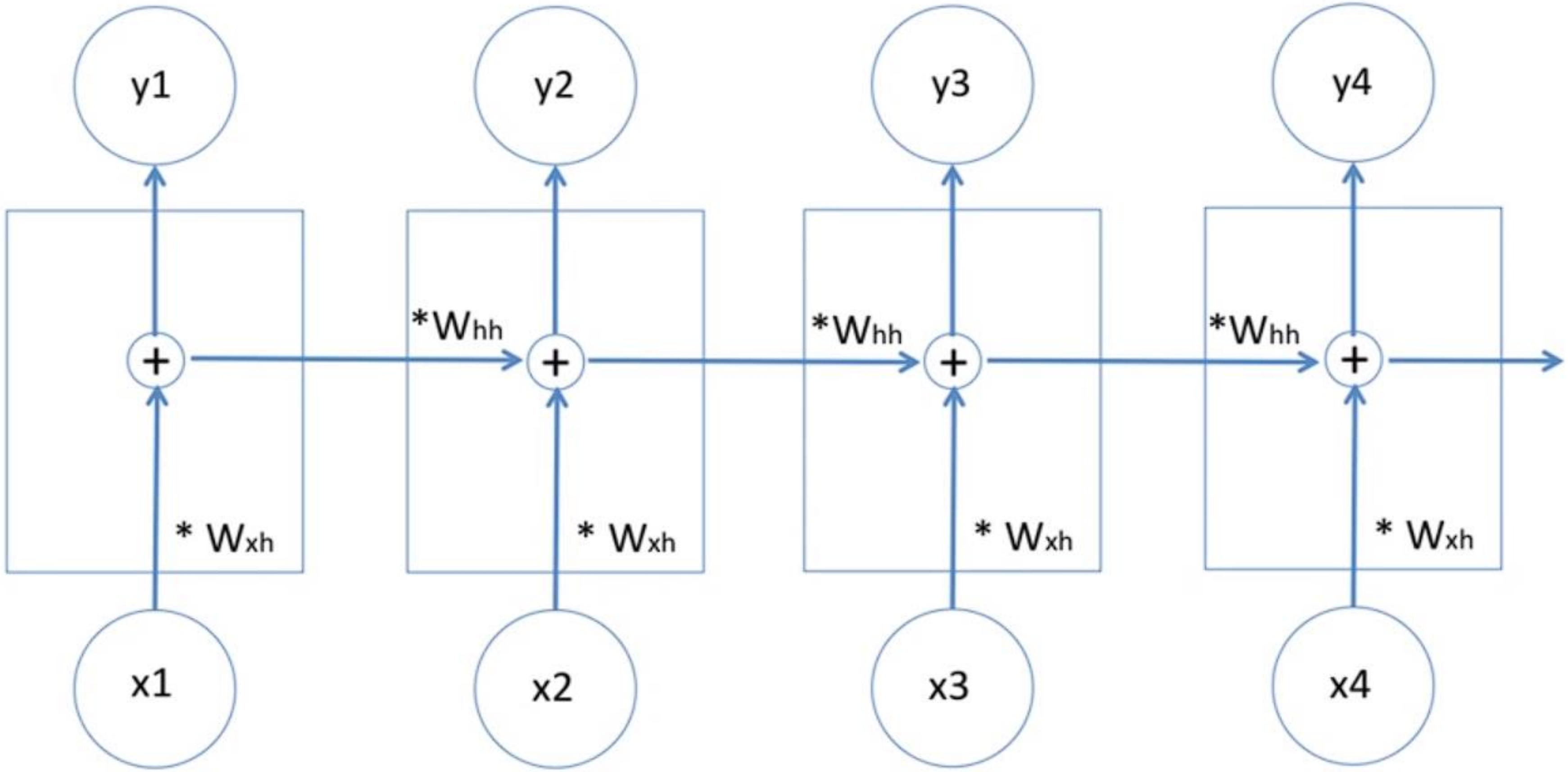
RNN

Recurrent Neural Network (RNN)

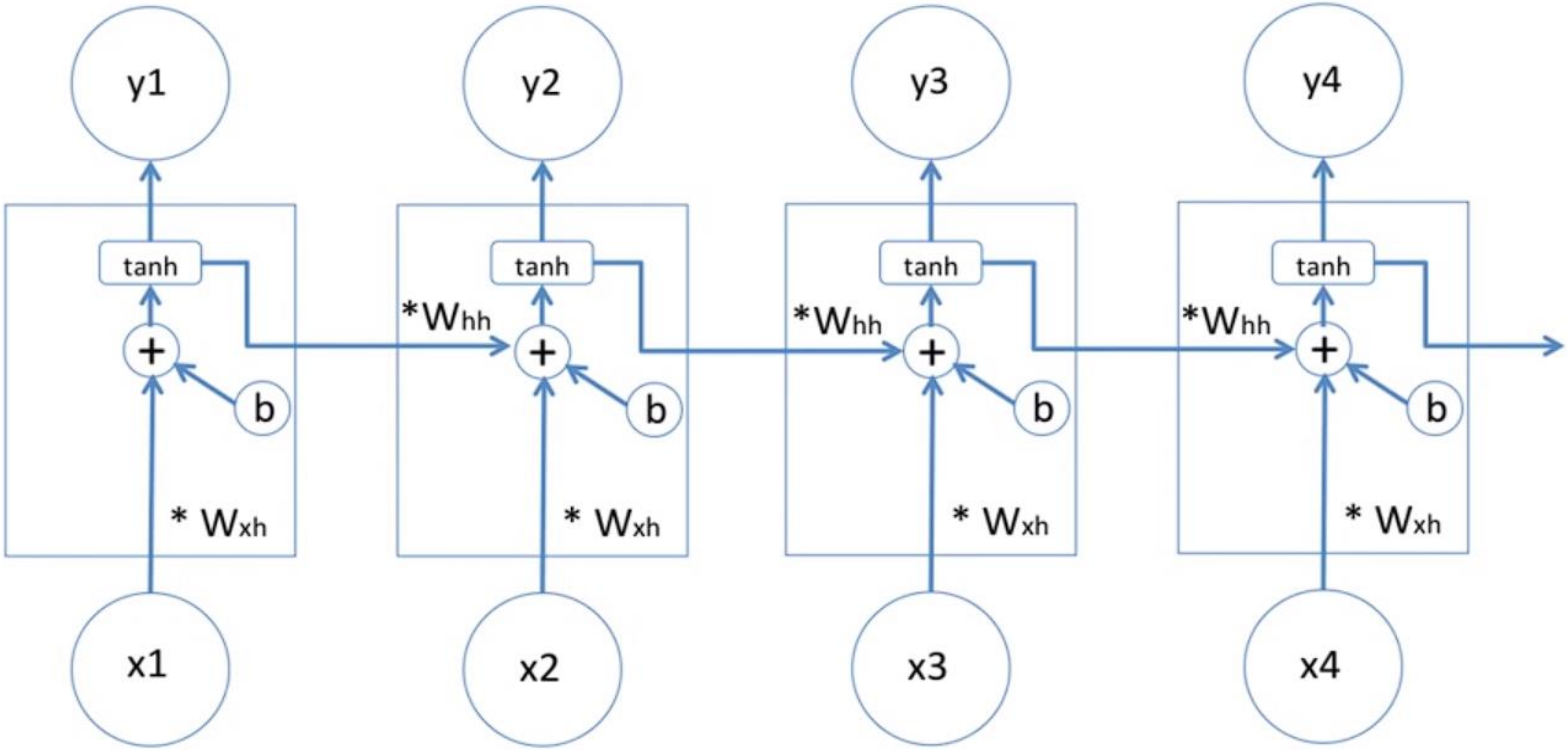
순환신경망(Recurrent Neural Network)

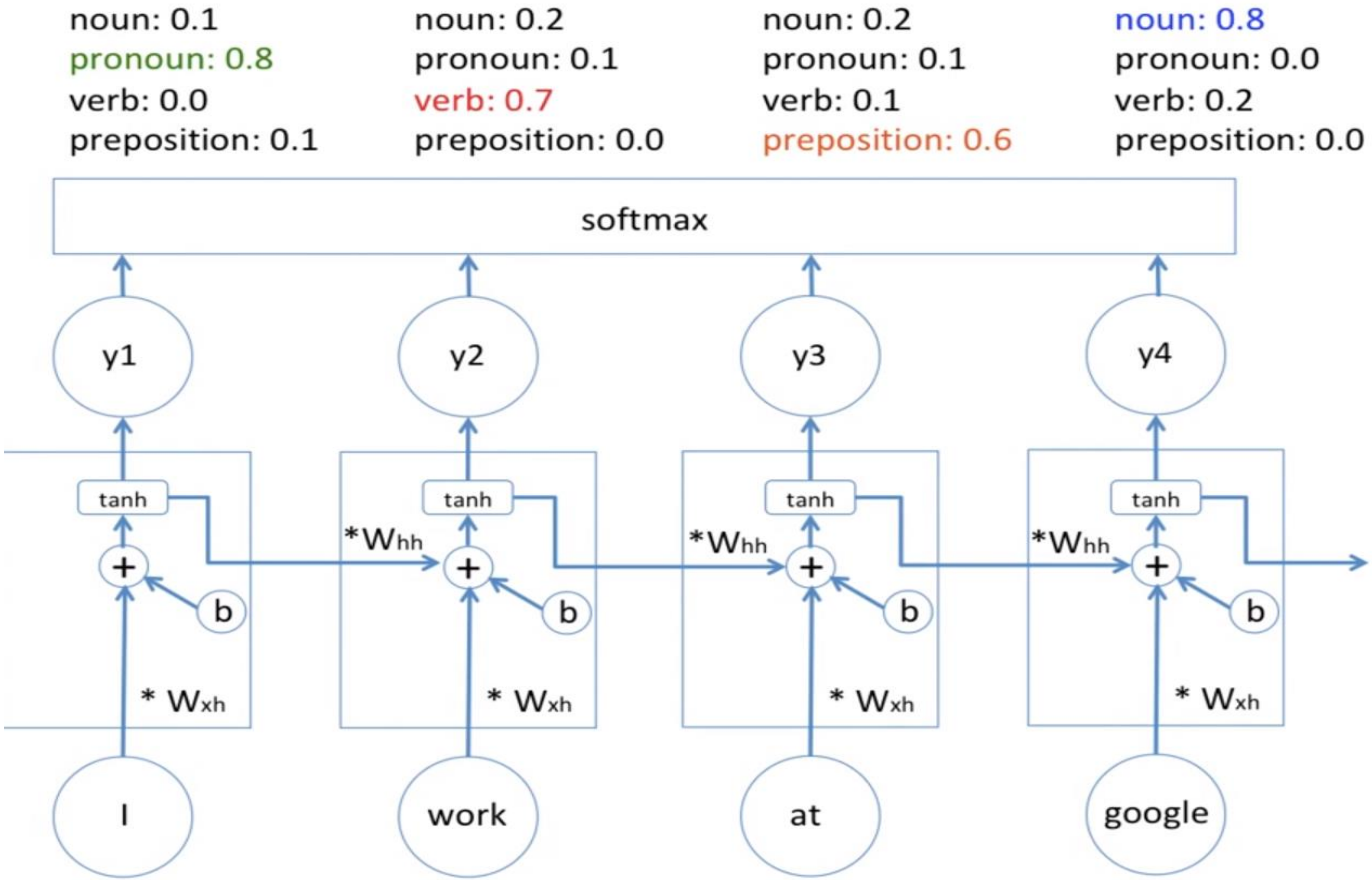
- 여러개의 데이터가 순서대로 입력되었을때 앞서 입력받은 데이터를 잠시 기억해 놓는 방법
- 모든 입력 값에 이 작업이 순서대로 이뤄지며 같은 층을 맴도는 것으로 보여 순환 신경망이라 부른다

Sequence is important for POS tagging



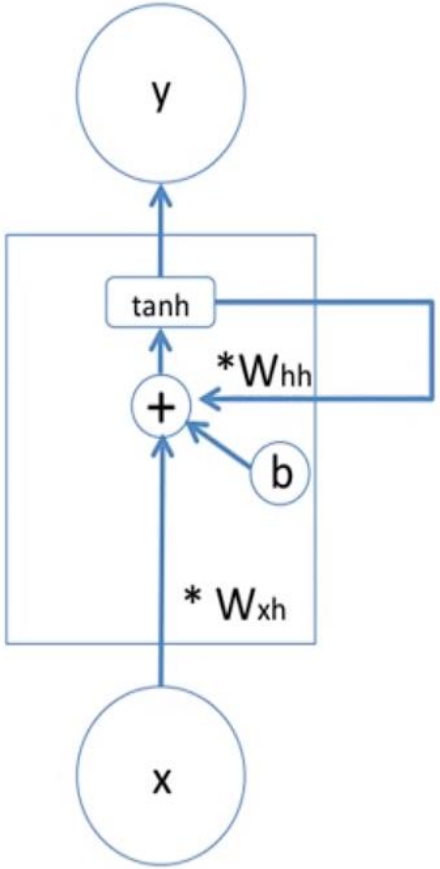
Sequence is important for POS tagging





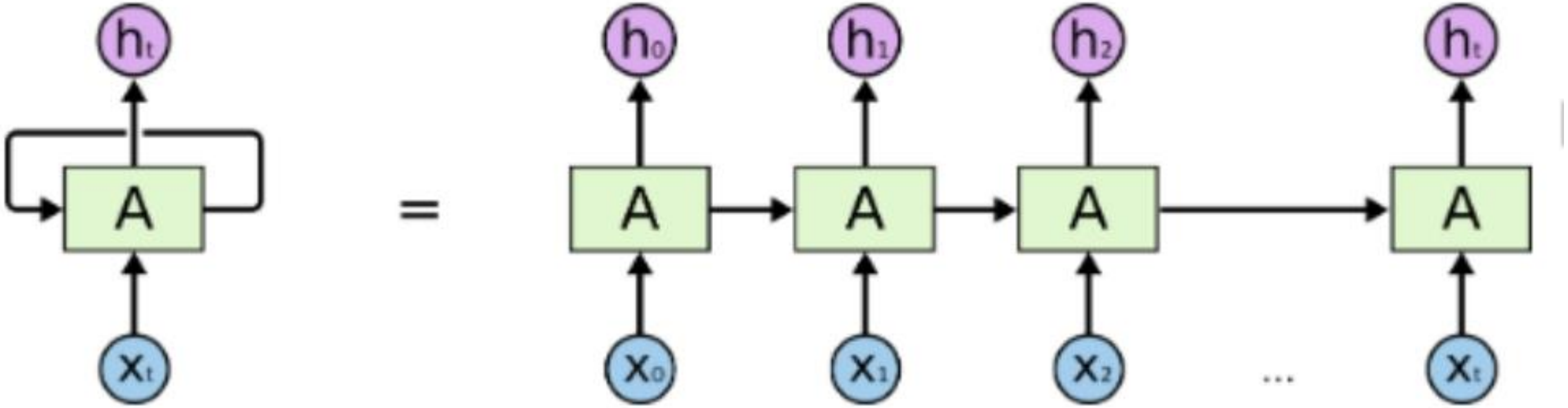
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

simplify of model diagram



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

Recurrent Neural Network



An unrolled recurrent neural network.

