RNN (Recurrent Neural Network)

노민준

nmh0408@g.skku.edu

NLP Study

2024/03/26

TRAIN AND TEST

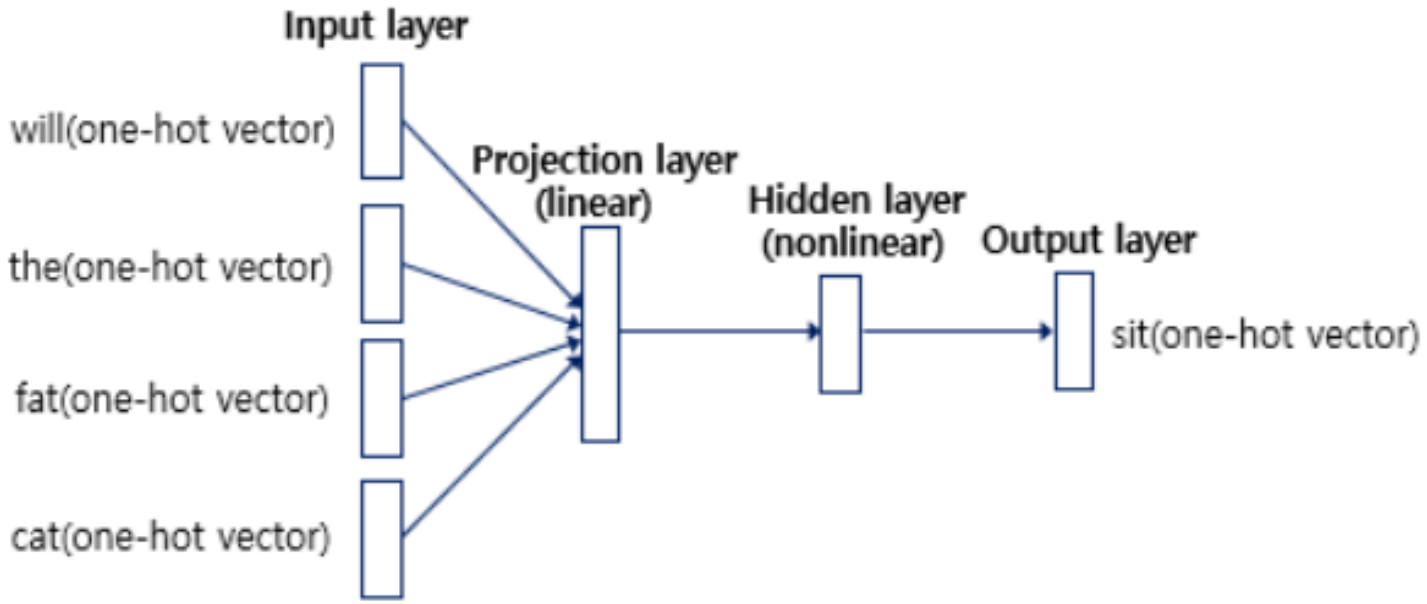
Contents

TRAIN AND TEST

피드 포워드 신경망 언어 모델 (NNLM)

NNLM

- Input layer – Projection layer – Hidden layer – Output layer 로 구성



TRAIN AND TEST

피드 포워드 신경망 언어 모델 (NNLM)

Input Layer

- 모든 단어들을 One-Hot vector로 생성
- Window의 크기만큼 Input Layer에 vector들을 넣어줌

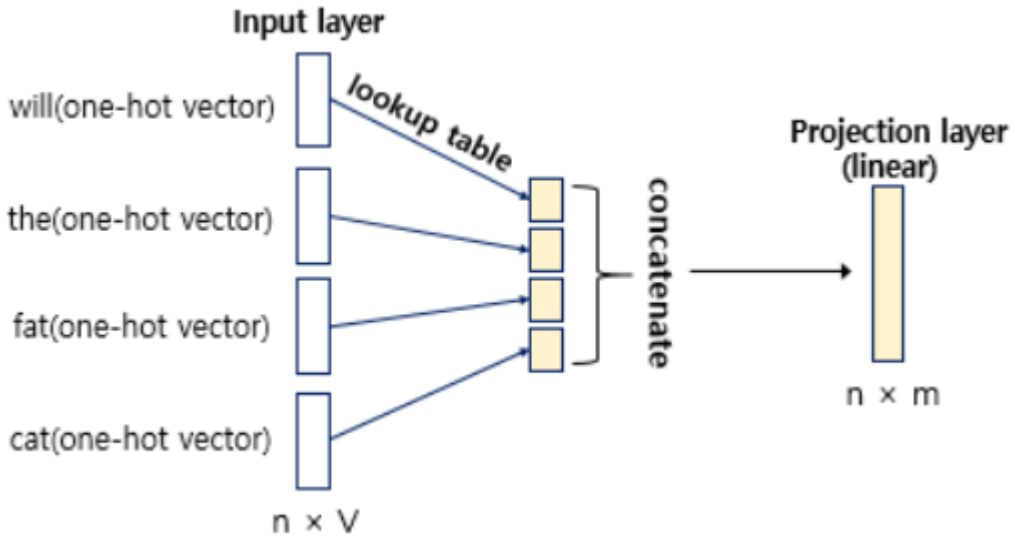
```
what = [1, 0, 0, 0, 0, 0, 0]
will = [0, 1, 0, 0, 0, 0, 0]
the = [0, 0, 1, 0, 0, 0, 0]
fat = [0, 0, 0, 1, 0, 0, 0]
cat = [0, 0, 0, 0, 1, 0, 0]
sit = [0, 0, 0, 0, 0, 1, 0]
on = [0, 0, 0, 0, 0, 0, 1]
```

