Seq2Seq with Attention for Natural Language Understanding and Generation

주재걸 교수

KAIST 김재철AI대학원

DAVIAN

Data and Visual Analytics Lab

Recurrent Neural Networks

Recurrent Neural Networks (RNNs)

- Given a sequence data, we recursively run the same function over time.

- We can process a sequence of vectors **x** by applying a recurrence formula at every time step:

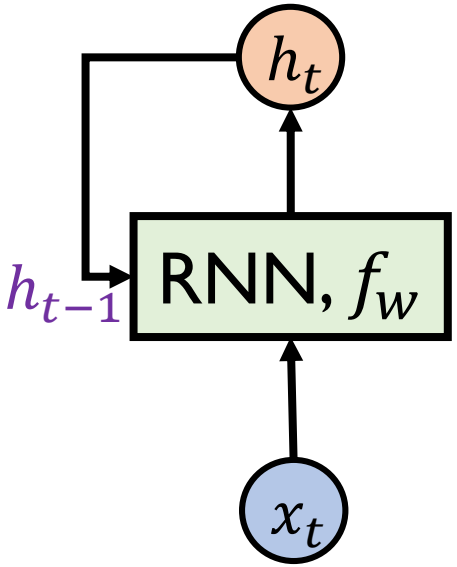
반복호출

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

New state — Some function with parameters W — Old state — Input vector at time step t

current hidden state vector

현재 Time step



Recurrent Neural Networks (RNNs)

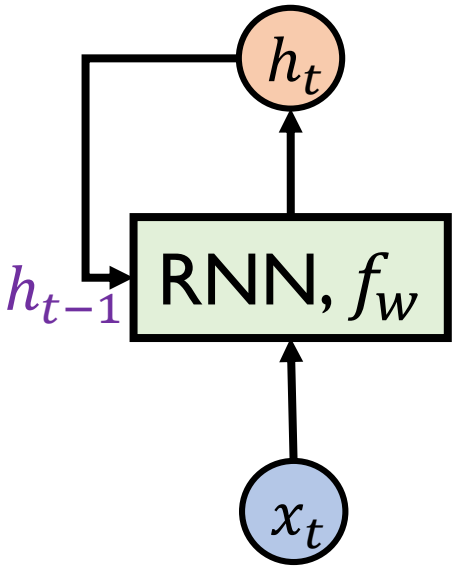
- Given a sequence data, we recursively run the same function over time.

- We can process a sequence of vectors **x** by applying a recurrence formula at every time step:

$$h_t = f_w(h_{t-1}, x_t)$$

Old state

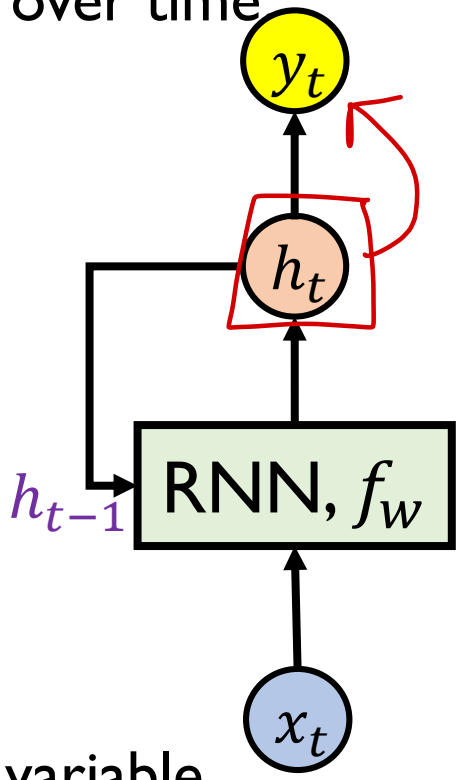
New state



- Note: the same function with the same set of parameters are used at every time step.

Recurrent Neural Networks (RNNs)

- Given a sequence data, we recursively run the same function over time



- We can process a sequence of vectors $\mathbf{x}$ by applying a recurrence formula at every time step:

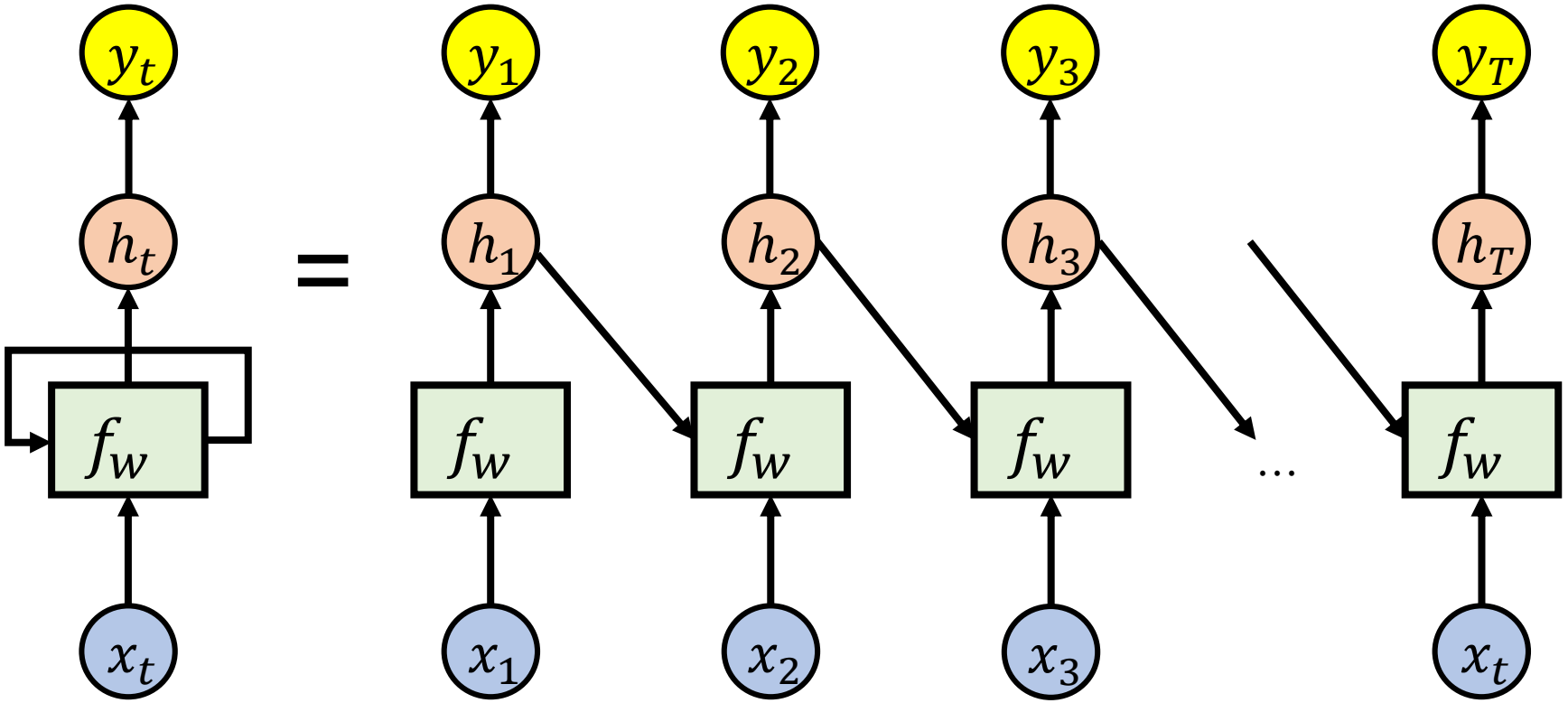
Old state

$$h_t = f_w(h_{t-1}, x_t)$$

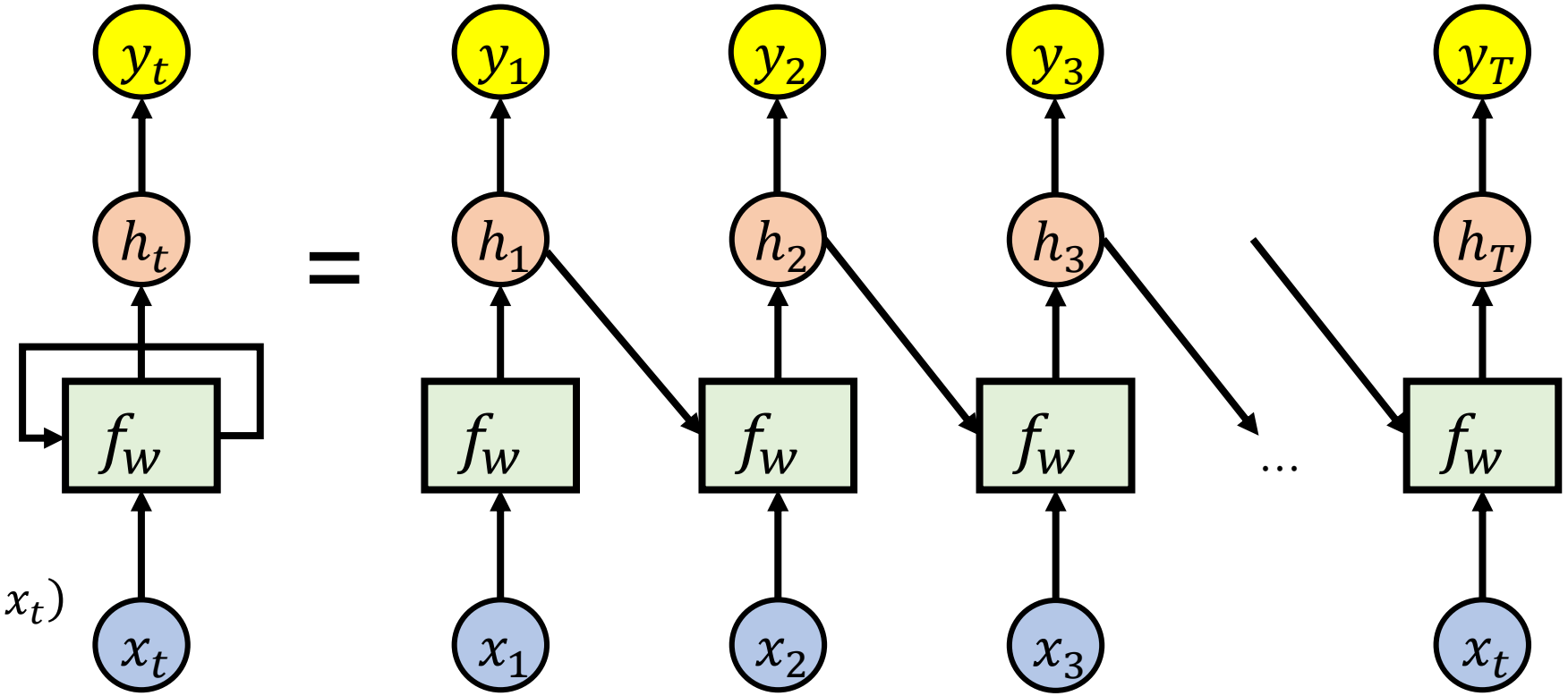
New state

- Optionally, at those time steps we have to predict the target variable, we use h_t as input to the output layer