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

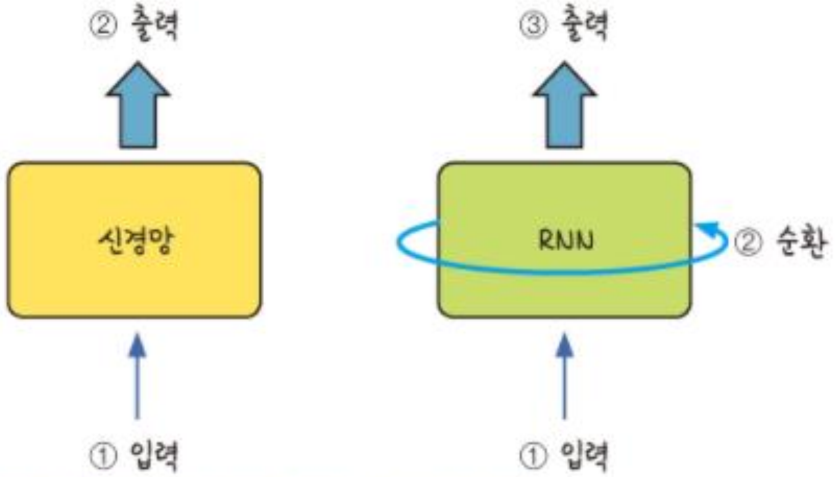


그림 17-1 일반 신경망과 순환 신경망의 차이

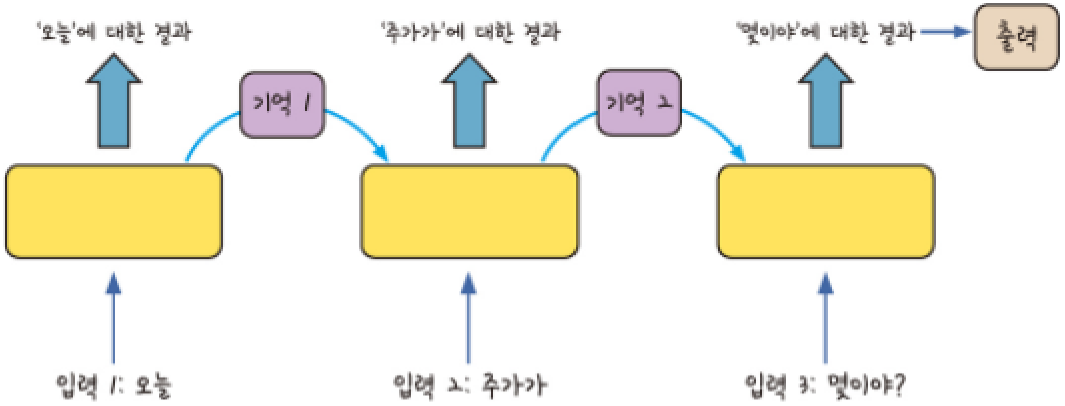
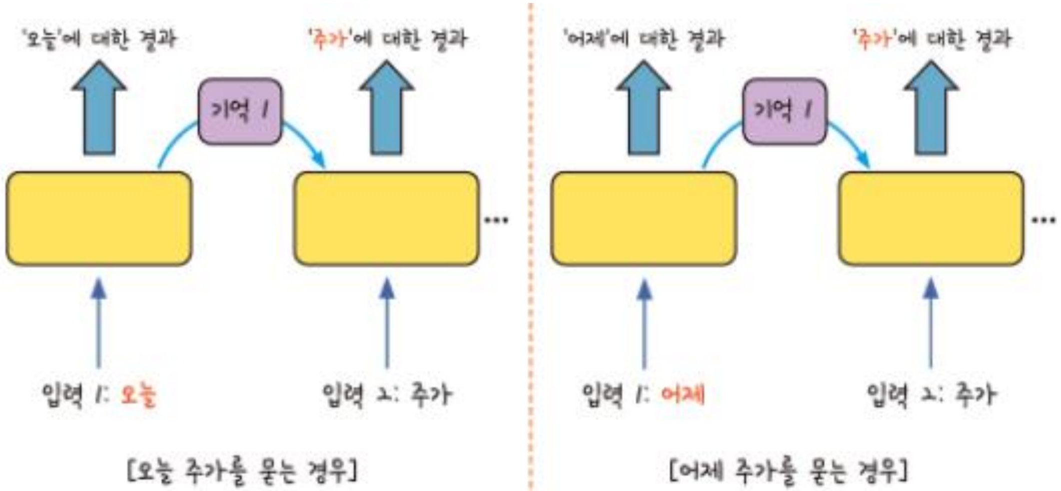


그림 17-2 "오늘 주가가 몇이야?"를 RNN이 처리하는 방식



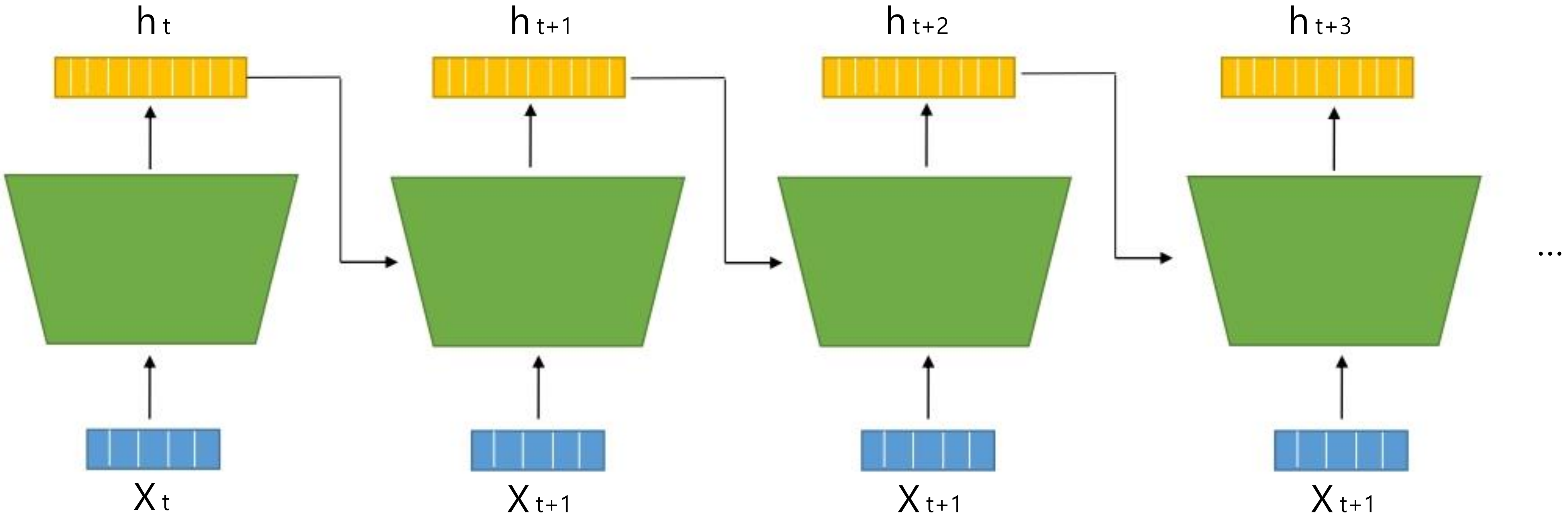
[오늘 주가를 묻는 경우]

[어제 주가를 묻는 경우]

그림 17-3 RNN을 사용하는 이유

Recurrent Neural Network

Process both new inputs and model output of previous input!



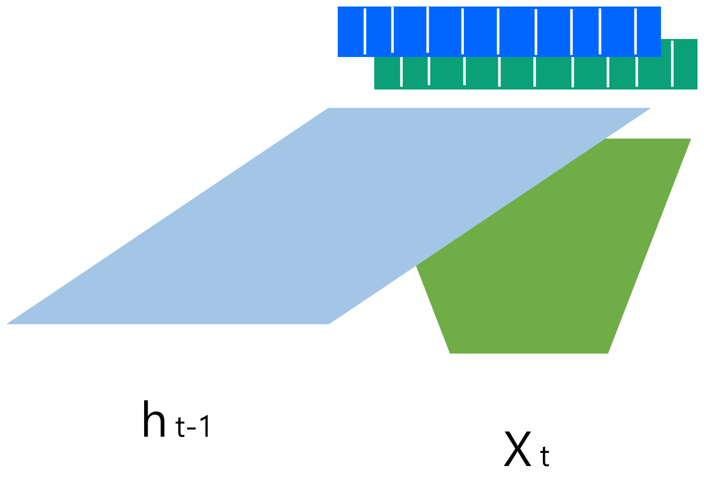
Recurrent Neural Network

But exactly, how can we combine new input and previous output?

X_t

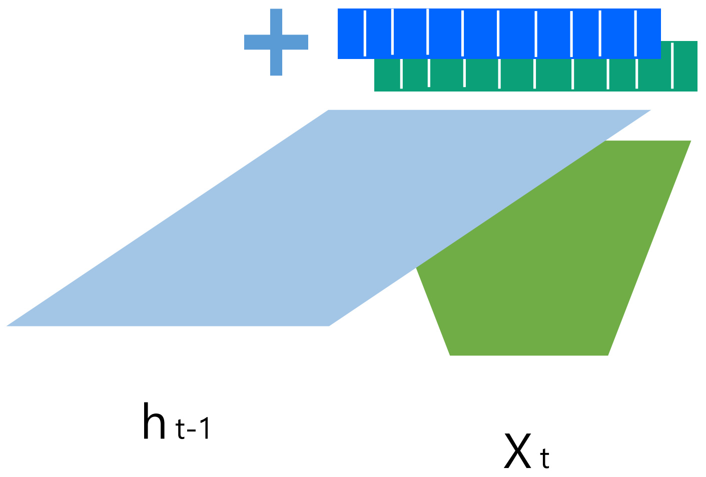
Recurrent Neural Network

But exactly, how can we combine new input and previous output?



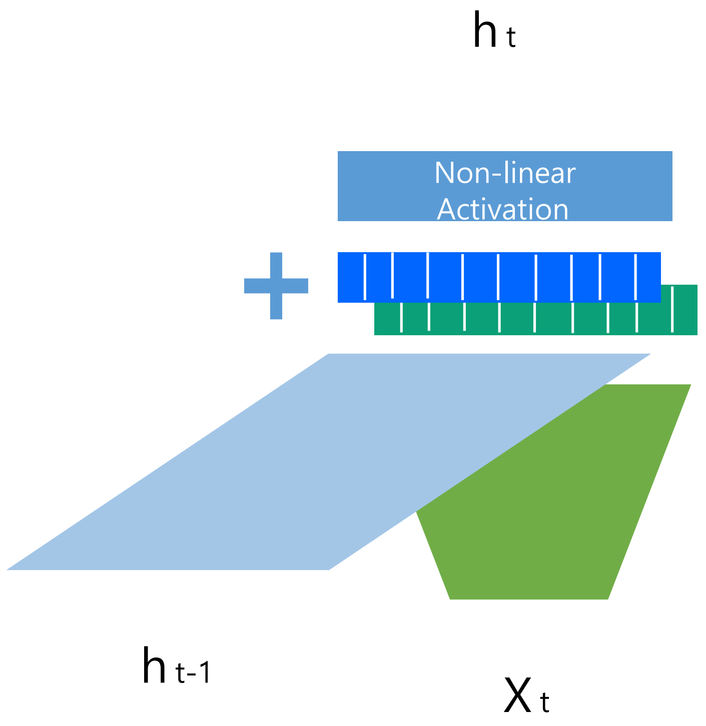
Recurrent Neural Network

But exactly, how can we combine new input and previous output?



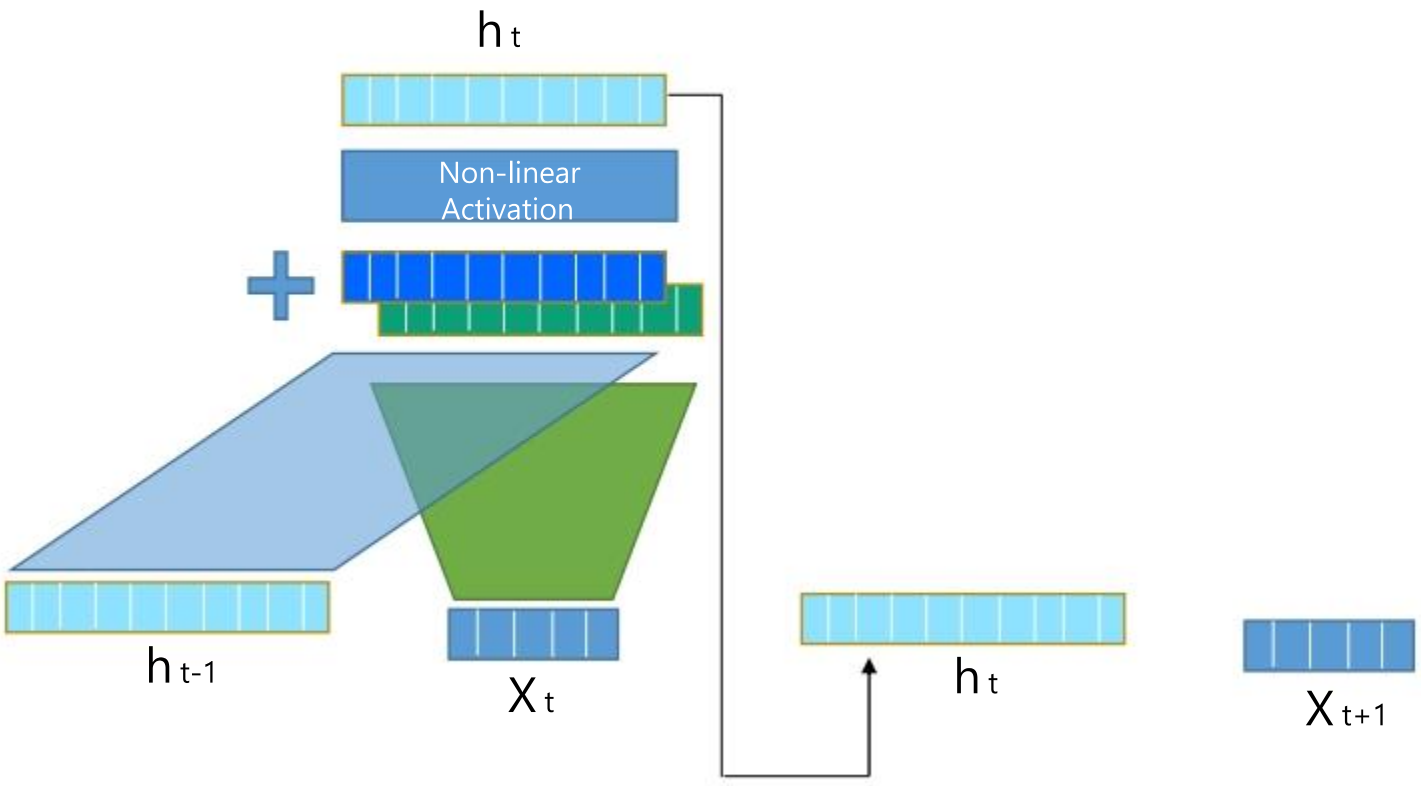
Recurrent Neural Network

But exactly, how can we combine new input and previous output?



