

RNN 이론

2024.01.30

이은주

Intro

Recurrent Neural Networks

Previous methods

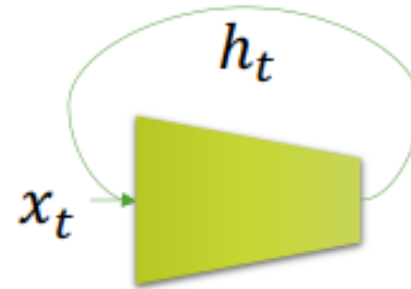
- e.g. Tabular Data, Image



$$y = f(x; \theta)$$

Recurrent Neural Networks

- e.g. Sequential Data, Time-series



$$h_t = f(x_t, h_{t-1}; \theta)$$



y

Sequential Data vs Time Series

Sequential data : 데이터의 **순서** 정보가 중요

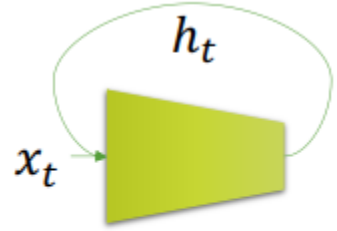
- 텍스트(단어의 순서)
- (샘플링 주기가 일정한) 영상/음성

"I am so happy" : am은 i없이 처리 할 수 x. happy는 am과 so없이 처리 할 수 x.

Time series : 해당 데이터가 발생한 **시각** 정보가 중요

- 주식 데이터(가격의 순서 및 발생 시점)
- 센서 데이터

RNN

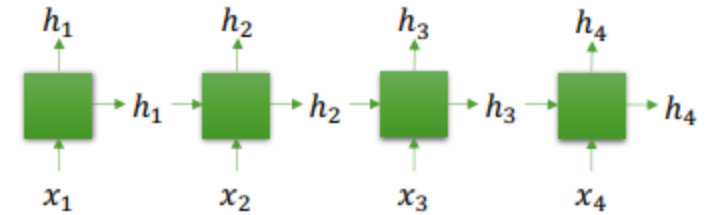


$$\hat{y}_t = h_t = f(x_t, h_{t-1}; \theta)$$

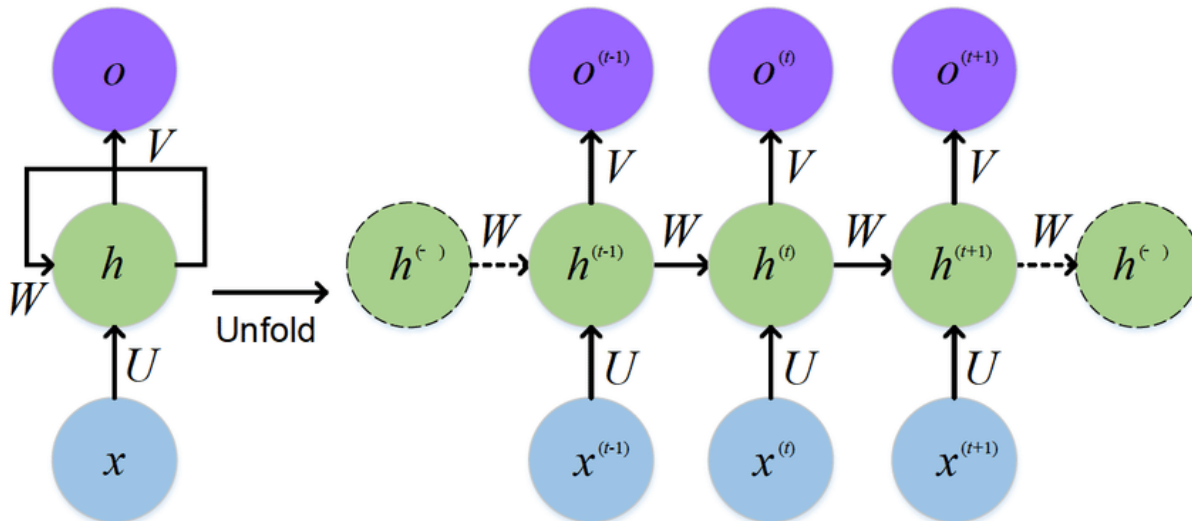
$$= \tanh(W_{ih}x_t + b_{ih} + W_{hh}h_{t-1} + b_{hh})$$

where $\theta = \{W_{ih}, b_{ih}, W_{hh}, b_{hh}\}$.

