

Lecture 12

RNN

Sung Kim <hunkim+mr@gmail.com>
<http://hunkim.github.io/ml/>

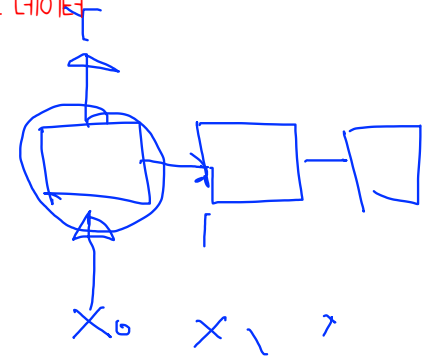
Sequence data



하나의 단어만 이해한다고 이해하는것 아니다

전체적인 단어 맥락을 이해해야한다 시퀀스 데이터

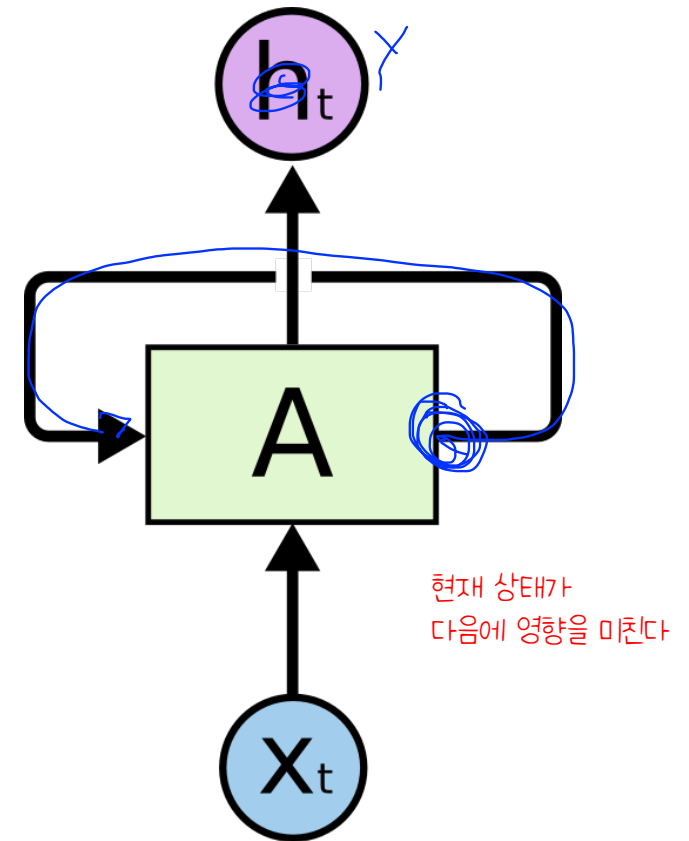
- We don't understand one word only
- We understand based on the previous words + this word. (time series)
- NN/CNN cannot do this



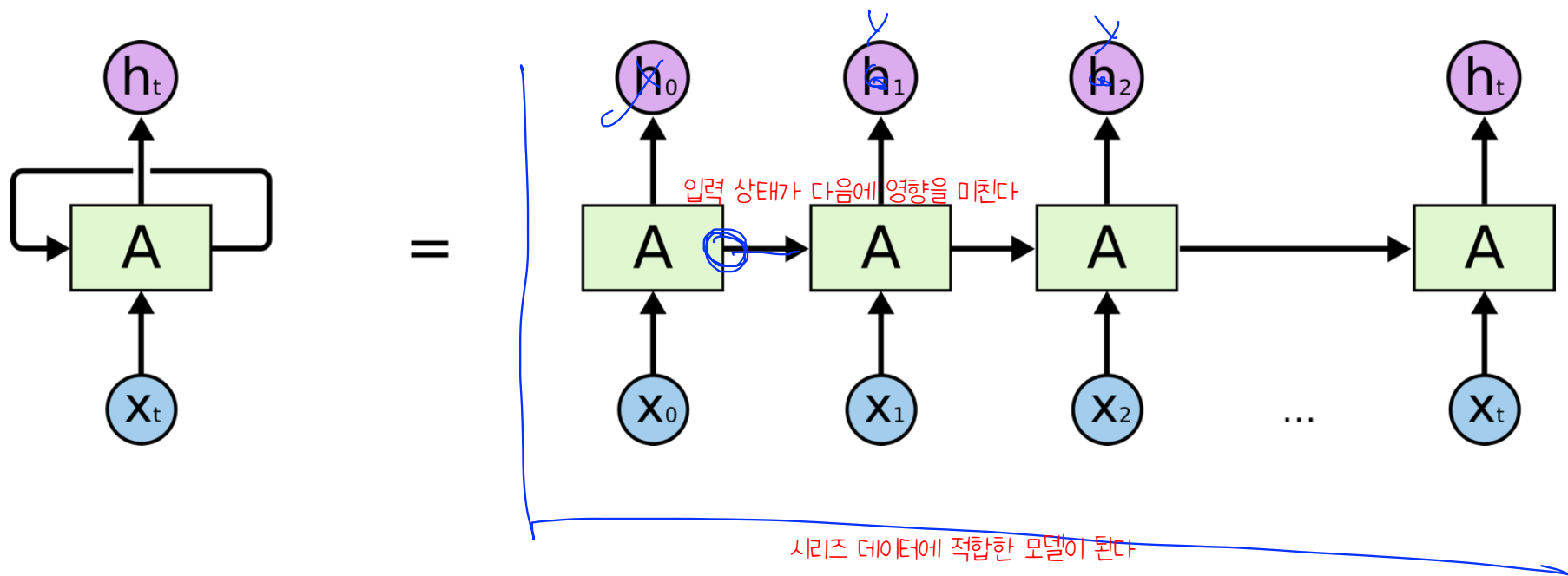
그래서 많은 사람들이 고민 끝에 만들어 냄

Sequence data

- We don't understand one word only
- We understand based on the previous words + this word. (time series)
- NN/CNN cannot do this