Unrolled Illustration of RNNs



Basic Architecture of RNNs



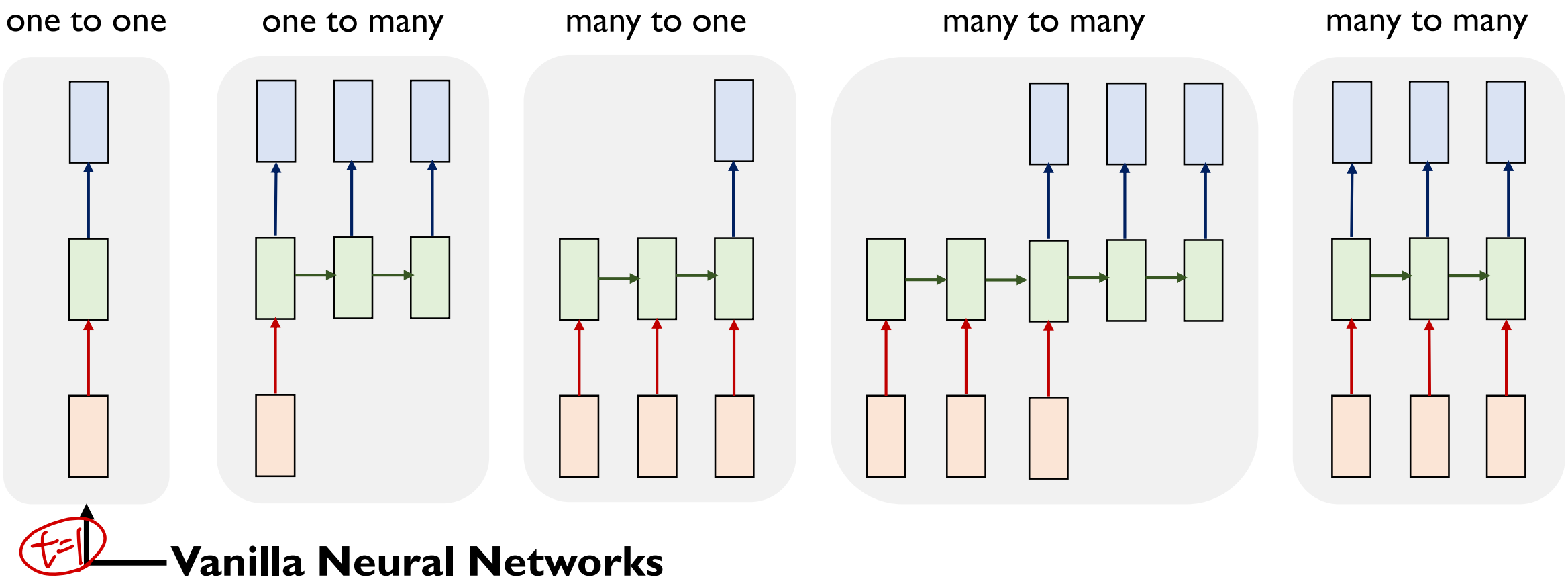
$$h_t = f_W(h_{t-1}, x_t)$$

$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$
$$y_t = W_{hy} h_t$$

t=3

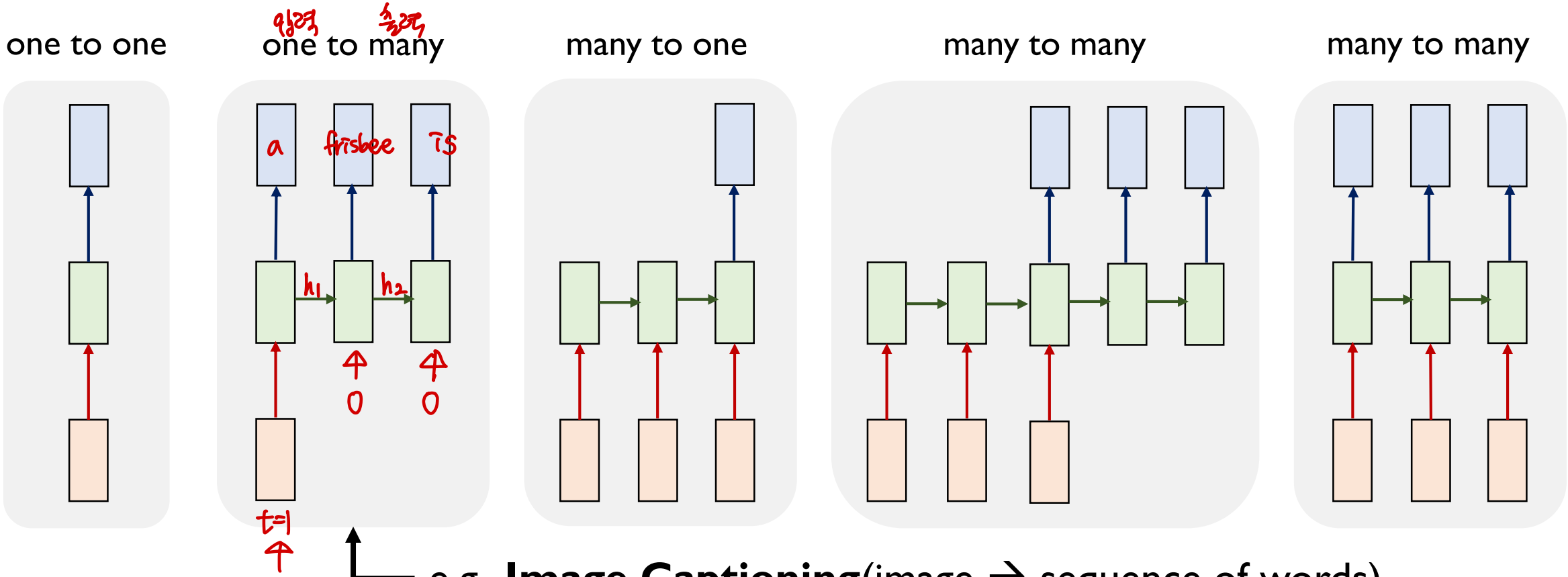
Various Problem Settings of RNN-based Sequence Modeling

- Vanilla neural networks



Various Problem Settings of RNN-based Sequence Modeling

- one-to-many



e.g., **Image Captioning**(image → sequence of words)

Image Captioning Examples

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.

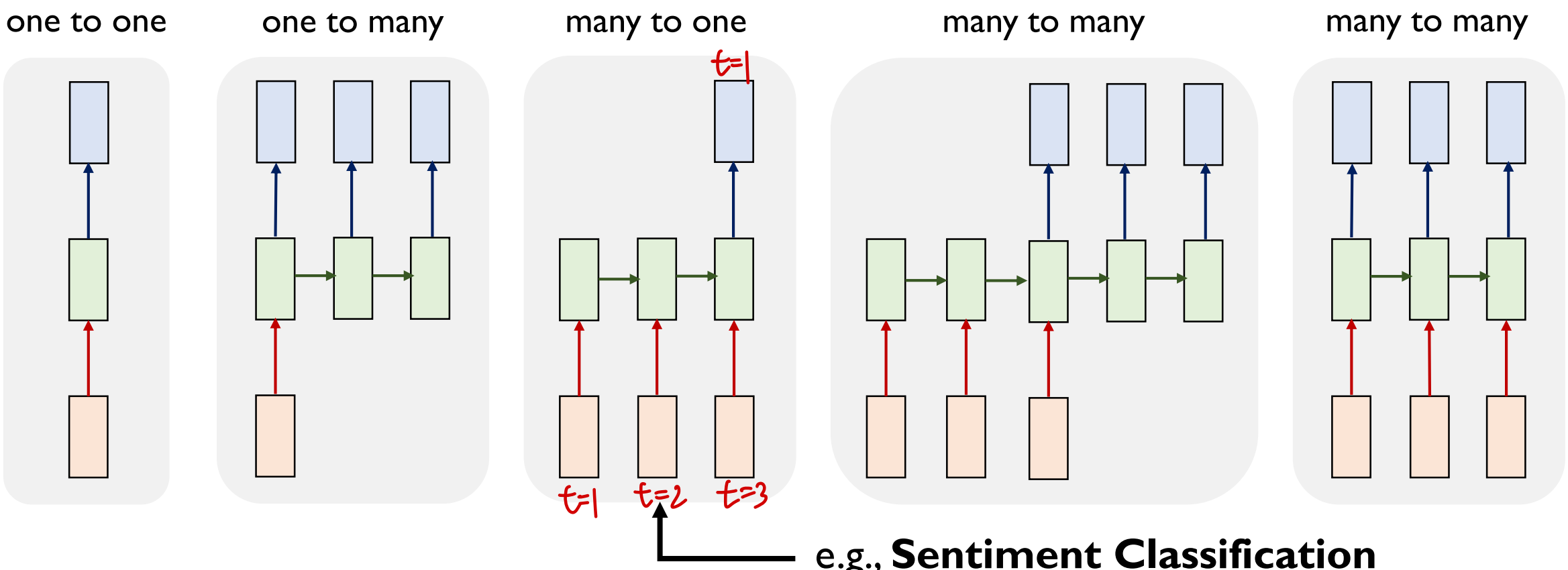
A group of people sitting on a boat in the water.

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

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, ICML'15

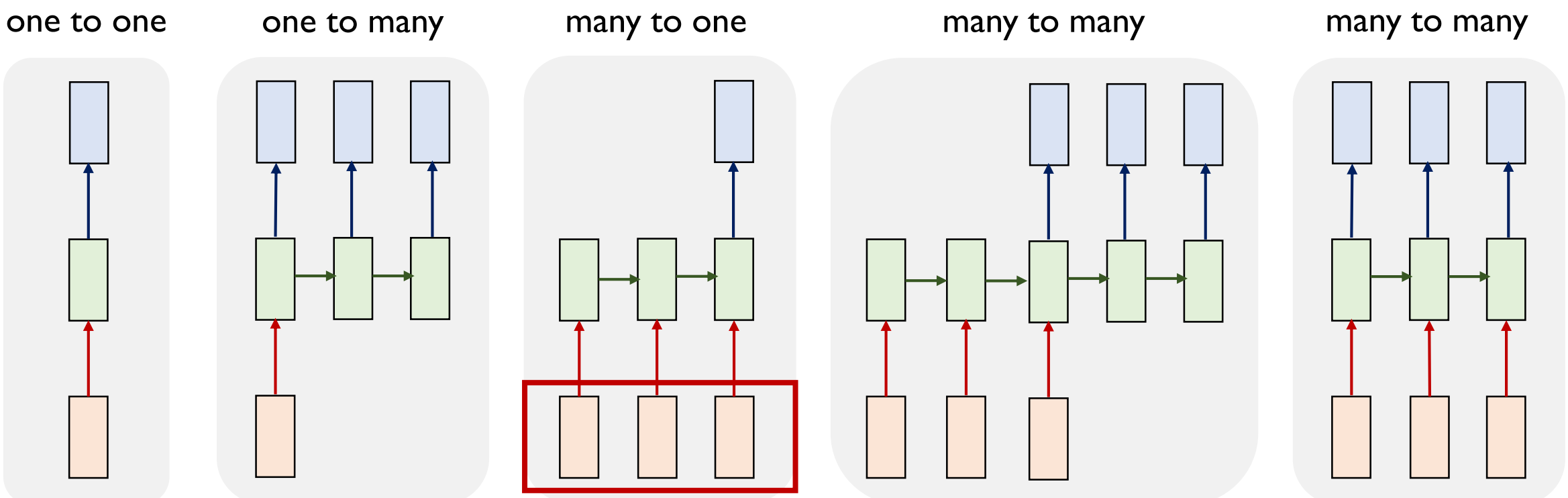
Various Problem Settings of RNN-based Sequence Modeling

- many-to-one



Various Problem Settings of RNN-based Sequence Modeling

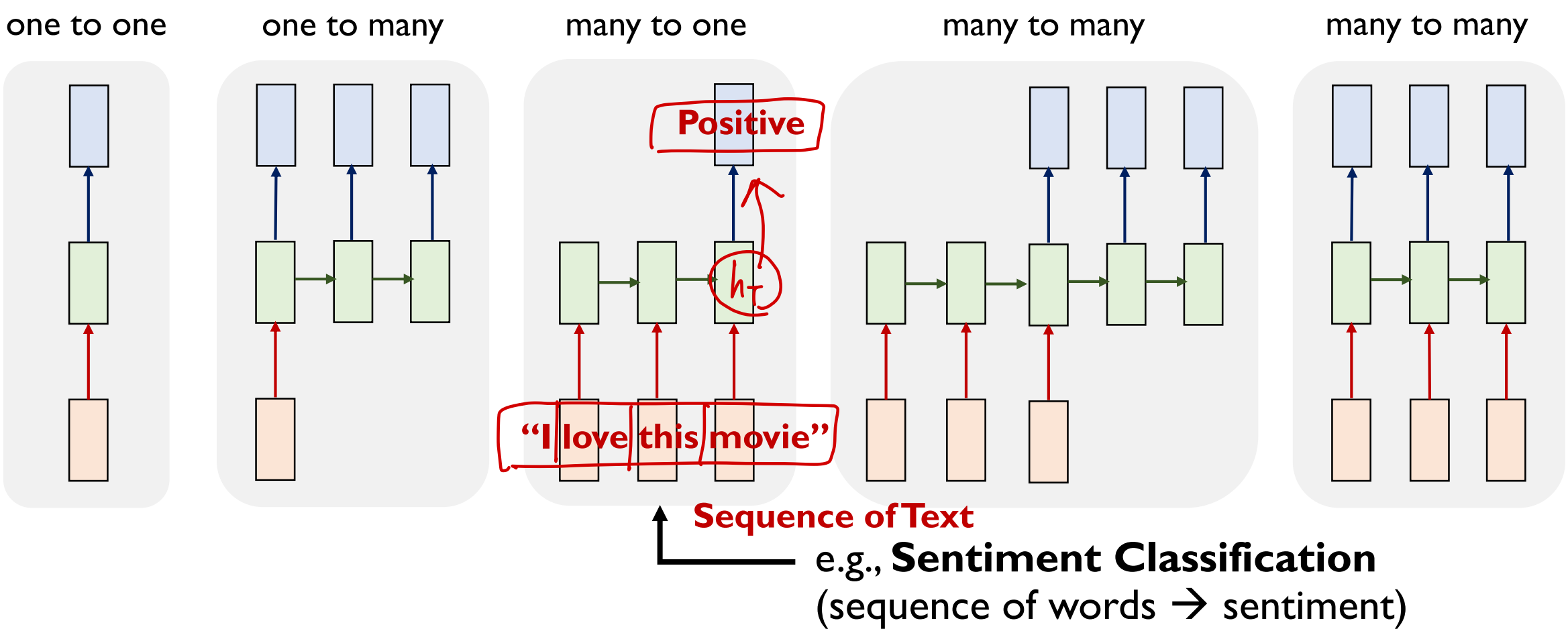
- many-to-one



e.g., **Sentiment Classification**
(sequence of words → sentiment)