$$h_t = f(x_t, h_{t-1}; \theta)$$



RNN은 자기자신에게 혹은 그 전 노드로 계산되는 구조

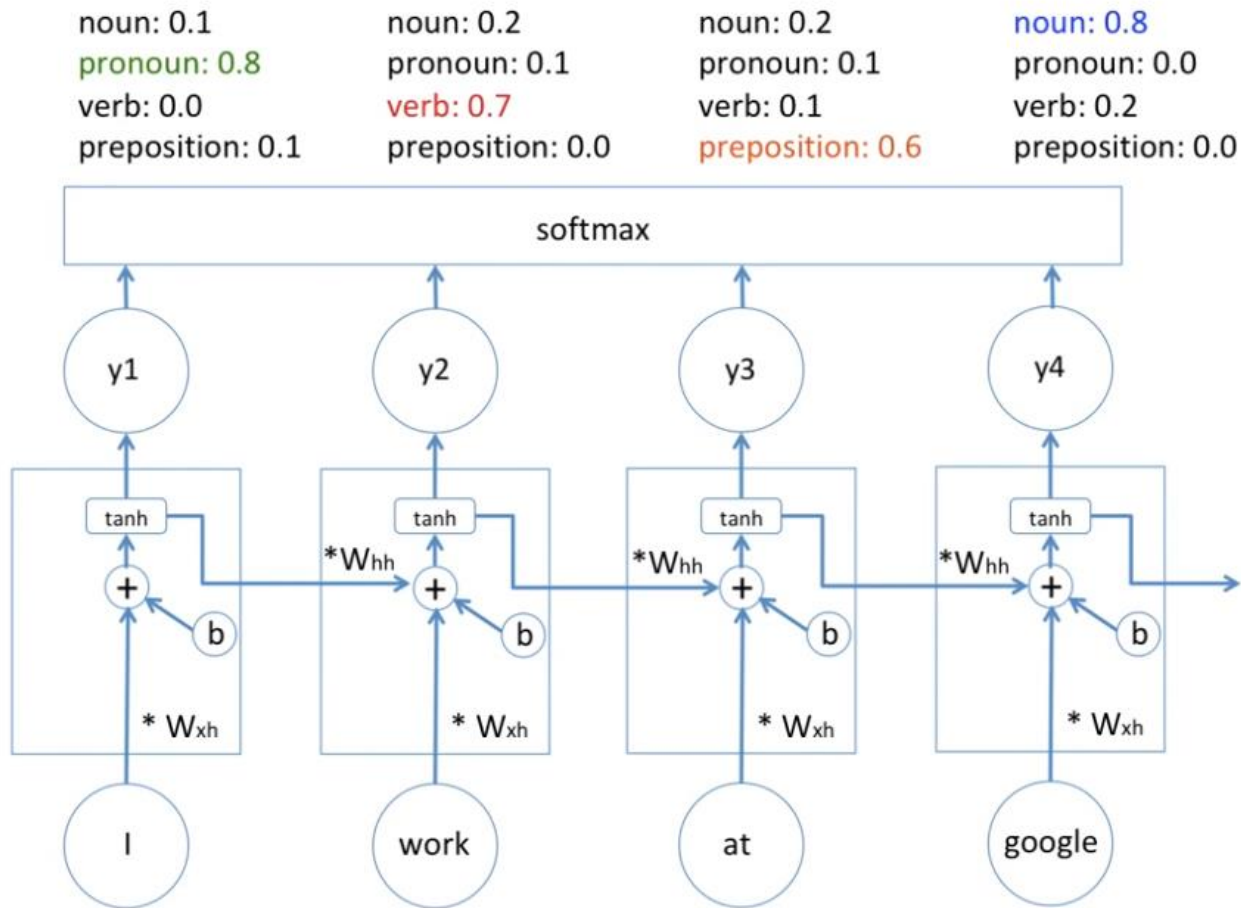


X : input

Y : output

Output과 state가 동일한 값을 가지고 나감

RNN 학습과정



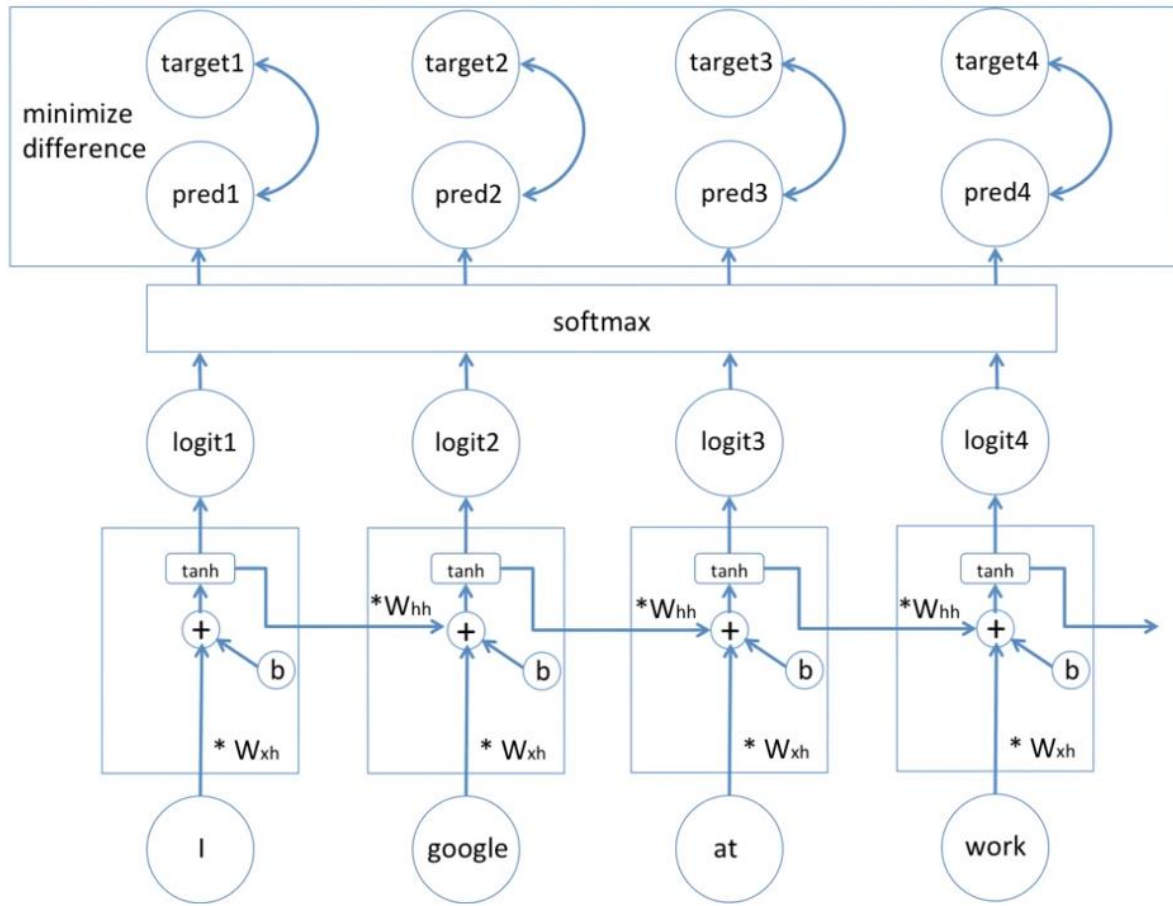
한 cell에는 두가지 weight이 존재 : W_{xh} , W_{hh} .

Input들어올 때 W_{xh} 와 곱해짐.

과거의 값이 들어올 때 W_{hh} 와 곱해짐.
두 개의 값을 더해 h가 나옴.

Softmax함수를 통해 나온 결과로 예측.
각 확률의 총 합은 1.

RNN 학습과정



Softmax input : logit

Softmax output : pred

Pred과 target을 줄여나가면서 학습 진행.

➔ W_{xh} , b , W_{hh} 최적화 시켜준다.