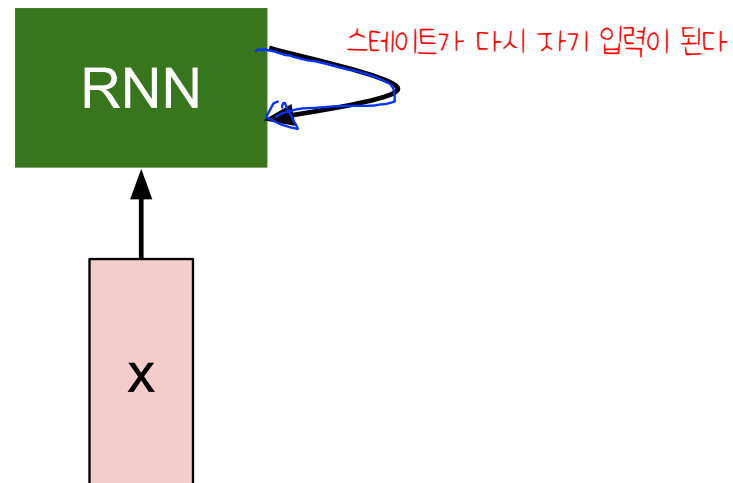


<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

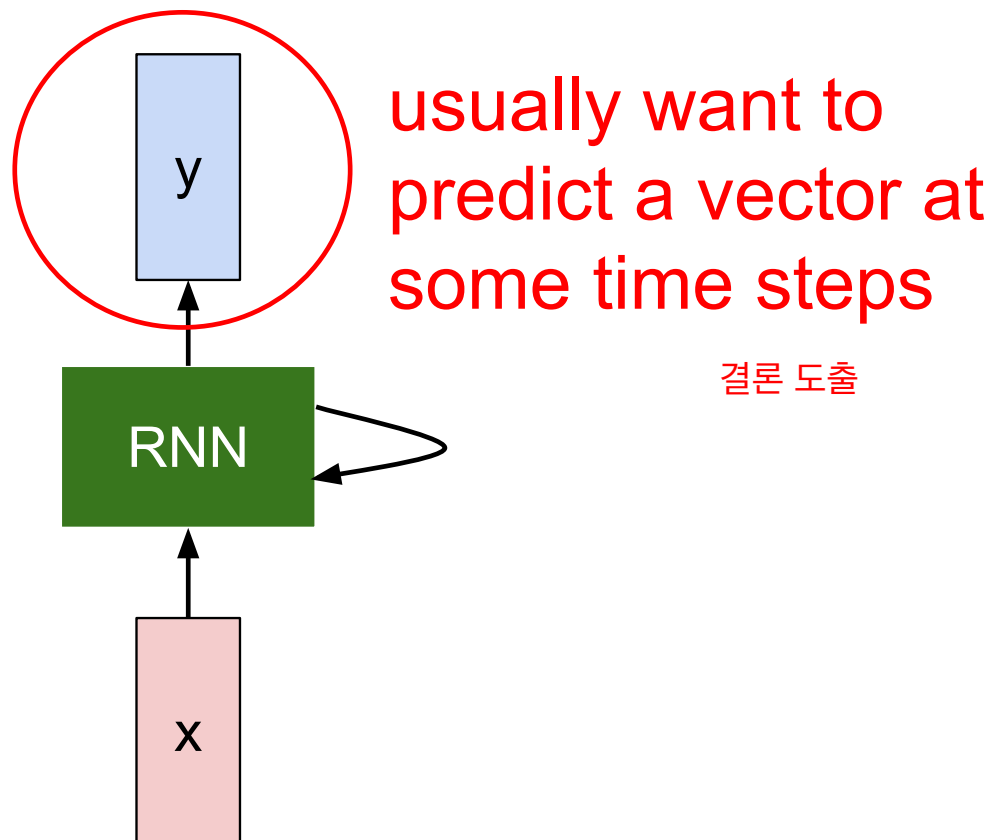


<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Recurrent Neural Network

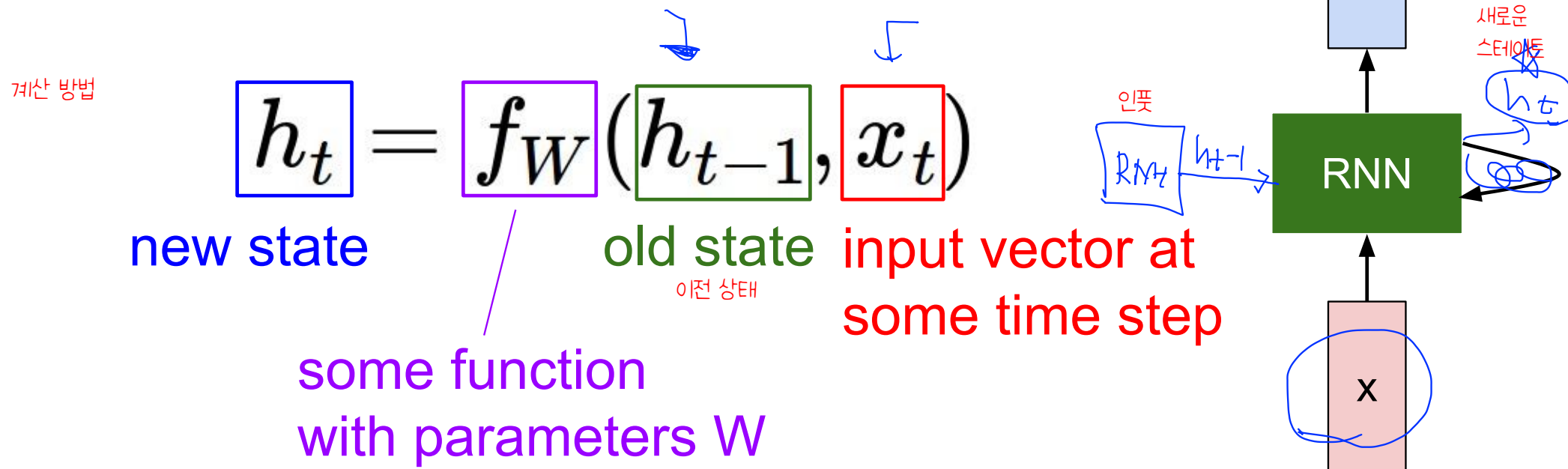


Recurrent Neural Network



Recurrent Neural Network

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



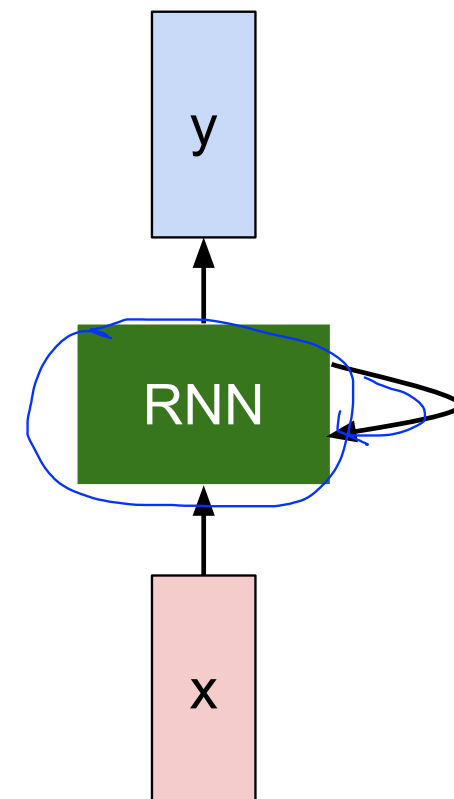
Recurrent Neural Network

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

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

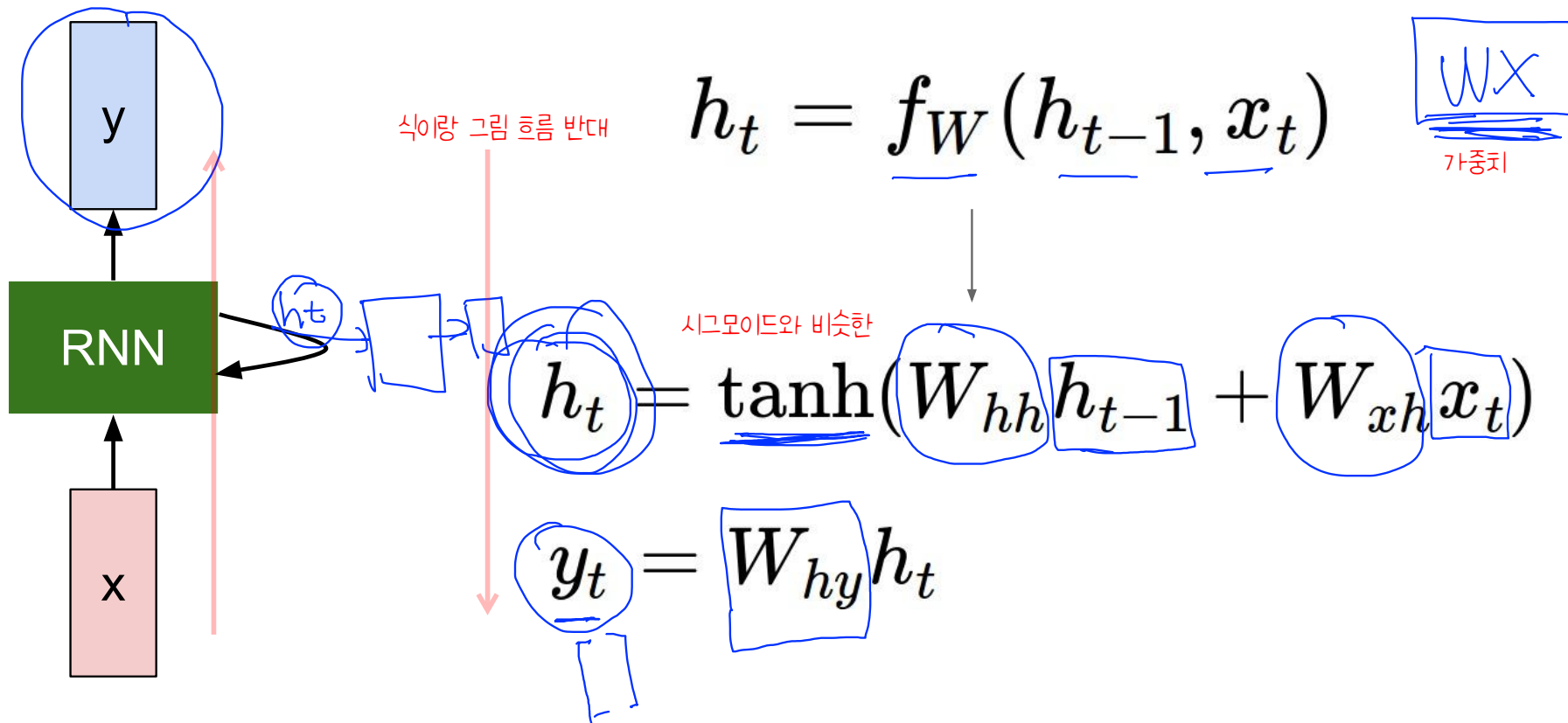
Diagram illustrating the recurrence formula with handwritten annotations: a blue arrow points from the text "이전스태이트" (previous state) to h_{t-1} , and another blue arrow points from the text "인풋" (input) to x_t . To the right of the formula, there are three blue-outlined boxes connected by horizontal lines, representing a sequence of hidden states.

