DSAC Module4

Deep Learning (4-1)

2023년 11월 18일~12월 16일

권오준

(ojkwon@deu.ac.kr)

RNN

순환 신경망 (RNN) – CNN

- 합성곱 신경망(CNN)
  - 2차원(공간) 필터에서 좋은 성과 입증, 이미지 학습 분야에서 널리 사용

- 그러나 CNN은 시간적인 차이를 두고 연속적으로 발생하는 데이터 분석에는 성능이 부족
  - 연속된 단어로 구성되는, 문장 분석, 자연어 처리, 자동 번역, 시계열 데이터 분석 등

- 이런 데이터에는 순환신경망(Recurrent Neural Network)이 잘 동작

순환 신경망 (RNN) – 텍스트 처리

- 기존 텍스트 분석에서 BOW를 이용
  - 문장 내에 어떤 단어들이 존재하는지는 파악하지만
  - 단어의 배열 정보는 사라지고 이용하지 못함

- 단어의 순서 정보도 분석하려면
  - BoW, 단어 벡터의 도입만으로는 부족
  - 단어의 발생 순서 정보를 활용할 수 있는 신경망 모델을 사용해야 한다
  - 이를 위해서 순환신경망(RNN)이 널리 사용

순환 신경망 (RNN) – 과거 정보 기억

- MLP, CNN에서는 입력 데이터를 한 번 신경망에 통과시키면 데이터가 한 방향으로만 흘러가는 구조

- RNN은 과거의 데이터에서 추출한 정보를 지속적으로 모델 내에서 저장하고 이를 학습에 재사용
  - 사람이 대화를 나누거나 책을 읽을 때 단어를 순차적으로 듣게 되고 최신의 입력 단어들 뿐 아니라 과거의 정보를 계속 내부적으로 사용해야 하는 것과 같은 원리

- 즉, CNN에서는 과거의 기억(상태정보)을 사용하지 않지만 RNN은 과거의 기억을 사용하는 구조를 제공

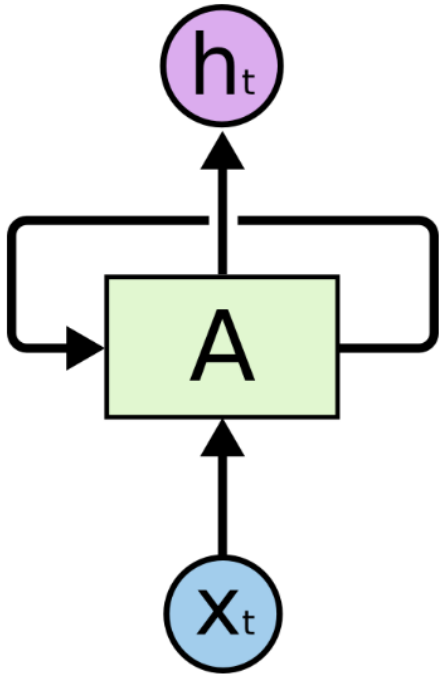
순환 신경망 (RNN) – 응용

- 주요 응용
  - 음성인식
  - 텍스트 분석
  - 자연어 처리
  - 챗봇
  - 감성 분석
  - 언어 모델링(language modeling)