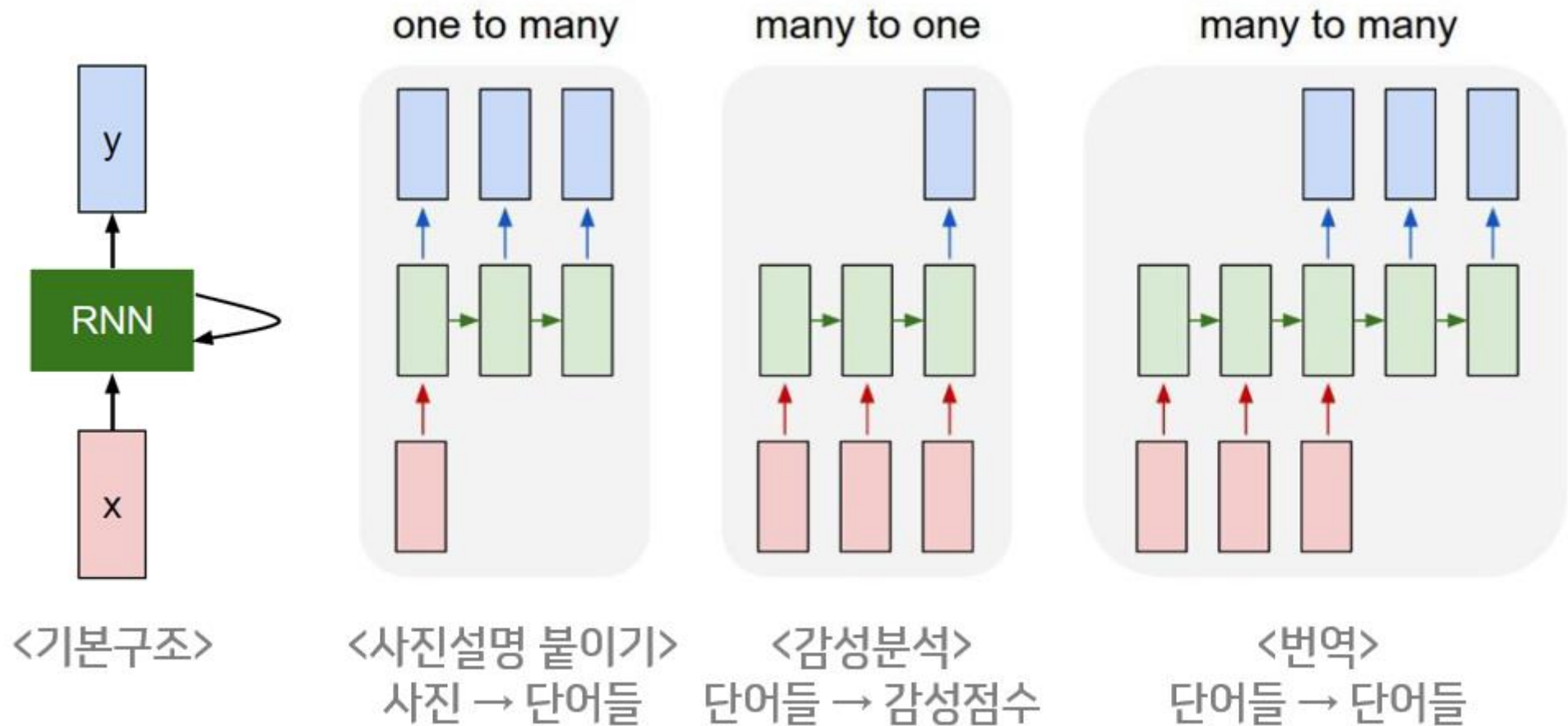
➔ Gradient descent로 조금씩 변형시켜줌.

파라미터(W_{xh} , b , W_{hh}) 각각은 하나의 변수.
여기서 W_{xh} 가 4번 나왔다고 W_{xh} 변수가 4개인 것 x.

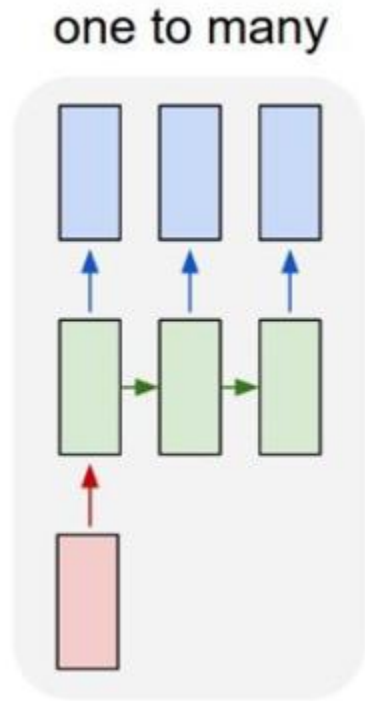
하나의 변수의 변형이므로

Backpropagation Through Time (BPTT)라 불린다.

