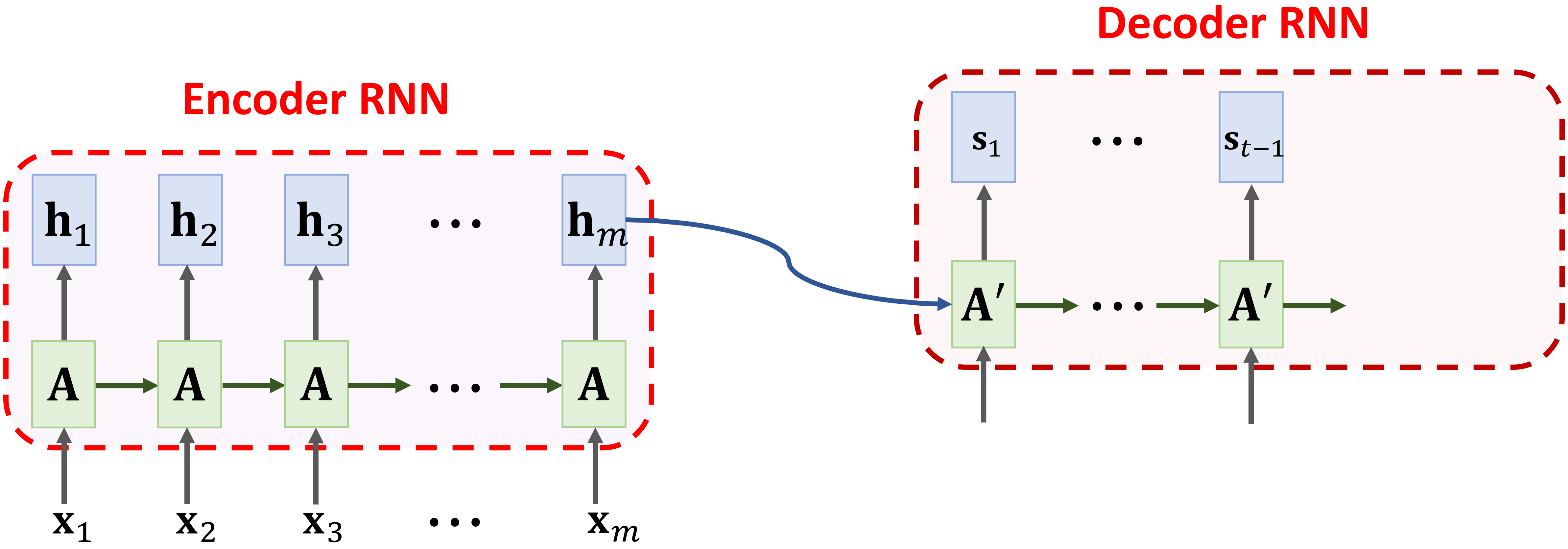
Original paper:

- Bahdanau, Cho, & Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

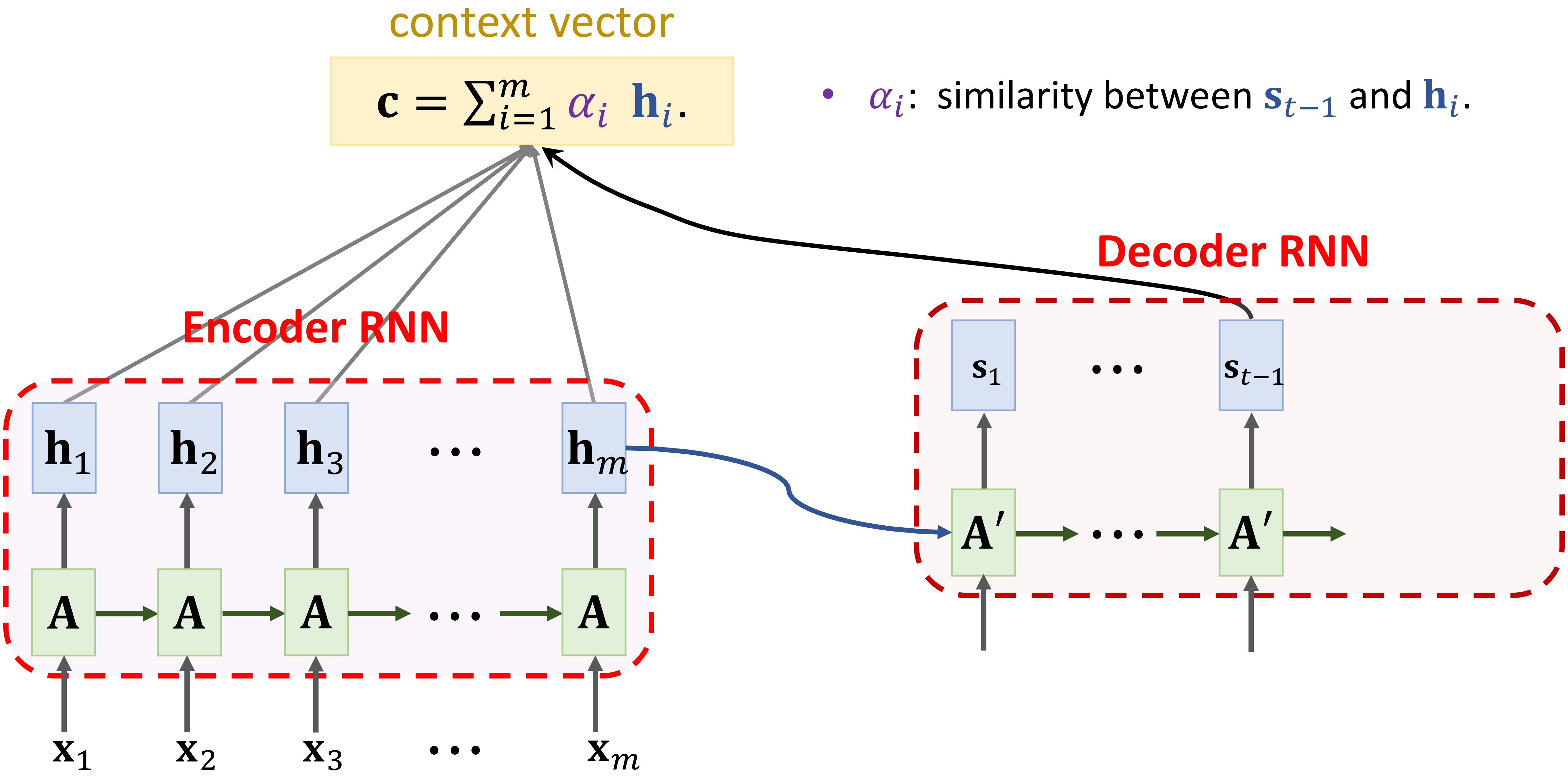
Seq2Seq Model



Seq2Seq Model



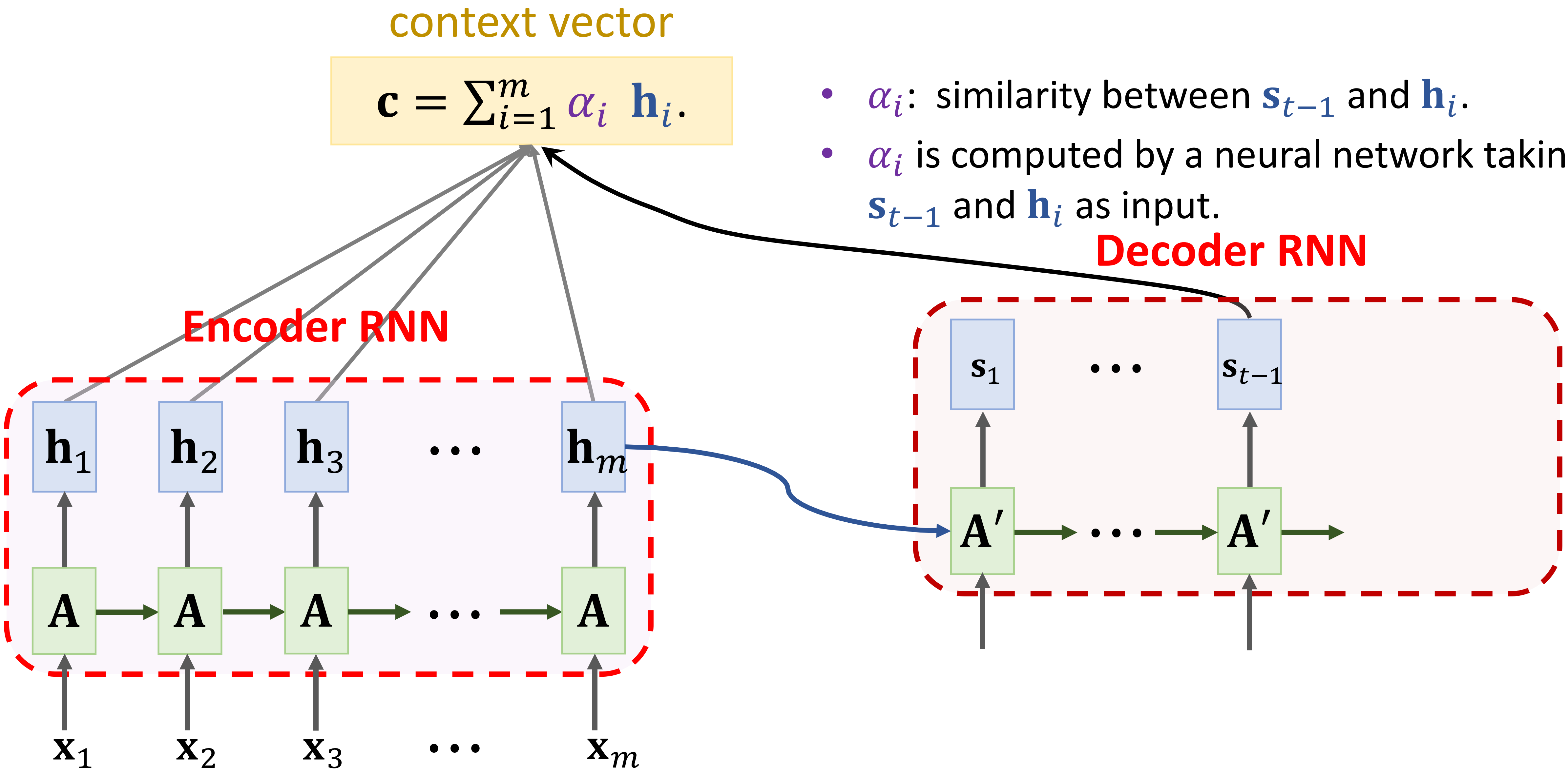
Attention



- α_i: similarity between $\mathbf{s}_{t-1}$ and $\mathbf{h}_i$.

Attention

- α_i: similarity between $\mathbf{s}_{t-1}$ and $\mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{s}_{t-1}$ and $\mathbf{h}_i$ as input.