  등에 널리 사용

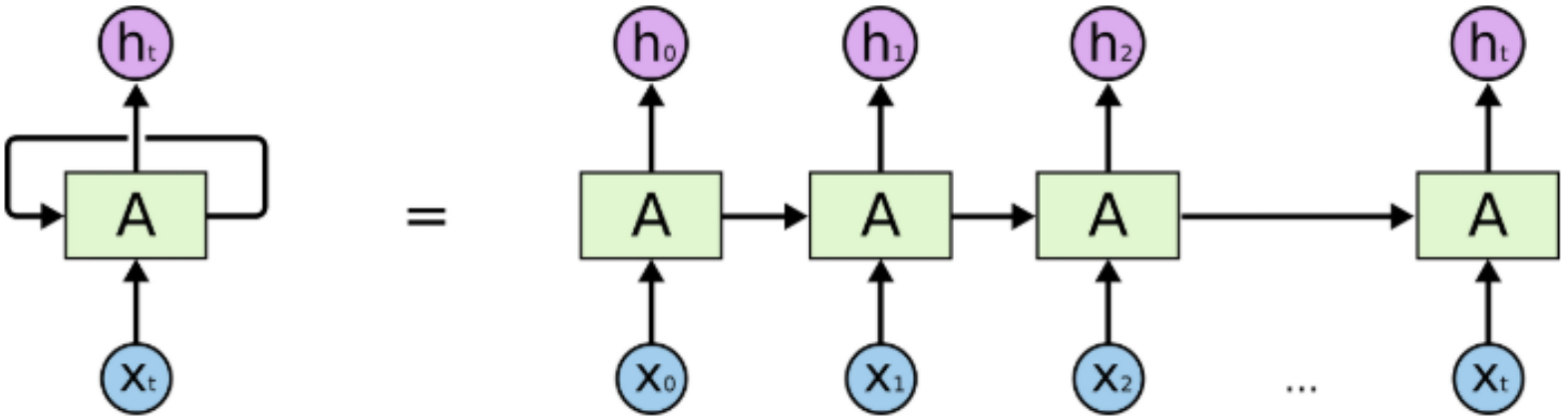
RNN 기본 구조

- RNN은 loop가 들어 있고
- 과거의 데이터가 미래에 영향을 줄 수 있는 구조



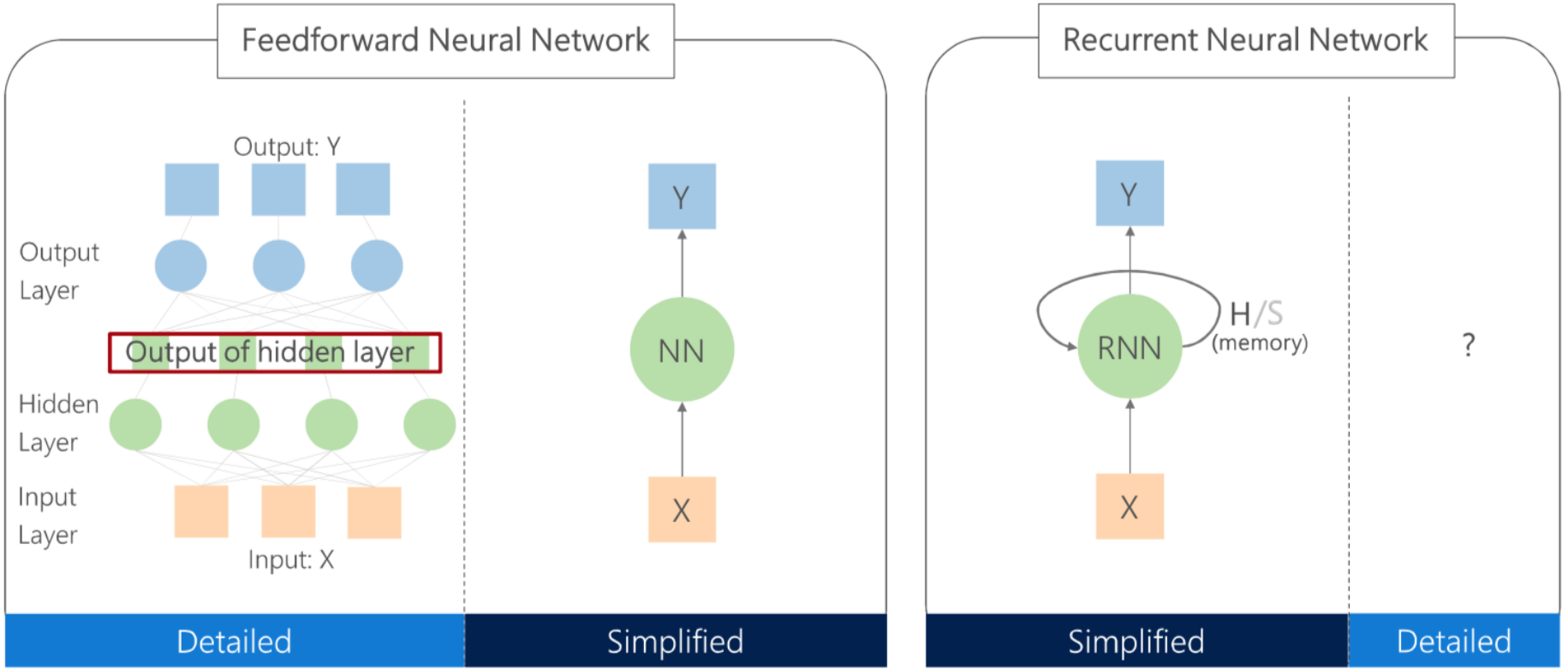
RNN unrolled

- 좌측
  - ➢ 순환신경망 하나의 유닛(셀)을 그린 것
  - ➢ 현재의 상태값이 다음번 상태값을 구하는데 순환적으로 사용
- 우측
  - ➢ 이해하게 쉽게 시간축으로 나누어 그린 것