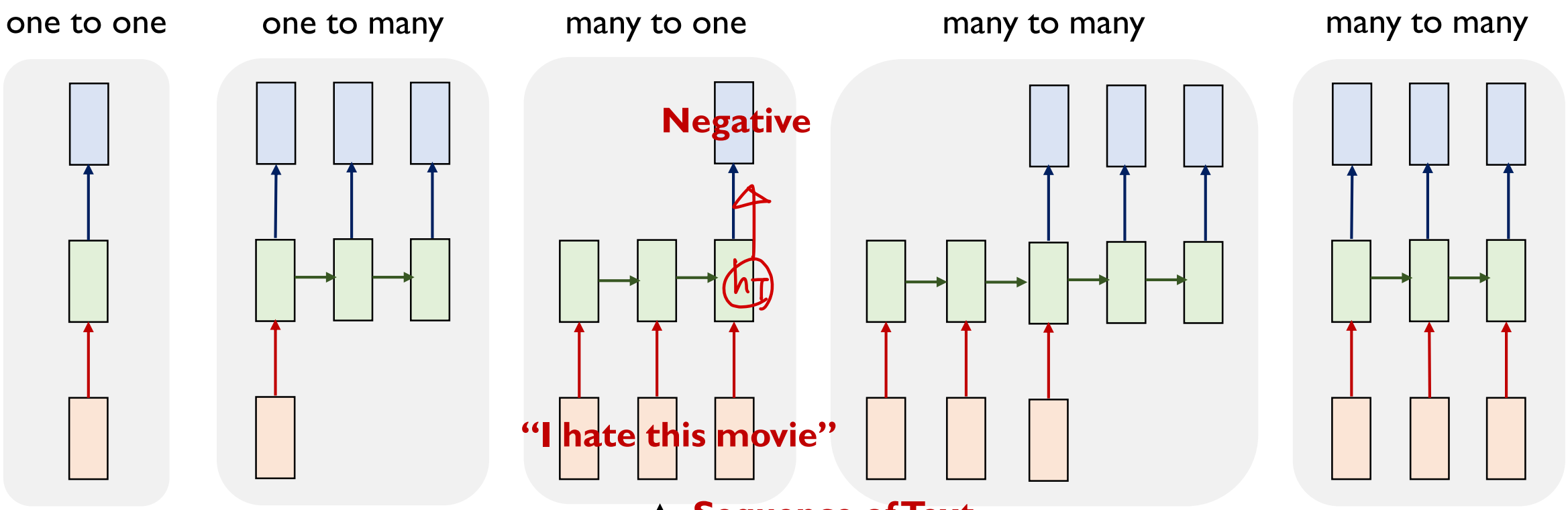
Various Problem Settings of RNN-based Sequence Modeling

- many-to-one



Various Problem Settings of RNN-based Sequence Modeling

- many-to-one

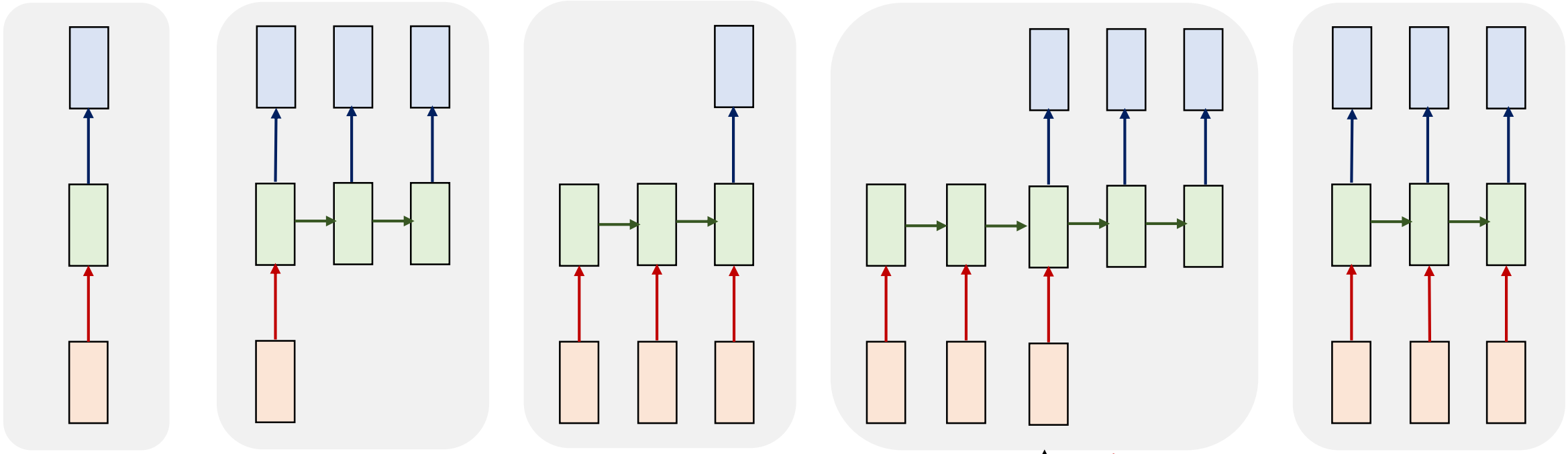


Various Problem Settings of RNN-based Sequence Modeling

- Sequence-to sequence

입출력 둘다 sequence

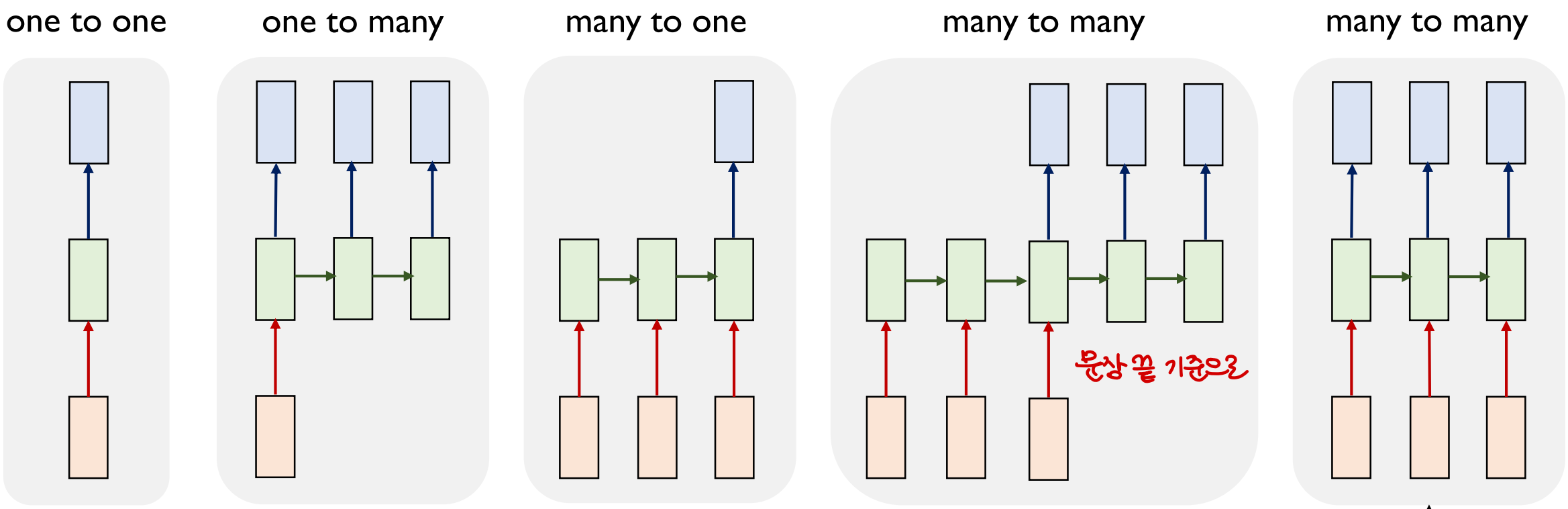
one to one | one to many | many to one | many to many | many to many



e.g., **Machine Translation** ← 번역
(sequence of words → sequence of words)

Various Problem Settings of RNN-based Sequence Modeling

- Sequence-to sequence

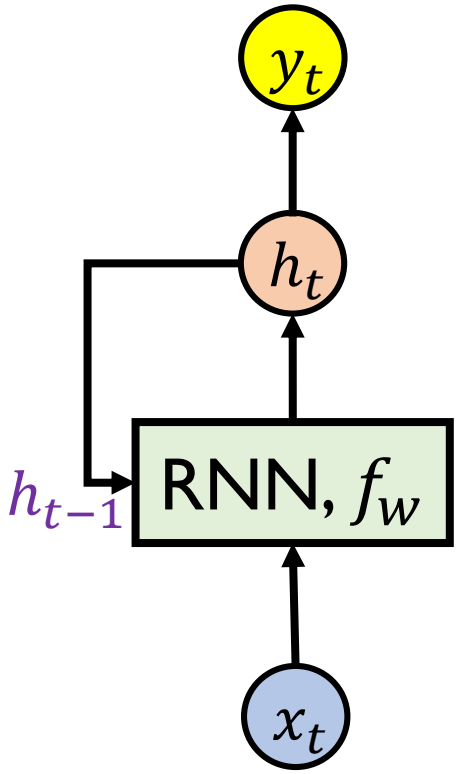


e.g., **Video Classification on Frame Level**

Character-level Language Model

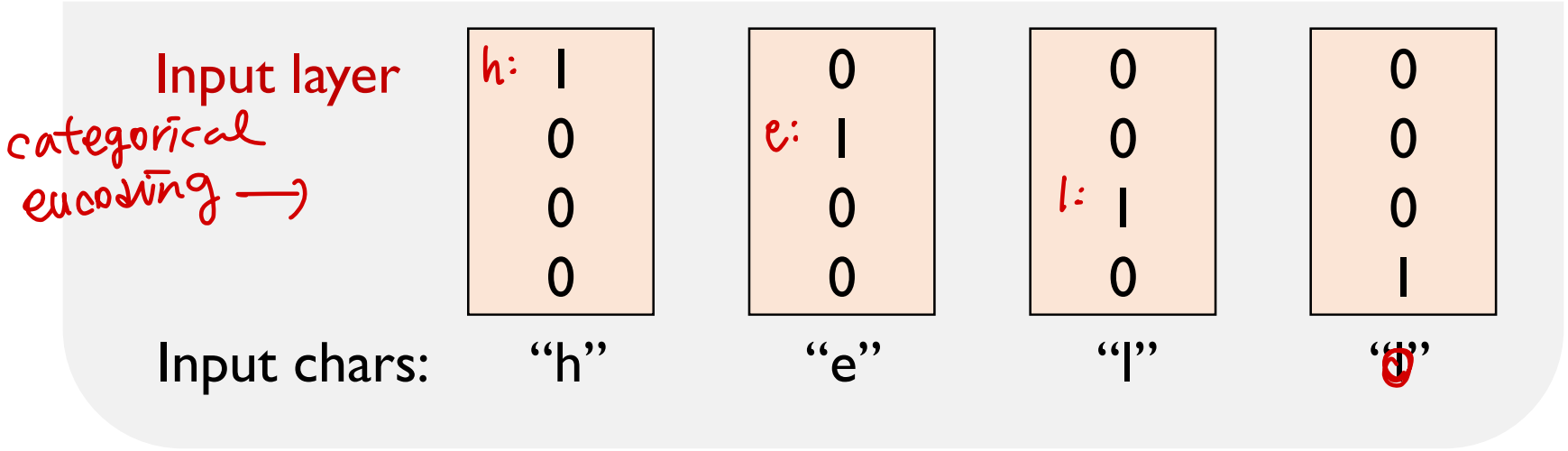
- Character-level language model example:
- Vocabulary: [h, e, l, o]
- Example training sequence: "hello"