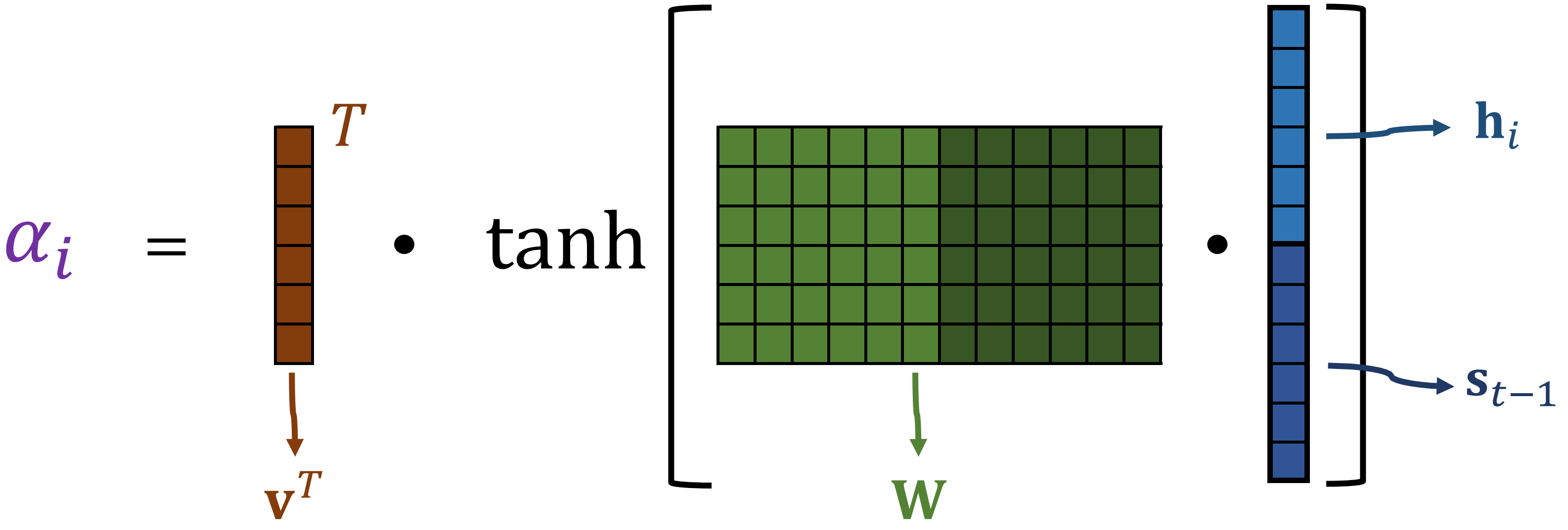


Attention

context vector

$$\mathbf{c} = \sum_{i=1}^{m} \alpha_i \ \mathbf{h}_i .$$

- α_i: similarity between $\mathbf{s}_{t-1}$ and $\mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{s}_{t-1}$ and $\mathbf{h}_i$ as input.
- $\mathbf{v}$ and $\mathbf{W}$ are trainable parameters.



Attention

context vector

$$\mathbf{c} = \sum_{i=1}^{m} \alpha_i \ \mathbf{h}_i .$$

- α_i: similarity between $\mathbf{s}_{t-1}$ and $\mathbf{h}_i$.

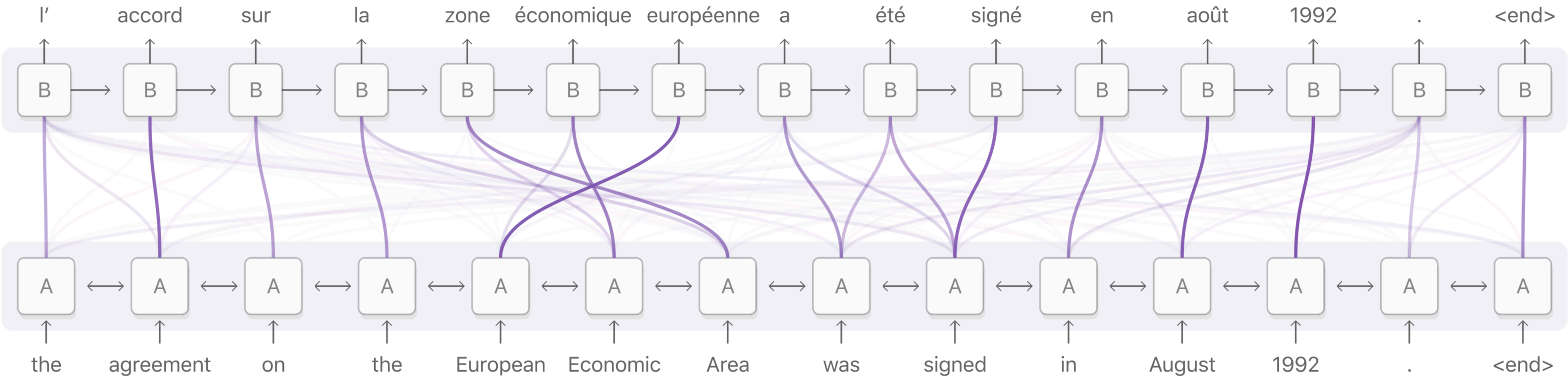


Figure is from https://distill.pub/2016/augmented-rnns/

Attention

context vector

$$\mathbf{c} = \sum_{i=1}^{m} \alpha_i \ \mathbf{h}_i .$$

- α_i: similarity between $\mathbf{s}_{t-1}$ and $\mathbf{h}_i$.

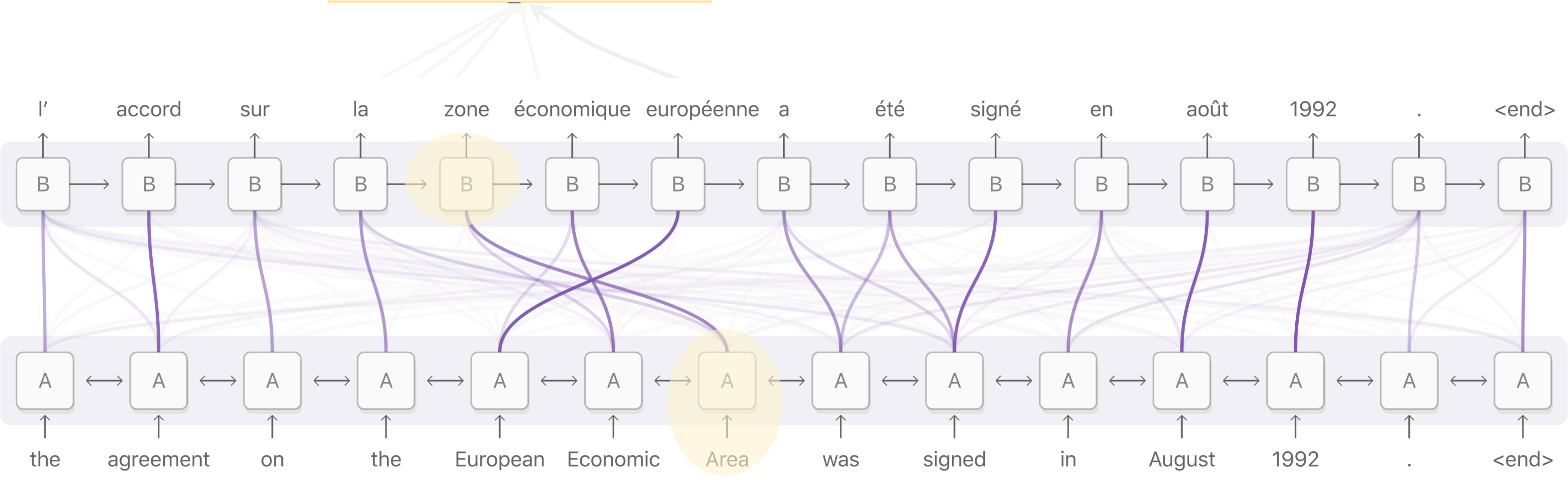
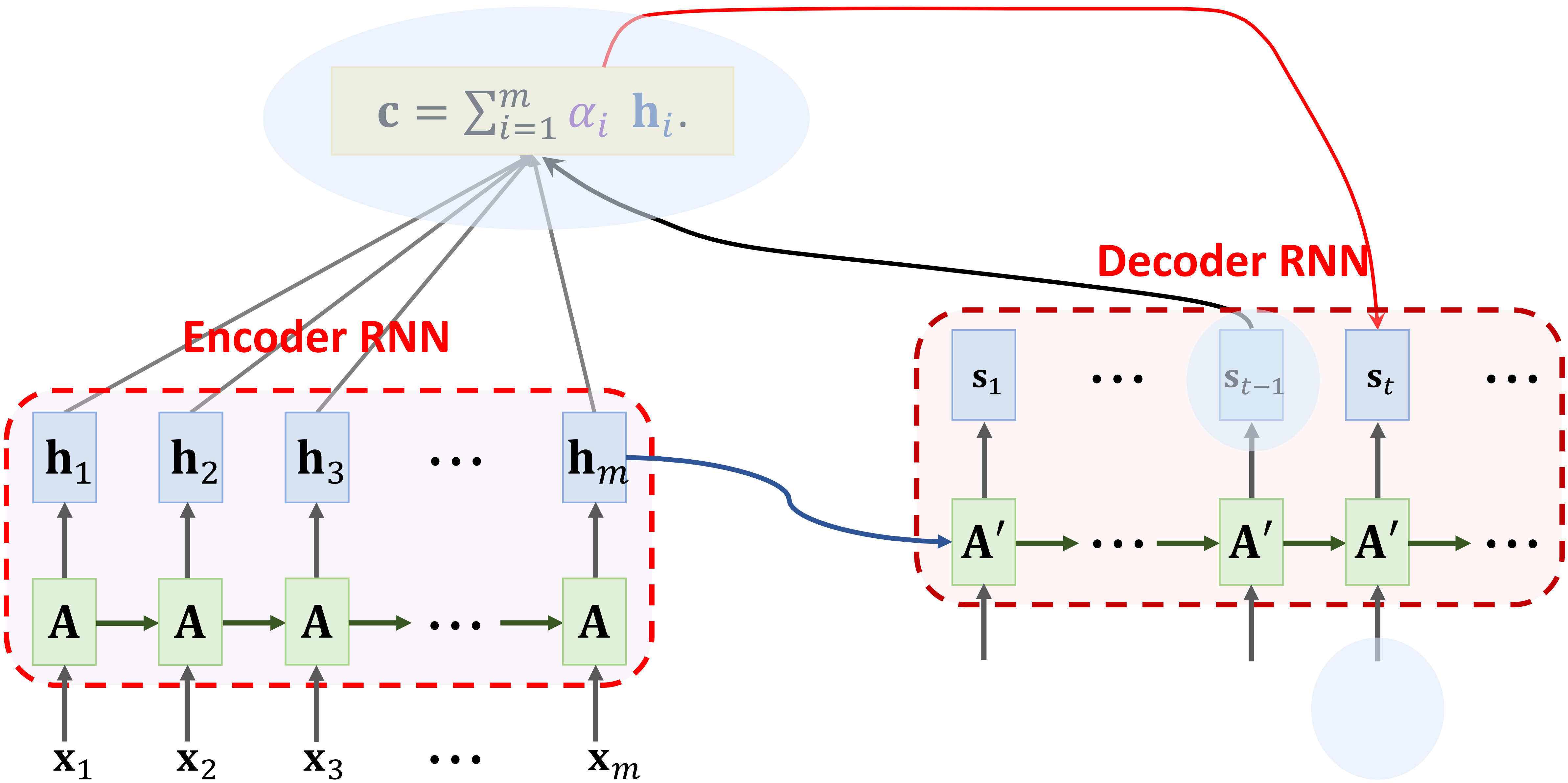