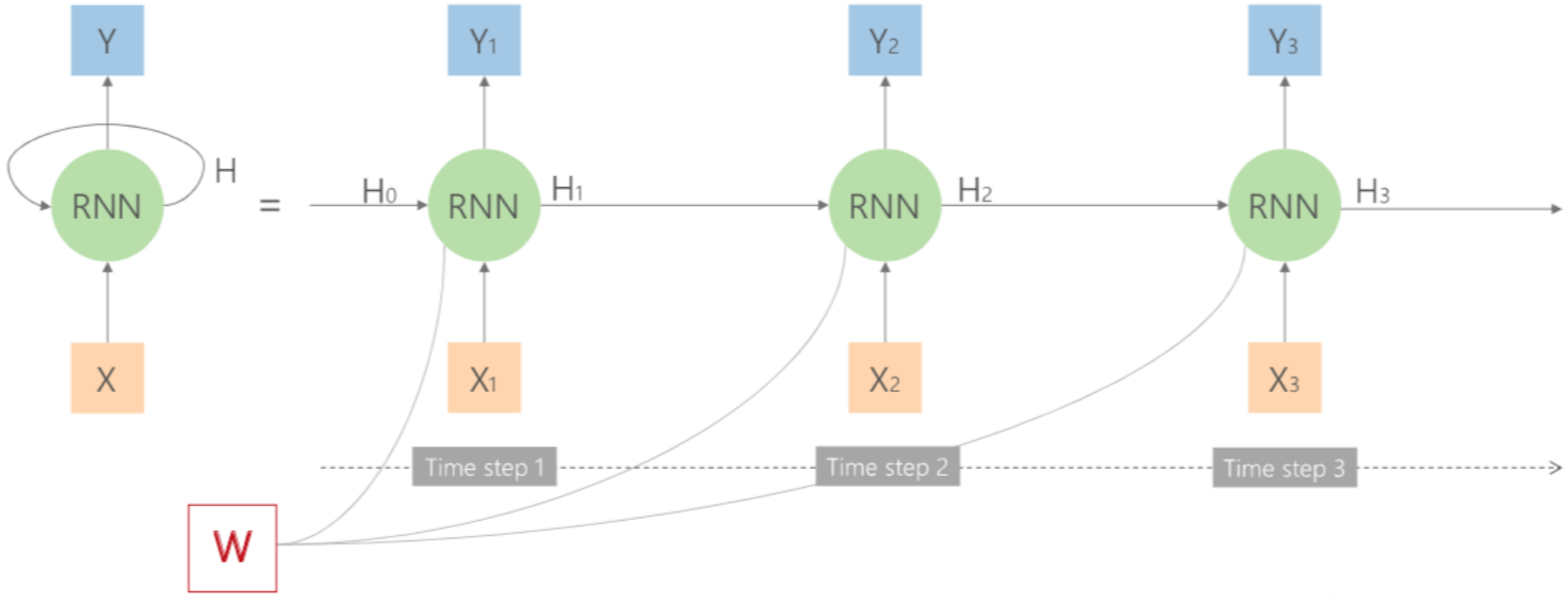
RNN vs FF 신경망

RNN has internal hidden state which can be fed back to network

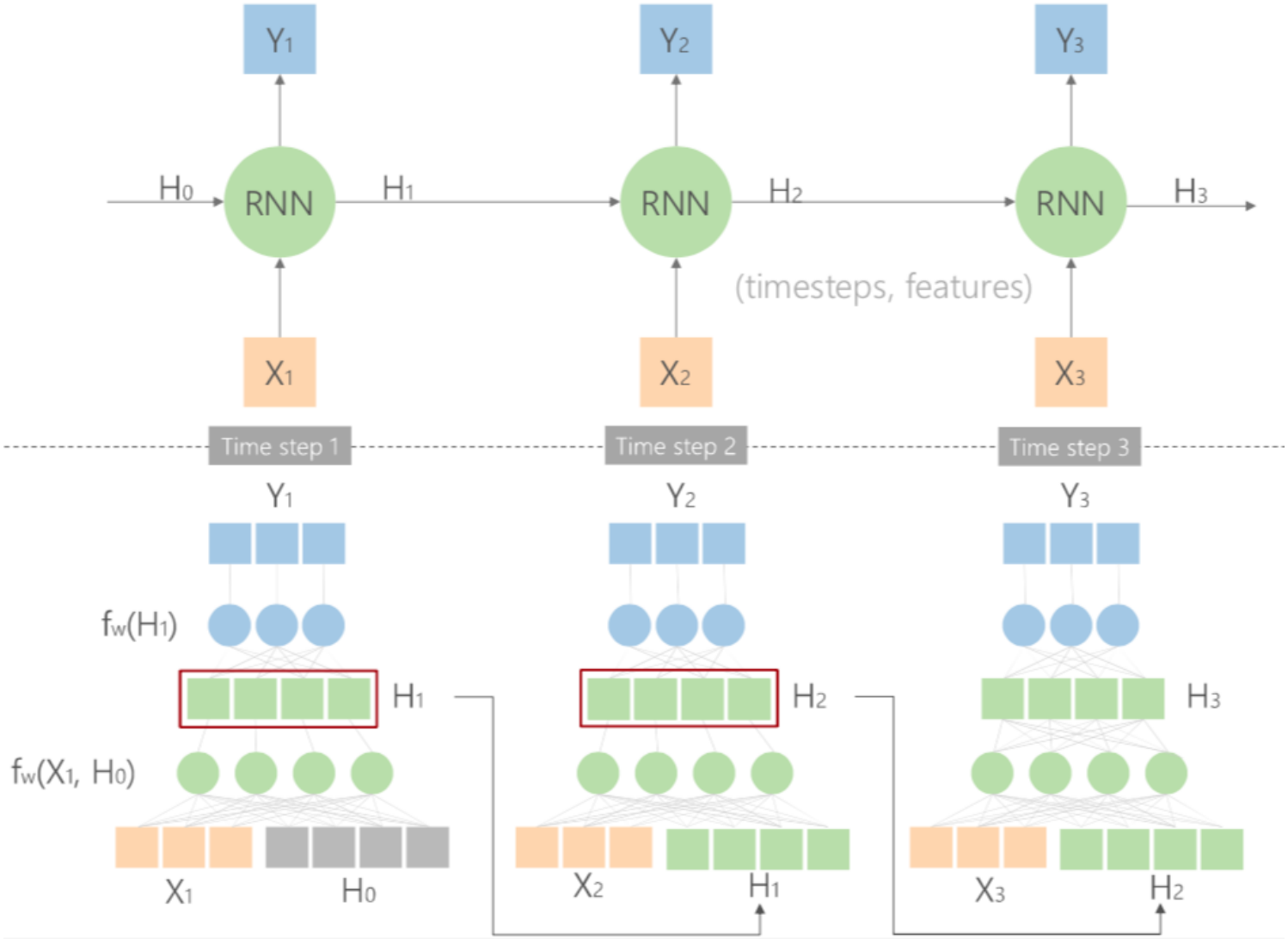


RNN unrolled

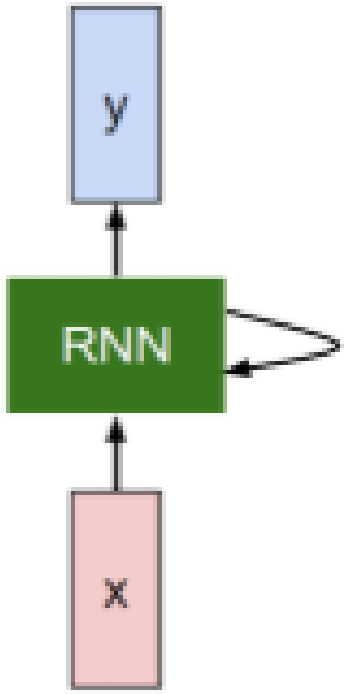
- 매 timesteps에 동일한 가중치를 사용



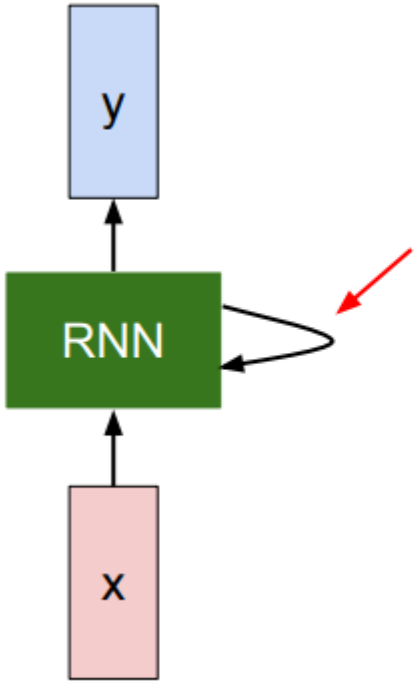
상세 동작


Y1
Y2
Y3
H0
RNN
H1
RNN
H2
RNN
H3
(timesteps, features)
X1
X2
X3
Time step 1
Time step 2
Time step 3
Y1
Y2
Y3
fw(H1)
H1
H2
H3
fw(X1, H0)
X1
H0
X2
H1
X3
H2

RNN 기본 구조



RNN 기본 구조 – 내부 상태 보유



Key idea: RNNs have an "internal state" that is updated as a sequence is processed