↑ 5개의 character



Character-level Language Model

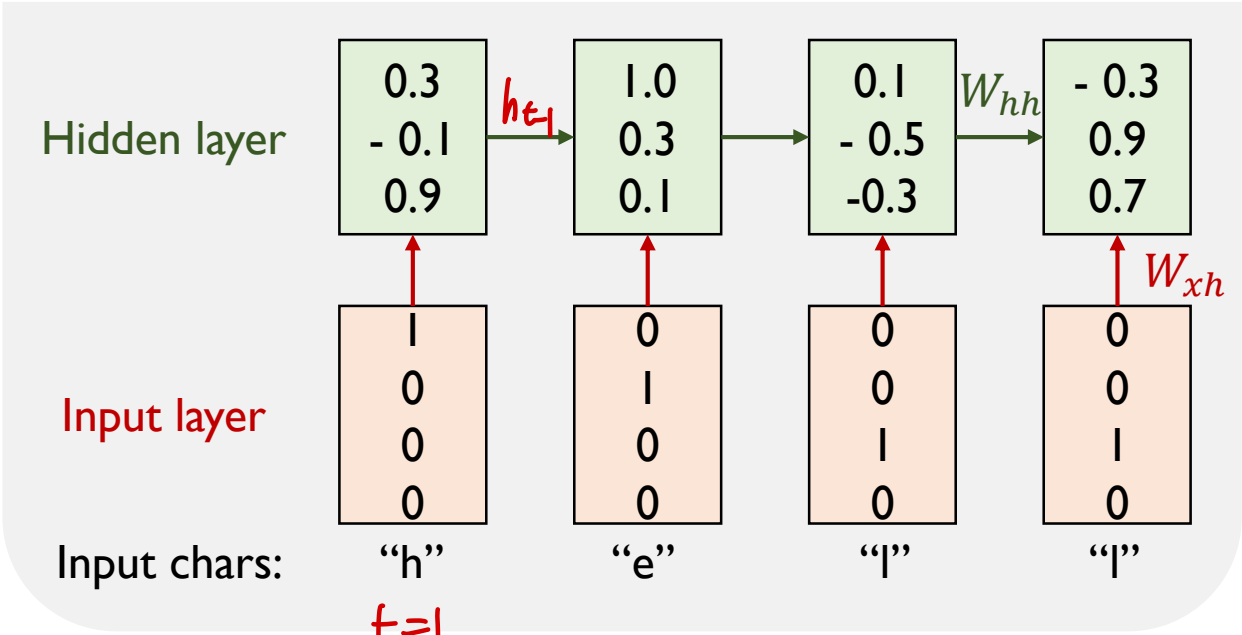
- Character-level language model example:
- Vocabulary: [h, e, l, o]
- Example training sequence: "hello"



Character-level Language Model

- Character-level language model example:
- Vocabulary: [h, e, l, o]
- Example training sequence: "hello"

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

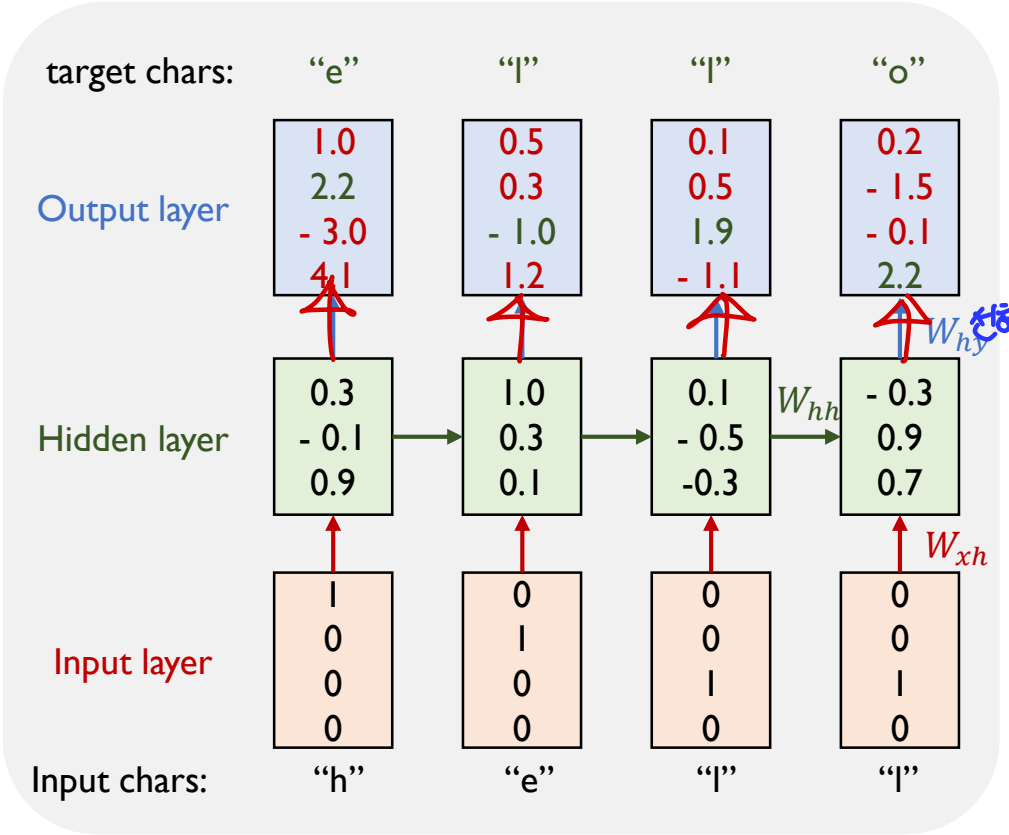


t=1

↳ t-1 존재 안할때 '0'

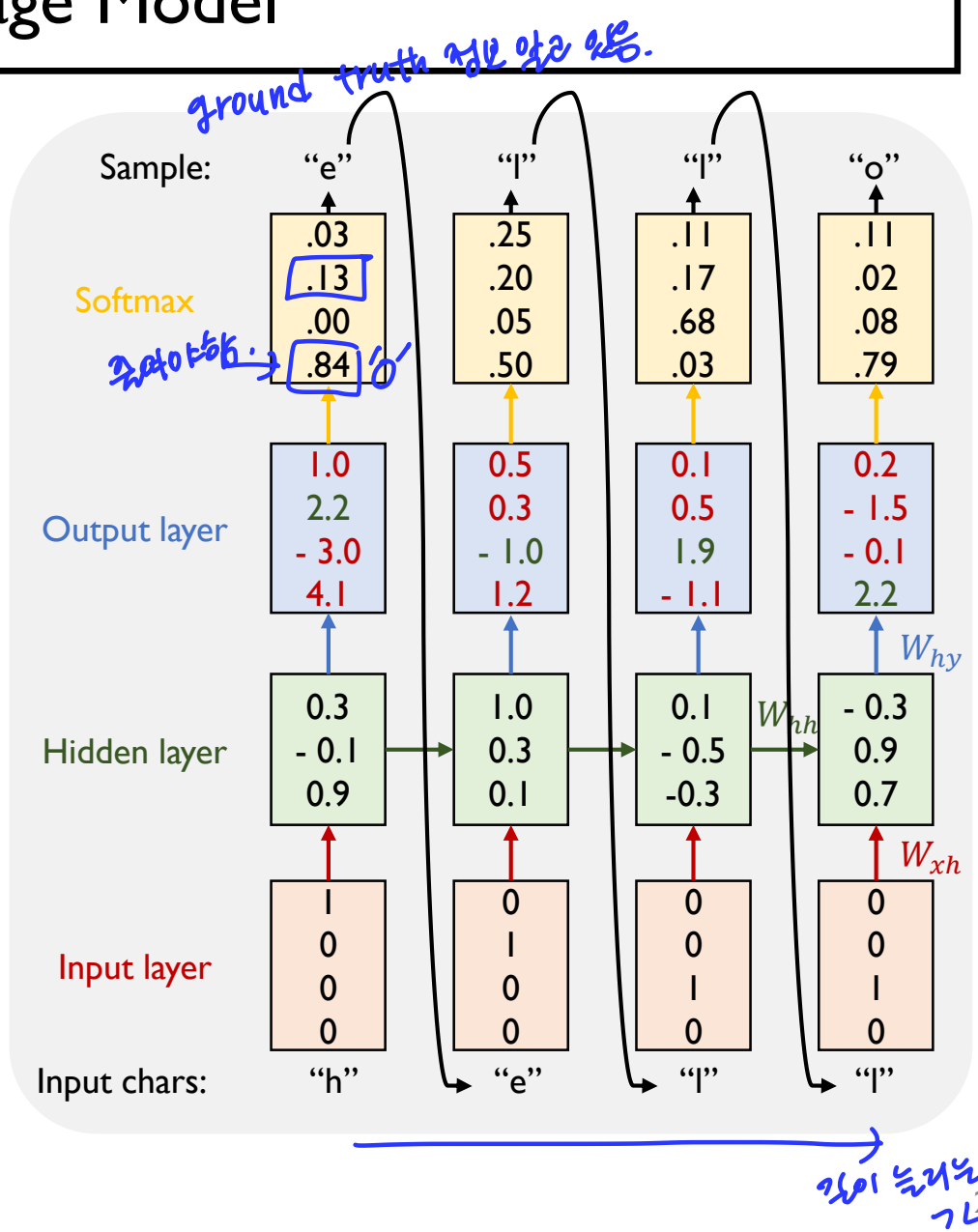
Character-level Language Model

- Character-level language model example:
- Vocabulary: [h, e, l, o]
- Example training sequence: "hello"



Character-level Language Model

- Character-level language model example:
- Vocabulary: [h, e, l, o]
- Example training sequence: "hello"
- At test-time sample characters one at a time, feed back as an input to the model at the next time step, which is called **auto-regressive model**