Figure is from https://distill.pub/2016/augmented-rnns/

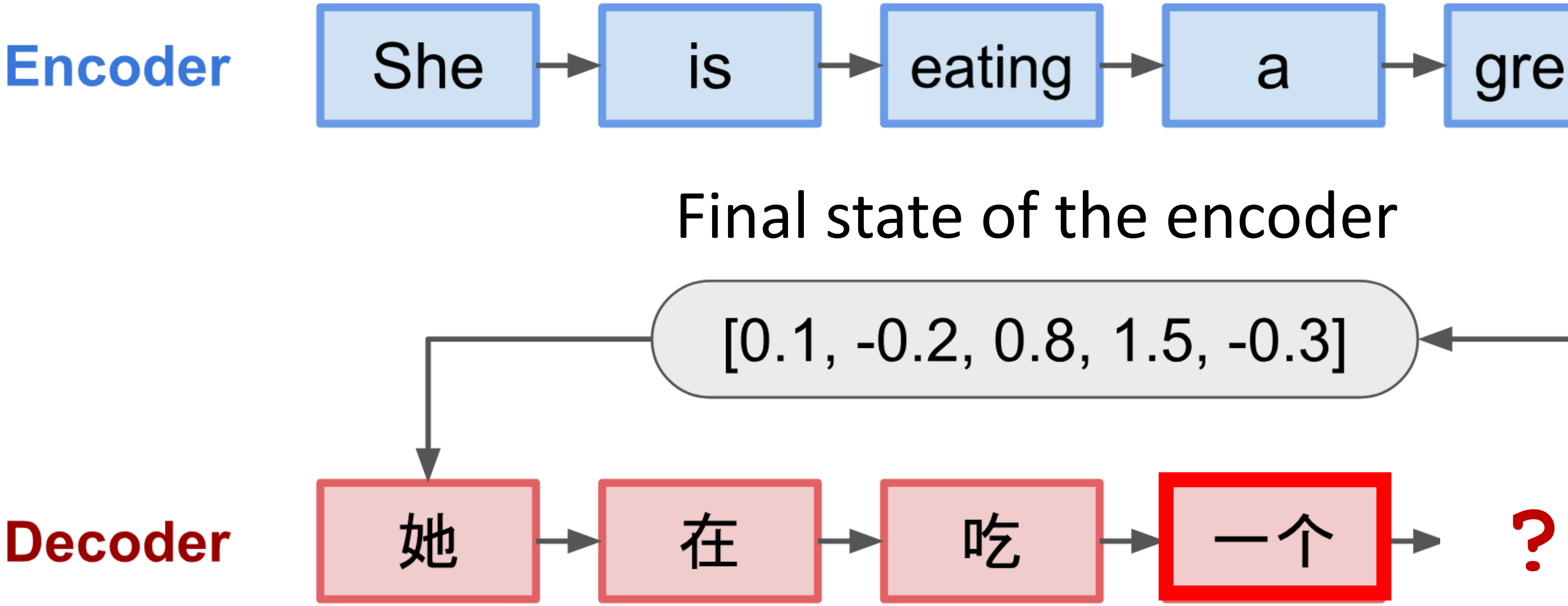
Attention



$\mathbf{c} = \sum_{i=1}^{m} \alpha_i \mathbf{h}_i.$
Encoder RNN
Decoder RNN
$\mathbf{h}_1$ $\mathbf{h}_2$ $\mathbf{h}_3$ ⋯ $\mathbf{h}_m$
A A A ⋯ A
$\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_3$ ⋯ $\mathbf{x}_m$
$\mathbf{s}_1$ ⋯ $\mathbf{s}_{t-1}$ $\mathbf{s}_t$ ⋯
A′ ⋯ A′ A′ ⋯

Summary

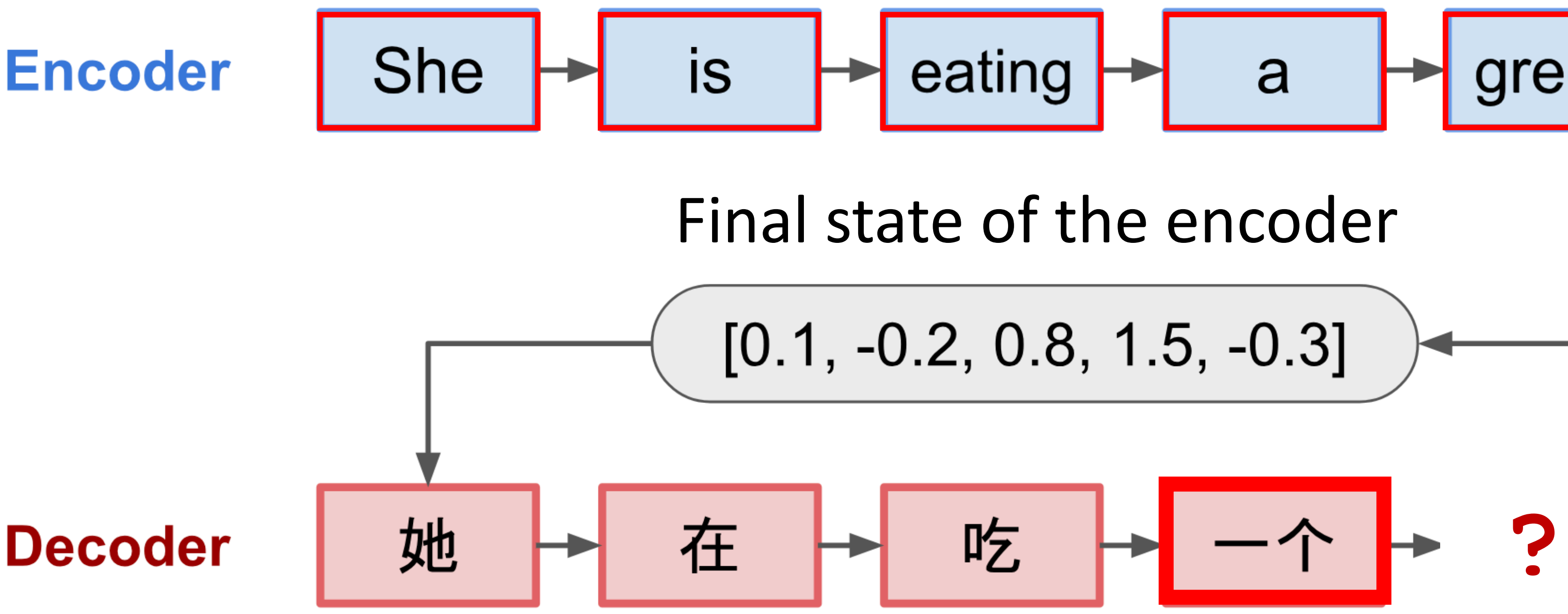
- Standard Seq2Seq model: the decoder looks at only its current state.



The figure is from blog lilianweng.github.io

Summary

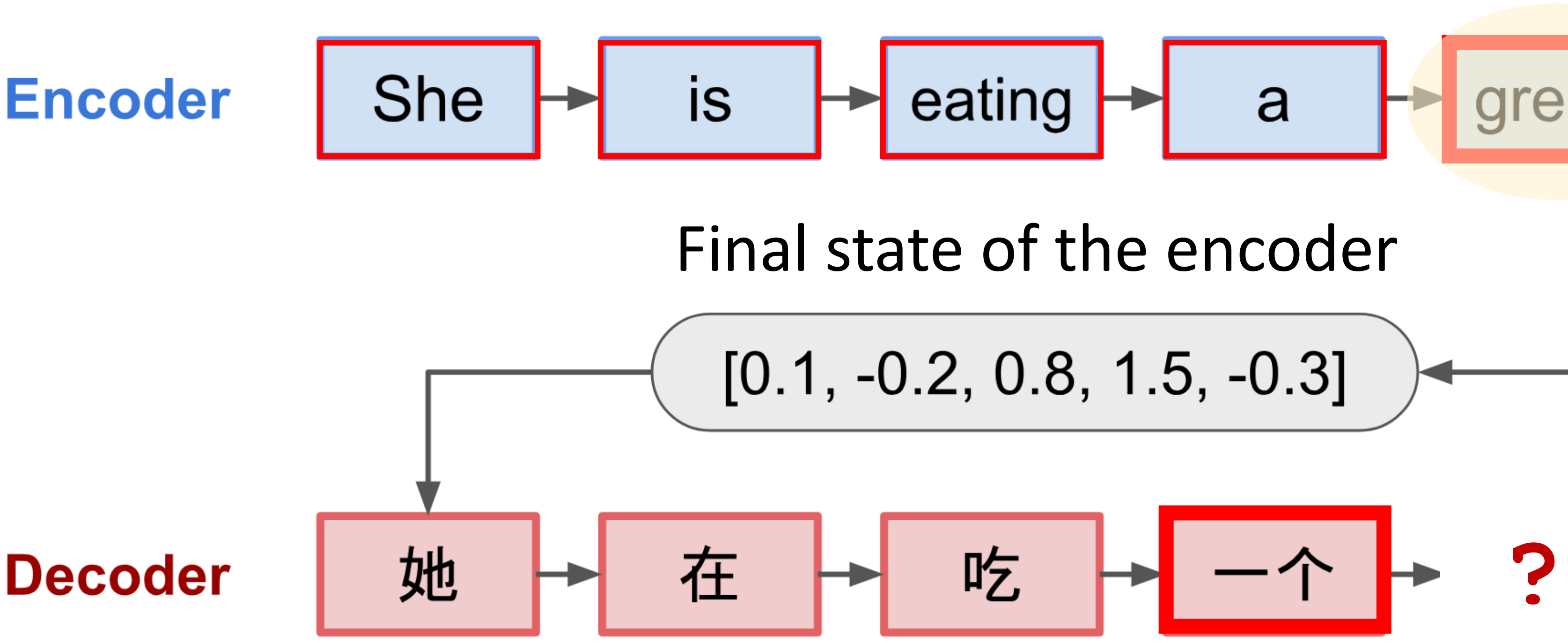
- Standard Seq2Seq model: the decoder looks at only its current state.
- Attention: decoder additionally looks at all the states of the encoder.



The figure is from blog lilianweng.github.io

Summary

- Standard Seq2Seq model: the decoder looks at only its current state.
- Attention: decoder additionally looks at all the states of the encoder.



The figure is from blog lilianweng.github.io

Summary

- Standard Seq2Seq model: the decoder looks at only its current state.
- Attention: decoder additionally looks at all the states of the encoder.