TRAIN AND TEST

피드 포워드 신경망 언어 모델 (NNLM)

Projection Layer

- One-Hot vector에 따라 가중치에서 Lookup Table 생성됨
- 생성된 Lookup Table들을 옆으로 이어 붙임 (Concatenate)



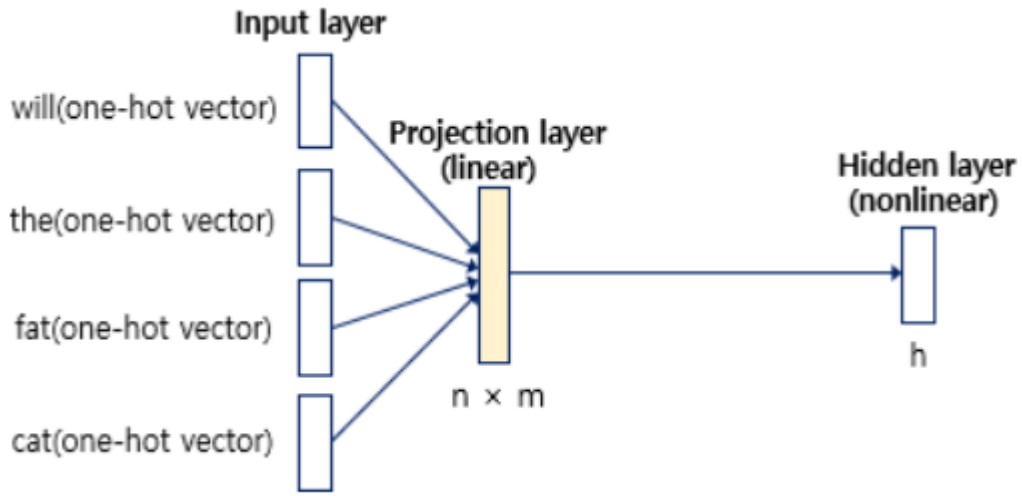
$$p^{layer} = (lookup(x_{t-n}); \ldots; lookup(x_{t-2}); lookup(x_{t-1})) = (e_{t-n}; \ldots; e_{t-2}; e_{t-1})$$

TRAIN AND TEST

피드 포워드 신경망 언어 모델 (NNLM)