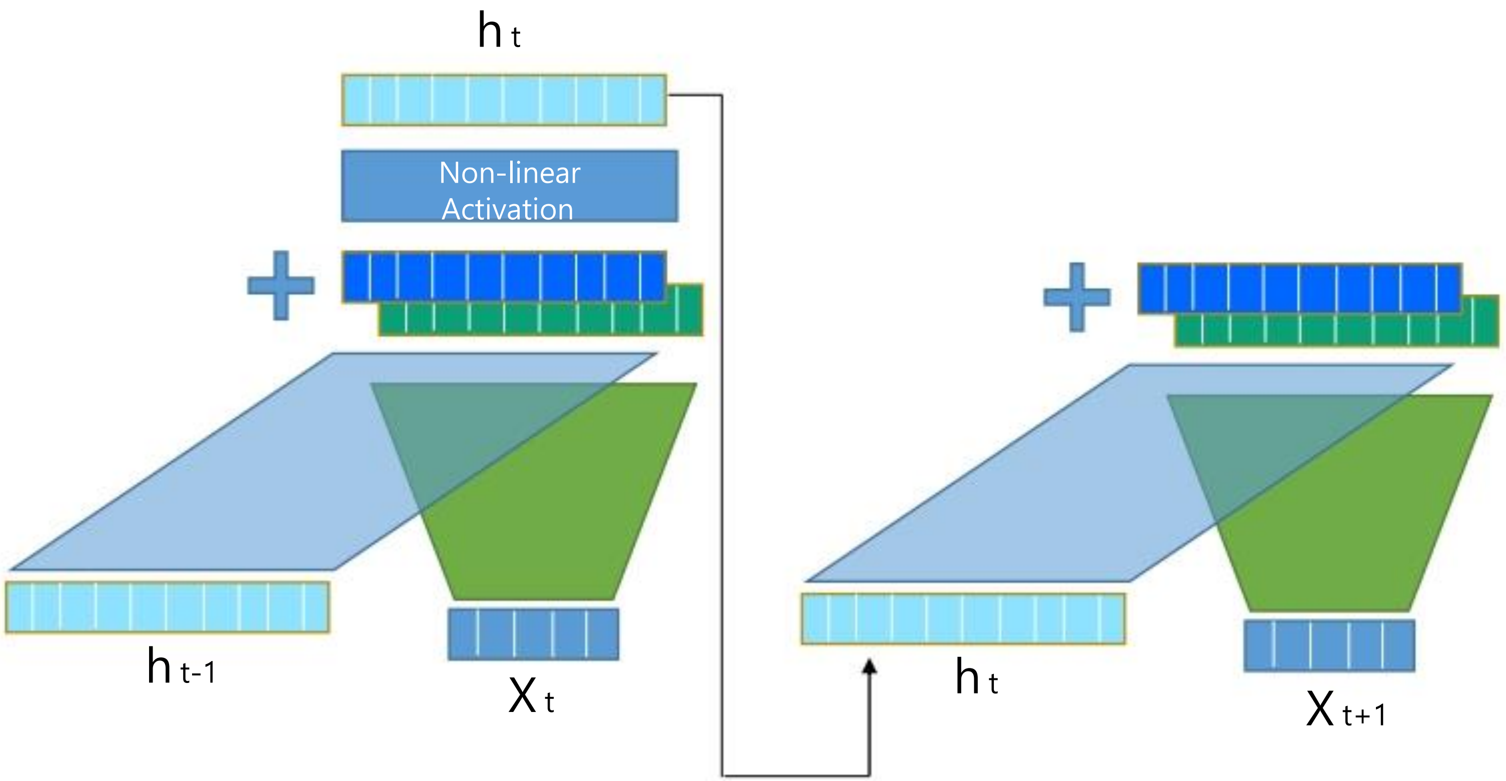
Recurrent Neural Network

But exactly, how can we combine new input and previous output?



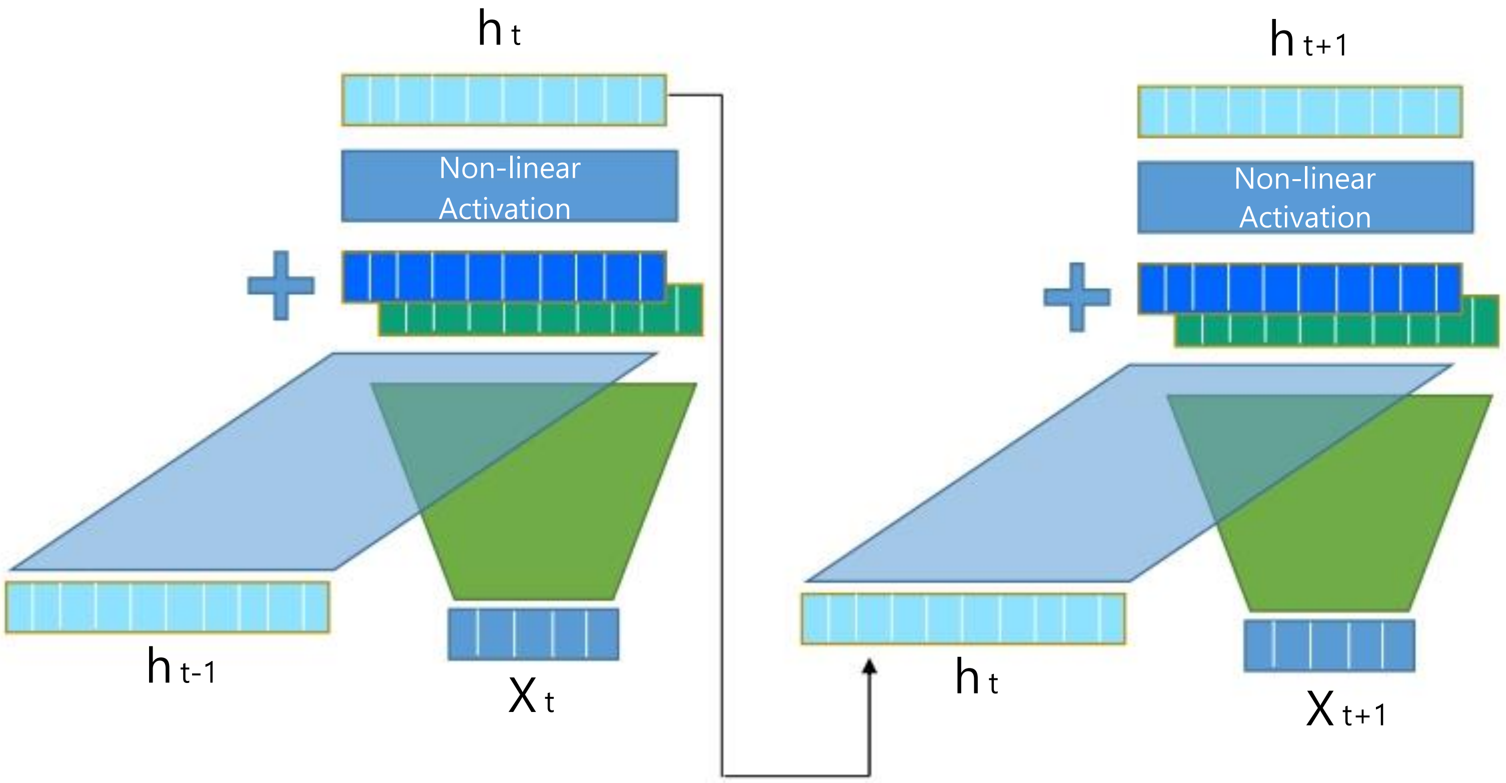
Recurrent Neural Network

But exactly, how can we combine new input and previous output?



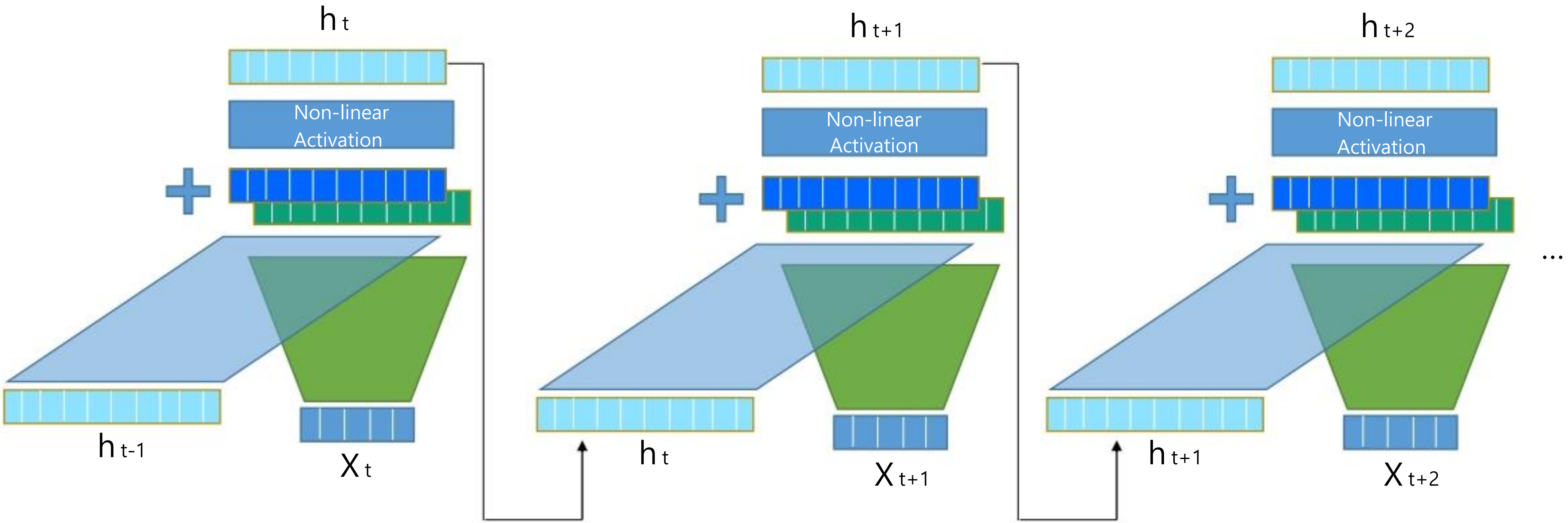
Recurrent Neural Network

But exactly, how can we combine new input and previous output?



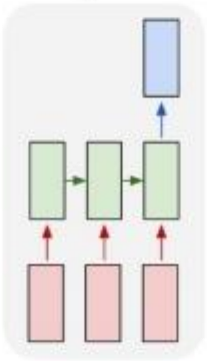
Recurrent Neural Network

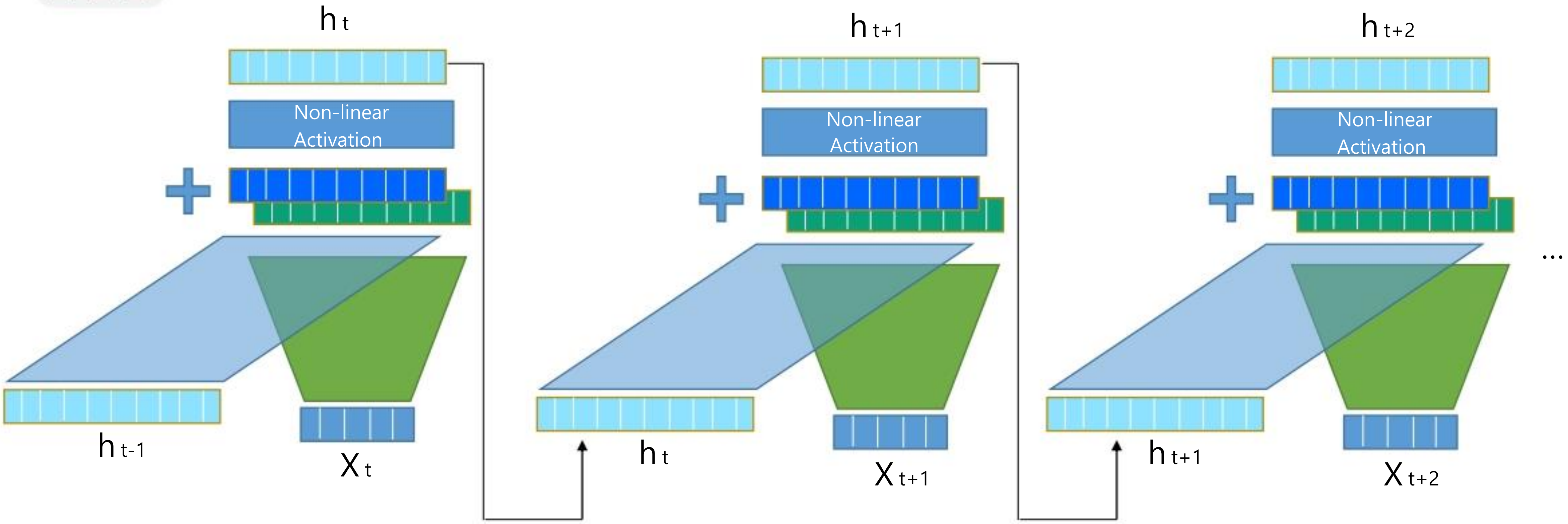
But exactly, how can we combine new input and previous output?