- Downside: higher time complexity.
  - l_1: source sequence length
  - l_2: target sequence length
  - Standard Seq2Seq: $O(l_1 + l_2)$ time complexity
  - Seq2Seq + attention: $O(l_1 \, l_2)$ time complexity

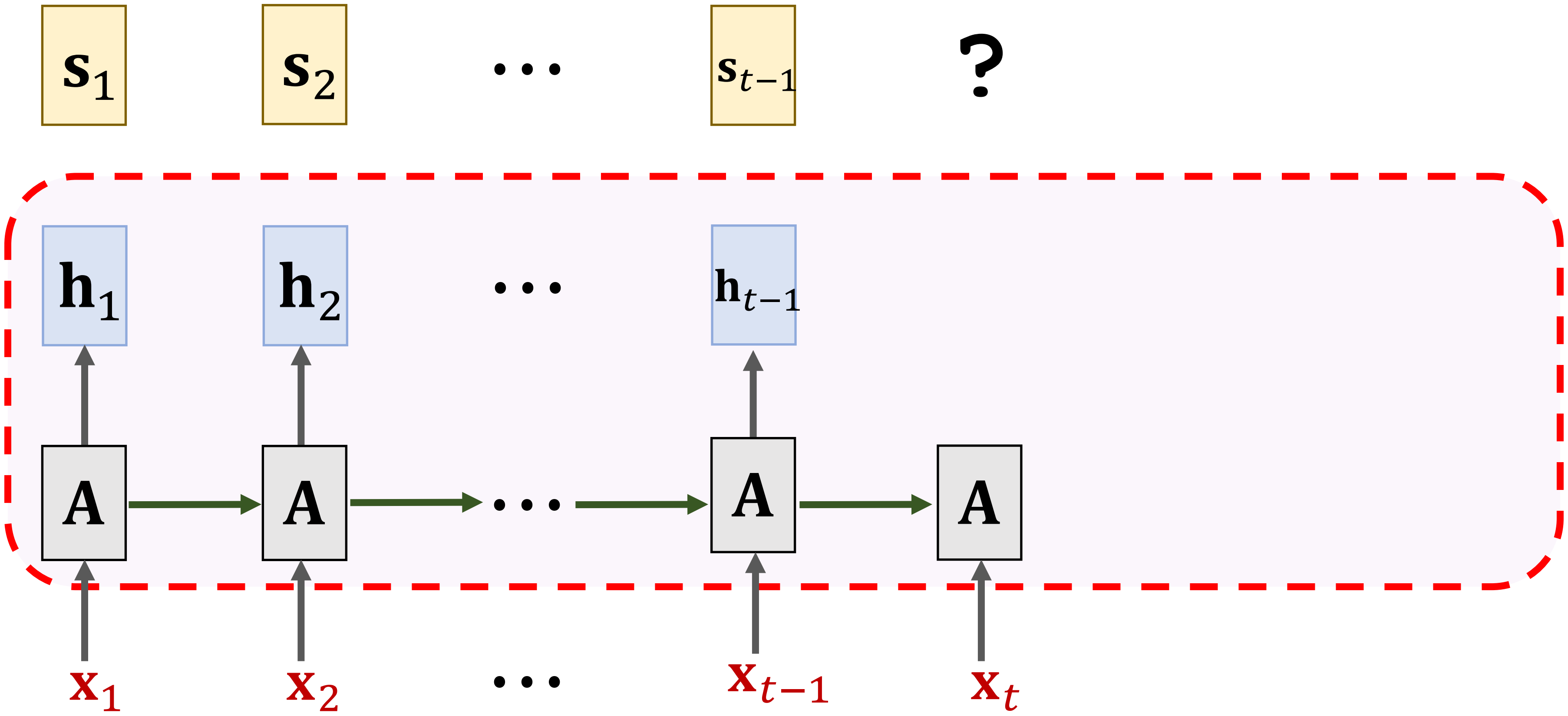
Self-Attention: Attention beyond Seq2Seq Models

Original paper:

- Cheng, Dong, & Lapata. Long Short-Term Memory-Networks for Machine Reading. In *EMNLP*, 2016.

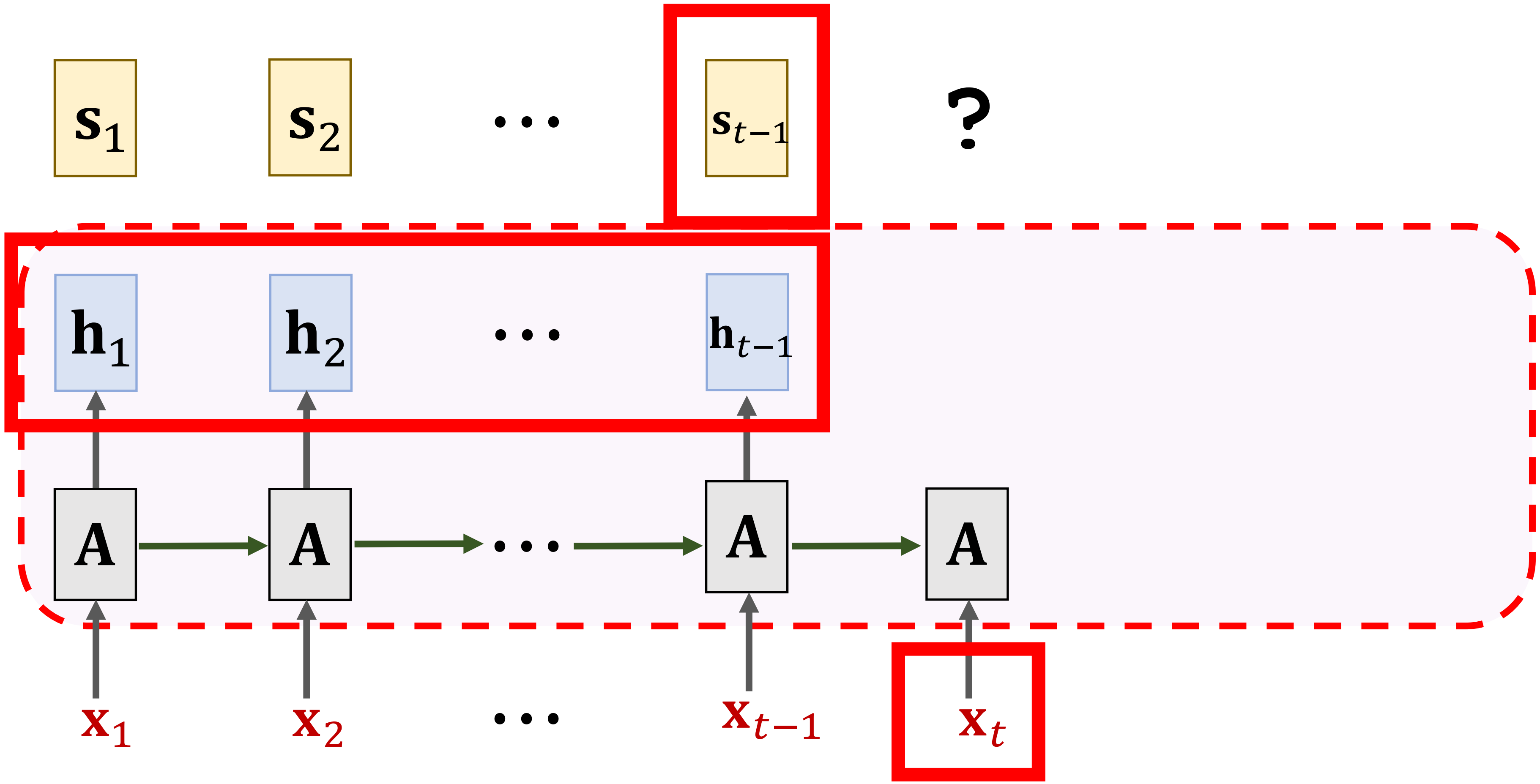
SimpleRNN with Self-Attention

- $\mathbf{s}_t = \sum_{i=1}^{t-1} \alpha_i \, \mathbf{h}_i$.



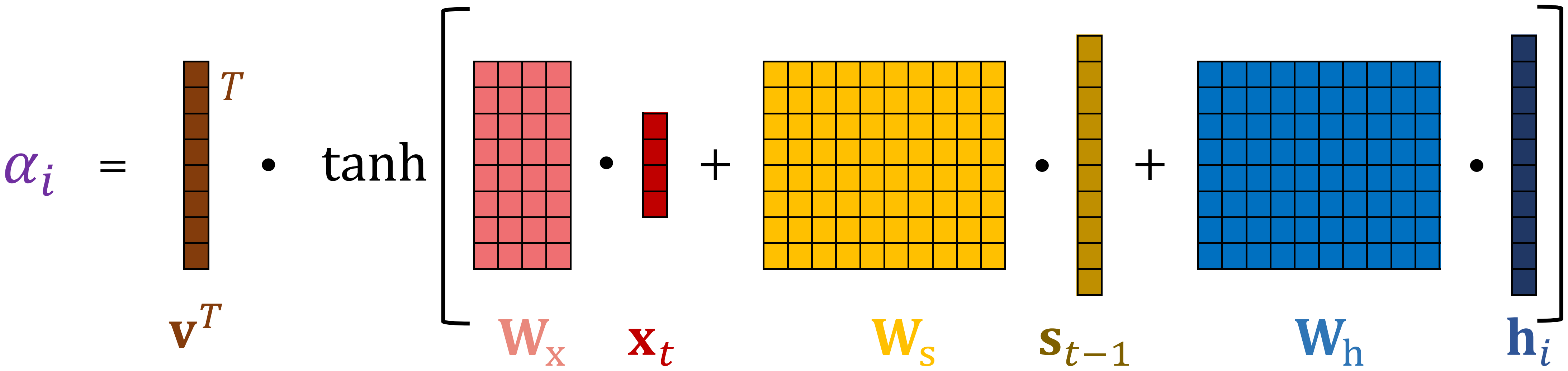
SimpleRNN with Self-Attention

- $\mathbf{s}_t = \sum_{i=1}^{t-1} \alpha_i \, \mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{x}_t$, $\mathbf{s}_{t-1}$, and $\mathbf{h}_i$ as inputs.



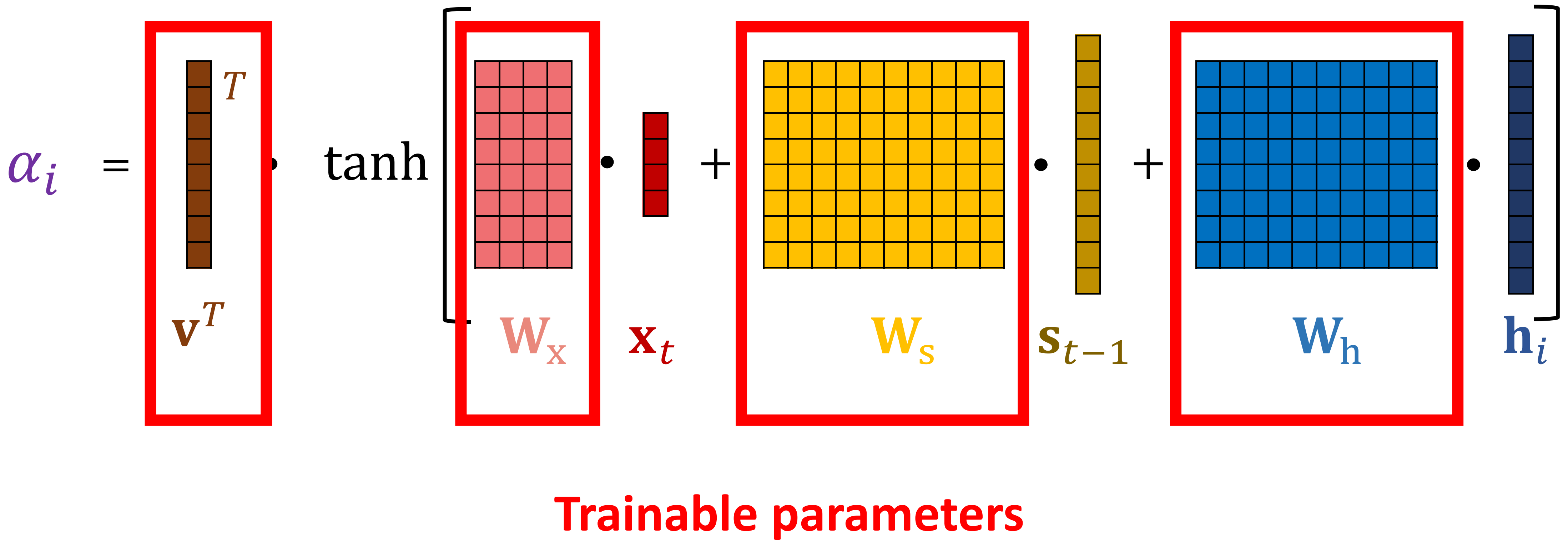
SimpleRNN with Self-Attention

- $\mathbf{s}_t = \sum_{i=1}^{t-1} \alpha_i \, \mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{x}_t$, $\mathbf{s}_{t-1}$, and $\mathbf{h}_i$ as inputs.