Notice: the same function and the same set of parameters are used at every time step.

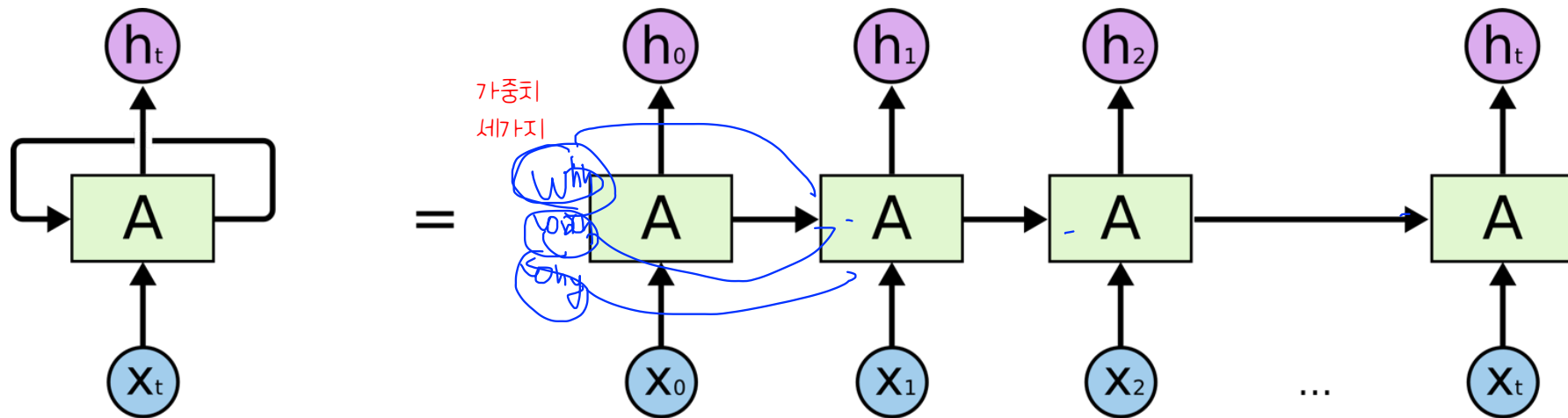


(Vanilla) Recurrent Neural Network

The state consists of a single “hidden” vector h :



가중치는 모두 같은 것으로 학습한다

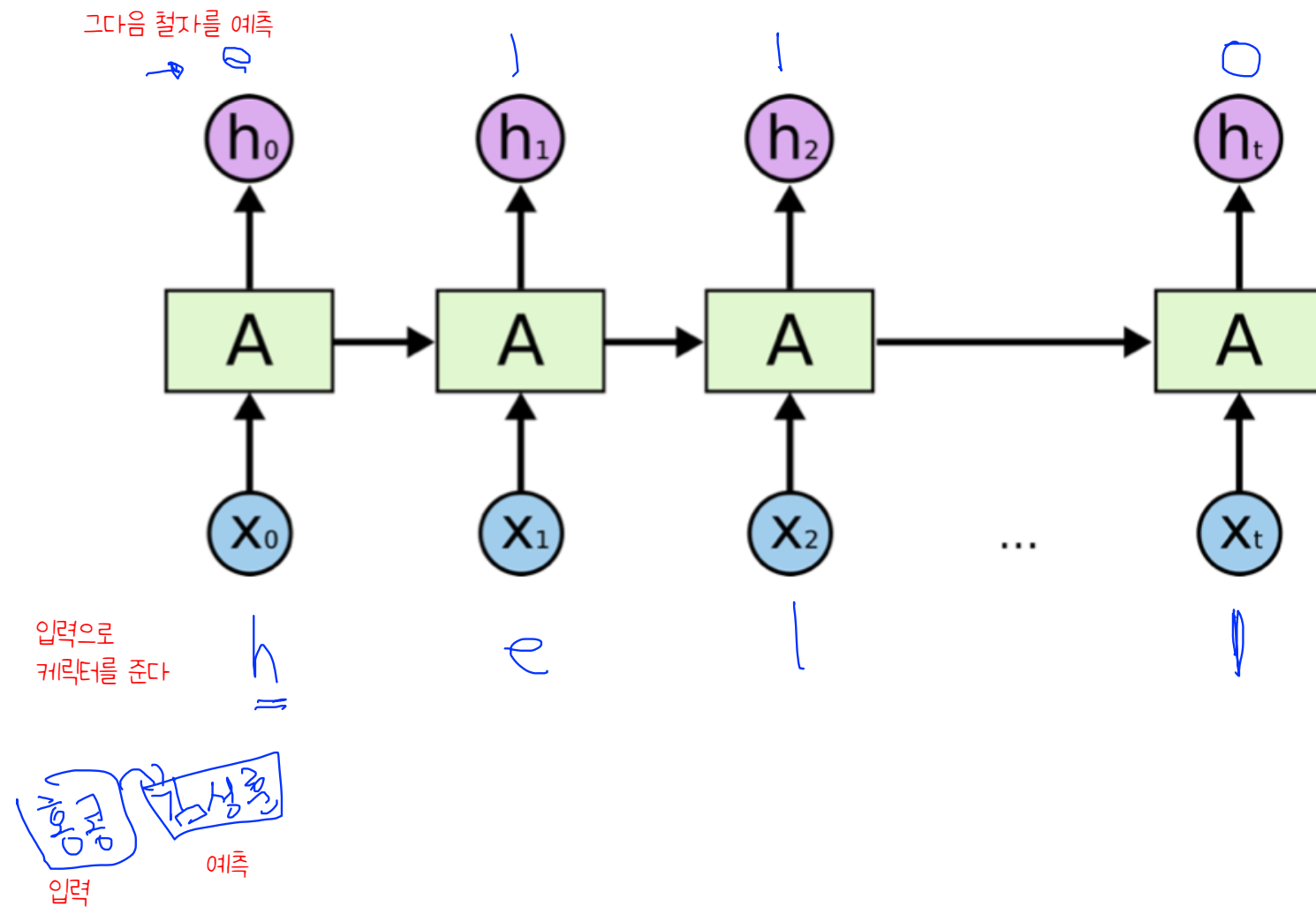


Notice: the same function and the same set of parameters are used at every time step.

Character-level language model example

Vocabulary:
[h,e,l,o]

Example training
sequence:
“hello”

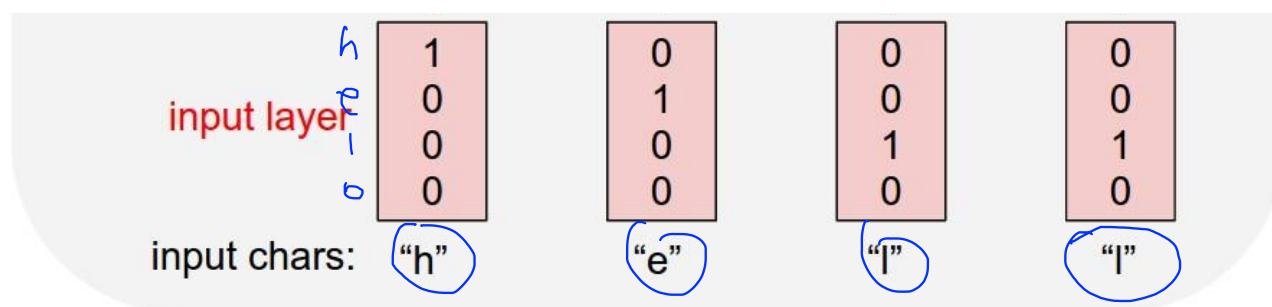


Character-level language model example

Vocabulary:
[h,e,l,o]