RNN 기본 구조 – 내부 상태 보유

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

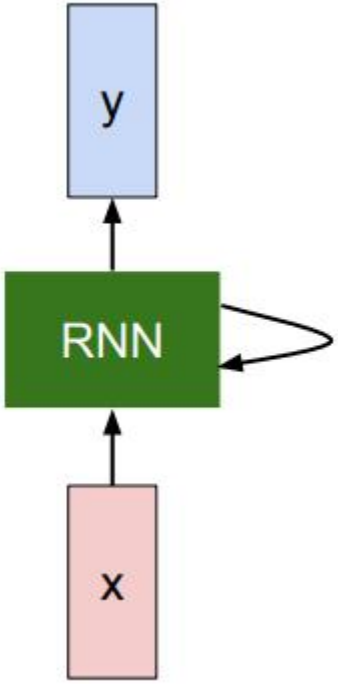
$$h_t = f_W(h_{t-1}, x_t)$$

new state

some function with parameters W

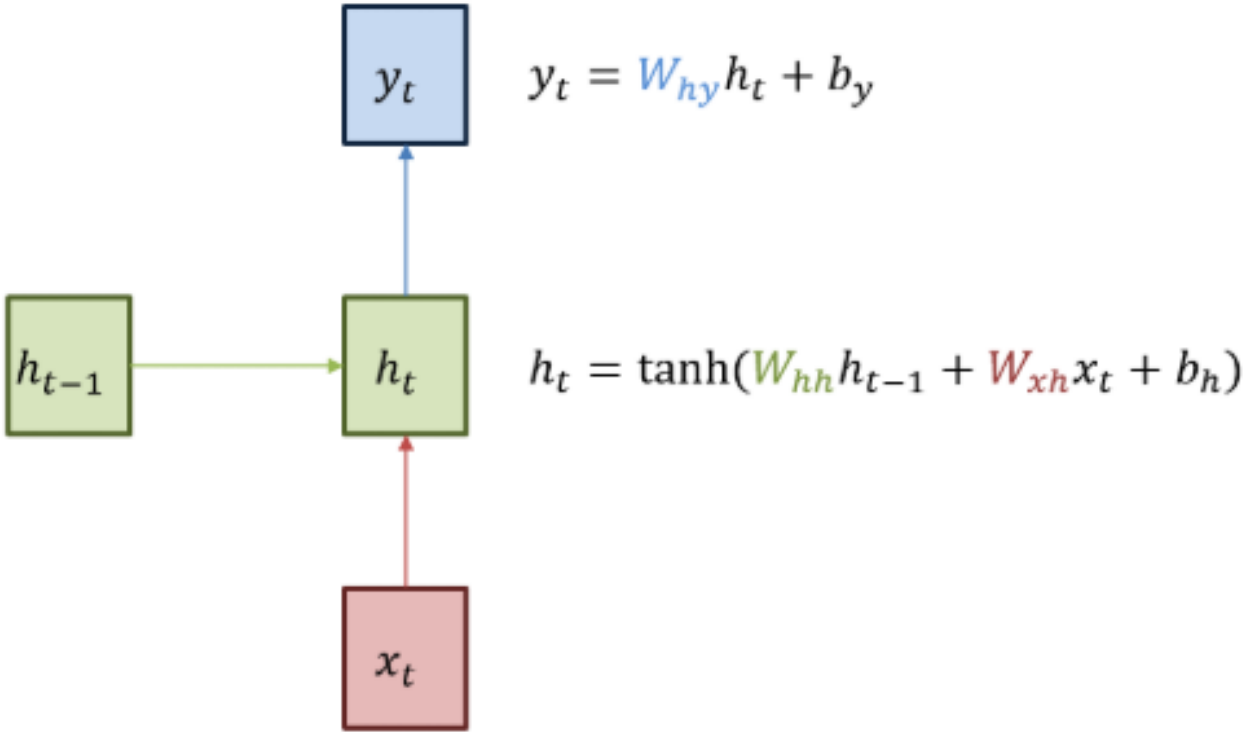
old state

input vector at some time step



기본 순환 신경망

- **현재의 출력**
  - 현재의 상태값으로부터 구한다



기본 순환 신경망 – 과거 정보 사용

- 모델의 현재 상태값을 얻기 위해서
  - **현재의 입력정보** 뿐 아니라 모델의 **이전의 상태 정보**도 동시에 사용

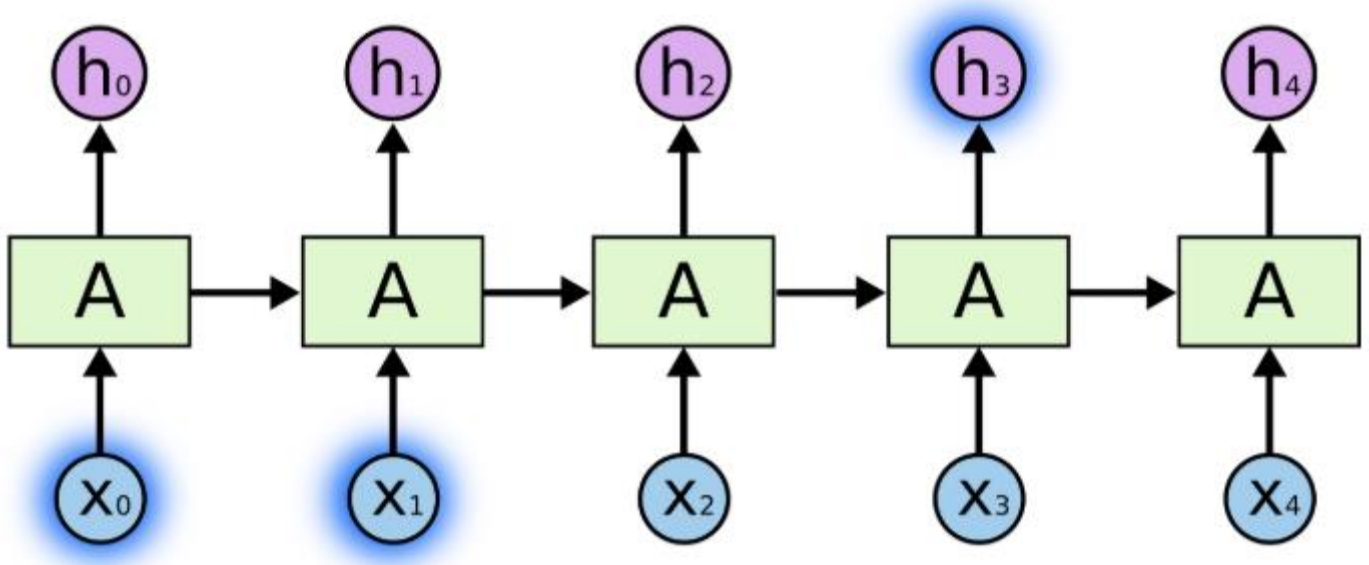
- 수식으로 표현

$$H_t = UX_t + WH_{t-1}$$

  - X(t) : 현재 입력 정보
  - H(t-1) : 이전 상태 정보
  - U, V : 가중치(계수) 매트릭스

- 이로써 과거의 정보를 내부적으로 재사용

기본 순환 신경망 – 과거 정보 사용



기본 순환 신경망 – 동일한 가중치 사용

- 모든 step에서 동일한 필터(가중치)를 사용
  - 계수 U, V, W는 시각(t)의 함수가 아니라, 모든 step에서 동일

- 순환신경망을 학습
  - 가중치 매트릭스 U, V, W의 값이 최적화되도록 학습하는 것

기본 순환 신경망 – 상태 , 출력

- **현재 상태 상태 활성화 함수**
  - Tanh함수를 주로 사용 (참고로 CNN에서는 ReLU)
  - W : 과거의 상태 정보를 얼마나 많이 활용하는지를 결정