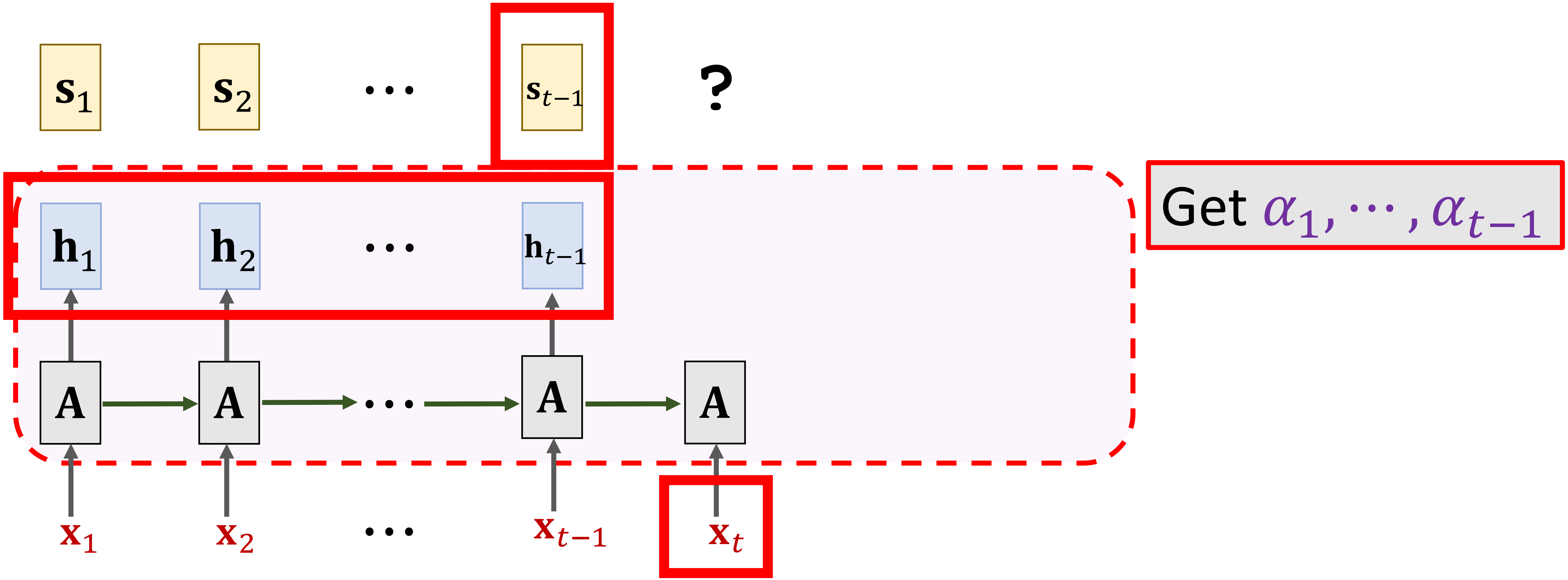
SimpleRNN with Self-Attention

- $\mathbf{s}_t = \sum_{i=1}^{t-1} \alpha_i \, \mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{x}_t$, $\mathbf{s}_{t-1}$, and $\mathbf{h}_i$ as inputs.



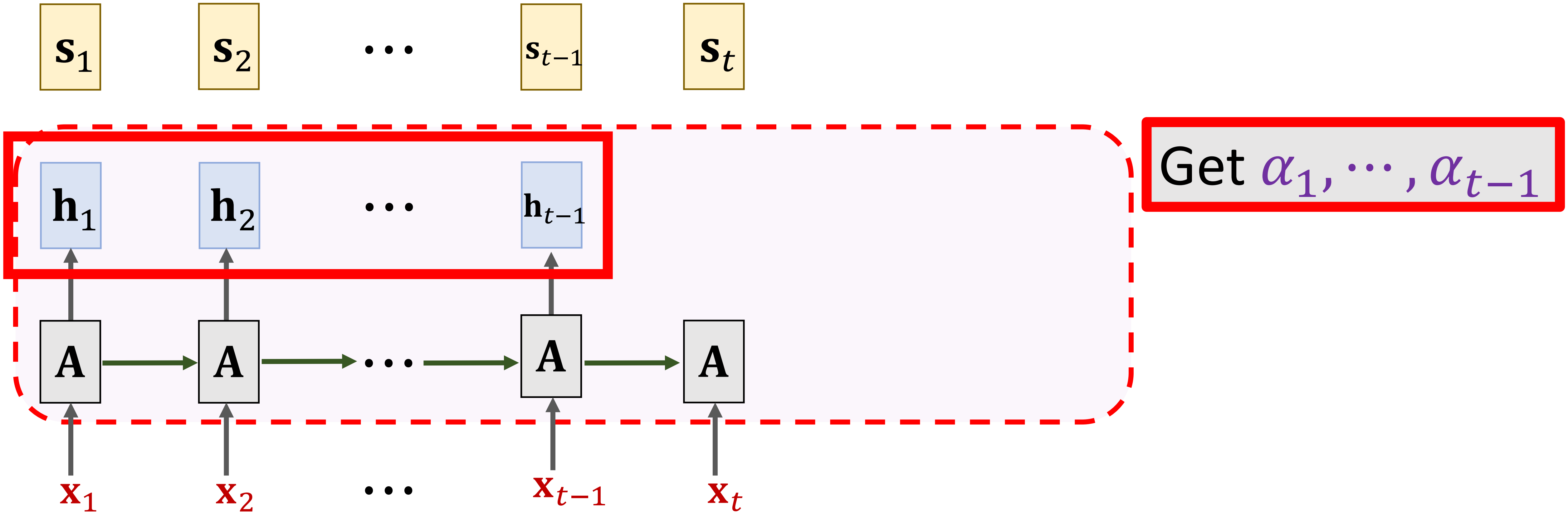
SimpleRNN with Self-Attention

- $\mathbf{s}_t = \sum_{i=1}^{t-1} \alpha_i \, \mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{x}_t$, $\mathbf{s}_{t-1}$, and $\mathbf{h}_i$ as inputs.



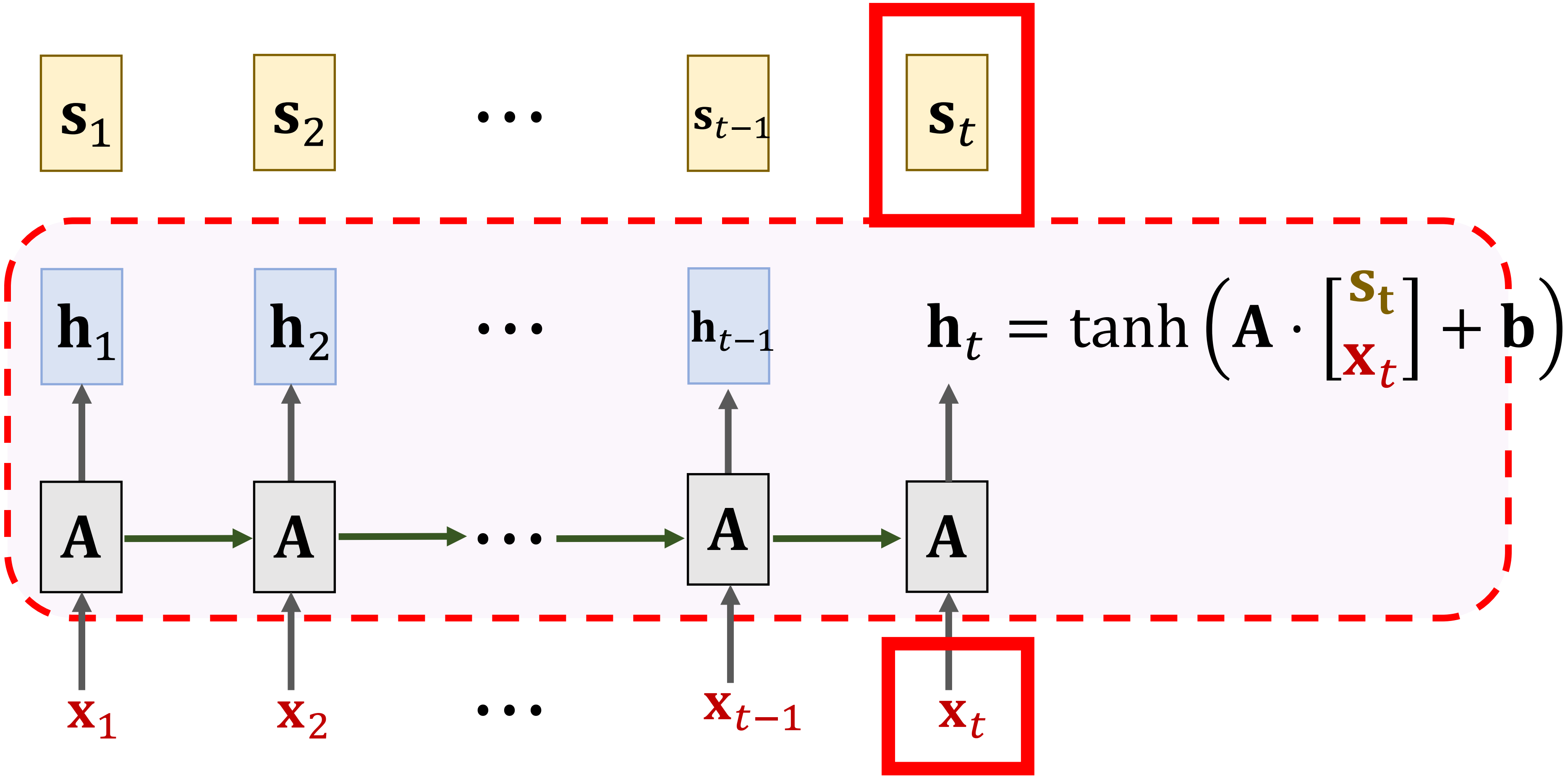
SimpleRNN with Self-Attention

- $\mathbf{s}_t = \sum_{i=1}^{t-1} \alpha_i \, \mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{x}_t$, $\mathbf{s}_{t-1}$, and $\mathbf{h}_i$ as inputs.