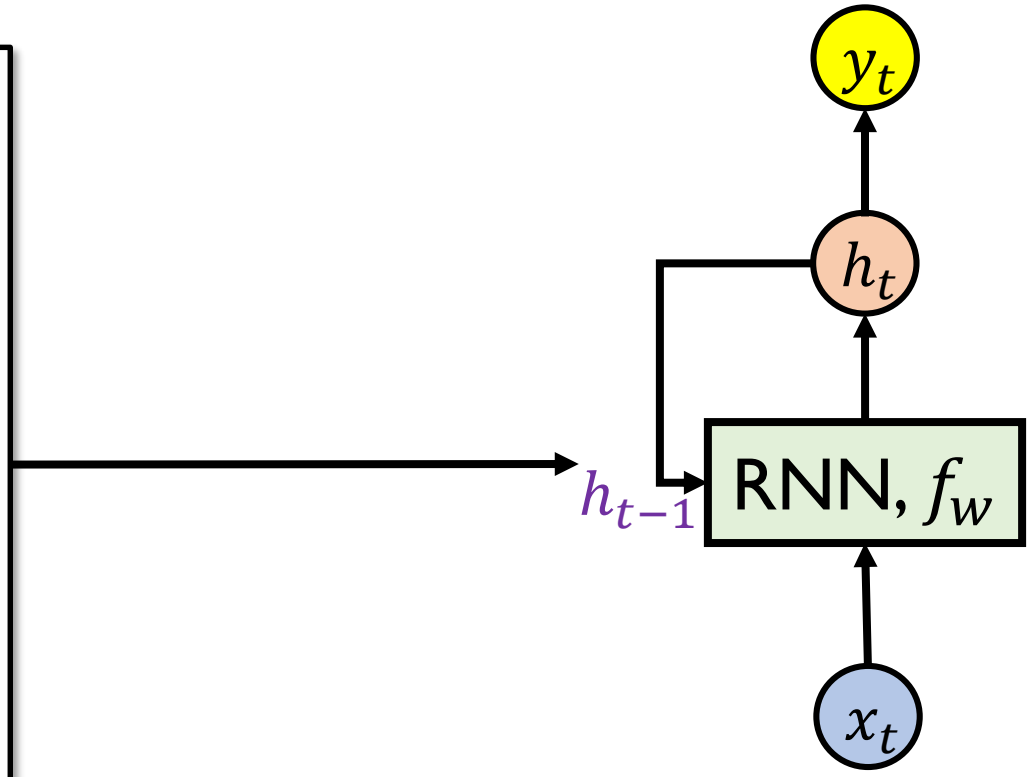


Character-level Language Model

- Training an RNN on Shakespeare's plays

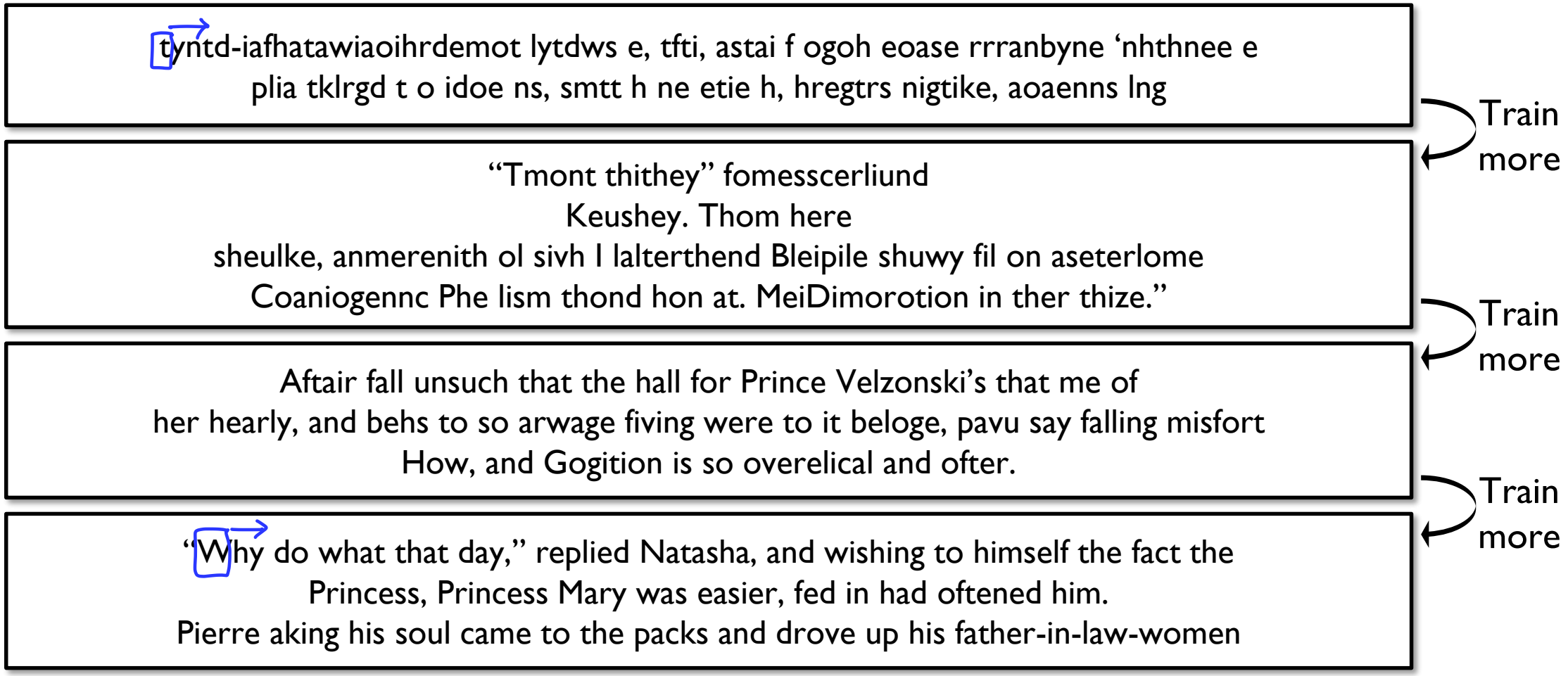
Sonnet 116 – Let me not …
by *William Shakespeare*

Let me not to the marriage of true minds
Admit impediments. Love is not love
Which alters when it alteration finds,
Or bends with the remover to remove:
O no! it is an ever-fixed mark
That looks on tempests and is never shaken;
It is the star to every wandering bark,
Whose worth's unknown, although his height be taken.
Love's not Time's fool, though rosy lips and cheeks
Within his bending sickle's compass come:
Love alters not with his brief hours and weeks,
But bears it out even to the edge of doom.
If this be error and upon me proved,
I never writ, nor no man ever loved.



Character-level Language Model

- Training process of RNN



Character-level Language Model

- Results of trained RNN

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

이름생성 ㄱㅆ

```
VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.
```

Character-level Language Model

- A paper written by RNN

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

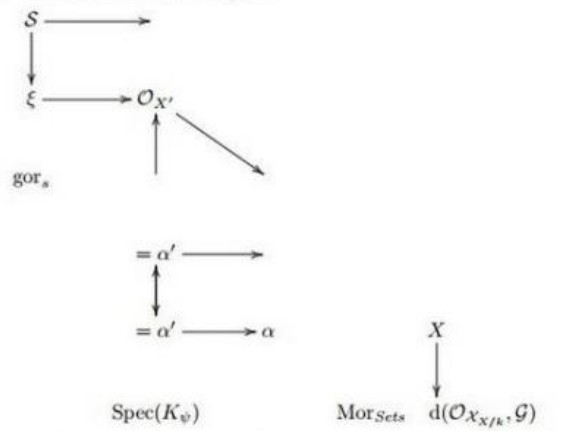
be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram



is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \quad \text{-1}(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_\ell}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{n}})$$

is an isomorphism of covering of $\mathcal{O}_{X_i}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

Character-level Language Model

- C code generated by RNN

```
static void do_command(struct seq_file *m, void *v)
{
        int column = 32 << (cmd[2] & 0x80);
        if (state)
                cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
        else
                seq = 1;
        for (i = 0; i < 16; i++) {
                if (k & (1 << 1))
                        pipe = (in_use & UMXTHREAD_UNCCA) +
                                ((count & 0x00000000fffffff8) & 0x000000f) << 8;
                if (count == 0)
                        sub(pid, ppc_md.kexec_handle, 0x20000000);
                pipe_set_bytes(i, 0);
        }
        /* Free our user pages pointer to place camera if all dash */
        subsystem_info = &of_changes[PAGE_SIZE];
        rek_controls(offset, idx, &soffset);
        /* Now we want to deliberately put it to device */
        control_check_polarity(&context, val, 0);
        for (i = 0; i < COUNTER; i++)
                seq_puts(s, "policy ");
}
```

Gradient Vanishing or Exploding Problem of Vanilla RNNs

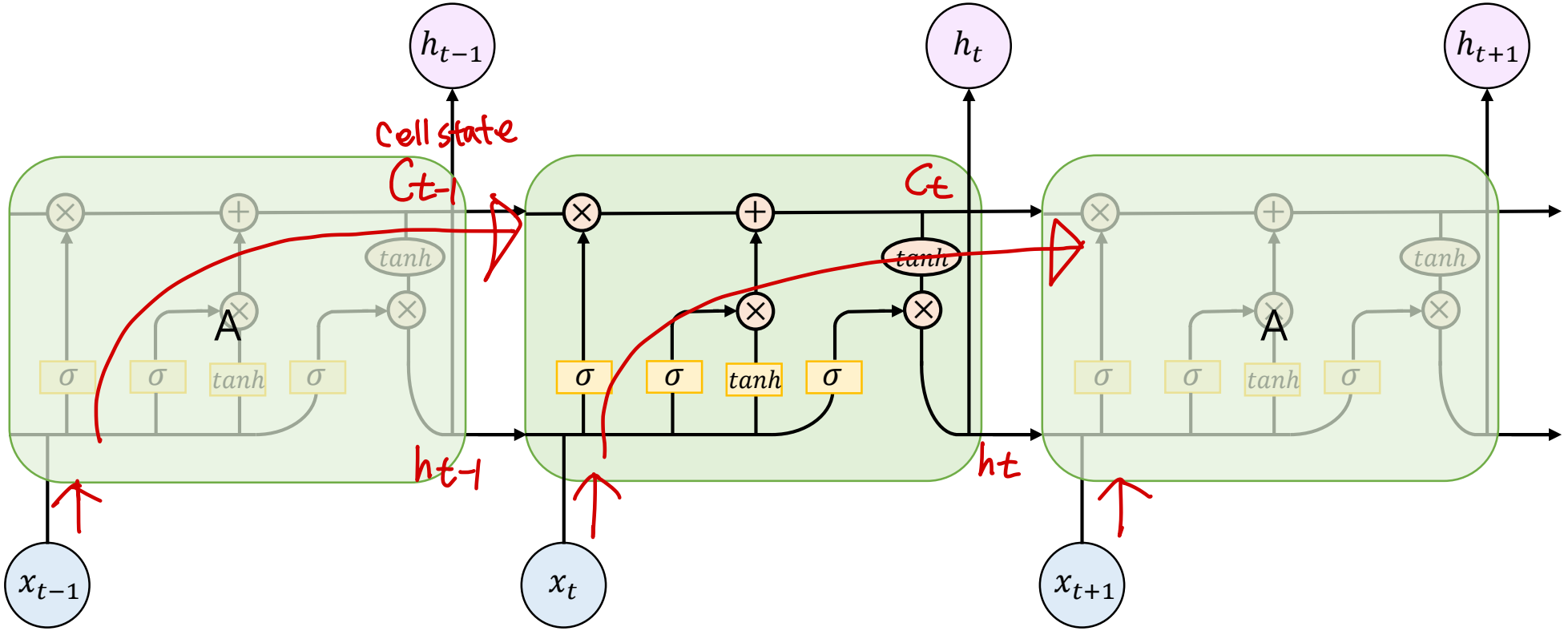
- Vanilla RNNs are simple but don't work very well due to a gradient vanishing or exploding problem.

- Thus, an advanced RNN models such as LSTM or GRU are often used in practice.

long short term

Long Short-Term Memory (LSTM)

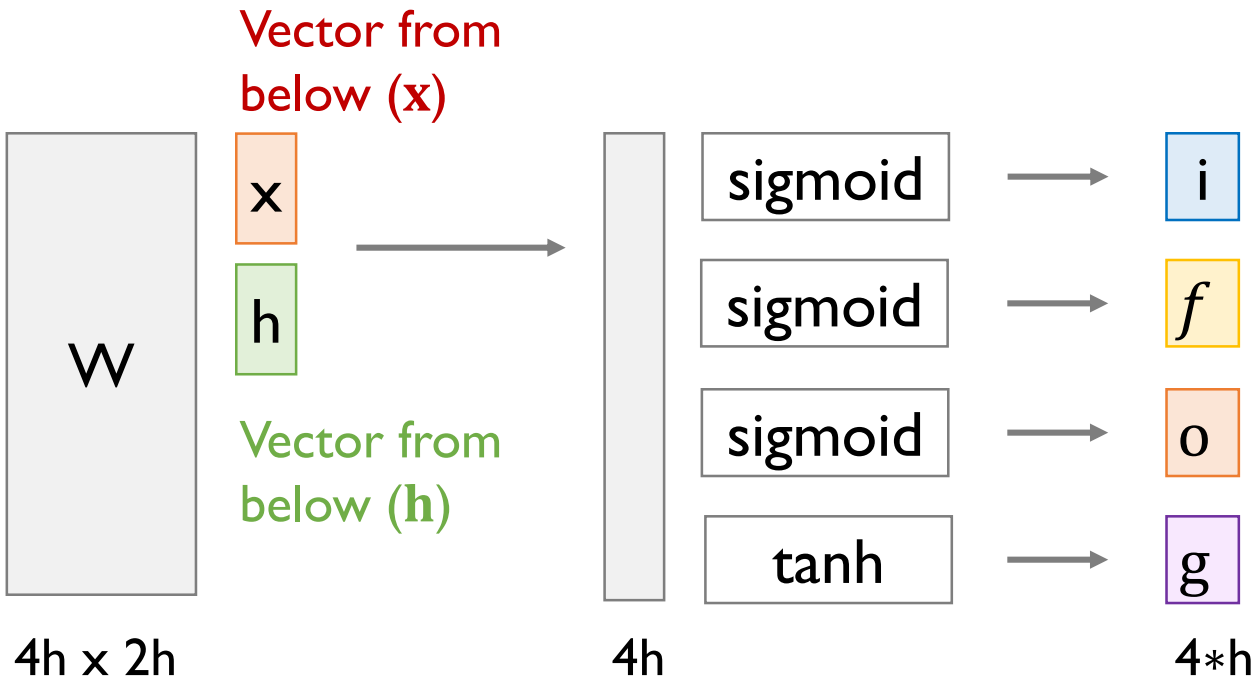
- What is LSTM (Long Short-Term Memory)?
- The repeating module in an LSTM contains four interacting layers