$$H_t = \tanh(WH_{t-1} + UX_t)$$

- **현재의 출력**
  - 현재의 상태값으로부터 구한다

$$y_t = VH_t + c$$

기본 순환 신경망 – 상태 , 출력

- 우리가 다른 사람의 말을 듣고 대답을 한다면,
  - 먼저 머릿속에 상대방의 말을 이해하는 상태를 만들어야 하고(즉, H_t)
  - 이 상태에 기반하여 출력(y_t)를 만들어내는 것과 유사한 개념

- 입력(X)의 단위
  - 한 단어일 수도 있고
  - 10개의 단어일 수도 있고, 한 문장일 수도

- 현재 출력을 구하는데 이전의 상태 정보를 순환적으로 사용
  - 결국 최초의 상태값을 포함해서 오래된 상태 정보를 조금이라도 활용한다고 볼 수 있다

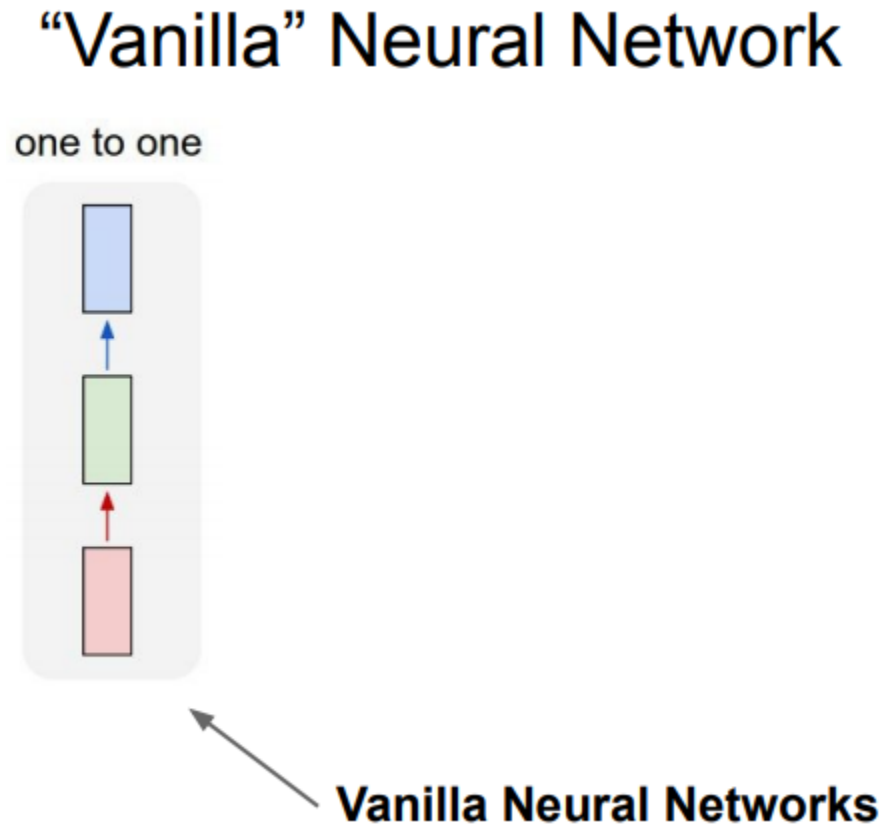
기본 순환 신경망 – 기울기 소실 문제

- 과거의 오래된 신호를 적은 비중으로 일괄적으로 반영하면 신호의 업데이트가 너무 적어져서 기울기 소실 문제가 발생

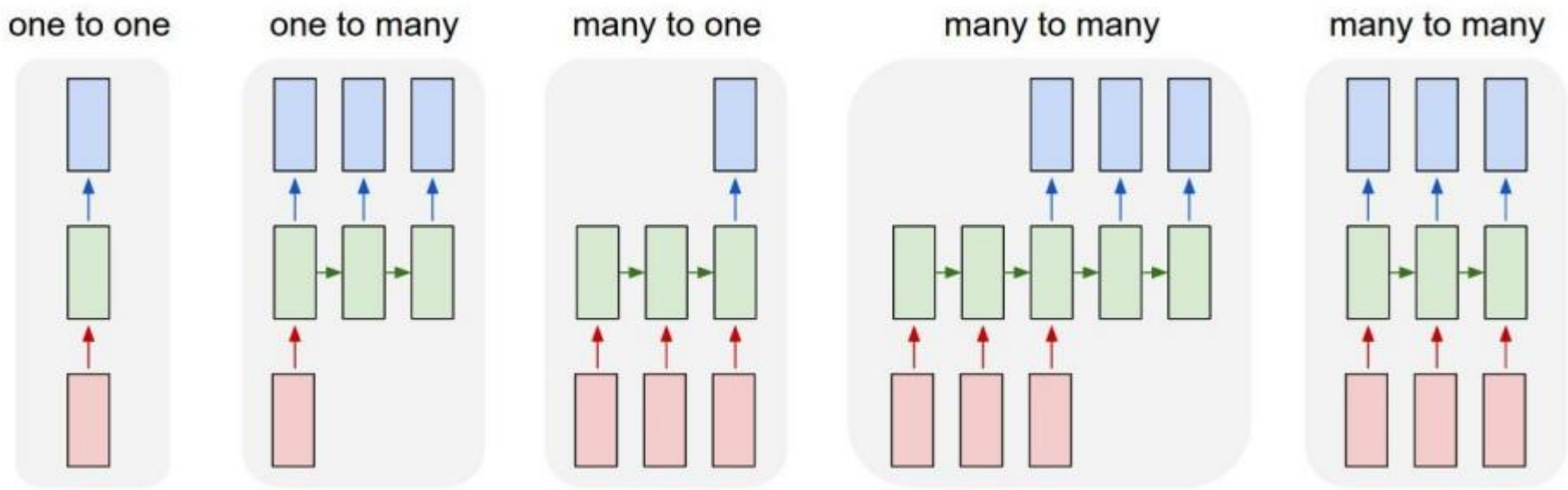
- 과거의 신호를 너무 많이 반영하면 신호가 발산하는 문제가 발생

- 해결책
  - LSTM(Long-Short Term Memory) 기법이 널리 사용
  - 오랜 기간 중요도를 유지할 정보와 그럴 필요 없이 망각해야 할 신호를 구분하여 따로 처리