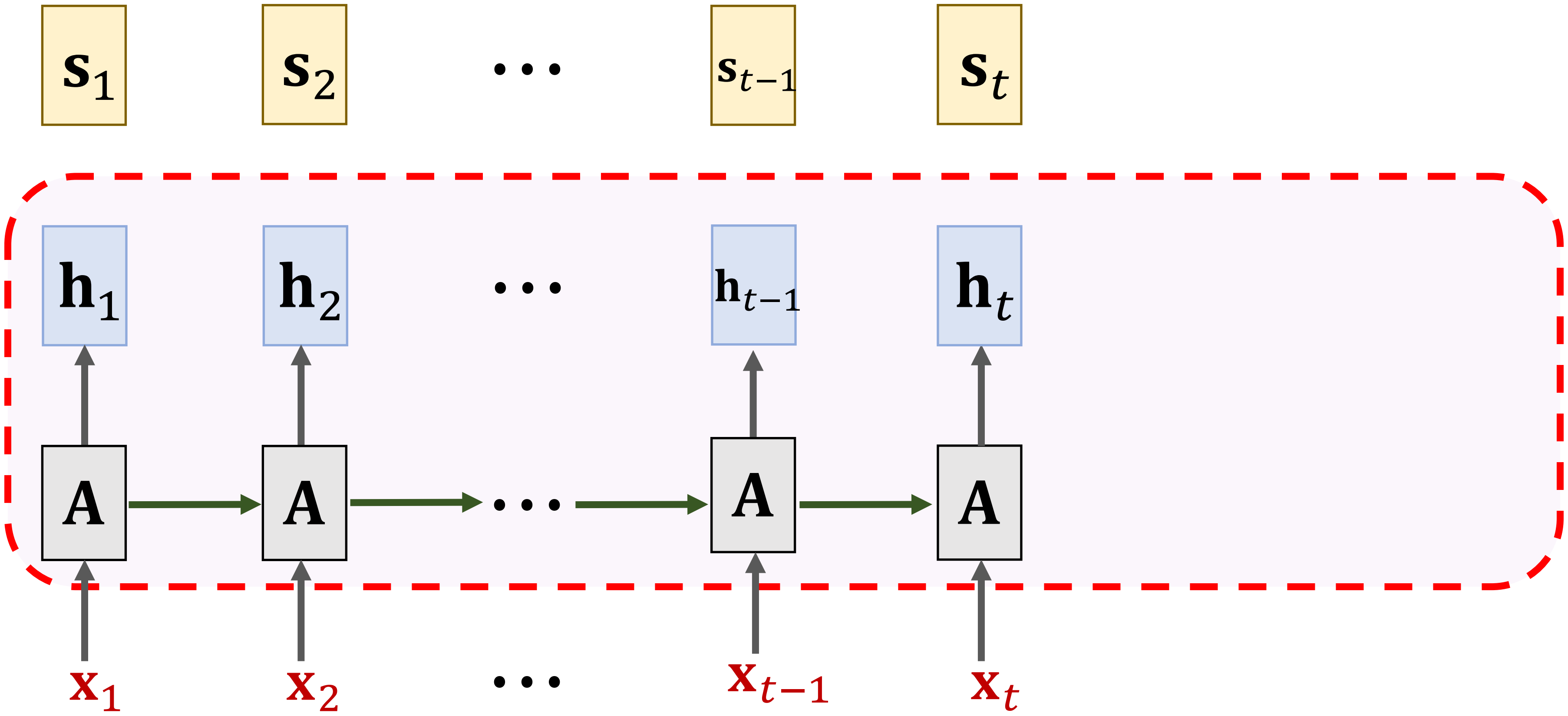
SimpleRNN with Self-Attention

- $\mathbf{s}_t = \sum_{i=1}^{t-1} \alpha_i \, \mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{x}_t$, $\mathbf{s}_{t-1}$, and $\mathbf{h}_i$ as inputs.



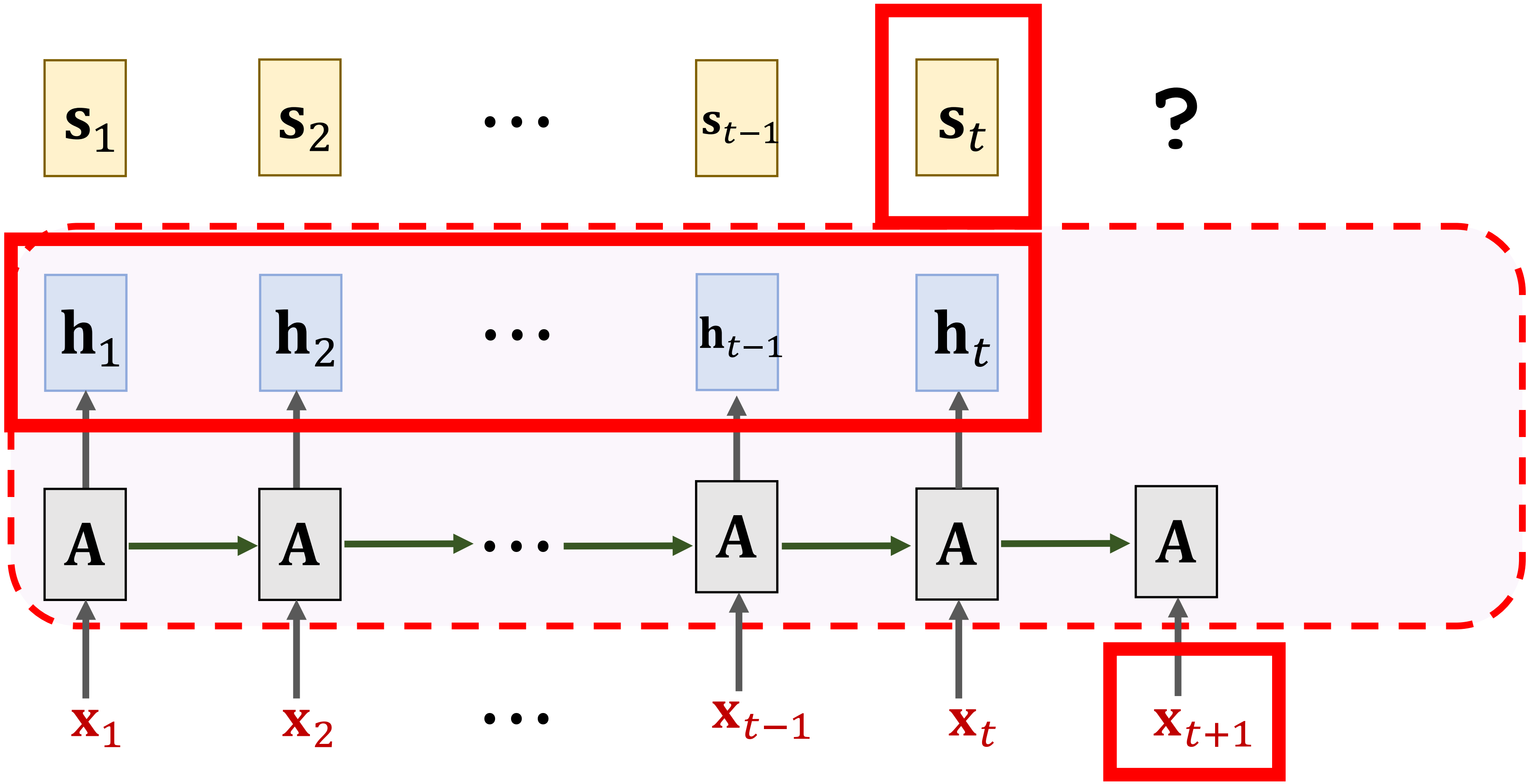
SimpleRNN with Self-Attention

- $\mathbf{s}_t = \sum_{i=1}^{t-1} \alpha_i \, \mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{x}_t$, $\mathbf{s}_{t-1}$, and $\mathbf{h}_i$ as inputs.



SimpleRNN with Self-Attention

- $\mathbf{s}_{t+1} = \sum_{i=1}^{t} \alpha_i \, \mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{x}_{t+1}$, $\mathbf{s}_t$, and $\mathbf{h}_i$ as inputs.



SimpleRNN with Self-Attention

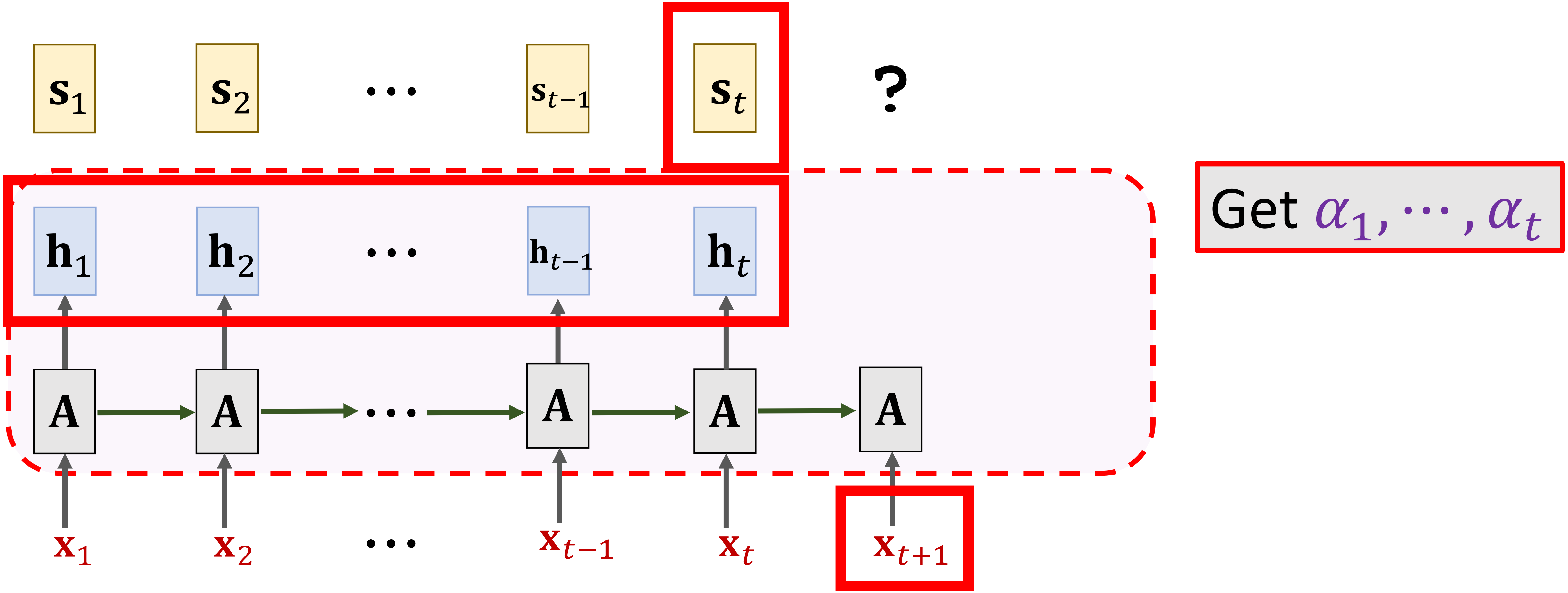
- $\mathbf{s}_{t+1} = \sum_{i=1}^{t} \alpha_i \, \mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{x}_{t+1}$, $\mathbf{s}_t$, and $\mathbf{h}_i$ as inputs.

- Not the same $\{\alpha_i\}$ computed previously.
- Because $\mathbf{x}_t \rightarrow \mathbf{x}_{t+1}$ and $\mathbf{s}_{t-1} \rightarrow \mathbf{s}_t$.

$$\alpha_i = \mathbf{v}^T \cdot \tanh \left[\mathbf{W}_\mathrm{x} \cdot \mathbf{x}_{t+1} + \mathbf{W}_\mathrm{s} \cdot \mathbf{s}_t + \mathbf{W}_\mathrm{h} \cdot \mathbf{h}_i \right]$$

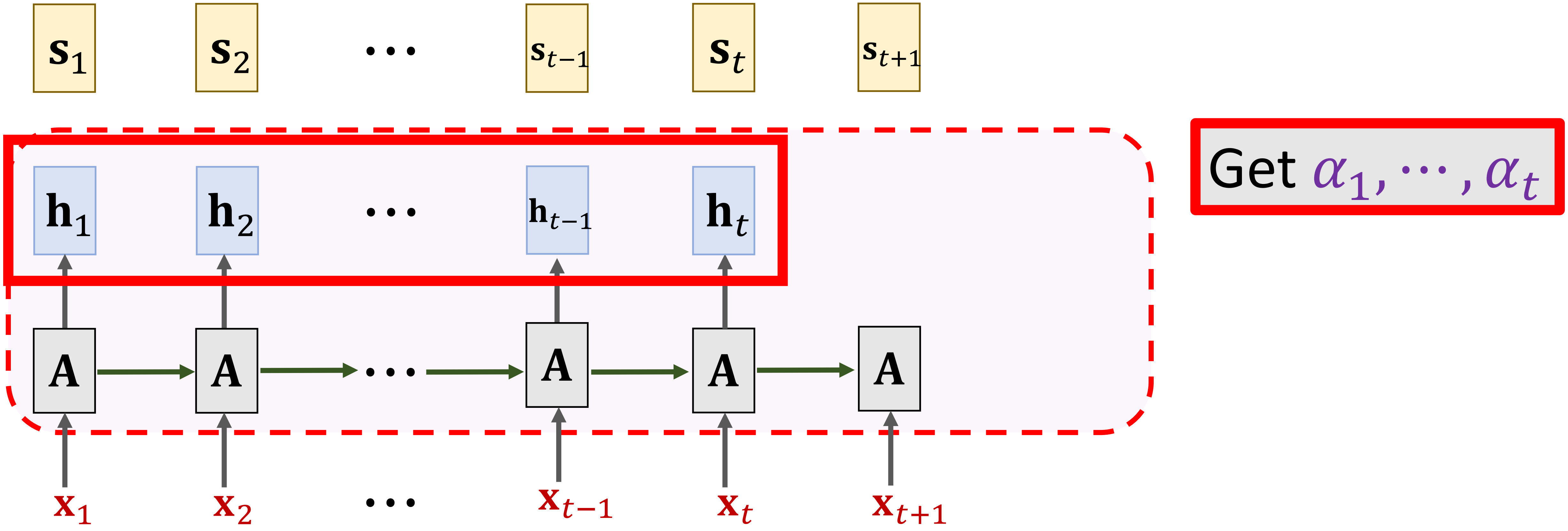
SimpleRNN with Self-Attention

- $\mathbf{s}_{t+1} = \sum_{i=1}^{t} \alpha_i \, \mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{x}_{t+1}$, $\mathbf{s}_t$, and $\mathbf{h}_i$ as inputs.