Example training
sequence:
“hello”

hello를 벡터로 표현해야한다 / 원 - 핫인코딩으로 표현



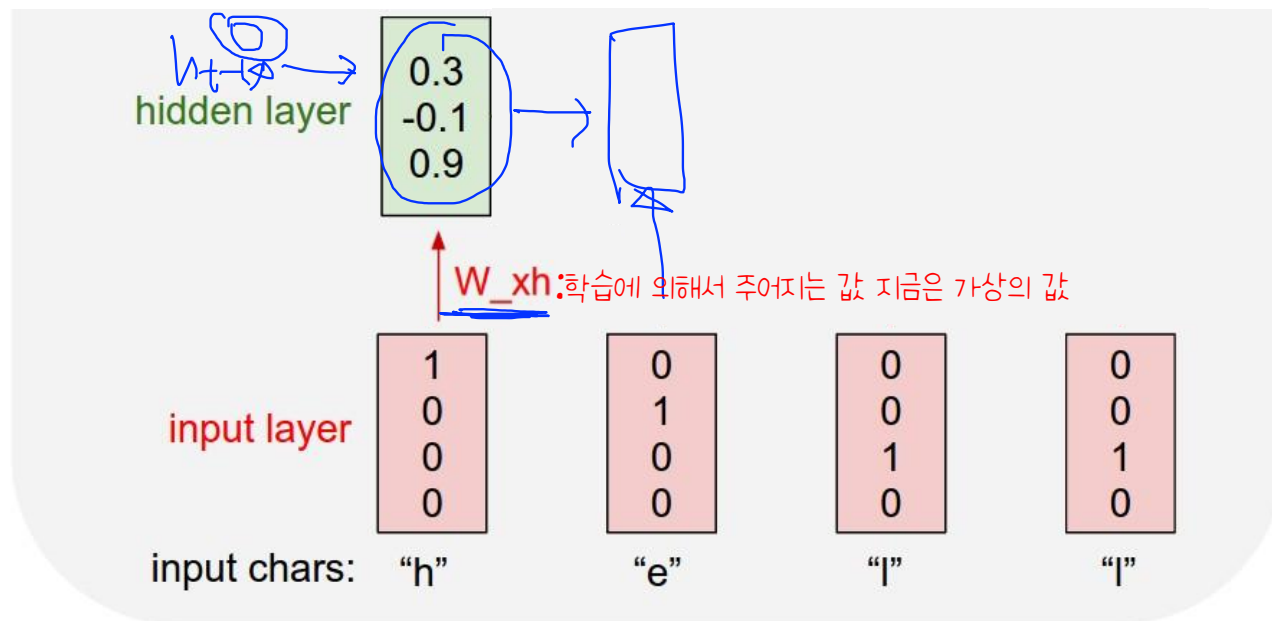
Character-level language model example

Vocabulary:
[h,e,l,o]

Example training
sequence:
“hello”

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

초기 항에는 0

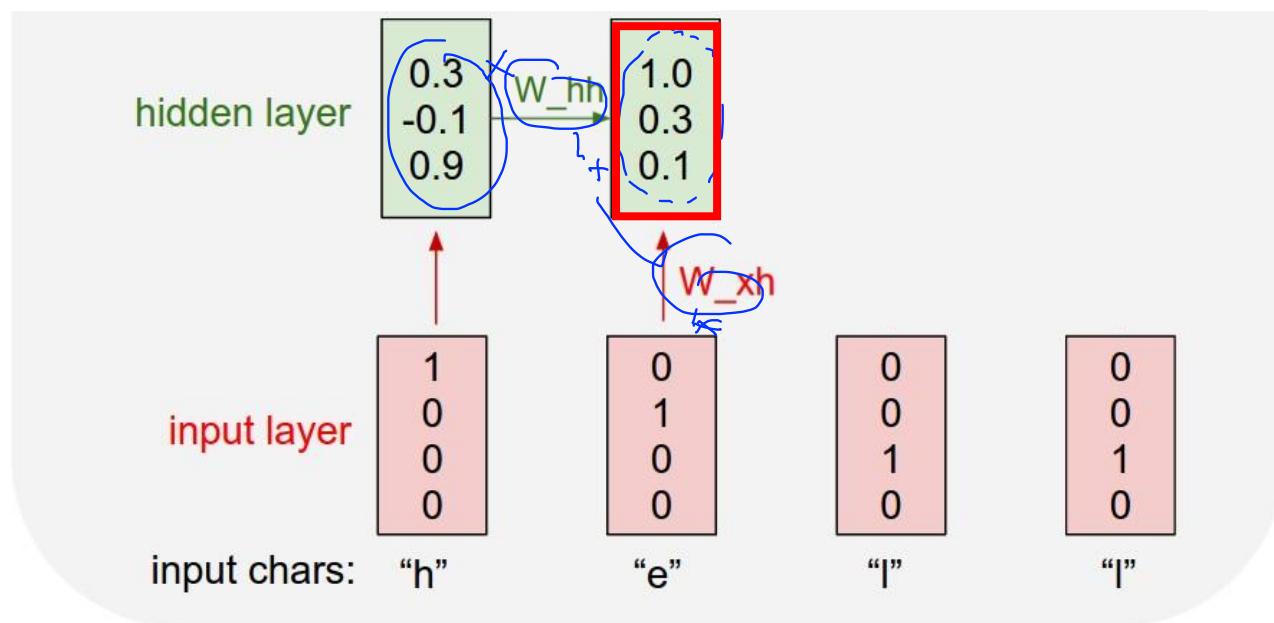


Character-level language model example

Vocabulary:
[h,e,l,o]

Example training
sequence:
“hello”