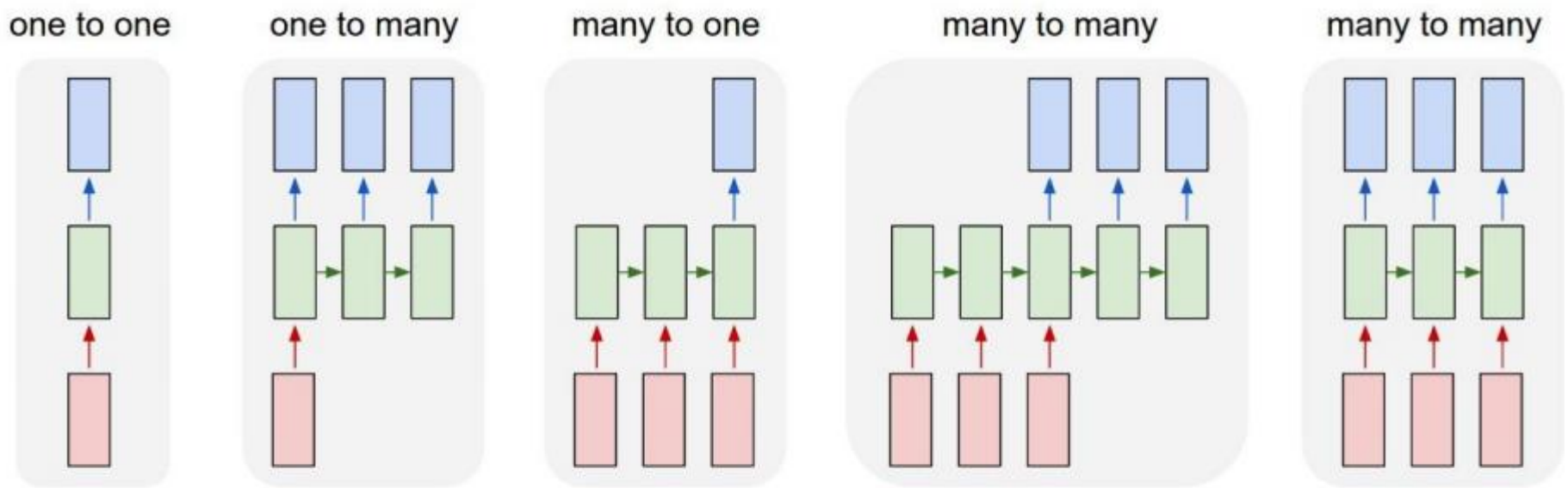
RNN 기본 모델 유형



RNN 기본 모델 유형


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

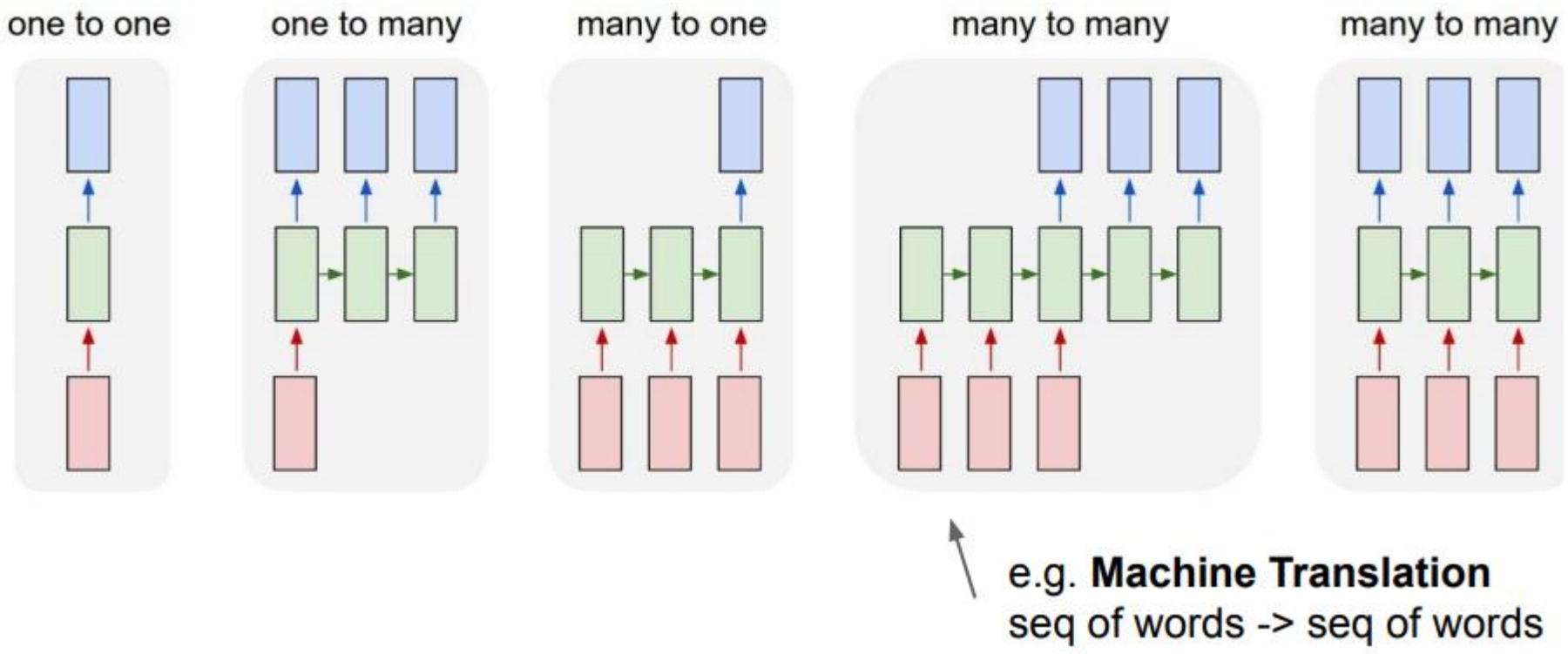
RNN 기본 모델 유형



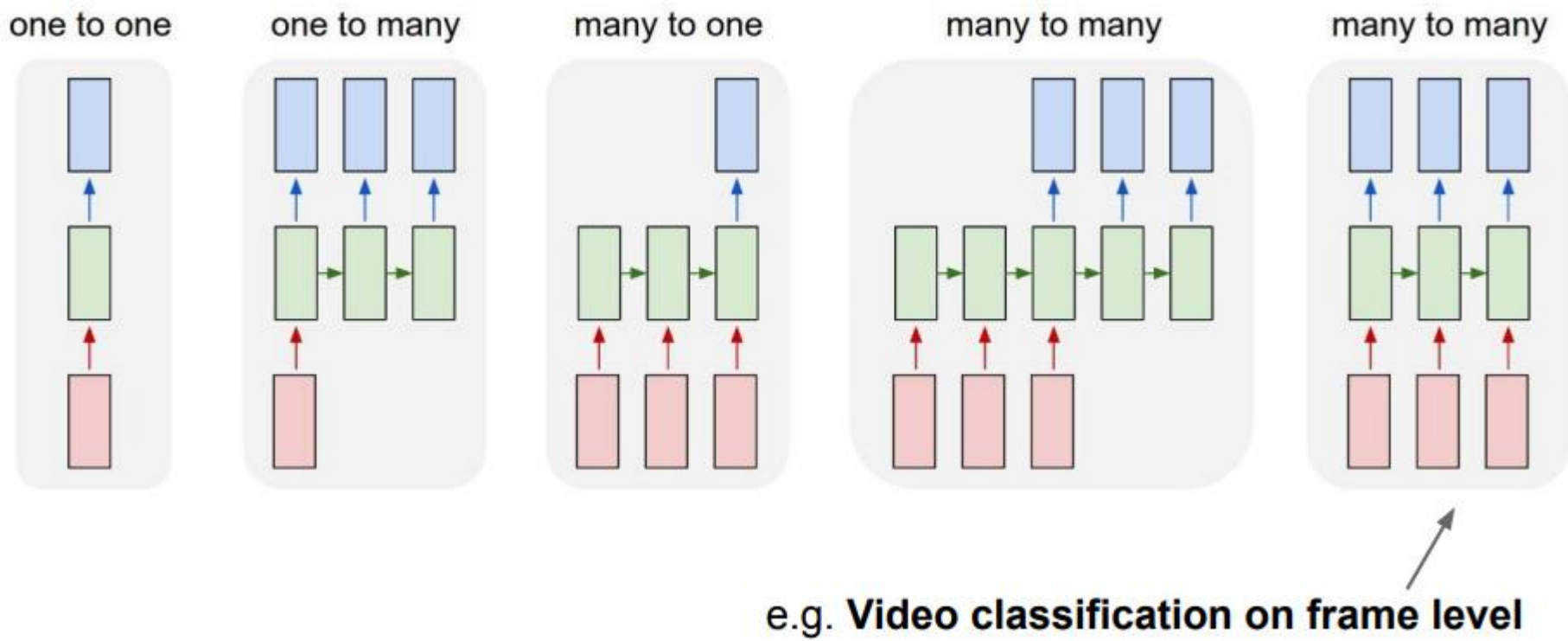
e.g. **Sentiment Classification**
sequence of words -> sentiment

RNN 기본 모델 유형

- 언어 모델(language model)
  - 일정 크기의 문장을 입력하면 이에 상응하여 한 글자씩 다음 위치에 있는 글자나 단어를 출력
  - 매 입력마다 출력이 발생



RNN 기본 모델 유형


one to one
one to many
many to one
many to many
many to many
e.g. **Video classification on frame level**

RNN 모델 – 인코더-디코더 (many to many)