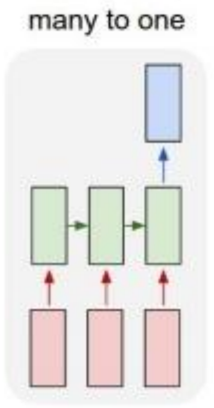
Recurrent Neural Network

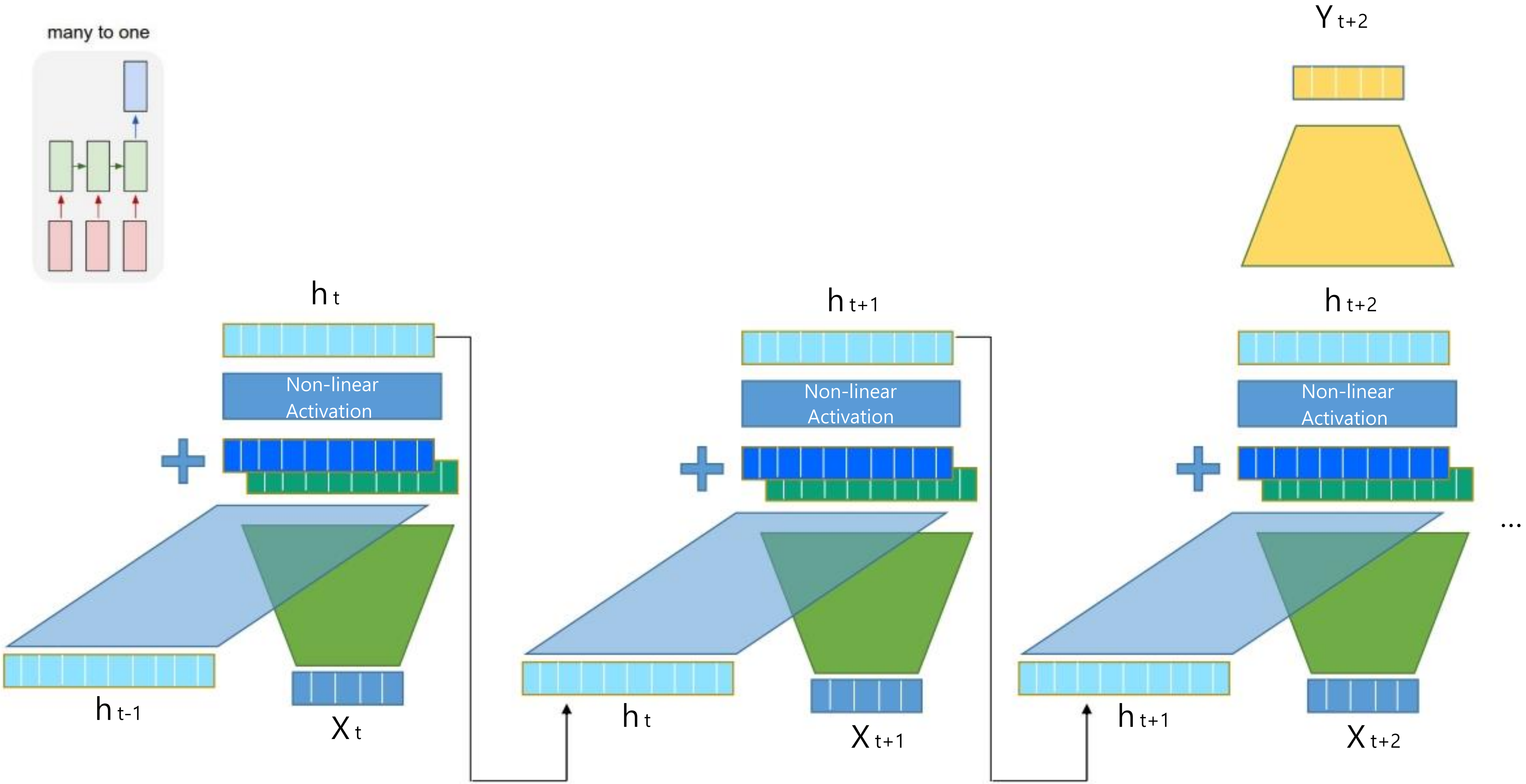
many to one



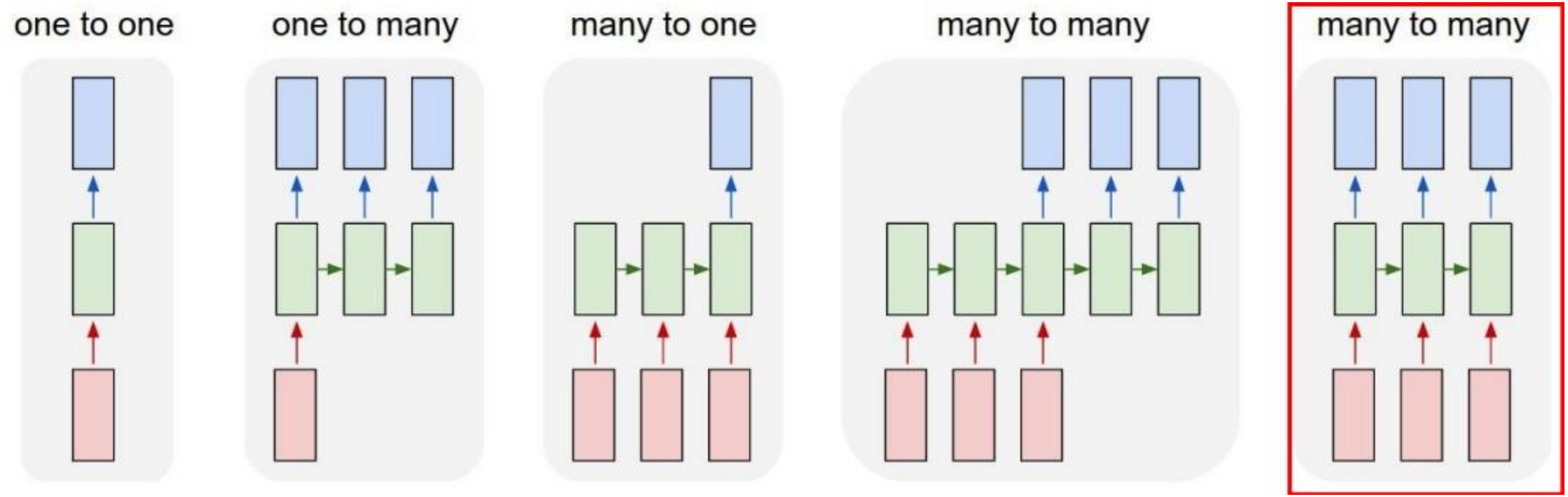


Recurrent Neural Network



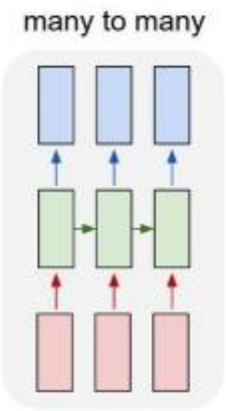


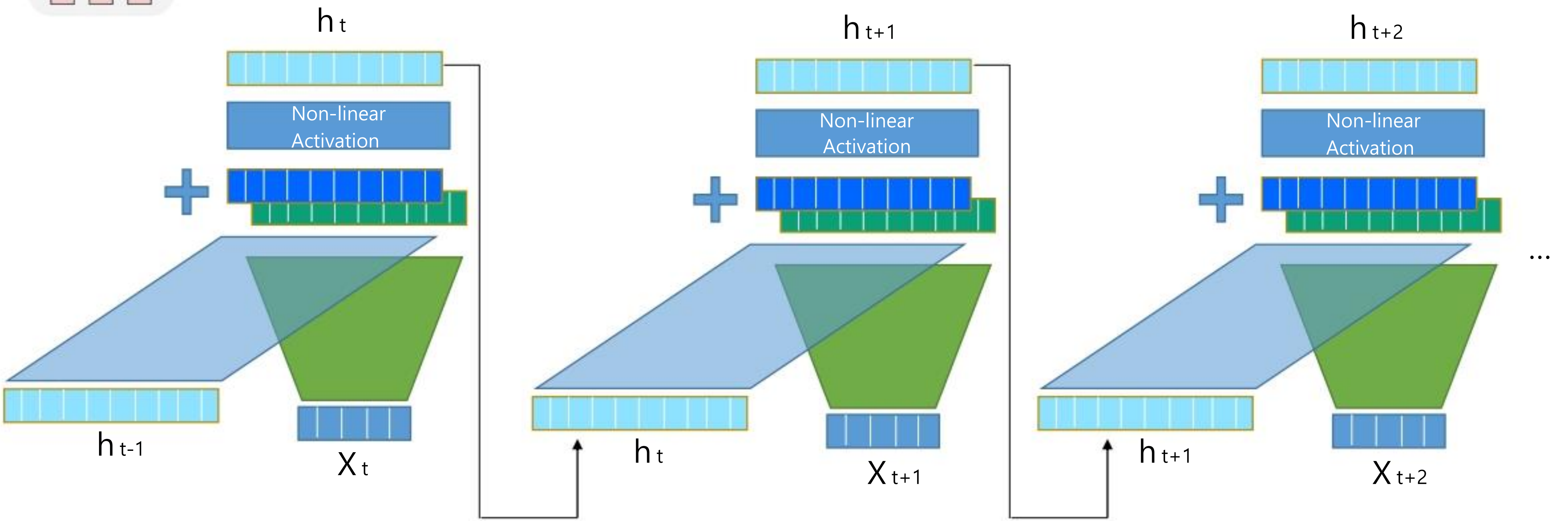
Types of Task Dealing with Sequential Data



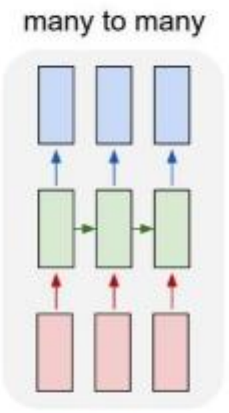
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