Long Short-Term Memory (LSTM)

- **f**: **Forget gate**, Whether to erase cell
- **i**: **Input gate**, Whether to write to cell
- **g**: **Gate gate**, How much to write to cell
- **o**: **Output gate**, How much to reveal cell

- **W**: weight (matrix)
- **H**: hidden state
- **c**: cell state
- **x**: input



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

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

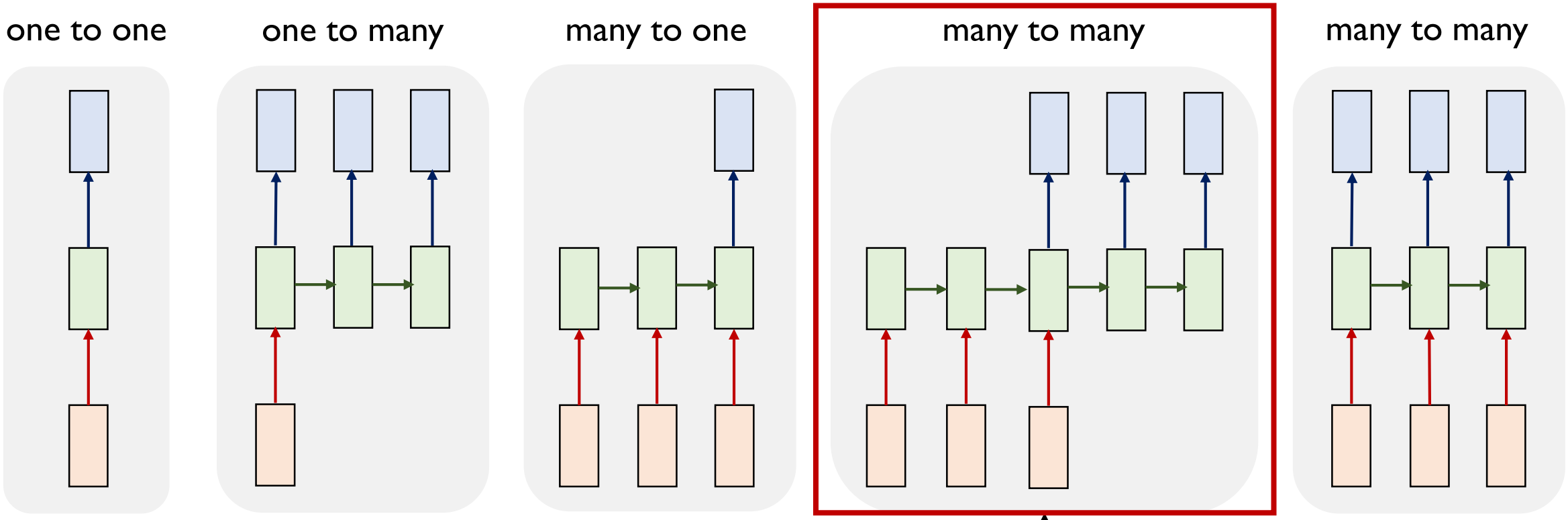
$$h_t = o \odot \tanh(c_t)$$

Seq2seq and Attention Model

Recall: Various Problem Settings of RNN-based Sequence Modeling

- **Sequence-to sequence**

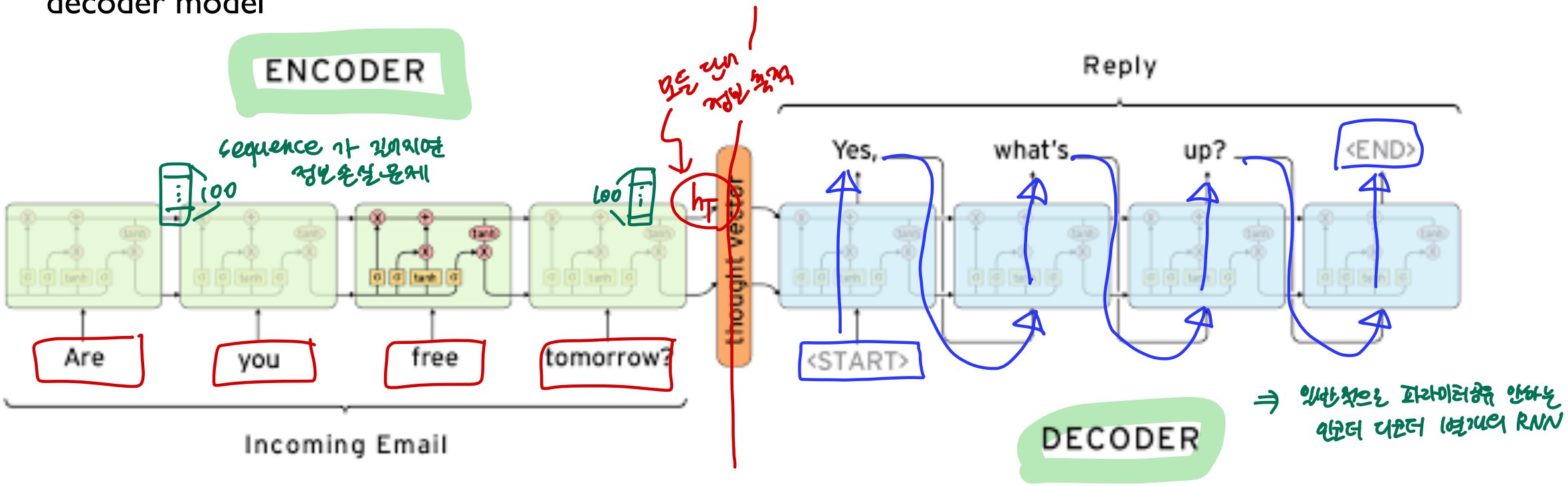
Seq2Seq



e.g., **Machine Translation**
(sequence of words → sequence of words)

Seq2Seq Model

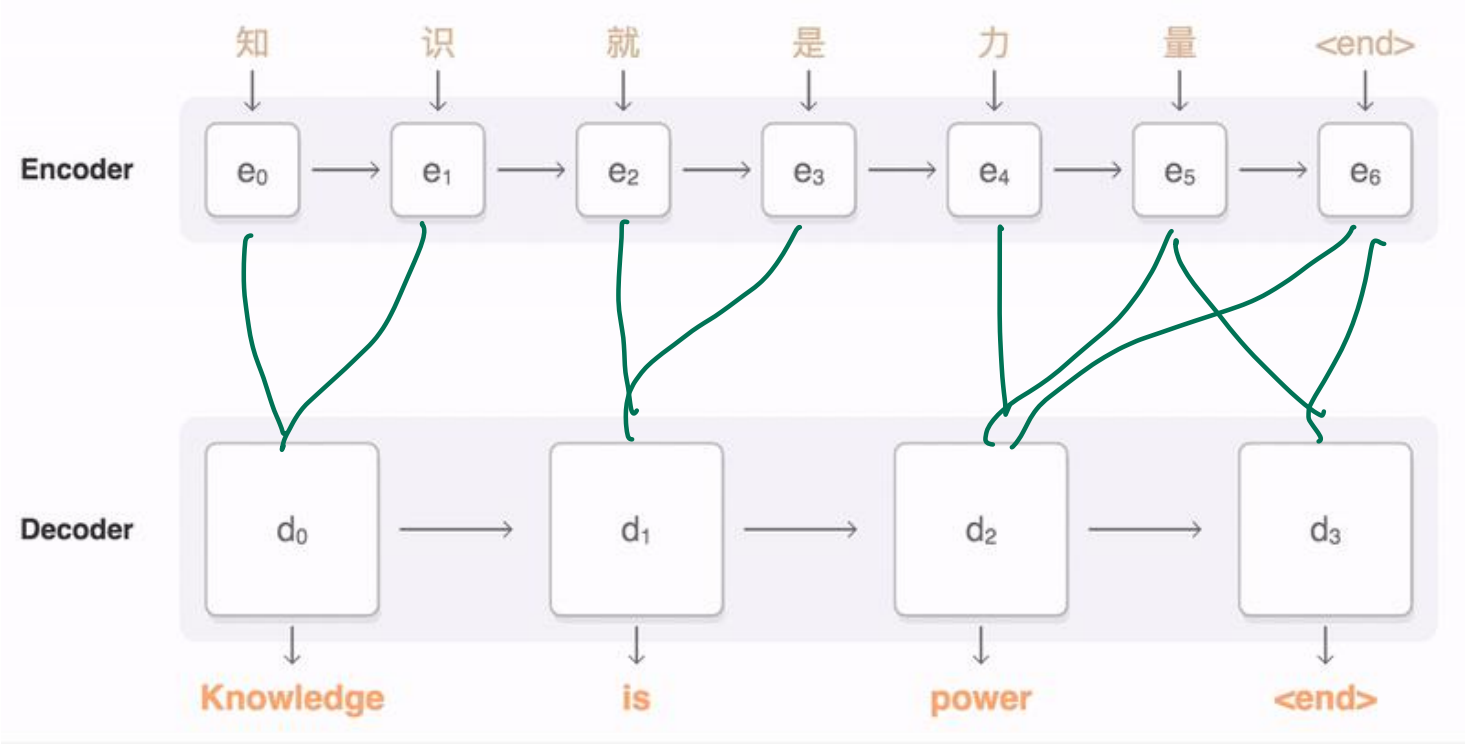
- It takes a sequence of words as input and gives a sequence of words as output
- It is composed of an encoder and a decoder
- Many NLP applications exist, e.g., machine translation, dialog systems, and so on
- Encode source into a fixed dimensional vector, use it as an initial hidden-state vector h_0 for a decoder model



Sequence to sequence learning with neural networks, ICML'14

Seq2Seq with Attention

- "Attention" provides a solution to the bottleneck problem of the original Seq2Seq
- Core idea: At each time step of the decoder, allow the decoder to utilize a different part of the source sequence
- Tensorflow official implementation: https://github.com/google/seq2seq

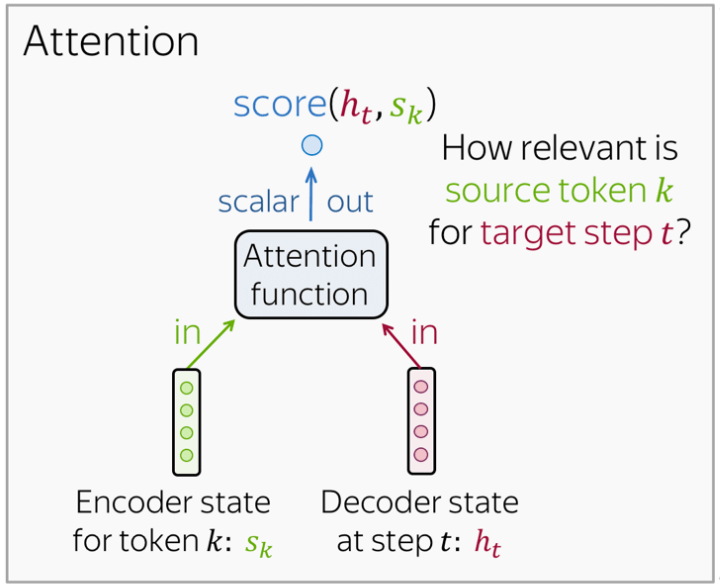


Seq2Seq with Attention

Attention output: weighted sum of encoder states with attention weights

Attention weights: distribution over source tokens

A model can learn to "pay attention" to the most relevant source tokens for each step