Hidden Layer

- Projection Layer에서 생성된 Vector를 활성화 함수에 넣어줌

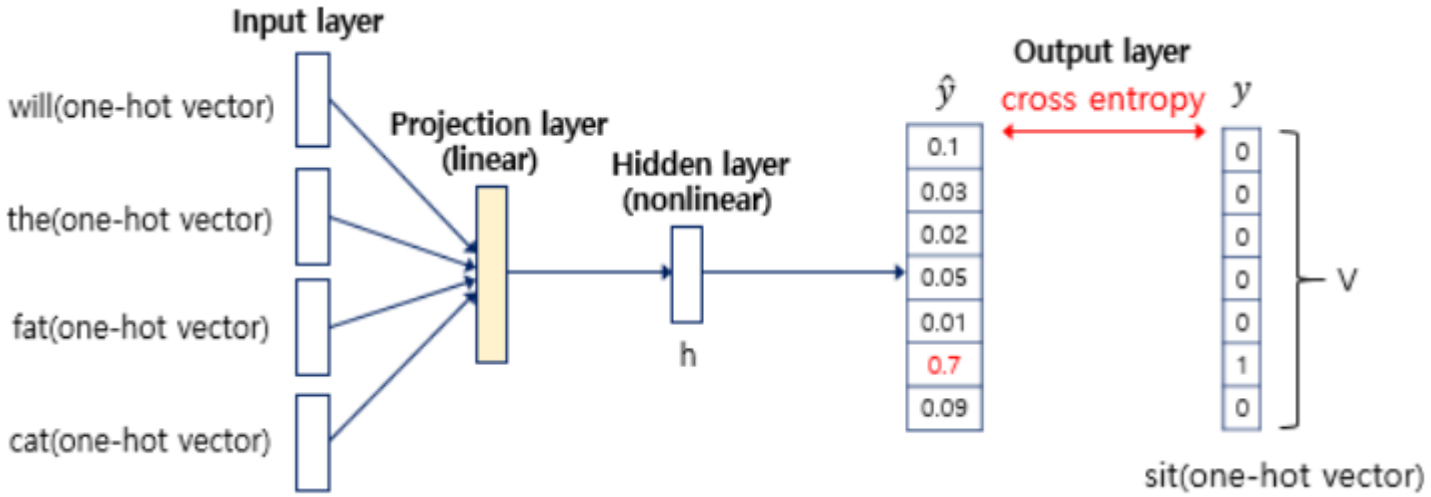


TRAIN AND TEST

피드 포워드 신경망 언어 모델 (NNLM)

Output Layer

- 손실함수를 이용해 가중치들을 학습



피드 포워드 신경망 언어 모델 (NNLM)

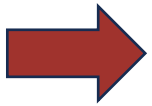
NNLM의 장단점

<장점>

- 충분한 양의 Corpus들을 학습한다면, 결국 유사한 목적으로 사용되는 단어들은 유사한 Embedding Vector값들을 가지게 될 것

<단점>

- Window의 크기를 정해주는 과정에서, 다음 단어를 예측할 때, 정해진 n개의 단어만 참고 가능



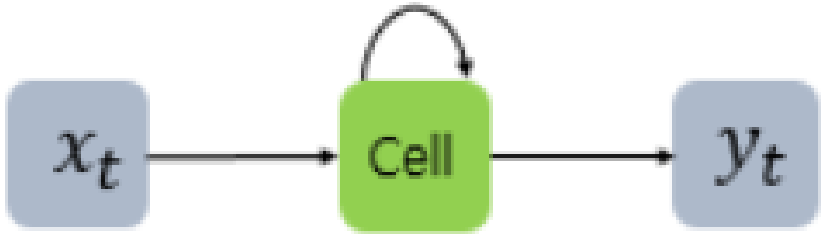
다양한 길이의 입력들을 처리할 수 있는 모델이 필요해짐

TRAIN AND TEST

순환 신경망(RNN)

RNN

- 기존의 신경망들은 전부 입력층에서 출력층 방향의 단방향으로만 이동
- RNN은 이와 달리, 이전 시점(t-1)에서 은닉층에서 나온 값을 다시 입력으로 받을 수 있음

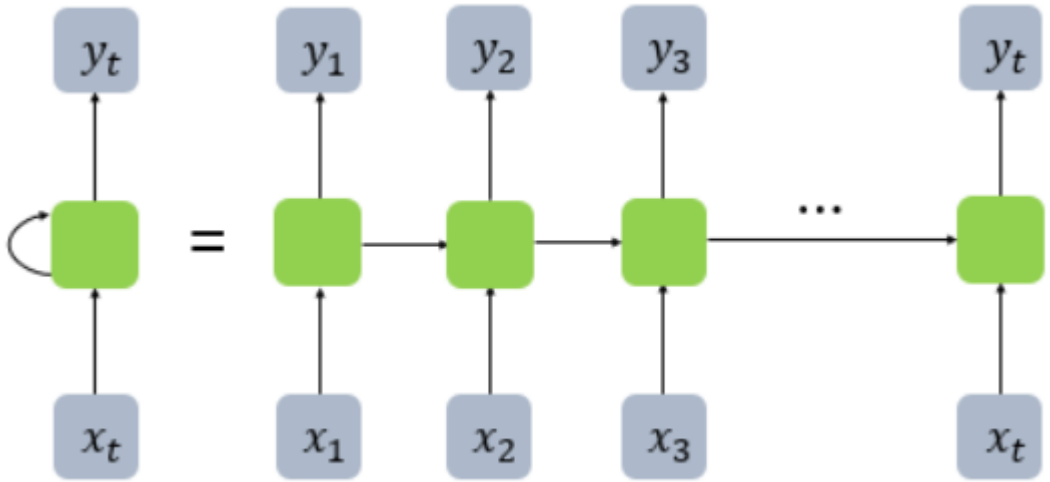


TRAIN AND TEST

순환 신경망(RNN)