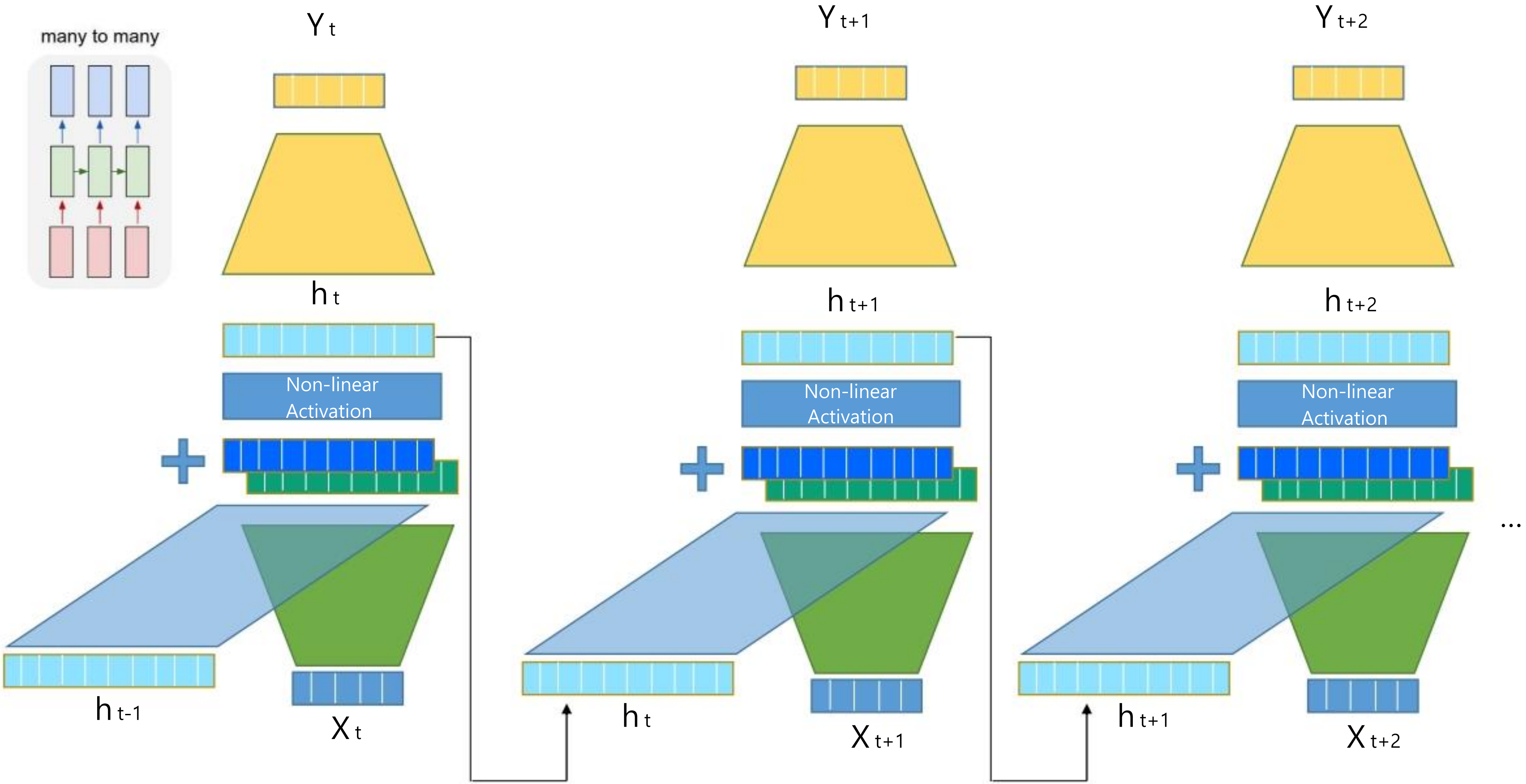
Recurrent Neural Network



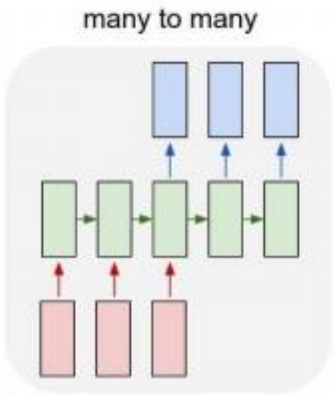


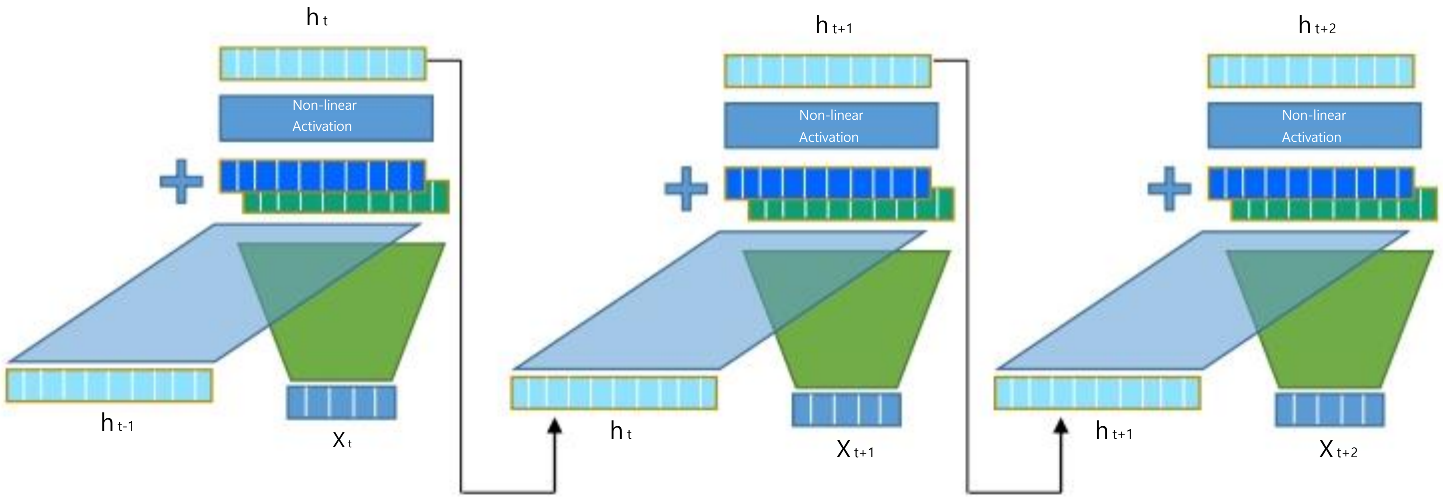
Recurrent Neural Network



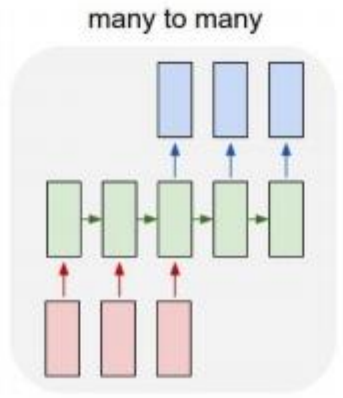


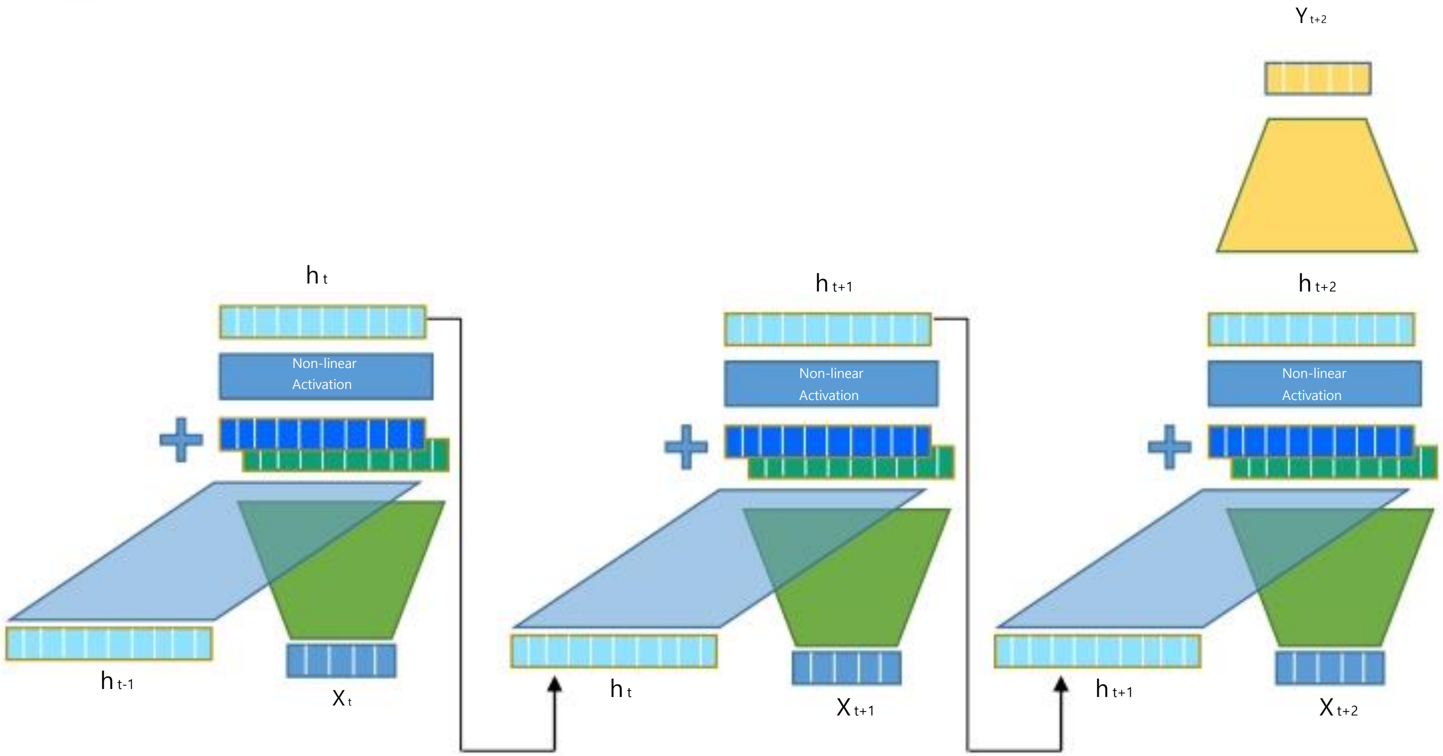
Recurrent Neural Network



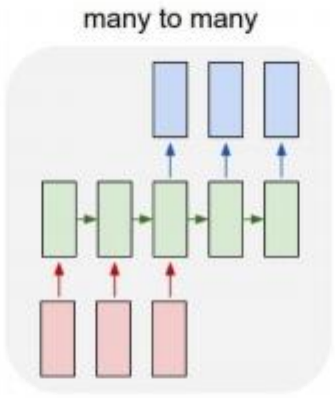


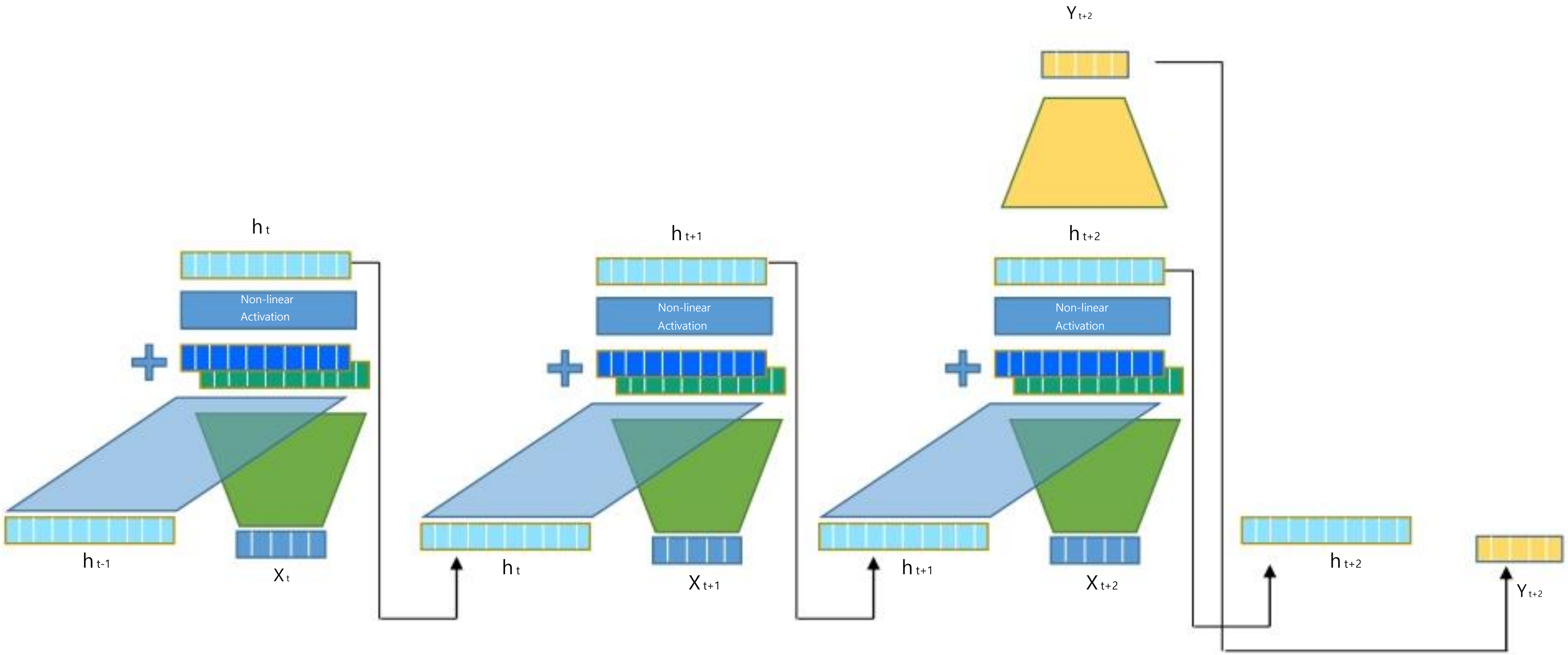
Recurrent Neural Network



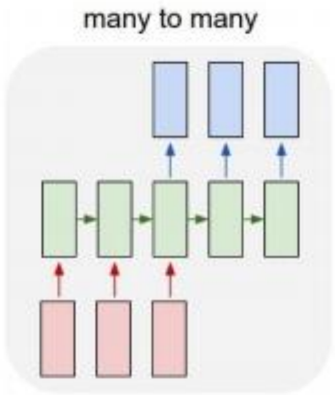


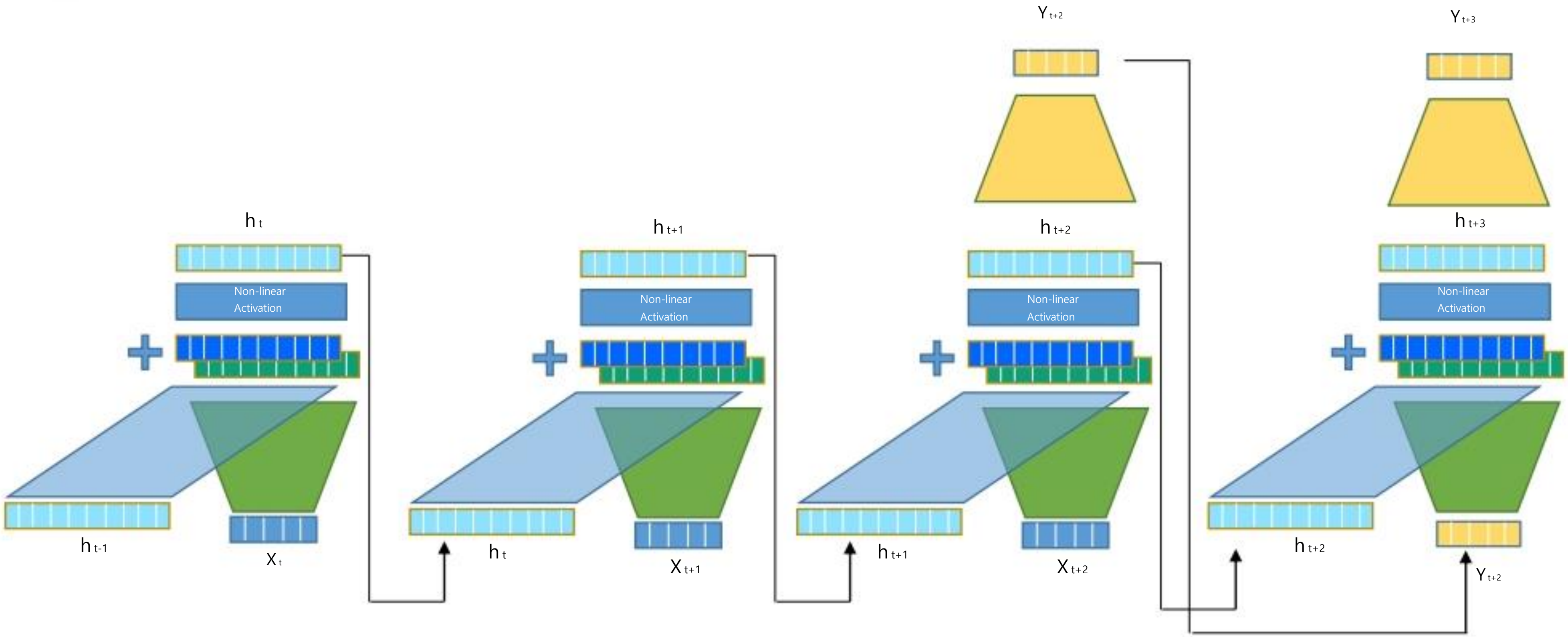
Recurrent Neural Network



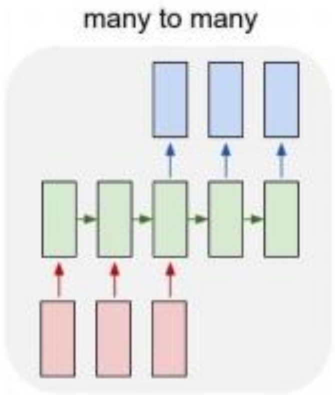


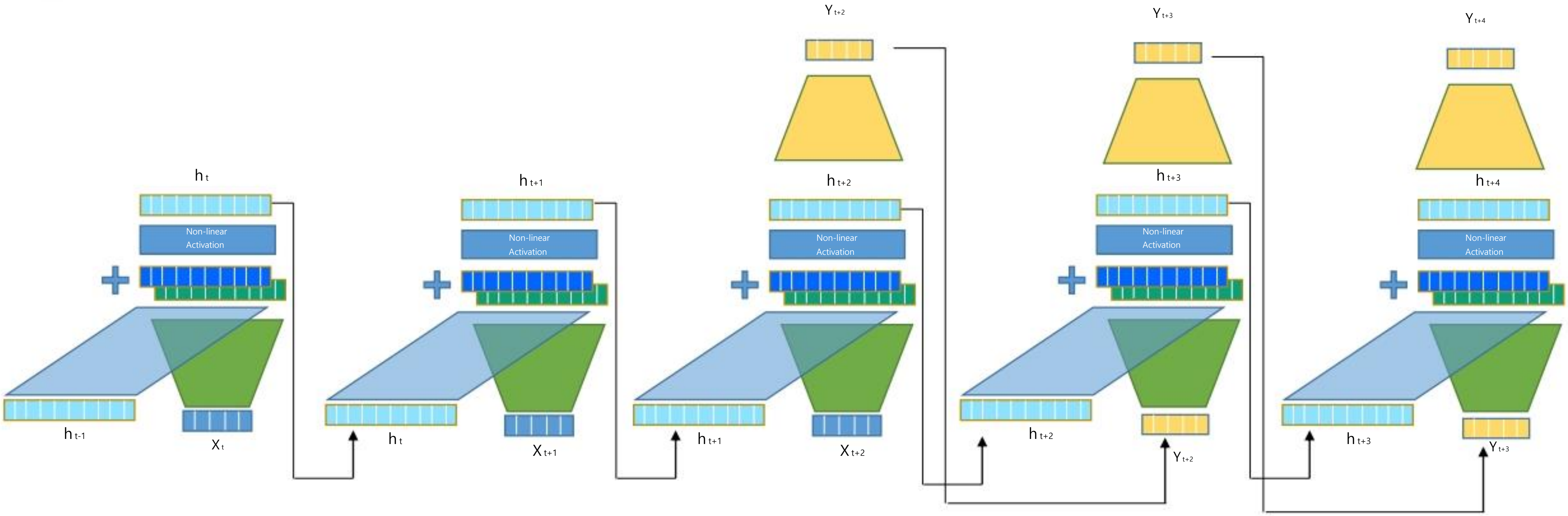
Recurrent Neural Network



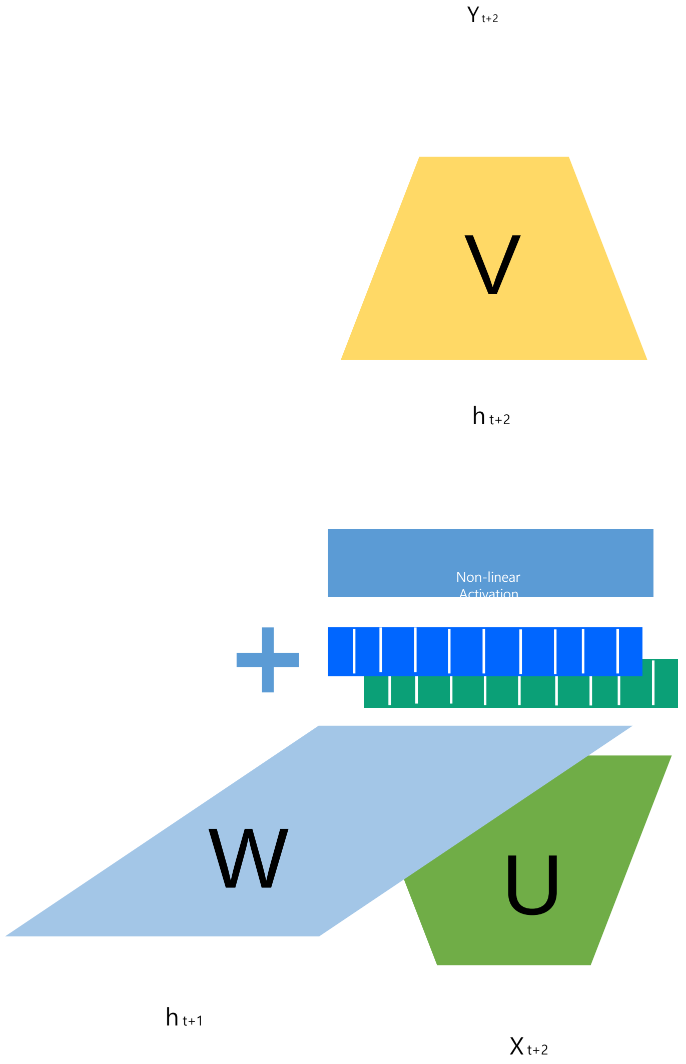


Recurrent Neural Network

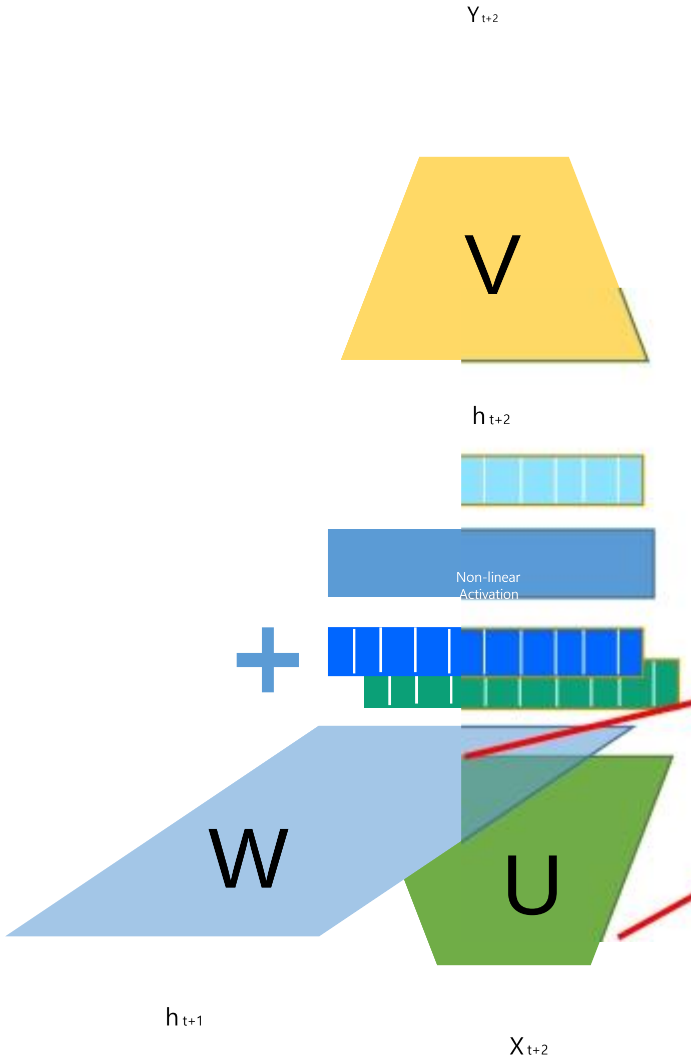




Recurrent Neural Network with Math