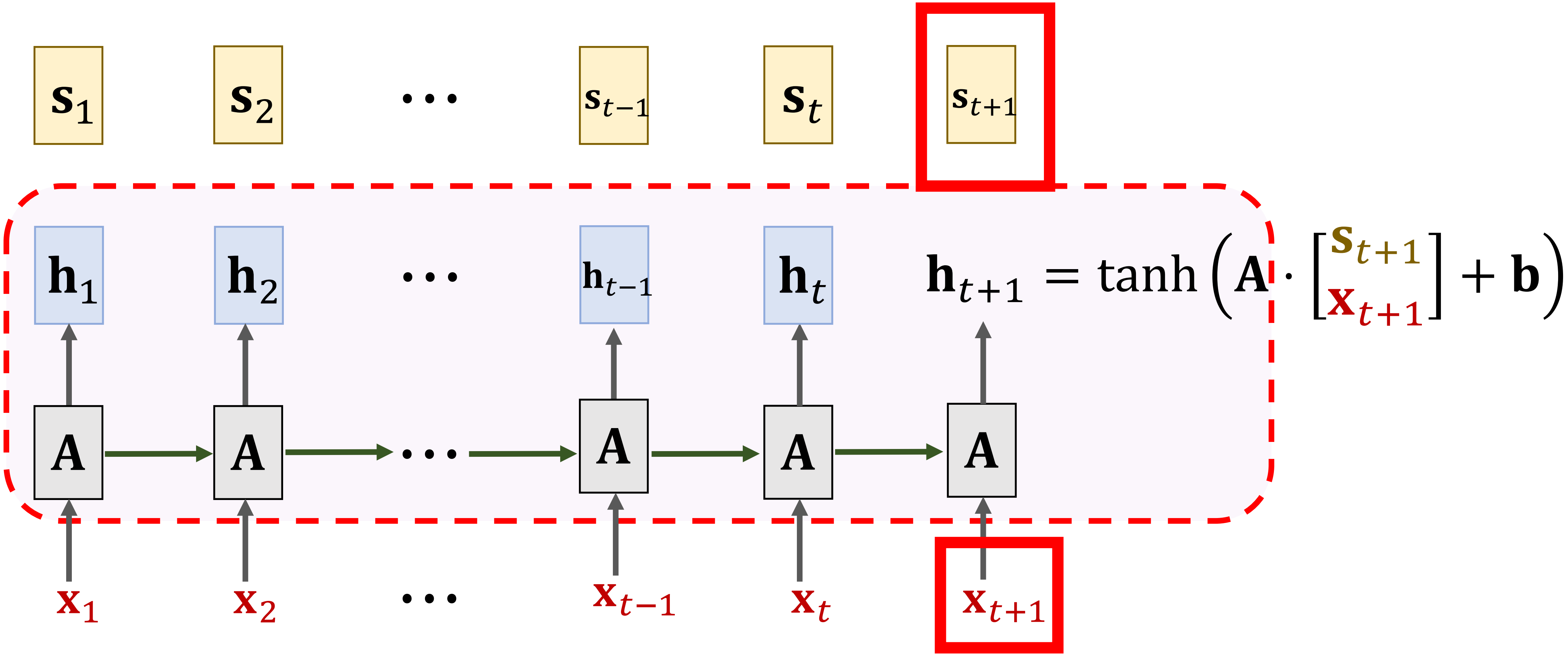
SimpleRNN with Self-Attention

- $\mathbf{s}_{t+1} = \sum_{i=1}^{t} \alpha_i \, \mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{x}_{t+1}$, $\mathbf{s}_t$, and $\mathbf{h}_i$ as inputs.



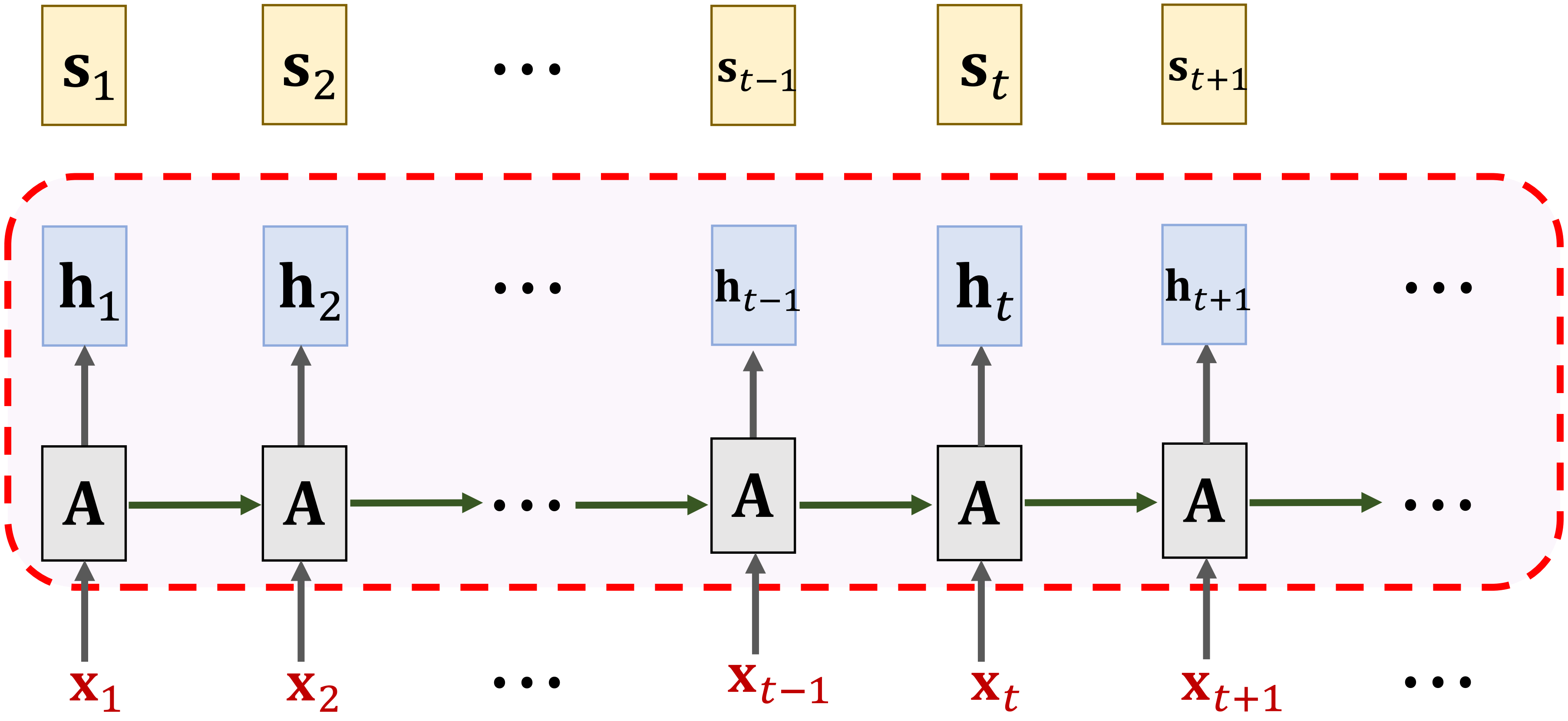
SimpleRNN with Self-Attention

- $\mathbf{s}_{t+1} = \sum_{i=1}^{t} \alpha_i \, \mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{x}_{t+1}$, $\mathbf{s}_t$, and $\mathbf{h}_i$ as inputs.

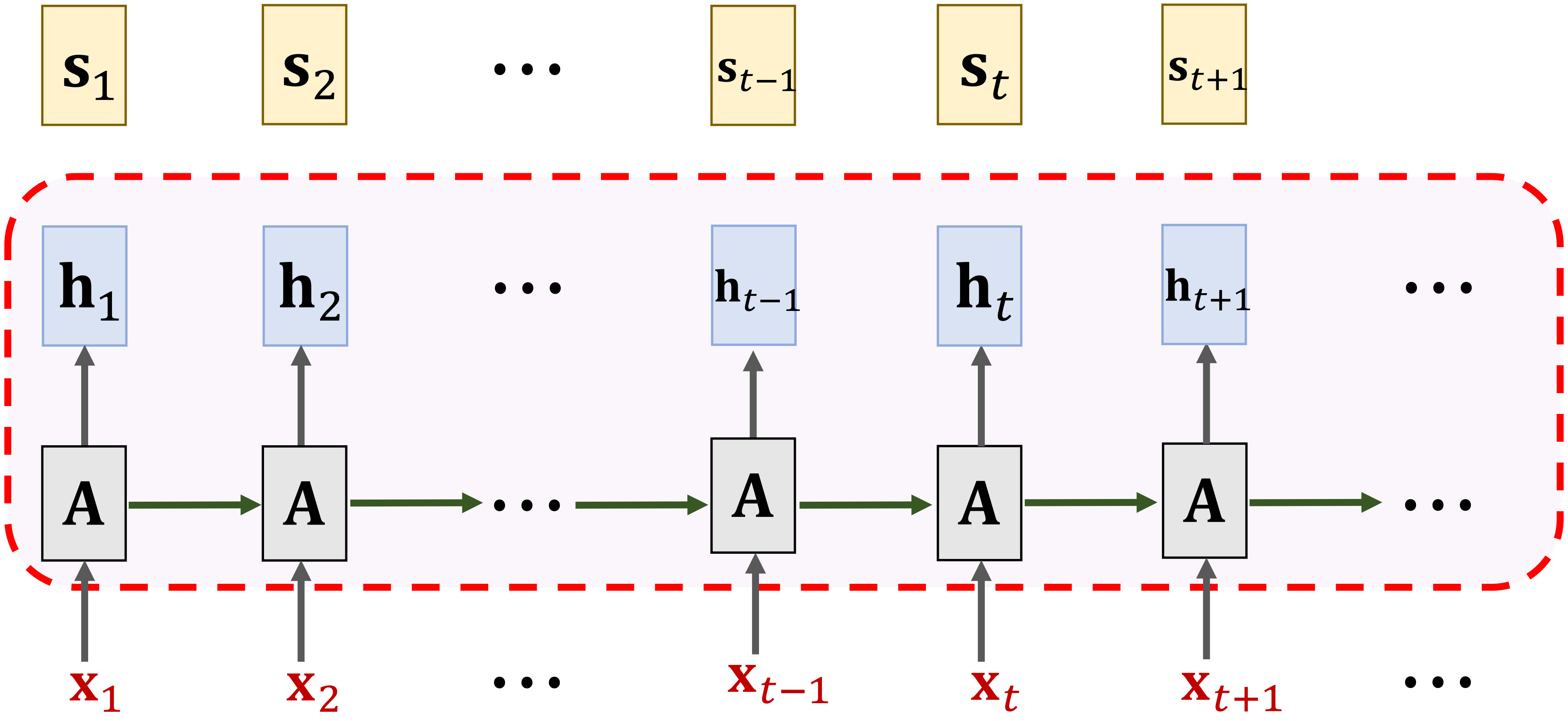


SimpleRNN with Self-Attention

- $\mathbf{s}_{t+1} = \sum_{i=1}^{t} \alpha_i \, \mathbf{h}_i$.
- α_i is computed by a neural network taking $\mathbf{x}_{t+1}$, $\mathbf{s}_t$, and $\mathbf{h}_i$ as inputs.