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

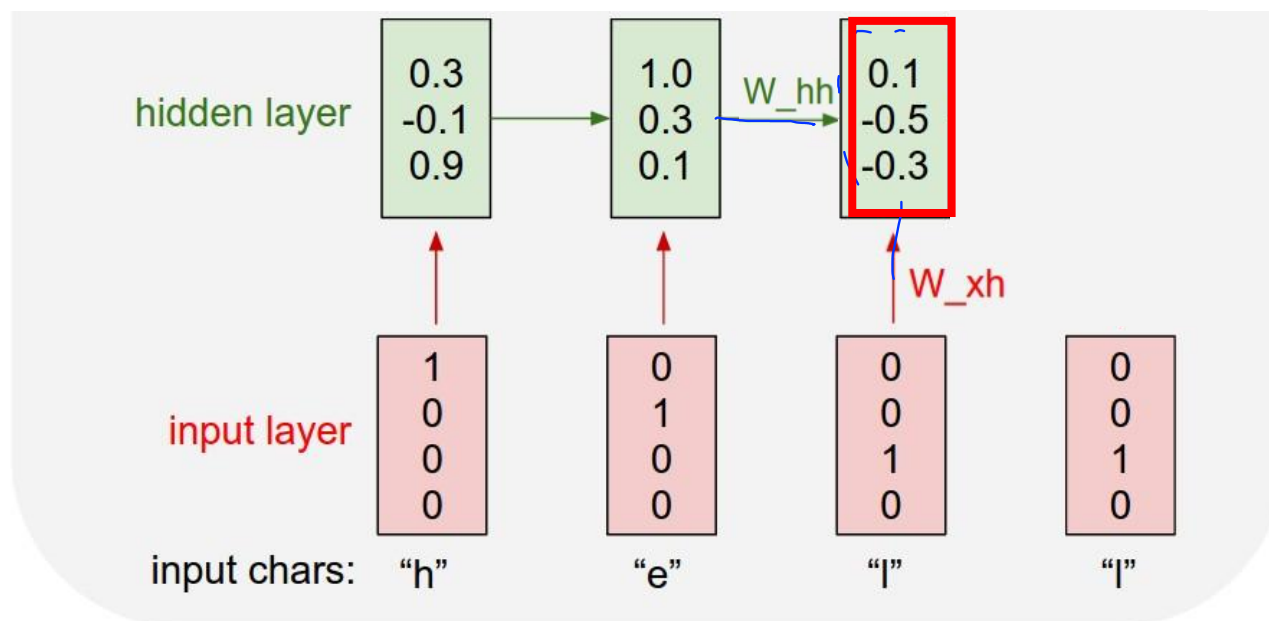


Character-level language model example

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

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

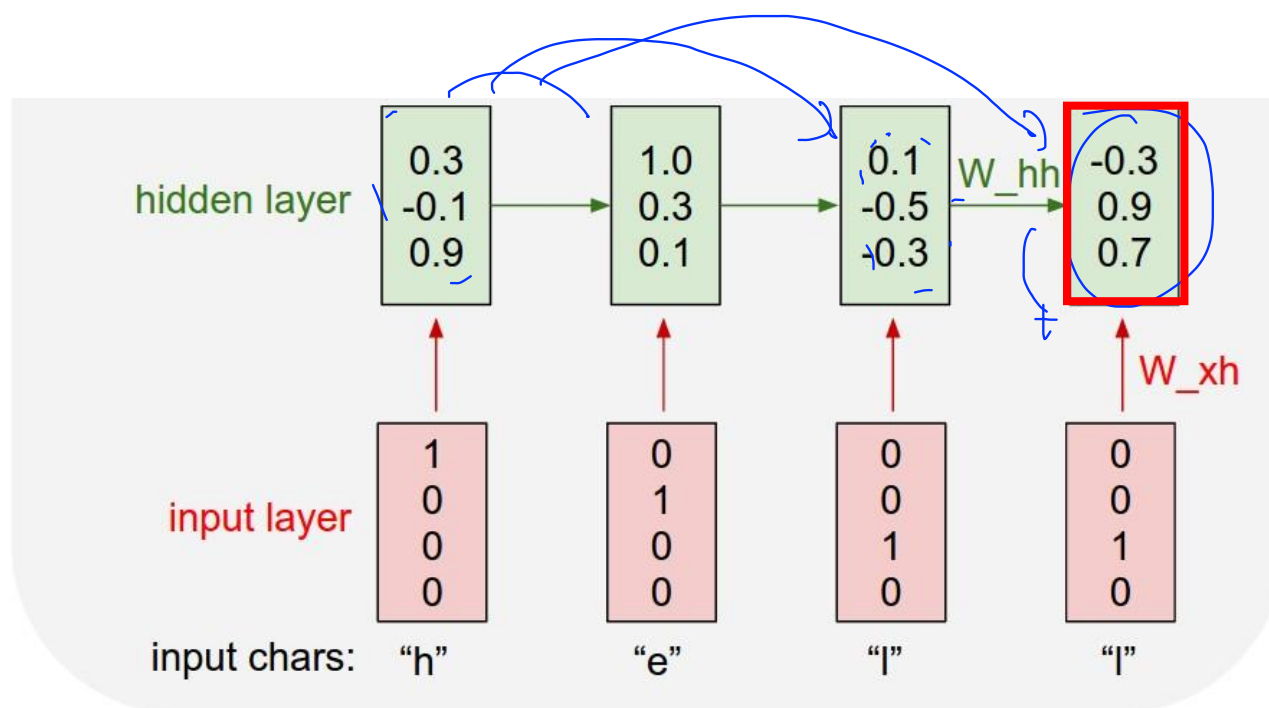


Character-level language model example

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

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

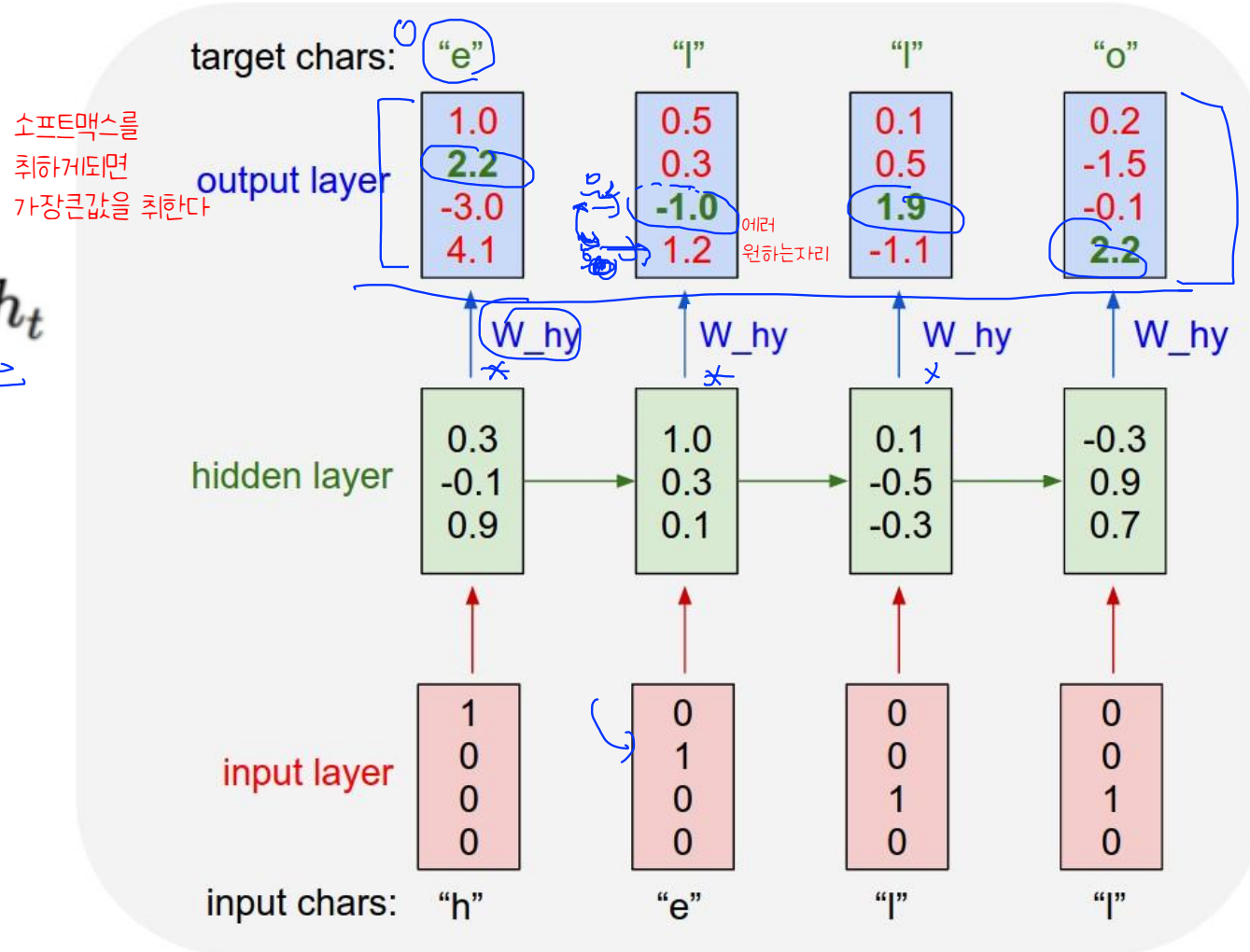


Character-level language model example

$$y_t = W_{hy} h_t$$

Vocabulary:
[h,e,l,o]