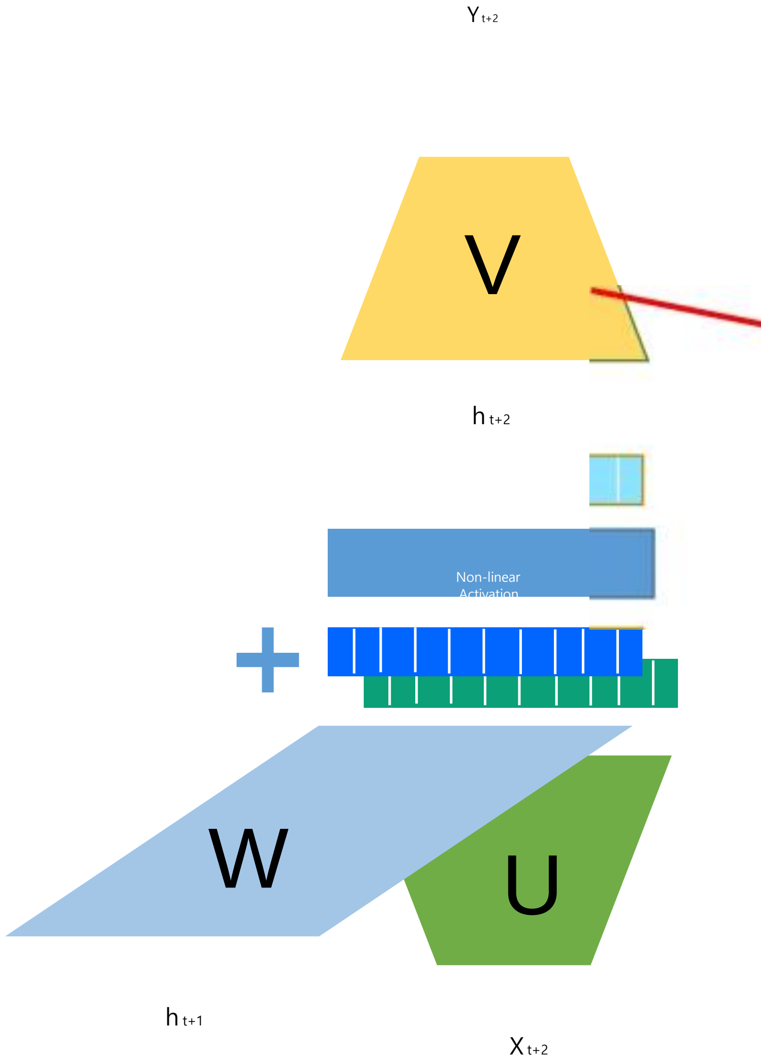


Recurrent Neural Network with Math



$$h_t = f(Ux_t + Wh_{t-1})$$

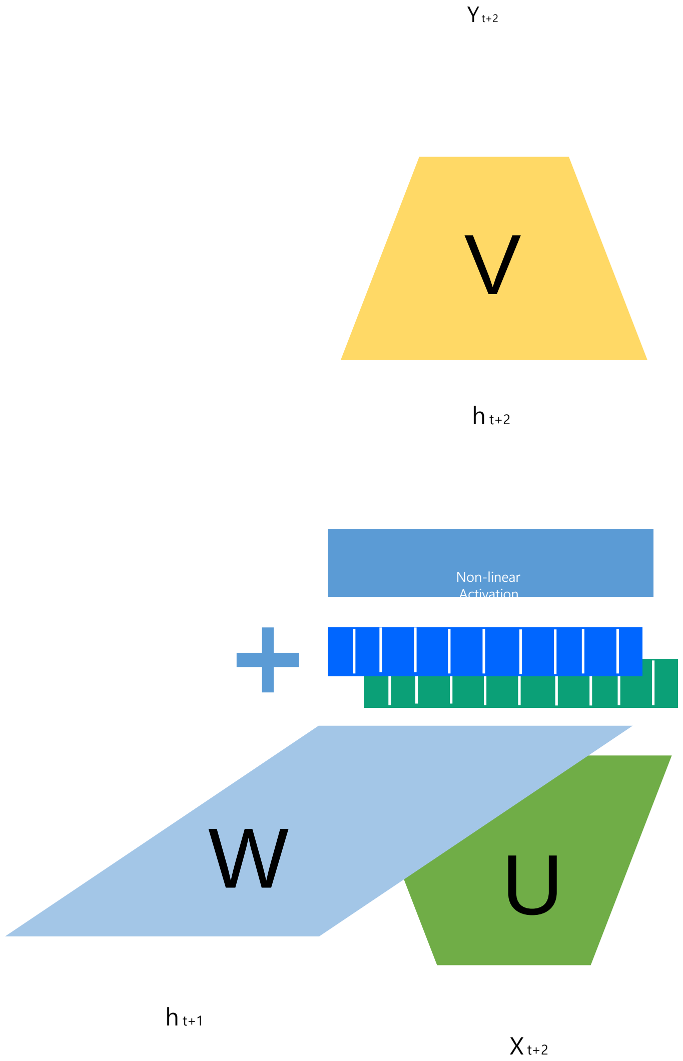
Recurrent Neural Network with Math



$$h_t = f(Ux_t + Wh_{t-1})$$

$$y_t = f(Vh_t)$$

Recurrent Neural Network with Math



f(x) = tanh(x)

$$h_t = f(Ux_t + Wh_{t-1})$$

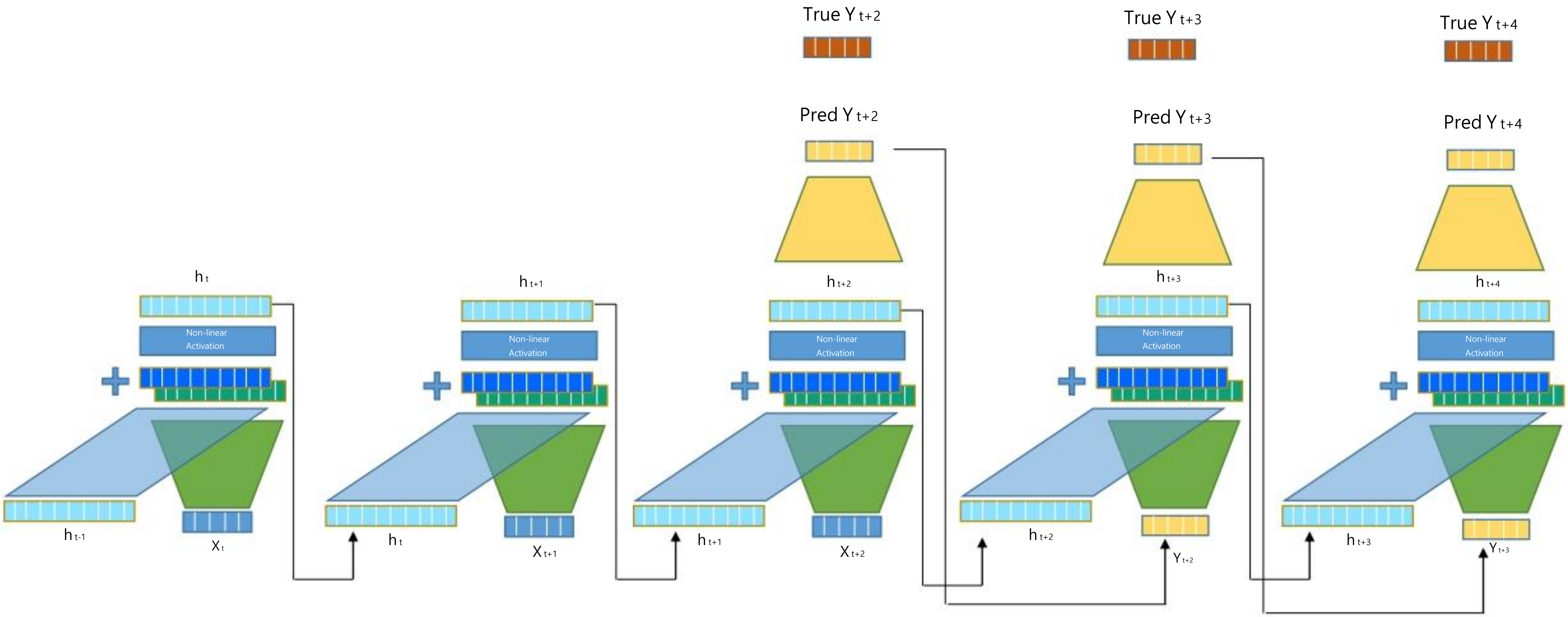
$$y_t = f(Vh_t)$$

f(x) = x

Okay, now we understand RNN model(hypothesis)

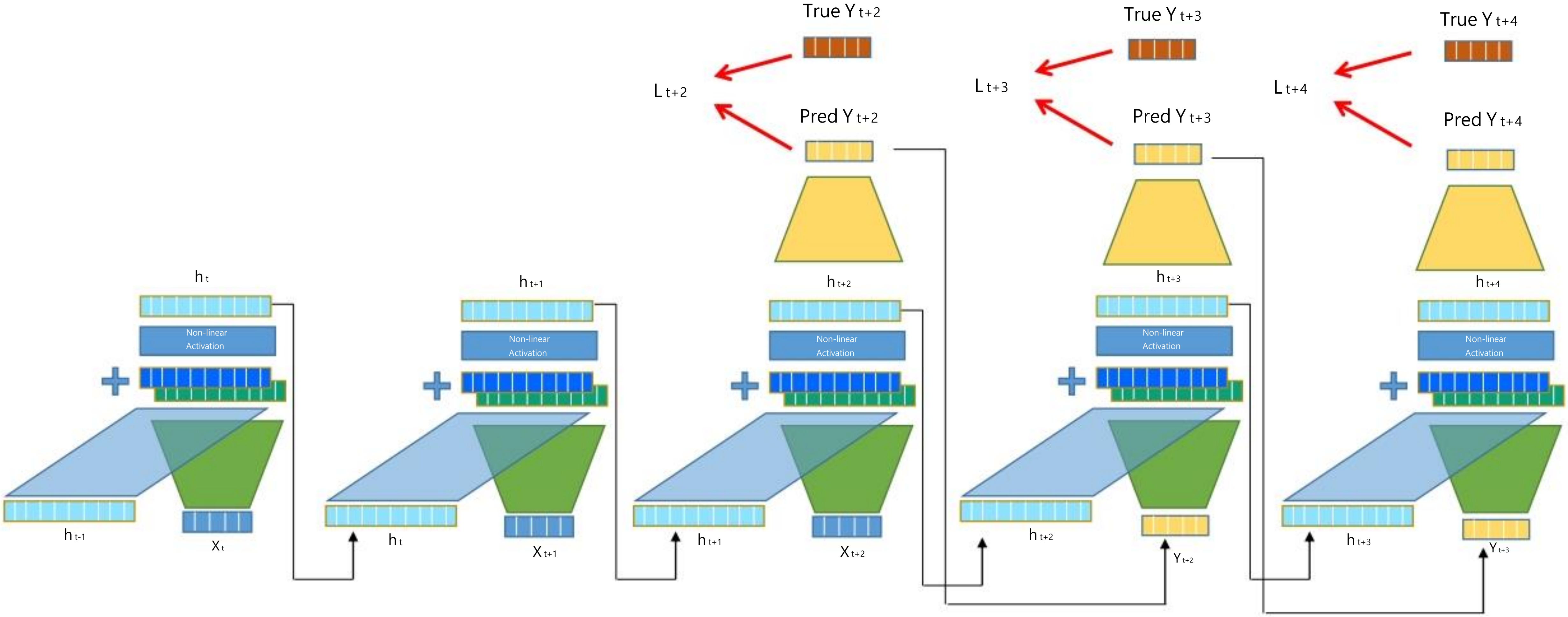
How can we evaluate it?

Calculate Loss of Recurrent Neural Network



Calculate Loss of Recurrent Neural Network

$$Loss(\theta) = \sum_t loss(y_{true,t}, y_{pred,t})$$

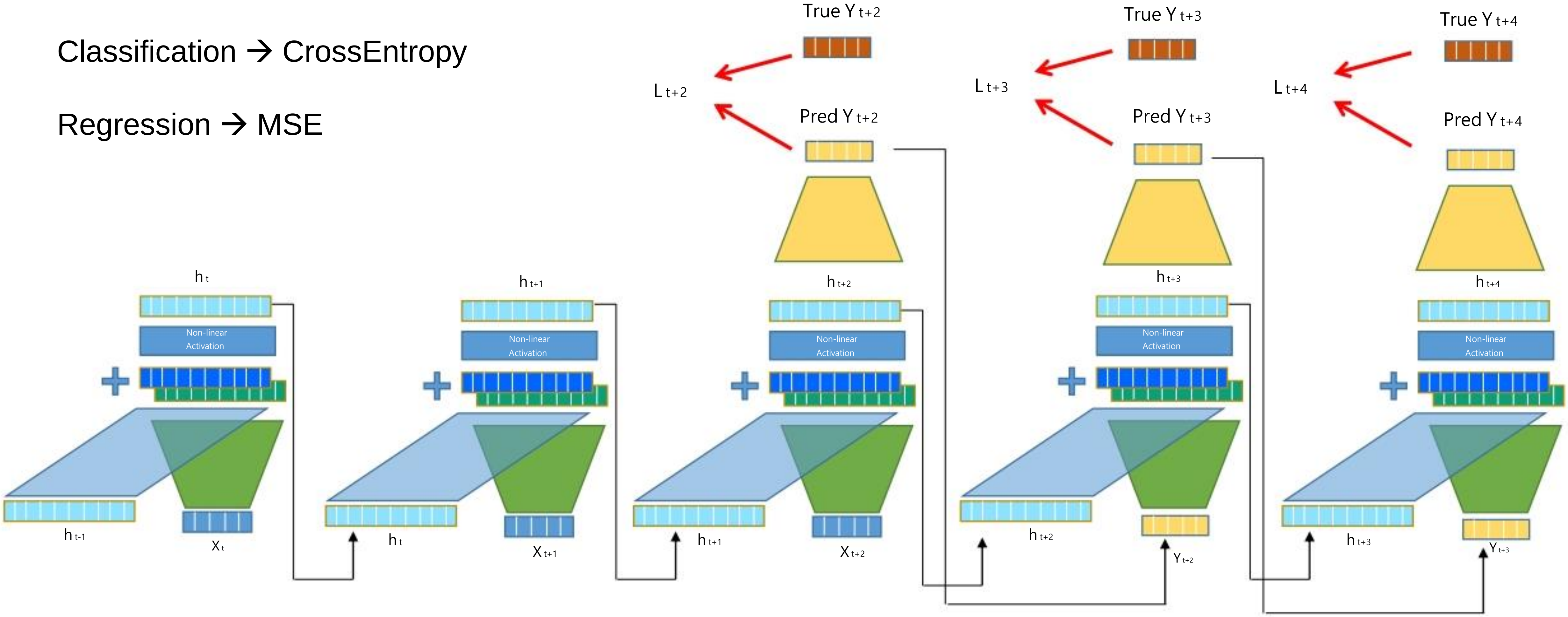


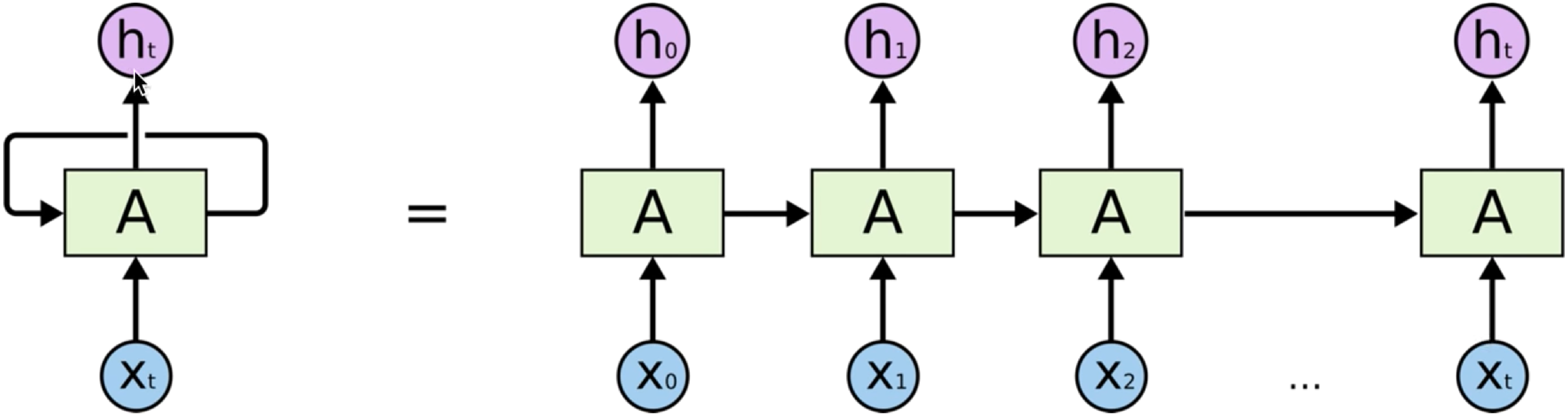
Calculate Loss of Recurrent Neural Network