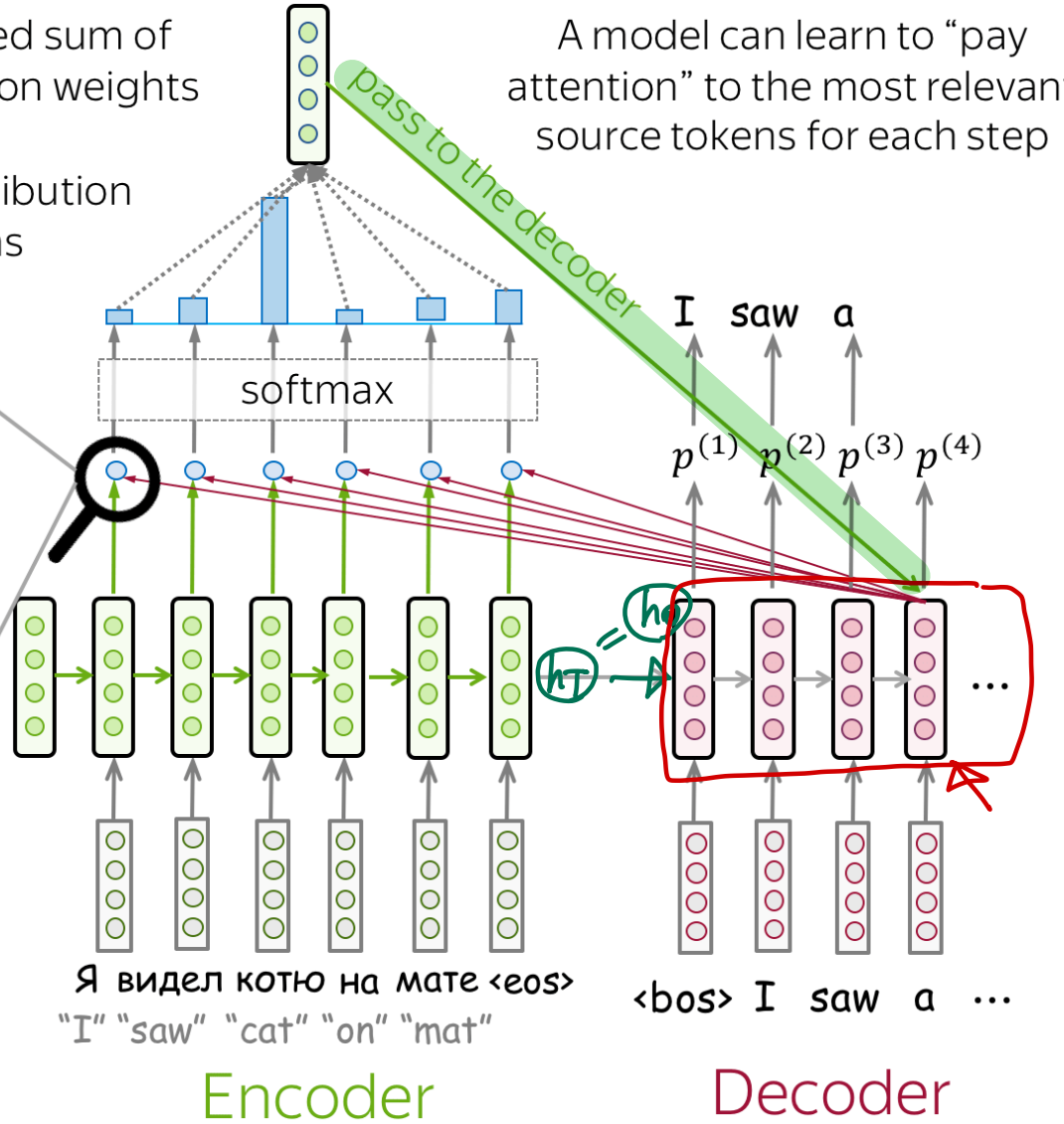
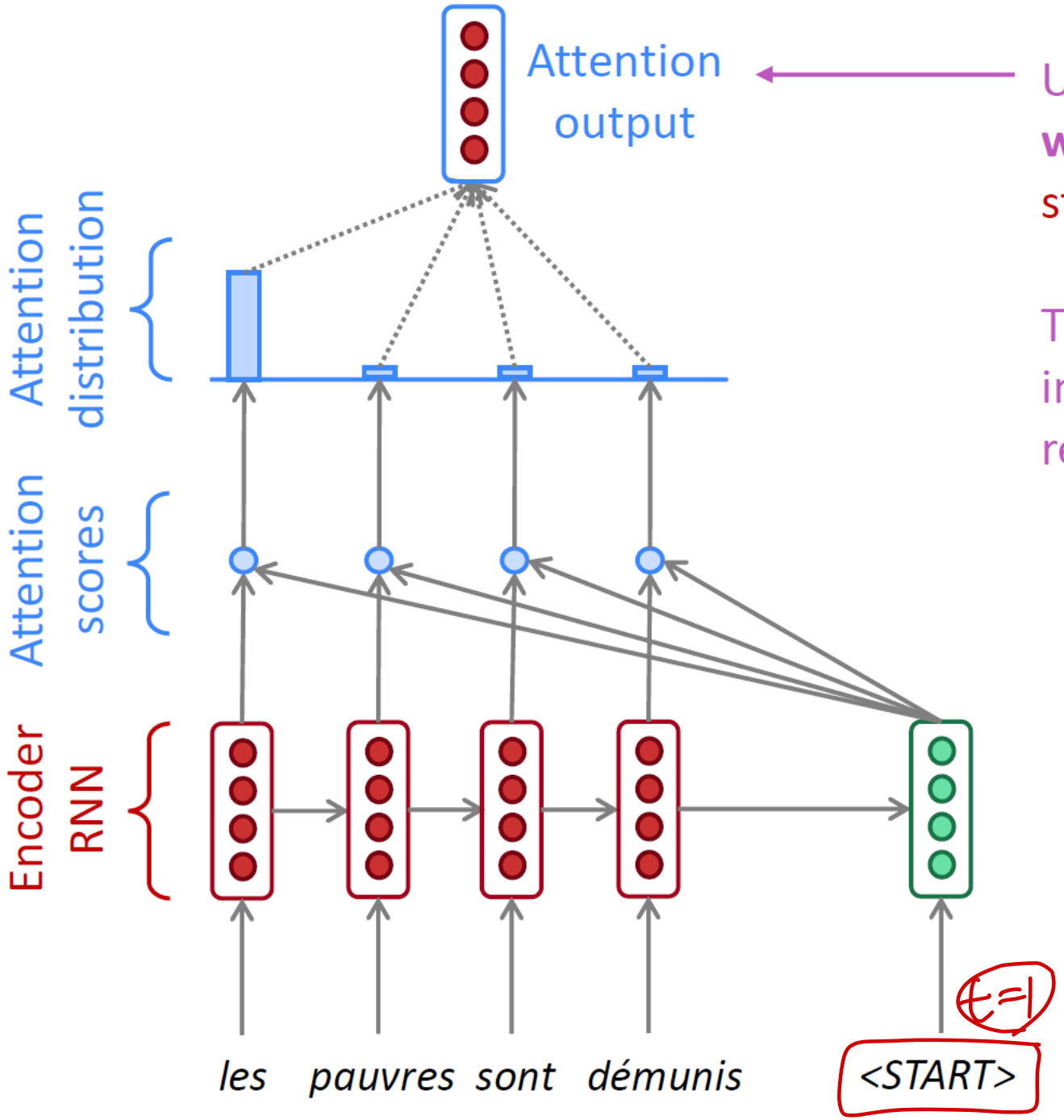
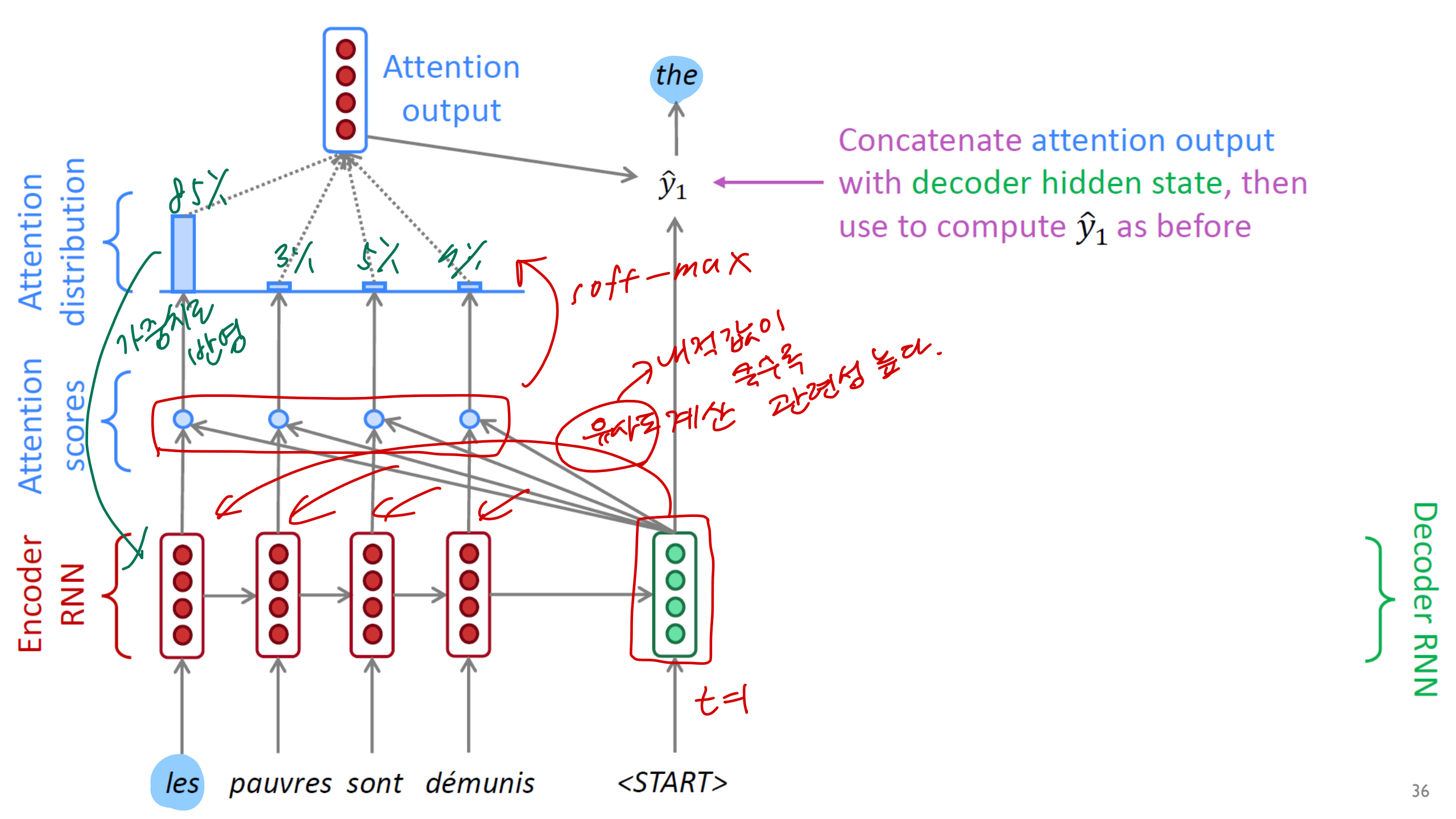