- 인코더-디코더 모델
  - 일종의 시퀀스-시퀀스 모델
  - 하나의 문장을 모두 듣고 이를 번역 또는 다른 대답을 하는 모델 등에 사용

- 시퀀스-시퀀스 모델
  - 하나의 시퀀스를 모드 입력하면 그 이후에 출력이 시퀀스 형태로 나오는 것

- 인코딩 된 결과는 디코더로 전달
  - context vector 또는 machine thought 벡터 형태로 전달

- teacher forcing
  - 예측한 출력이 아니라 실제 레이블 값으로 학습시키는 방법

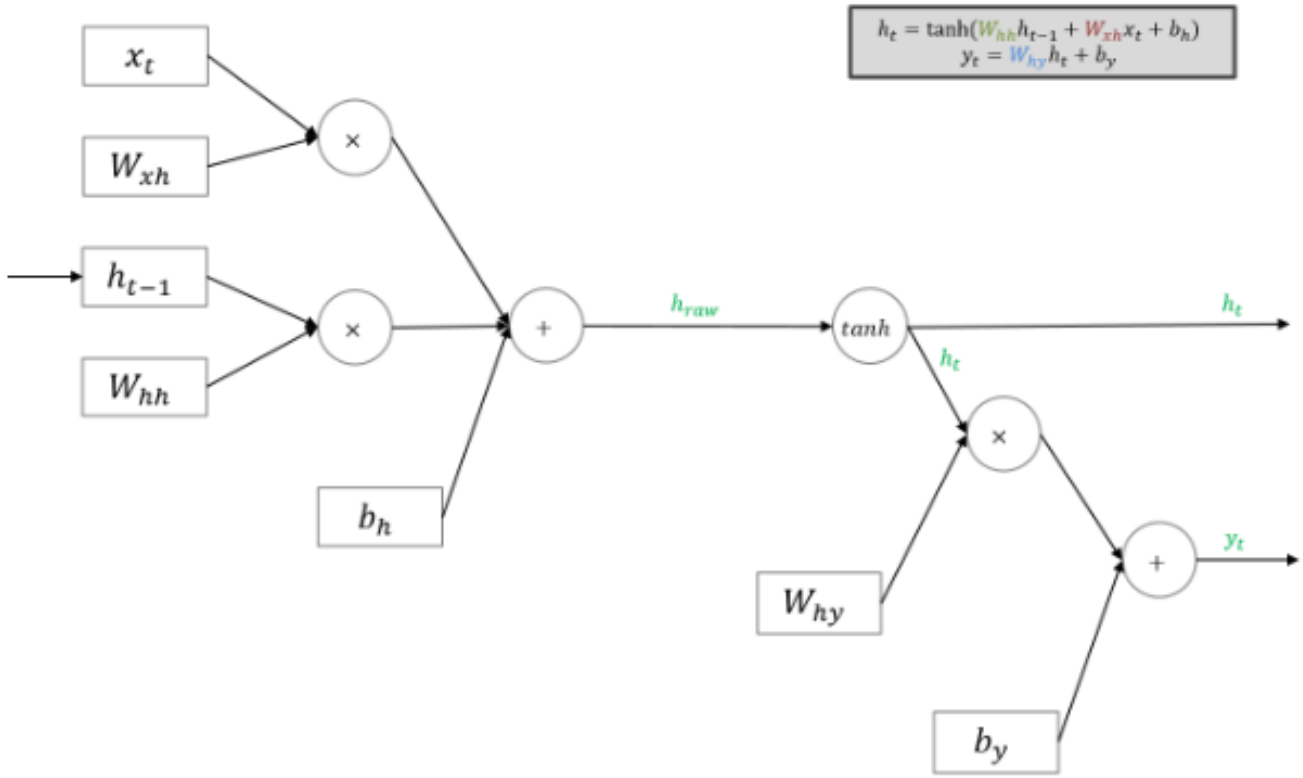
RNN 학습 – BPTT

- 경사하강(gradient descent) 방식의 학습을 주로 사용
  - 오류역전파 방식을 이용
  - 모델의 출력과 레이블 사이에 오류가 발생하며 이로부터 계산된 손실의 경사값을 앞 단쪽의 계층으로 전달하여 가중치를 조절

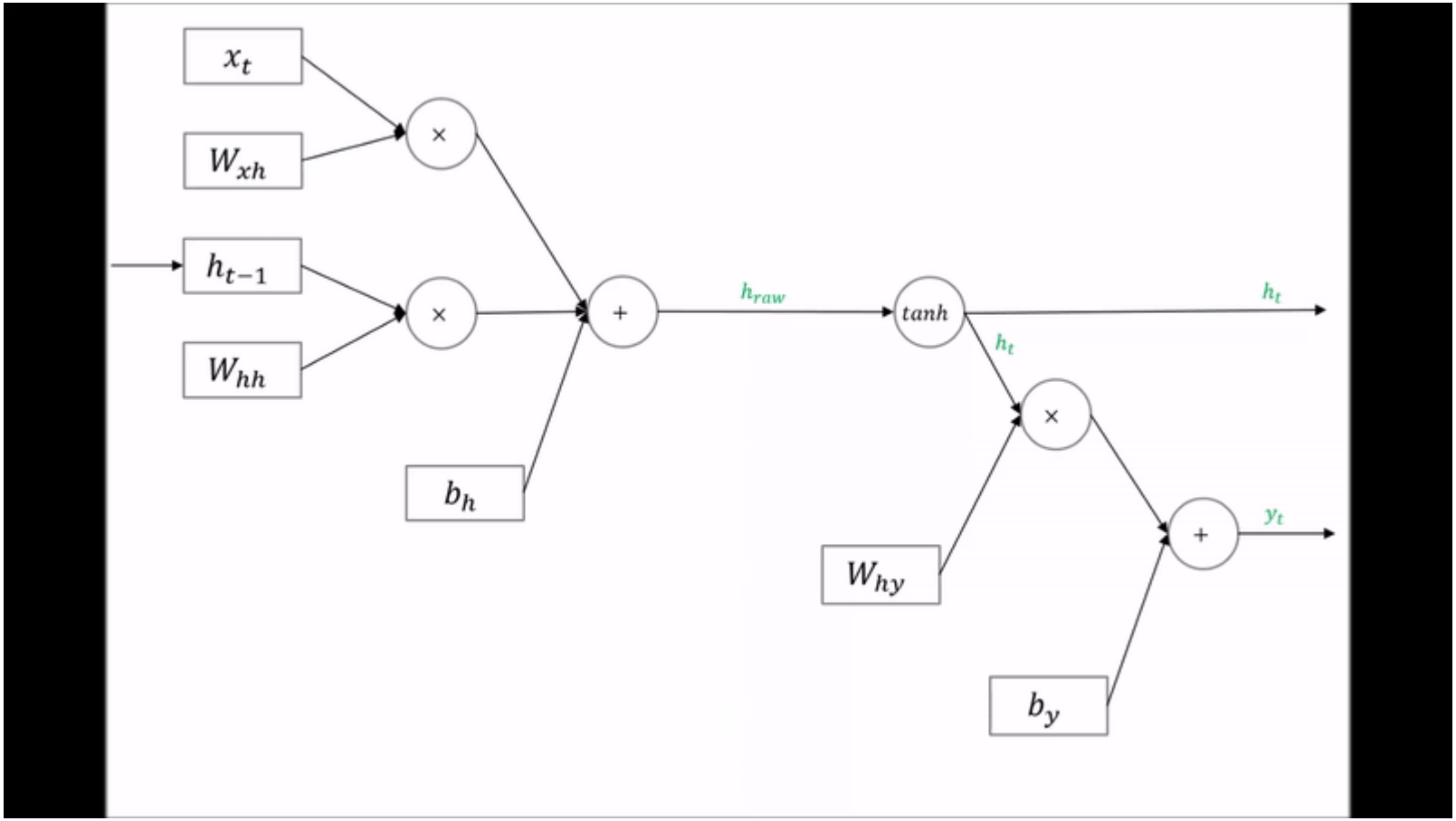
- 학습에 의해서 가중치 매트릭스 W, U, V가 업데이트 되는데 RNN에서는 모든 타임 스텝에서 동일한 가중치를 사용한다는 것을 주의해야 한다.

- 즉, 현재의 경사(gradient) 변화는 과거 스텝의 계산에도 영향을 미친다. 이러한 현상을 BPTT(backpropagation through time)이라고 함

RNN – 순전파


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

RNN – 역전파



RNN의 문제점

- RNN에서 여러 계층을 거치는 경우, 예를 들어 단어가 수십 개로 된 문장을 해석하는 경우, 오차 역전파를 하면서 경사값이 거의 사라지거나 또는 너무 큰 값으로 발산하여 RNN이 제대로 동작하지 못하기 쉽다
  - 오래된 정보를 모두 중요시하면 정보가 너무 많이 축적되어 발산할 우려
  - 오래된 정보를 약하게 반영하면 오래되었지만 중요한 정보를 캐치하지 못하는 즉, 소실되는 우려

- 이를 해결하기 위해서
  - LSTM(Long-Short Term Memory), GRU 기법이 제안
  - LSTM에서는 오랜기간 중요도를 유지할 정보와 그럴 필요 없이 망각해야 할 신호를 구분하여 따로 처리

Q&A