$$Loss(\theta) = \sum_{t} loss(y_{true,t}, y_{pred,t})$$

Classification → CrossEntropy

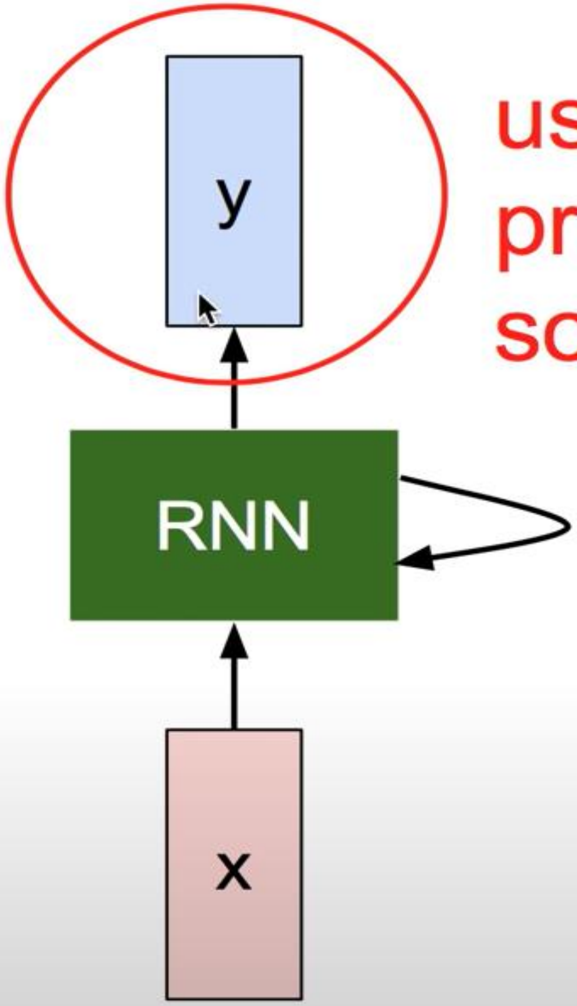
Regression → MSE





h_t
A
x_t
=
h_0
h_1
h_2
h_t
A
A
A
A
x_0
x_1
x_2
...
x_t

Recurrent Neural Network



usually want to predict a vector at some time steps

Recurrent Neural Network

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

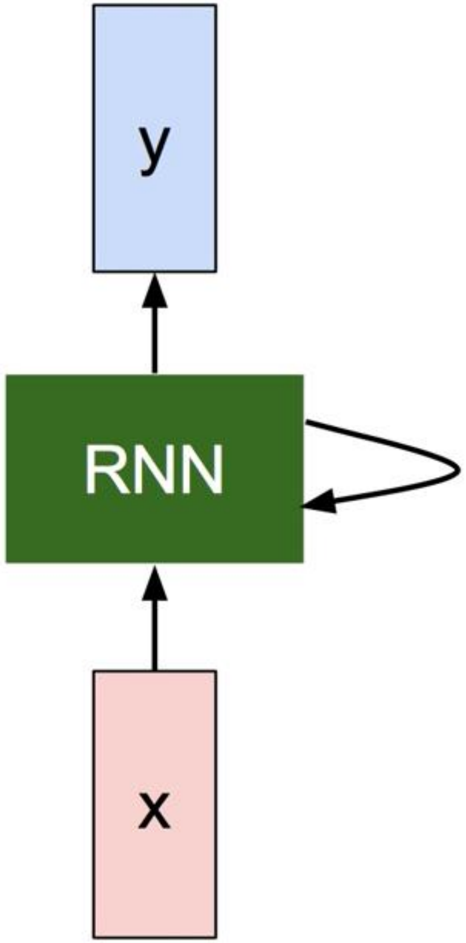
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

old state

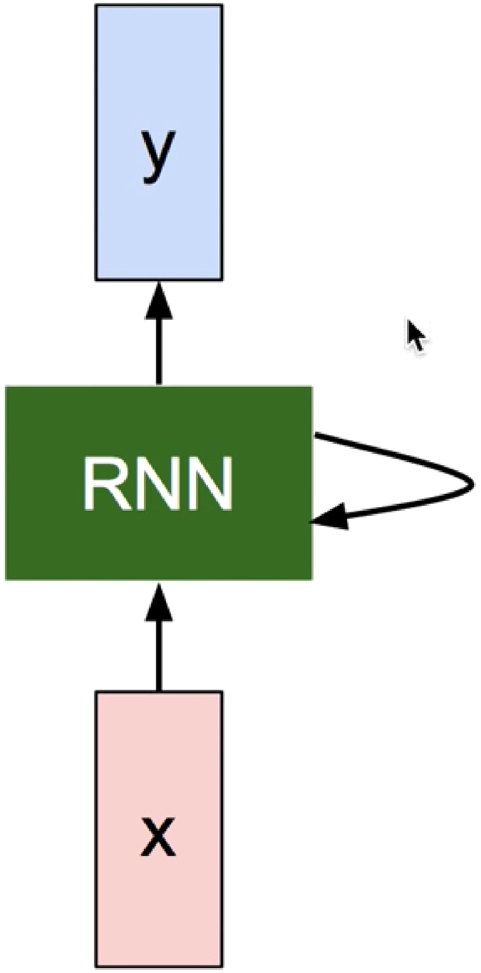
input vector at some time step

some function with parameters W



(Vanilla) Recurrent Neural Network

The state consists of a single *"hidden"* vector **h**:

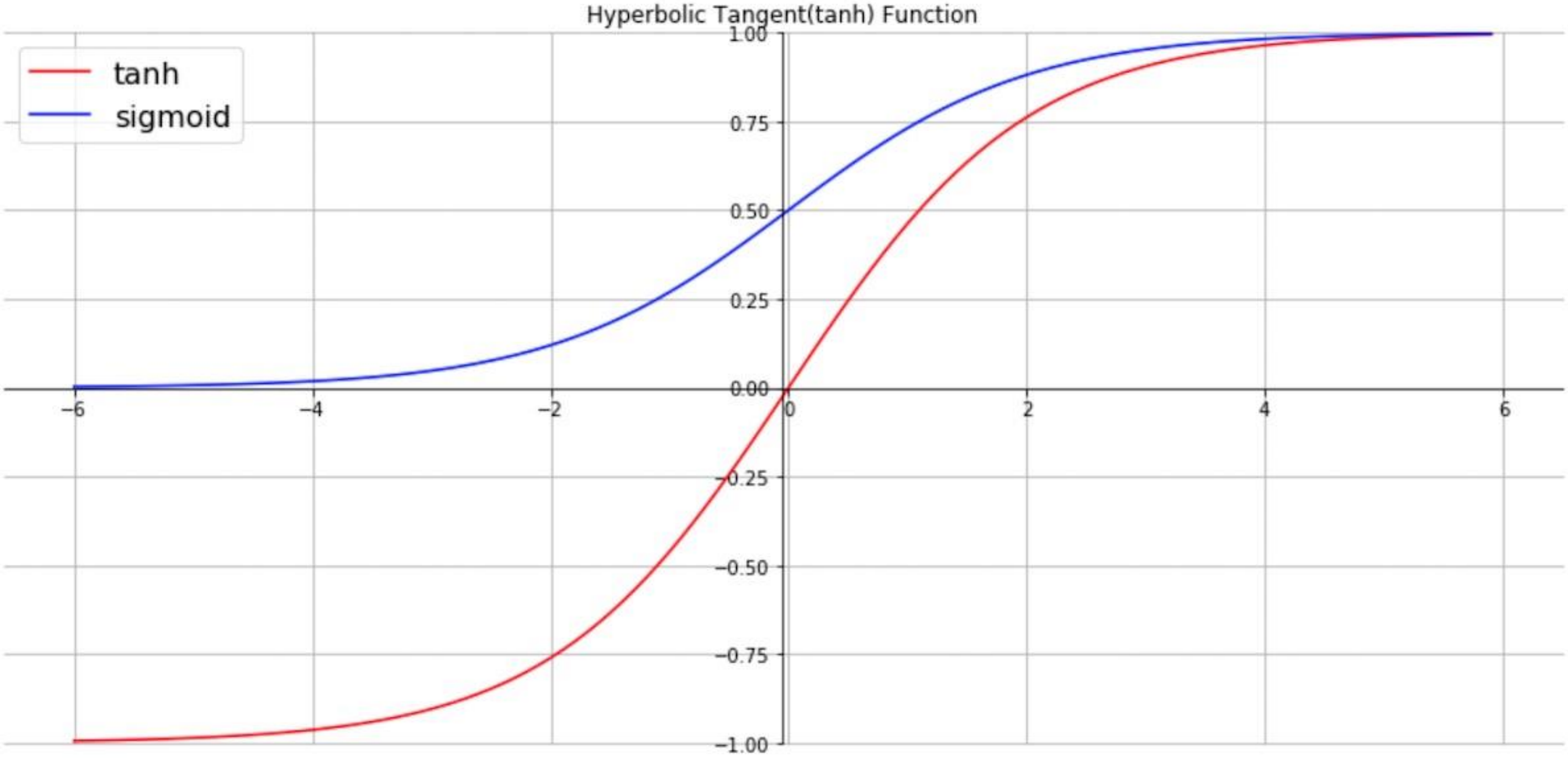


$$h_t = f_W(h_{t-1}, x_t)$$

$$\downarrow$$

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

$$y_t = W_{hy}h_t$$

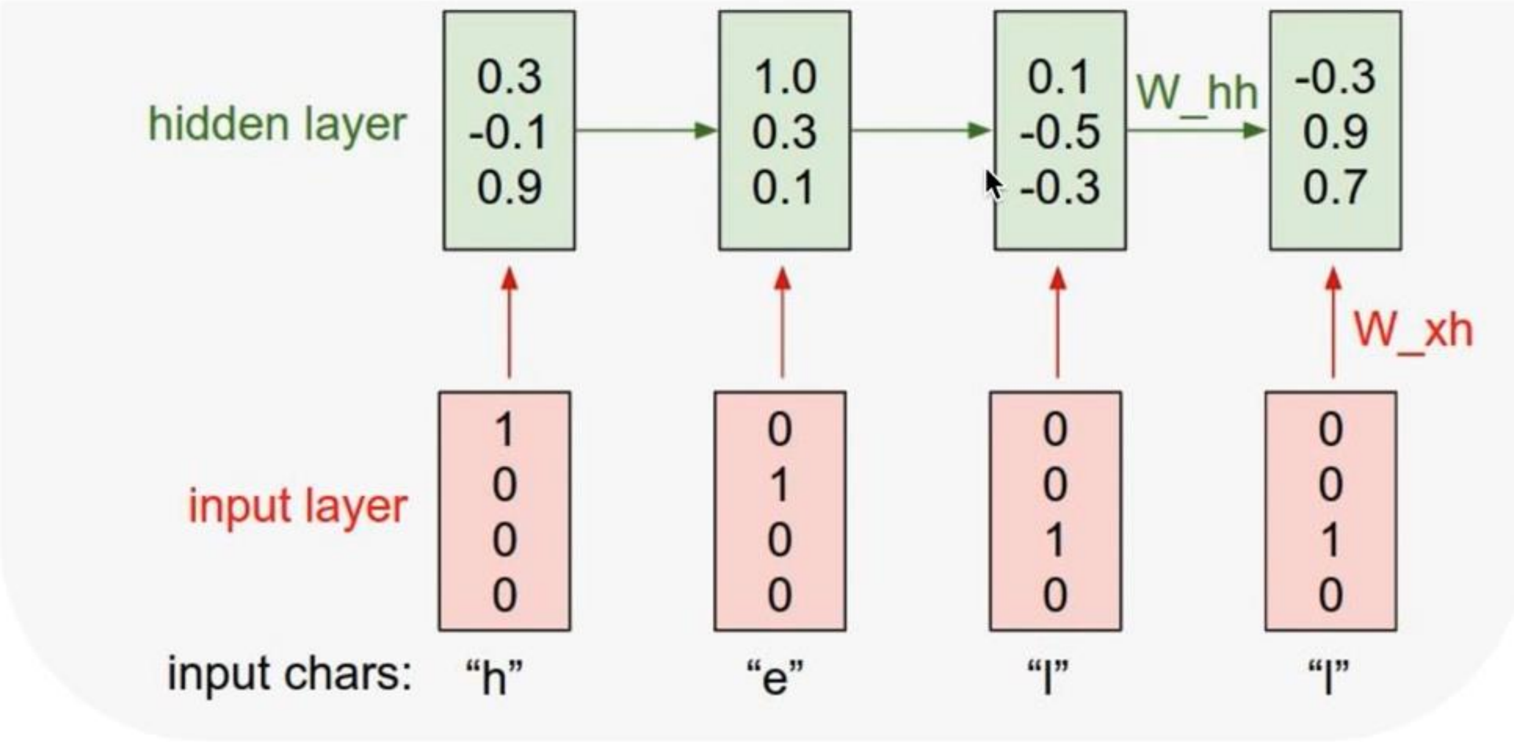

Hyperbolic Tangent(tanh) Function
tanh
sigmoid
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−6
−4
−2
0
2
4
6