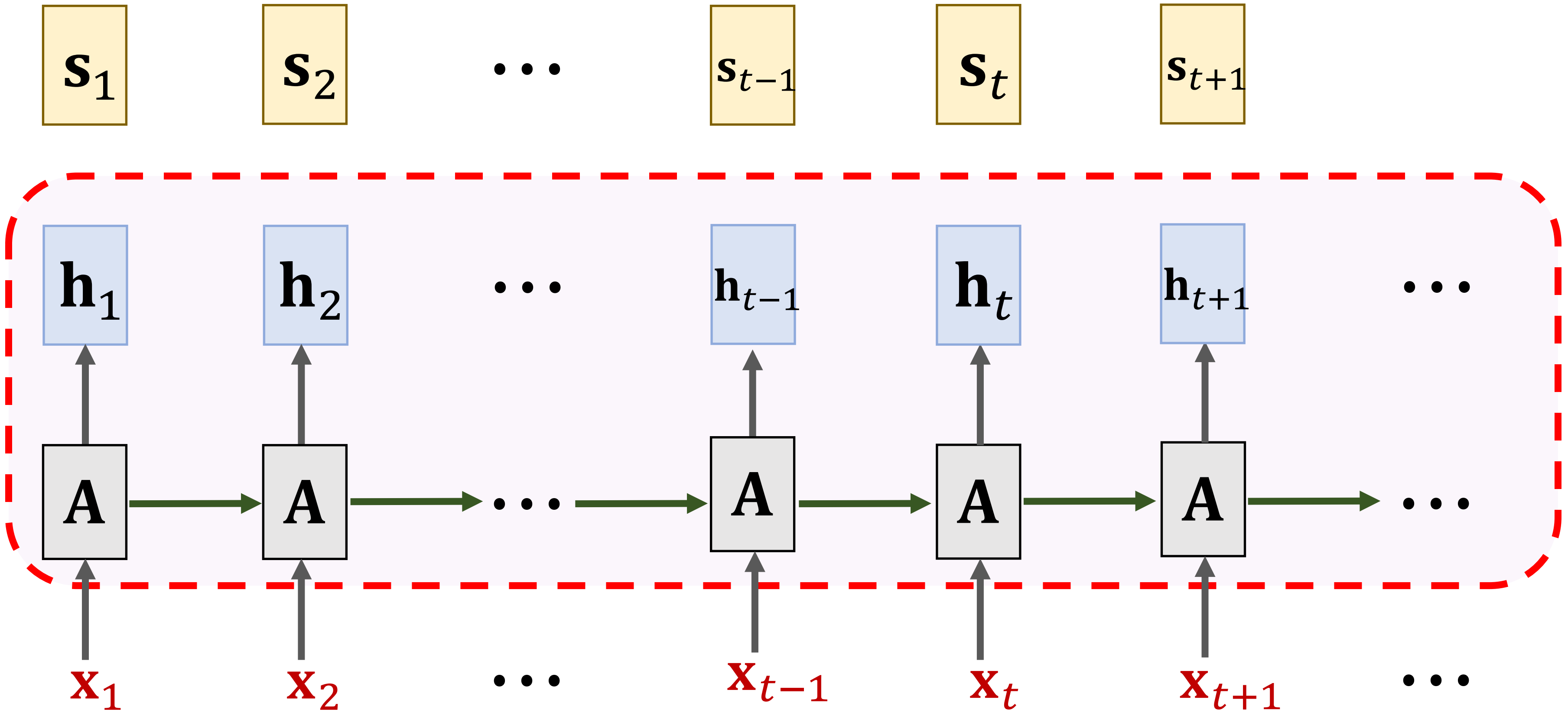


SimpleRNN with Self-Attention



LSTM with Self-Attention

- The update of LSTM with self-attention is analogous.



LSTM with Self-Attention

- The update of LSTM with self-attention is analogous.
- $\mathbf{s}_t = \sum_{i=1}^{t-1} \alpha_i \mathbf{h}_i.$

Exactly the same as SimpleRNN

LSTM with Self-Attention

- The update of LSTM with self-attention is analogous.
- $\mathbf{s}_t = \sum_{i=1}^{t-1} \alpha_i \ \mathbf{h}_i$.

Exactly the same as SimpleRNN

- $\mathbf{c}_t' = \sum_{i=1}^{t-1} \alpha_i \ \mathbf{c}_i$.

Conveyor belt

LSTM with Self-Attention

- The update of LSTM with self-attention is analogous.
- $\mathbf{s}_t = \sum_{i=1}^{t-1} \alpha_i \ \mathbf{h}_i.$
- $\mathbf{c}'_t = \sum_{i=1}^{t-1} \alpha_i \ \mathbf{c}_i.$
- $\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}'_t + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t.$

Forget gate, computed using $\mathbf{x}_t$ and $\mathbf{s}_t$.

New value, computed using $\mathbf{x}_t$ and $\mathbf{s}_t$.

Input gate, computed using $\mathbf{x}_t$ and $\mathbf{s}_t$.

LSTM with Self-Attention

- The update of LSTM with self-attention is analogous.
- $\mathbf{s}_t = \sum_{i=1}^{t-1} \alpha_i \; \mathbf{h}_i.$
- $\mathbf{c}'_t = \sum_{i=1}^{t-1} \alpha_i \; \mathbf{c}_i.$
- $\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}'_t + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t.$
- $\mathbf{h}_t = \mathbf{o}_t \circ \mathbf{c}_t.$

Output gate, computed using $\mathbf{x}_t$ and $\mathbf{s}_t$.

LSTM with Self-Attention

- The update of LSTM with self-attention is analogous.
- $\mathbf{s}_t \;=\; \sum_{i=1}^{t-1} \alpha_i \; \mathbf{h}_i.$
- $\mathbf{c}'_t \;=\; \sum_{i=1}^{t-1} \alpha_i \; \mathbf{c}_i.$
- $\mathbf{c}_t \;=\; \boxed{\mathbf{f}_t} \circ \mathbf{c}'_t \;+\; \boxed{\mathbf{i}_t} \circ \boxed{\tilde{\mathbf{c}}_t.}$
- $\mathbf{h}_t \;=\; \boxed{\mathbf{o}_t} \circ \mathbf{c}_t.$

Difference 1:

- In standard LSTM, the gates and new value are computed using $\mathbf{x}_t$ and $\mathbf{h}_{t-1}$.
- With self-attention, the gates and new value are computed using $\mathbf{x}_t$ and $\mathbf{s}_t$.

LSTM with Self-Attention

- The update of LSTM with self-attention is analogous.
- $\mathbf{s}_t = \sum_{i=1}^{t-1} \alpha_i \ \mathbf{h}_i.$
- $\mathbf{c}'_t = \sum_{i=1}^{t-1} \alpha_i \ \mathbf{c}_i.$
- $\mathbf{c}_t = \mathbf{f}_t \circ \boxed{\mathbf{c}'_t} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t.$
- $\mathbf{h}_t = \mathbf{o}_t \circ \mathbf{c}_t.$

Difference 2:

- In standard LSTM, apply forget gate to $\mathbf{c}_{t-1}$.
- With self-attention, apply forget gate to $\mathbf{c}'_t = \sum_{i=1}^{t-1} \alpha_i \ \mathbf{c}_i$.

Summary

- With self-attention, RNN is less likely to forget.

Summary

- With self-attention, RNN is less likely to forget.
- Pay attention to the context relevant to the new input.

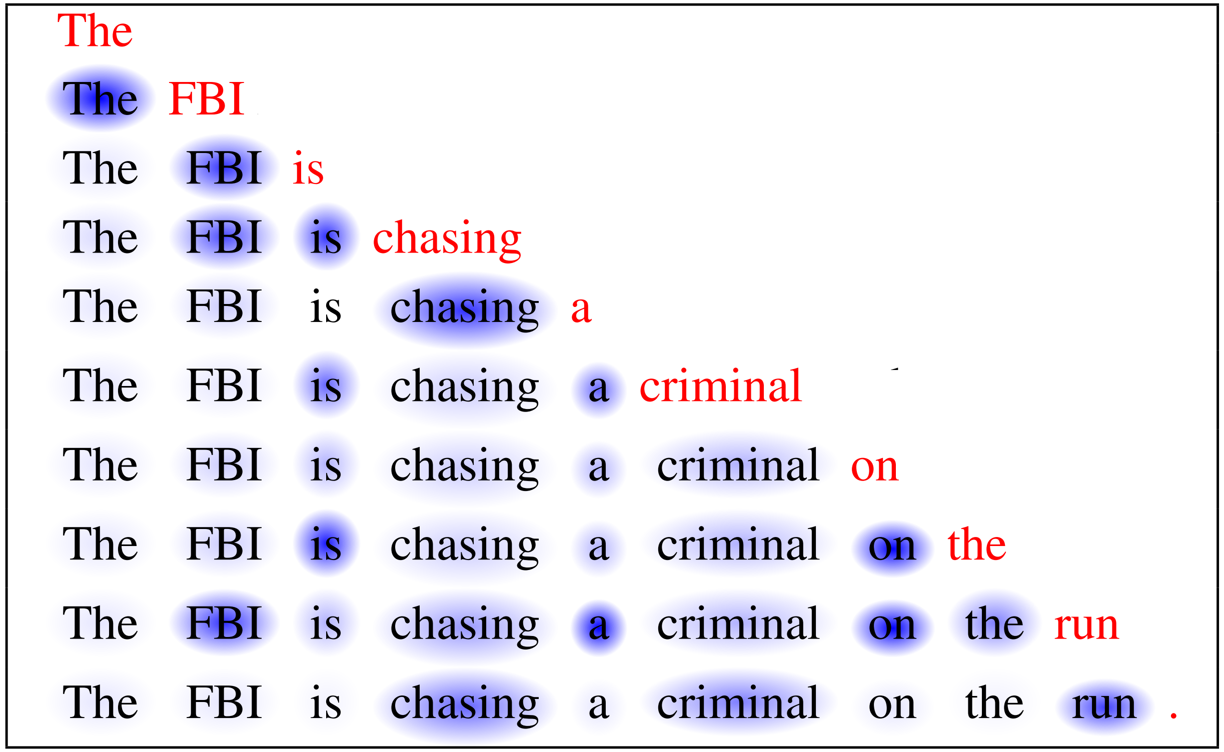


Figure is from the paper " Long Short-Term Memory-Networks for Machine Reading."