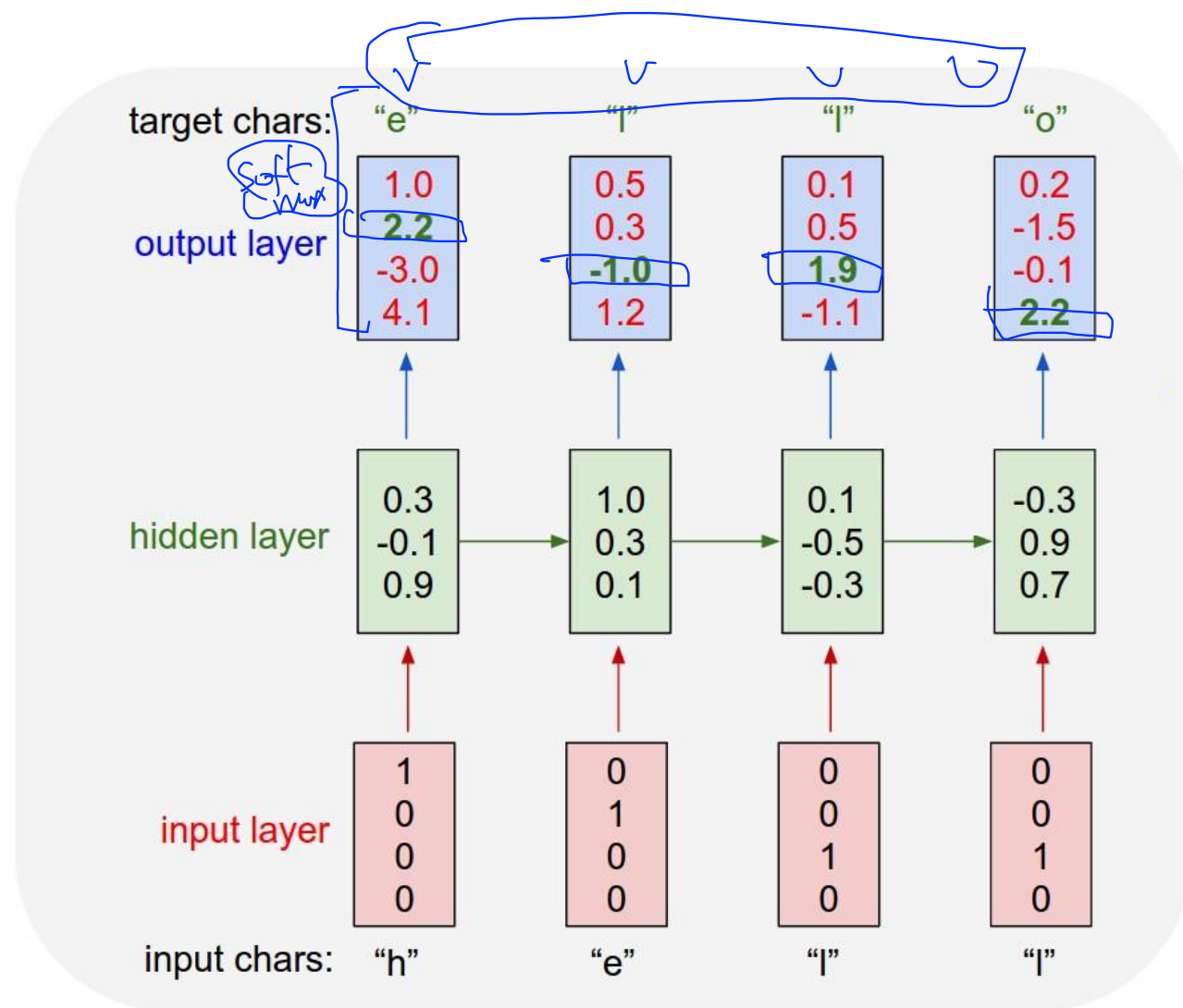
Example training
sequence:
“hello”



Character-level language model example

Vocabulary:
[h,e,l,o]

Example training
sequence:
“hello”



RNN applications

https://github.com/TensorFlowKR/awesome_tensorflow_implementations

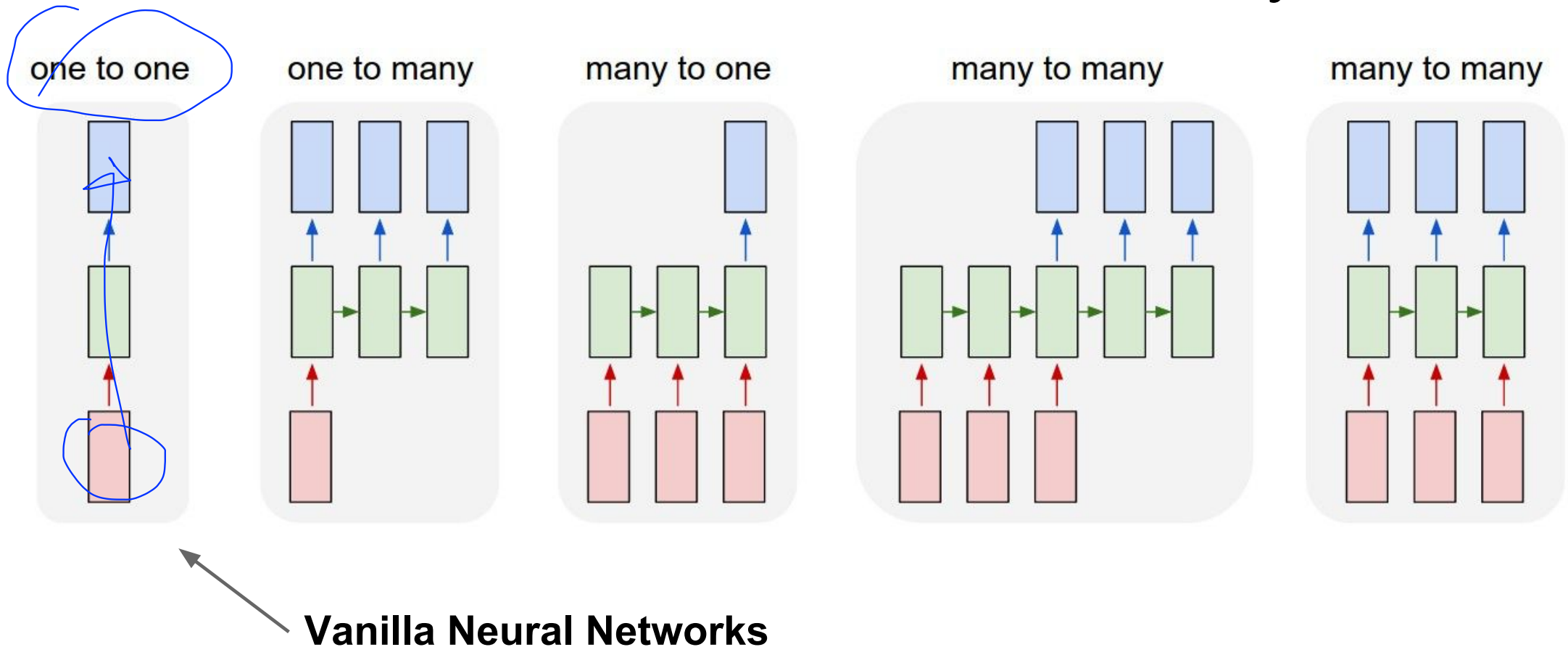
- Language Modeling
- Speech Recognition 이전 했던 말과 관련이 있기에
- Machine Translation 번역 기계를 만들 수 있음
- Conversation Modeling/Question Answering
- Image/Video Captioning 이미지 비디오 캡셔닝도 가능하다
- Image/Music/Dance Generation



RNN 활용 용도가 무궁무진하다

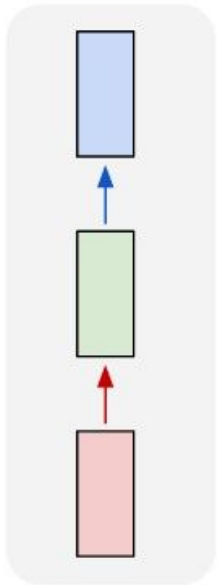
<http://jiwonkim.org/awesome-rnn/>

Recurrent Networks offer a lot of flexibility:

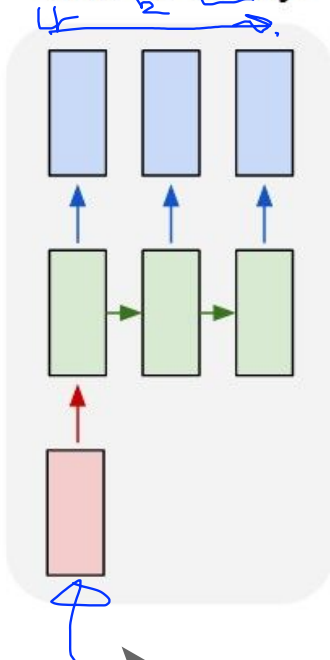


Recurrent Networks offer a lot of flexibility:

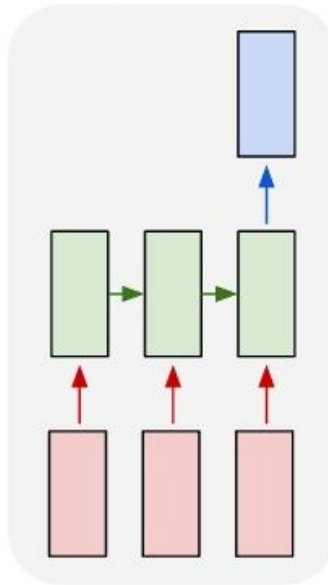
one to one



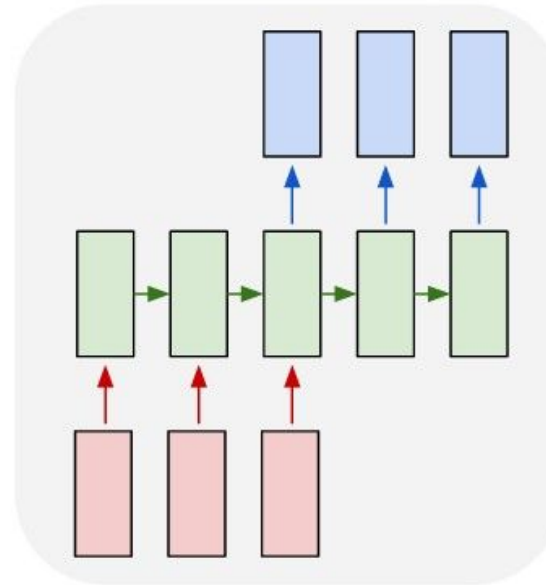
one to many



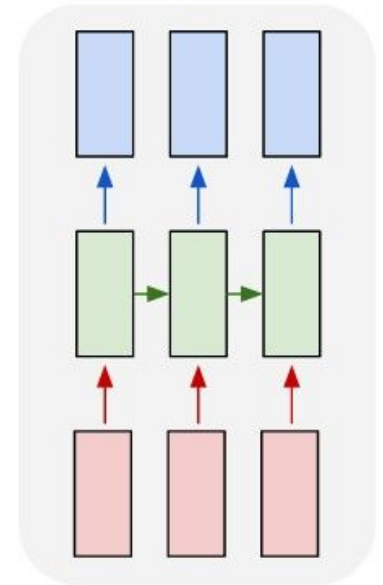
many to one



many to many



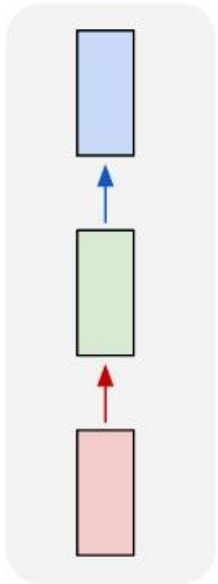
many to many



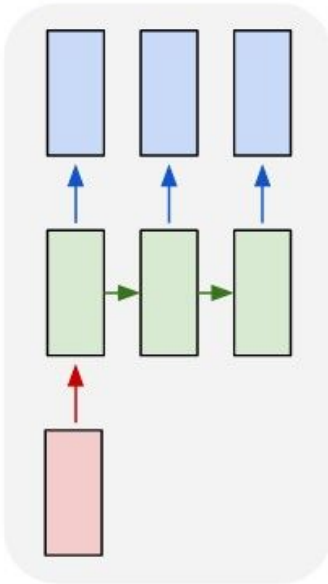
e.g. **Image Captioning**
image -> sequence of words

Recurrent Networks offer a lot of flexibility:

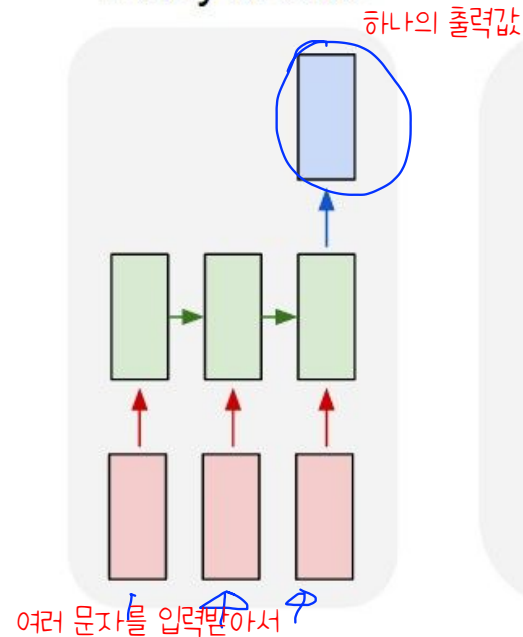
one to one



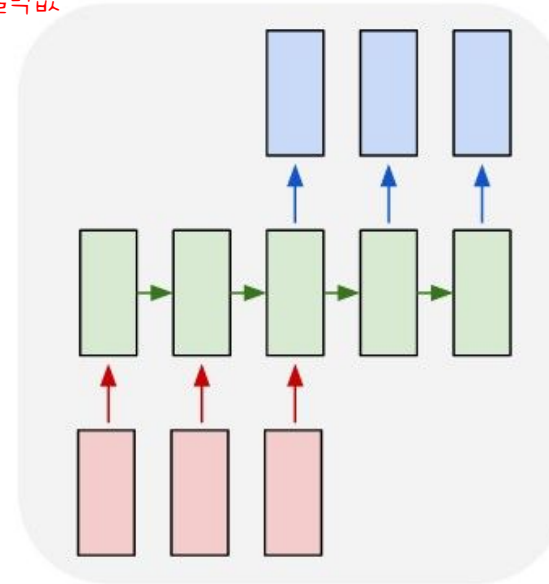
one to many



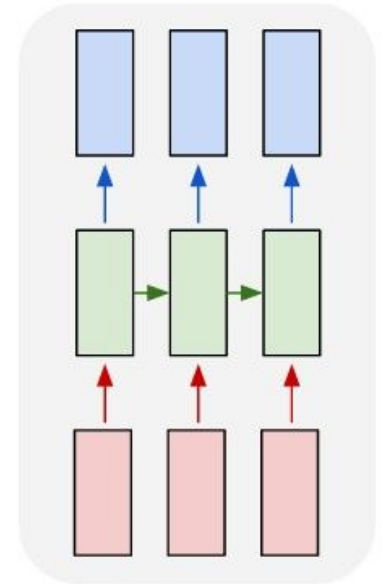
many to one



many to many



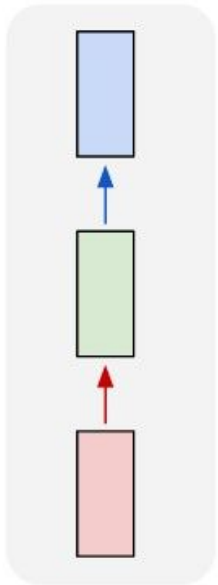
many to many



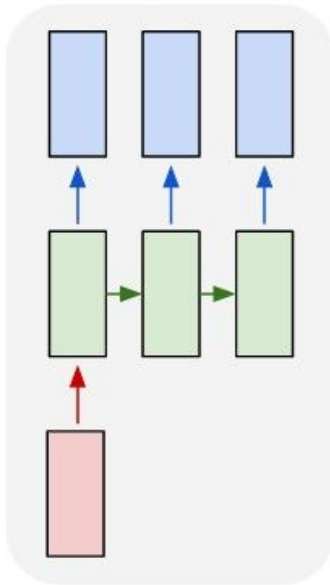
e.g. **Sentiment Classification**
sequence of words -> sentiment

Recurrent Networks offer a lot of flexibility:

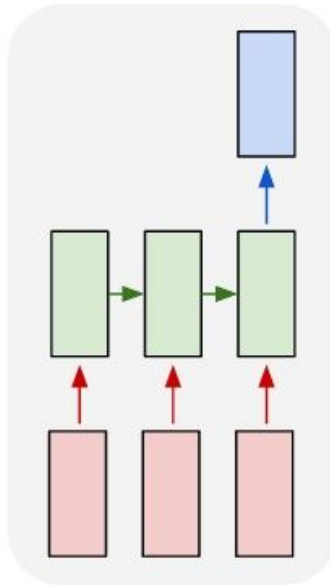
one to one



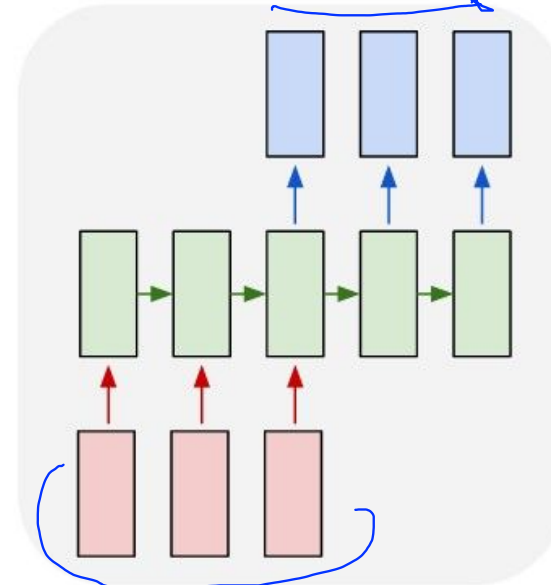
one to many



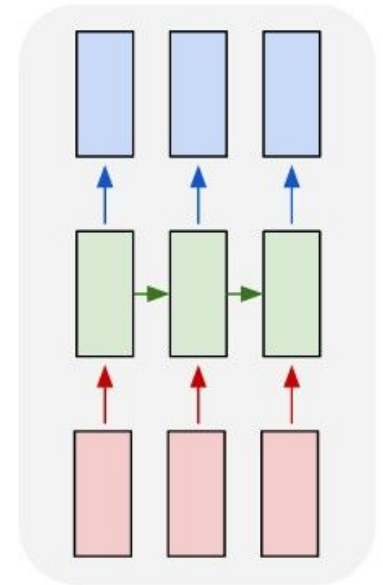
many to one



many to many



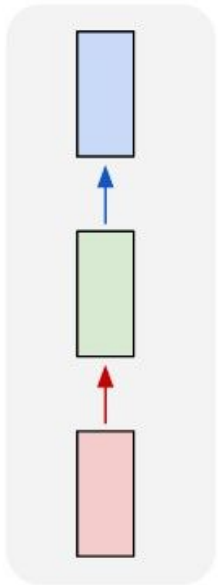
many to many



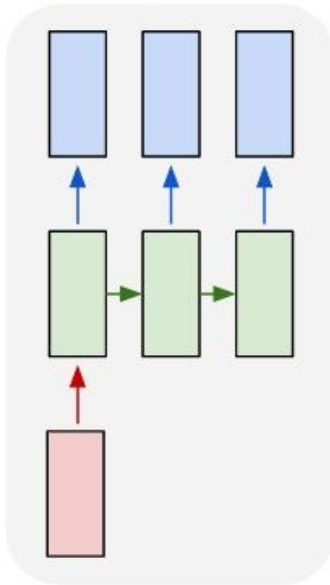
e.g. **Machine Translation**
seq of words -> seq of words

Recurrent Networks offer a lot of flexibility:

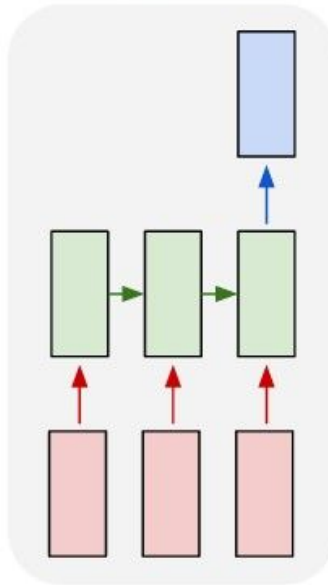
one to one



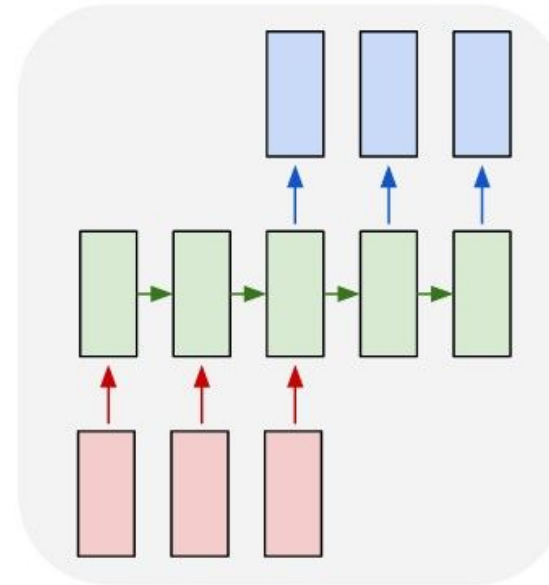
one to many



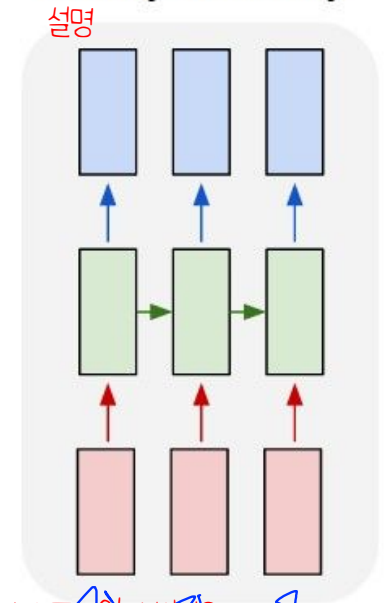
many to one



many to many



many to many

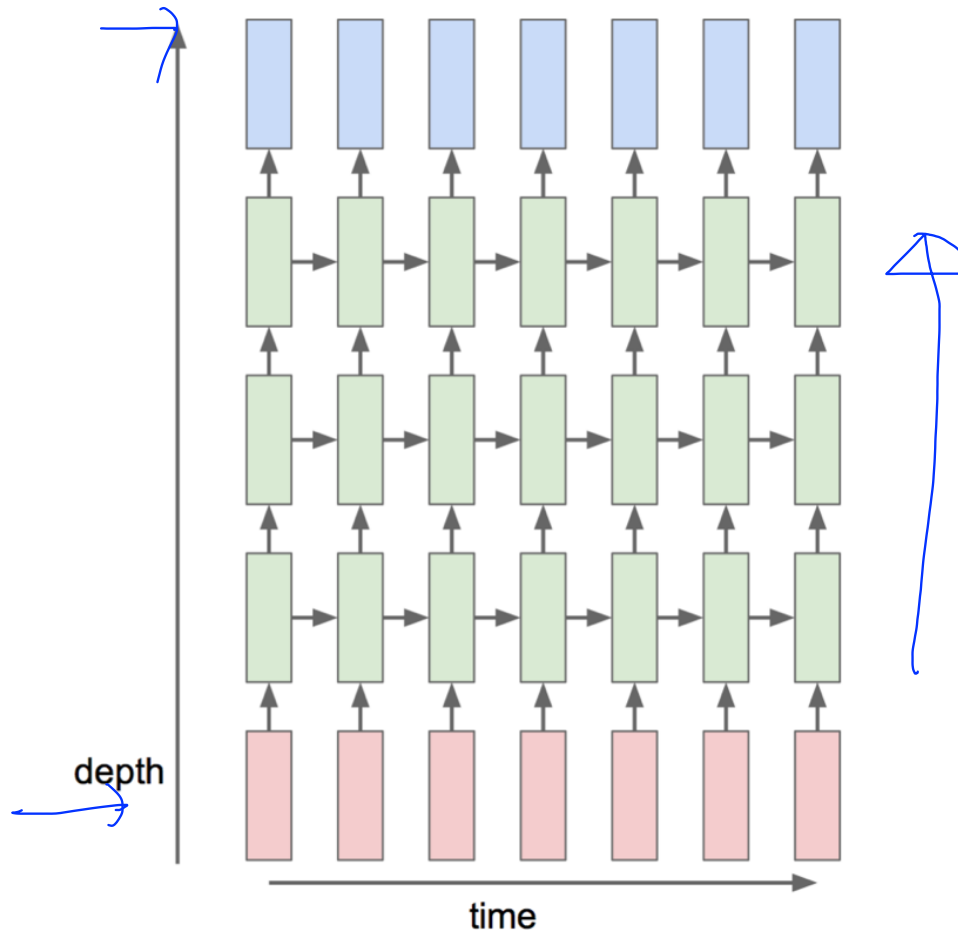


e.g. **Video classification on frame level**

Multi-Layer RNN

출력

입력



여러 레이어를 둘 수 있으며,
복잡한 학습이 가능하다

Training RNNs is challenging

- Several advanced models
 - Long Short Term Memory (LSTM)
 - GRU by Cho et al. 2014

RNN하면 대부분

LSTM GRU

둘중하나를 쓰게된다

Next
RNN in TensorFlow