Character-level language model example

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

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

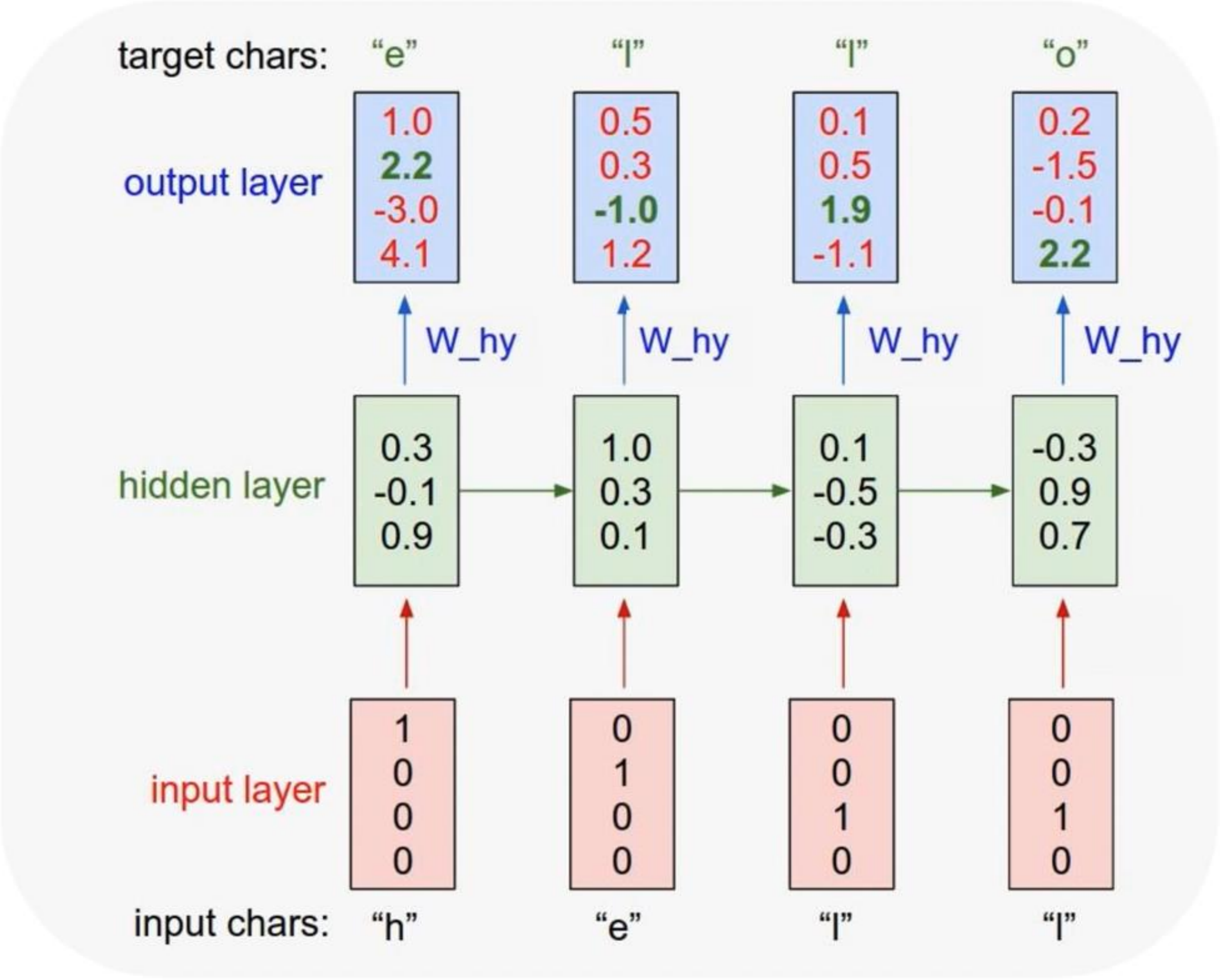


Character-level language model example

$$y_t = W_{hy}h_t$$

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

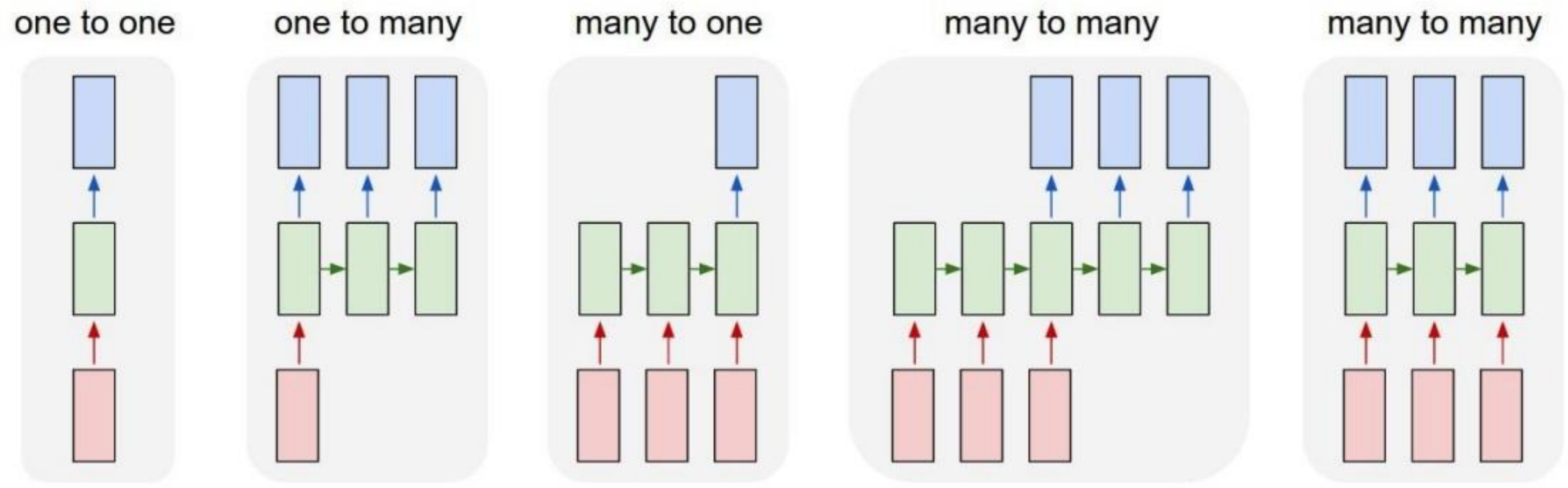


RNN applications

https://github.com/TensorFlowKR/awesome_tensorflow_implementations

- Language Modeling
- Speech Recognition
- Machine Translation
- Conversation Modeling/Question Answering
- Image/Video Captioning
- Image/Music/Dance Generation

Types of Task Dealing with Sequential Data



이미지로
부터단어
추출

감정분석
시계열

문자열로
부터 다른
문자열

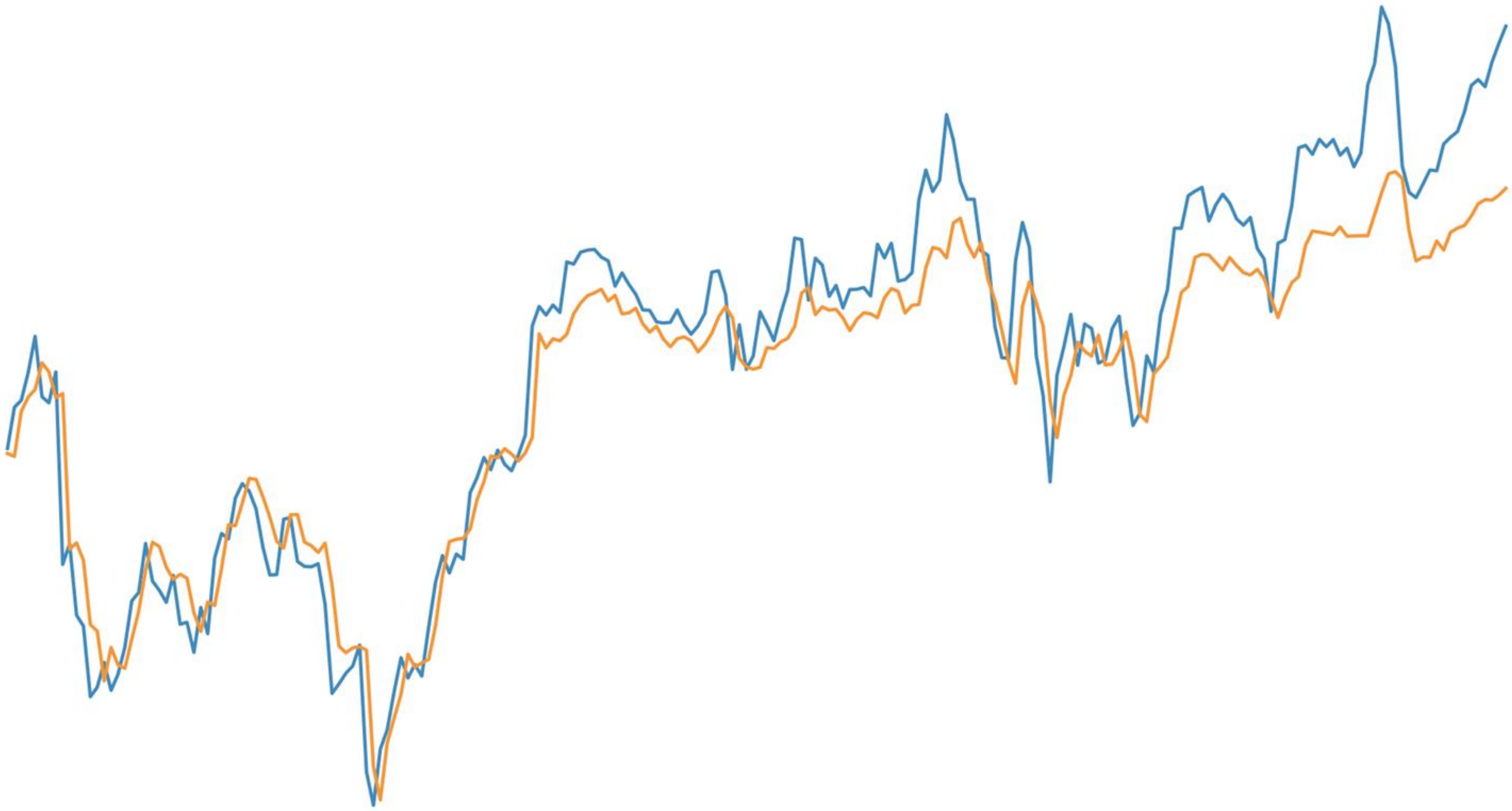
비디오프레임

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

LSTM, GRU

- 여러개의 데이터가 순서대로 입력되었을때 앞서 입력받은 데이터를 잠시 기억해 놓는 방법
- 기억된 데이터의 중요성을 판단하여 별도의 가중치를 줘서 다음 데이터로 넘긴다.
- 모든 입력 값에 이 작업이 순서대로 이뤄지며 같은 층을 맴도는 것으로 보여 순환 신경망이라 부른다

Time series data

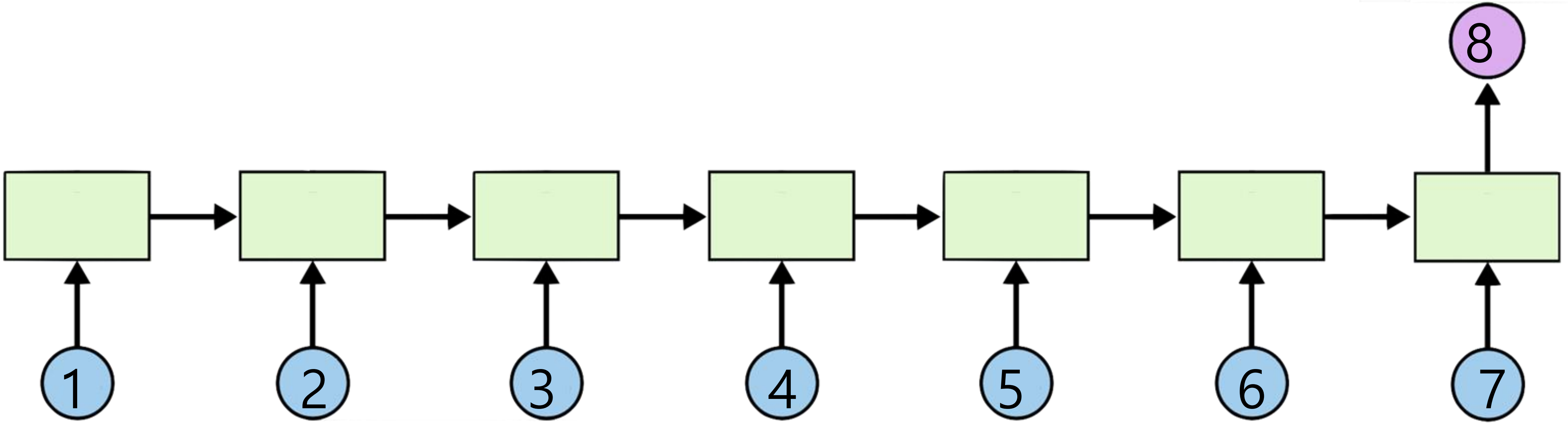


Time series data

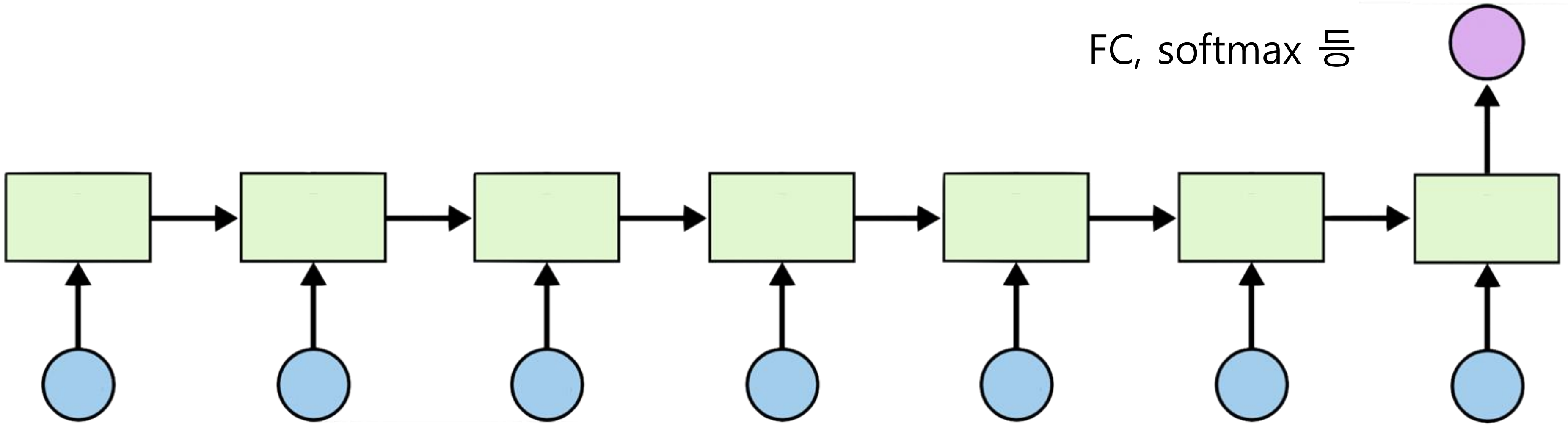
Open	High	Low	Volume	Close
828.659973	833.450012	828.349976	1247700	**831.659973**
823.02002	828.070007	821.655029	1597800	**828.070007**
819.929993	824.400024	818.97998	1281700	**824.159973**
819.359985	823	818.469971	1304000	**818.97998**
819	823	816	1053600	**820.450012**
816	820.958984	815.48999	1198100	**819.23999**
811.700012	815.25	809.780029	1129100	**813.669983**
809.51001	810.659973	804.539978	989700	**809.559998**
807	811.840027	803.190002	1155300	**808.380005**

'data-02-stock_daily.csv'

Many to one



https://github.com/hunkim/DeepLearningZeroToAll/blob/master/lab-12-5-rnn_stock_prediction



Open	High	Low	Volume	Close
828.659973	833.450012	828.349976	1247700	**831.659973**
823.02002	828.070007	821.655029	1597800	**828.070007**
819.929993	824.400024	818.97998	1281700	**824.159973**
819.359985	823	818.469971	1304000	**818.97998**
819	823	816	1053600	**820.450012**
816	820.958984	815.48999	1198100	**819.23999**
811.700012	815.25	809.780029	1129100	**813.669983**
809.51001	810.659973	804.539978	989700	**?**
807	811.840027	803.190002	1155300	**?**