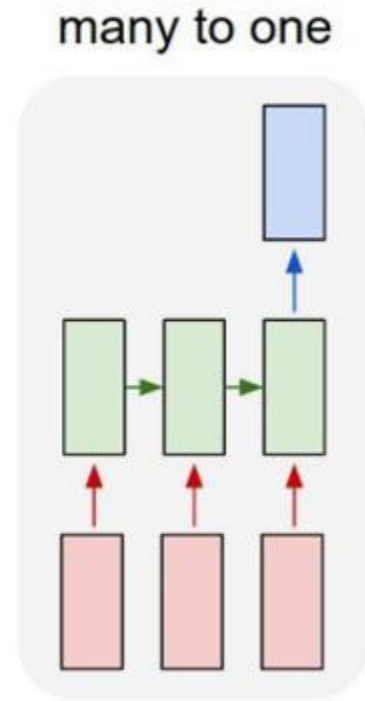
RNN의 다양한 구조



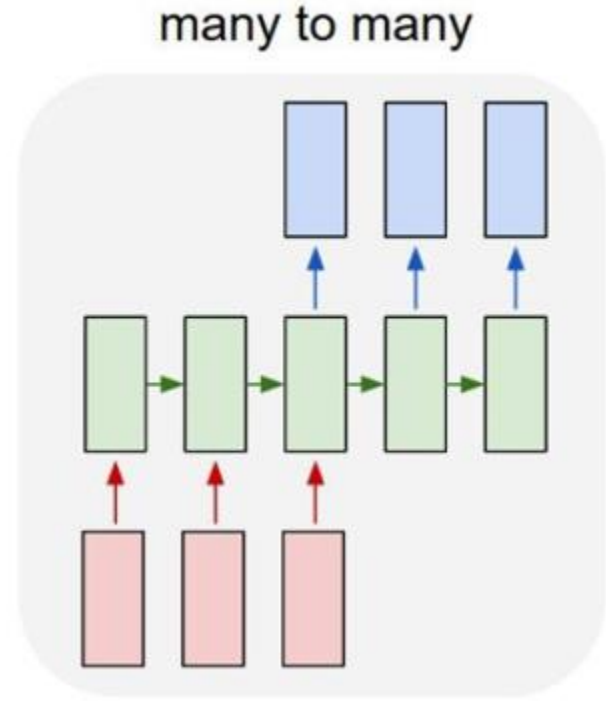
RNN의 다양한 구조



하나의 이미지 입력
-> 사진의 제목 출력
(제목은 단어들의 나열)



입력문장에 대해
긍정/부정 판별



번역, 챗봇, 품사 예측
문장 입력 -> 문장 출력

RNN의 장단점

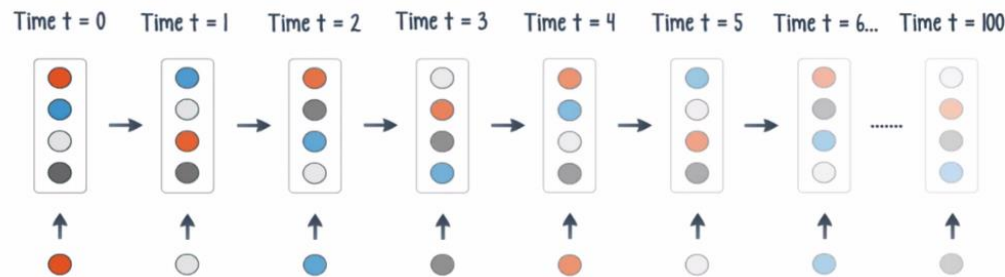
장점

- input의 길이에 제한 x
- 현재 상태 뿐만 아니라 이전 상태의 정보 활용

단점

- Long term dependency 문제 및 vanishing gradient 문제 :

Decay of information through time



input과 output 차이가 커질 경우 gradient 작아짐. -> gradient 사라진다.
=> LSTM, GRU로 해결