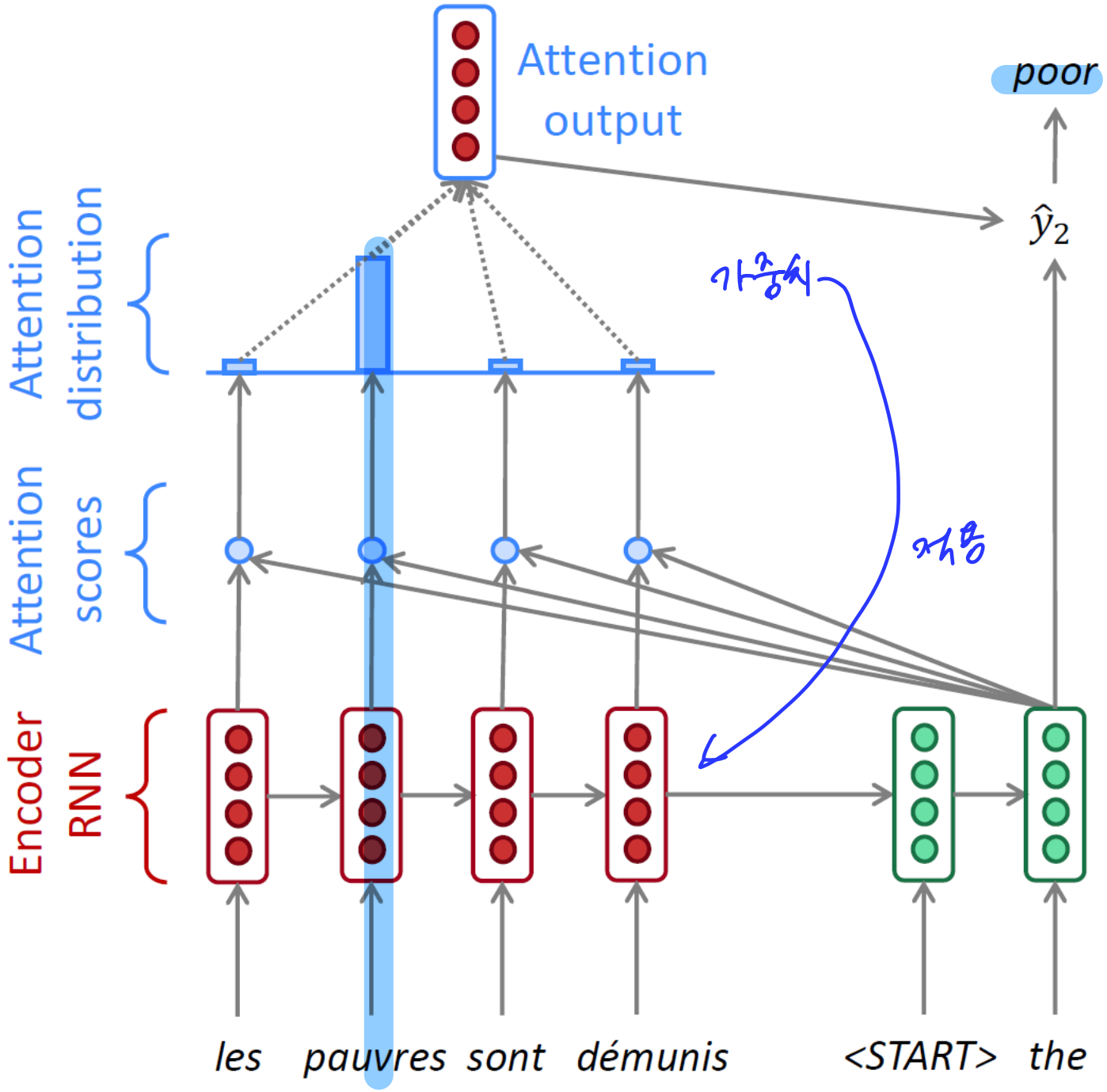
Image courtesy: https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

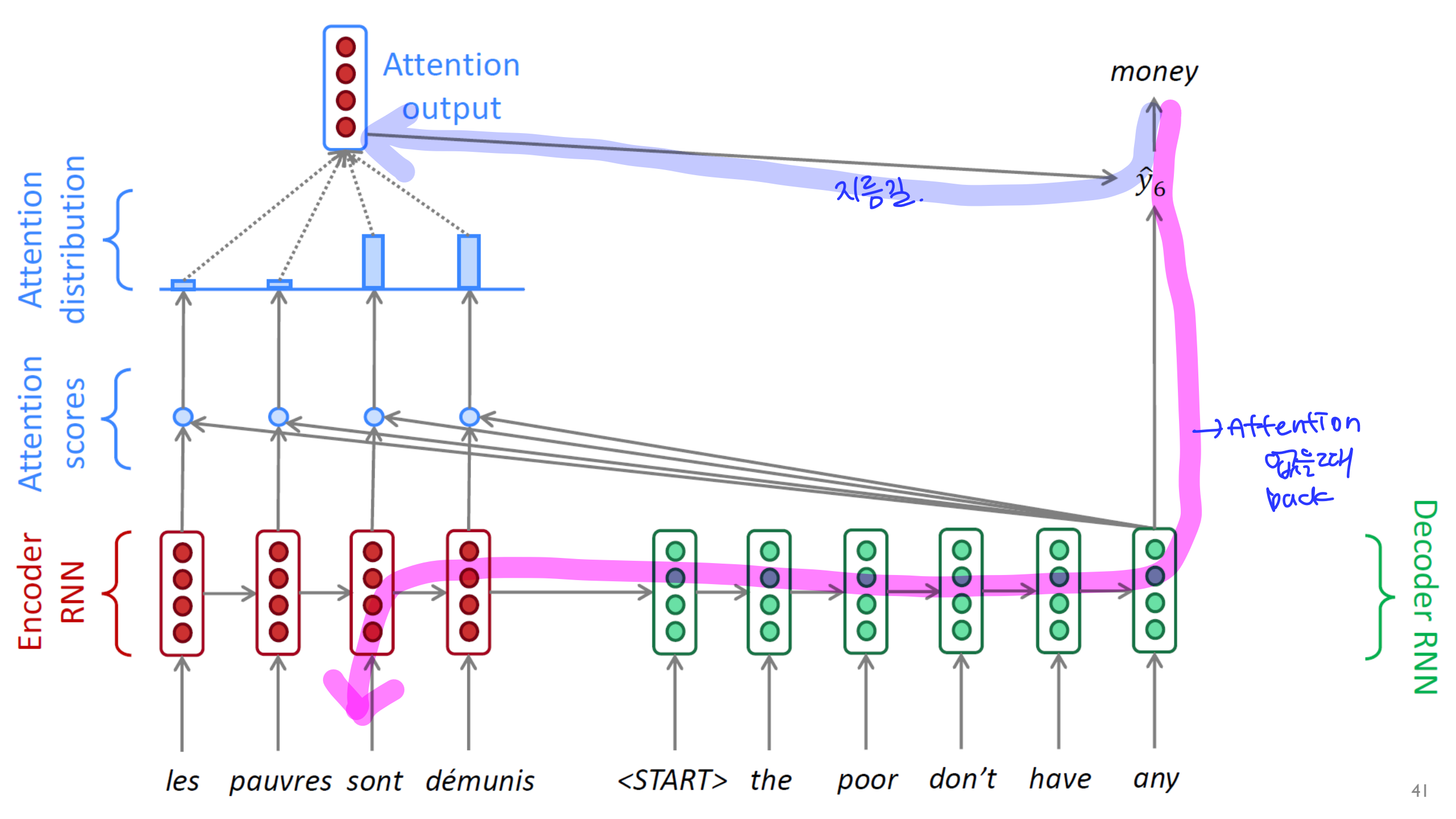


Use the attention distribution to take a **weighted sum** of the encoder hidden states.

The attention output mostly contains information the hidden states that received high attention.


Attention output
the
$\hat{y}_1$
Concatenate attention output with decoder hidden state, then use to compute $\hat{y}_1$ as before
Attention distribution
Attention scores
Encoder RNN
Decoder RNN
softmax
85%
3%
5%
4%
les
pauvres
sont
démunis
<START>
t=1


Attention output
poor
$\hat{y}_2$
Attention distribution
Attention scores
Encoder RNN
Decoder RNN
les
pauvres
sont
démunis
<START>
the

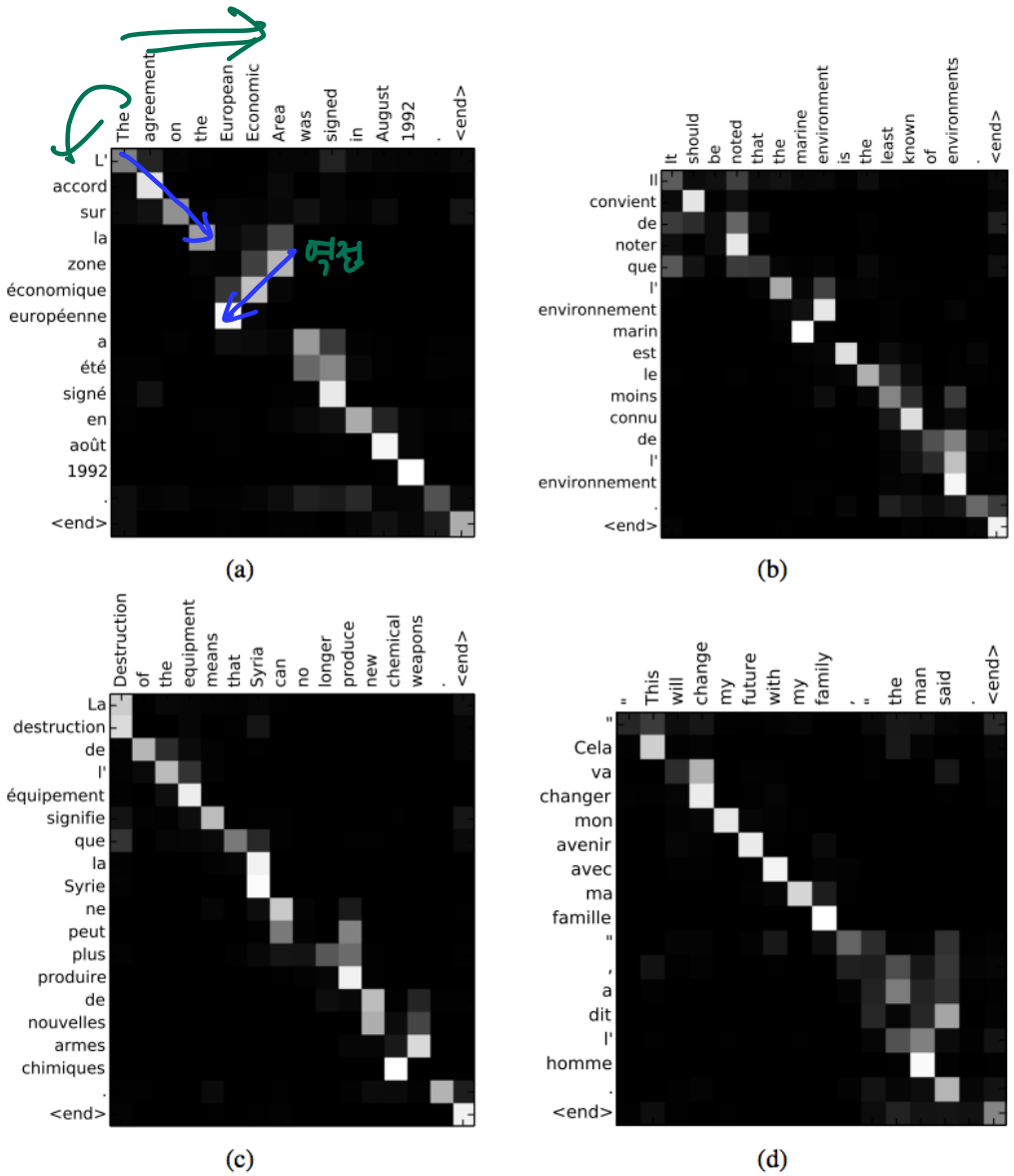

Attention output
money
$\hat{y}_6$
지름길.
Attention distribution
Attention scores
Encoder RNN
Decoder RNN
→ Attention 아이디어 back
les
pauvres
sont
démunis
<START>
the
poor
don't
have
any

Attention is Great!

- Attention significantly improves NMT performance
  - It is effective to allow the decoder to focus on particular part of the source sequence
- Attention solves the bottleneck problem
  - Attention allows the decoder to look directly at the source sequence, addressing the bottleneck problem
- Attention helps with vanishing gradient problem
  - Provides a shortcut to faraway states

Attention Examples in Machine Translation

- Attention provides interpretability
  - By inspecting attention distribution, we can see what the decoder was focusing on
  - We get alignment for for free, even though we never explicitly trained an alignment system
  - The network just learned alignment by itself
  - For example, it properly learns grammatical orders of words, and sometimes it skips unnecessary words such as an article



Advanced Attention Techniques

- Gating (using sigmoid instead of softmax)
  - Squeeze-and-excitation networks, gated convolutional networks
- Self-attention (serving a general-purpose sequence or set encoder)
  - Three important concepts: query, key, and value
  - Transformer, BERT, and their variants