RNN

- 기존의 신경망들은 전부 입력층에서 출력층 방향의 단방향으로만 이동
- RNN은 이와 달리, 이전 시점(t-1)에서 은닉층에서 나온 값을 다시 입력으로 받을 수 있음

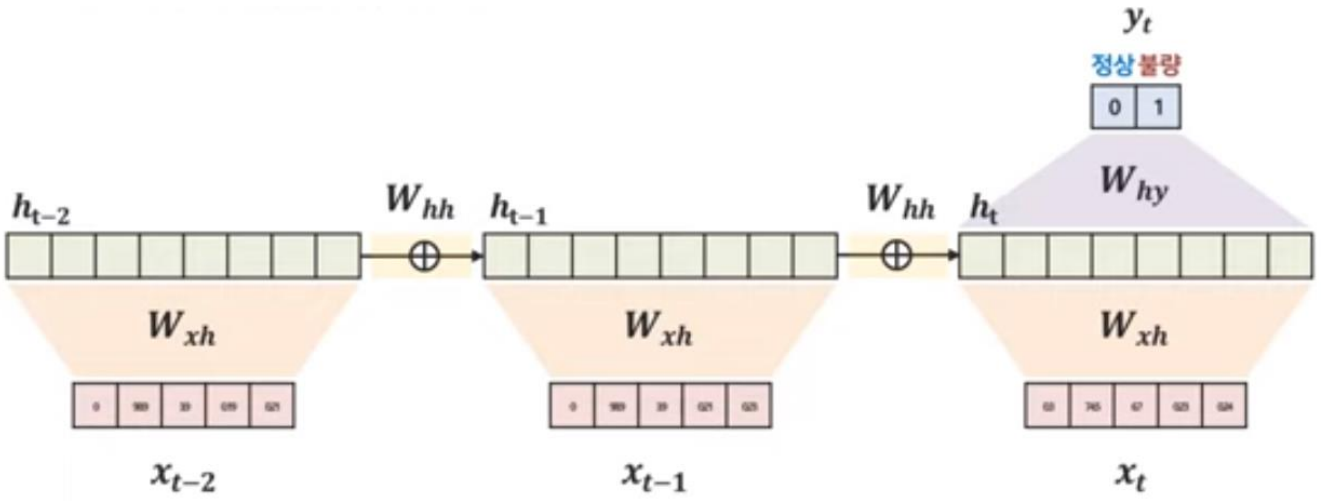


TRAIN AND TEST

순환 신경망(RNN)

RNN

- 기존의 신경망들은 전부 입력층에서 출력층 방향의 단방향으로만 이동
- RNN은 이와 달리, 이전 시점(t-1)에서 은닉층에서 나온 값을 다시 입력으로 받을 수 있음



$$h_{t-1} = f(W_{xh}x_{t-1} + W_{hh}h_{t-2})$$

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

$$y_t = g(W_{hy}h_t)$$

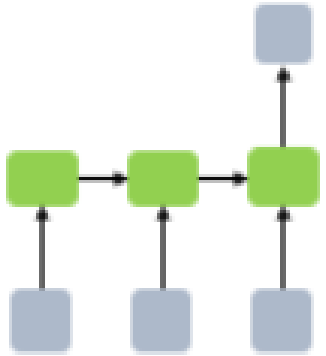
$$f(\cdot) = tanh\ ,\ g(\cdot) = softmax$$

TRAIN AND TEST

RNN의 다양한 구조

다 대 일 구조

- 여러 시점 X로 하나의 Y를 예측
- EX) 여러 시점의 센서 데이터를 이용해, 특정 시점의 제품 상태 예측, text classification



다 대 일(many-to-one)

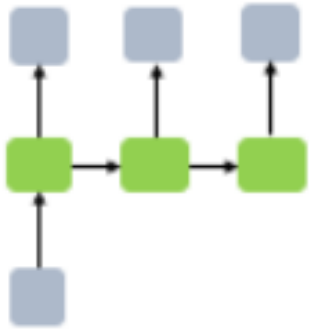
	시간	센서1	센서2	센서3	센서4	센서5	상태
$t-2$	1200	0	989	39	019	021	정상
$t-1$	1300	0	989	39	021	023	정상
t	1400	03	745	67	023	024	불량

TRAIN AND TEST

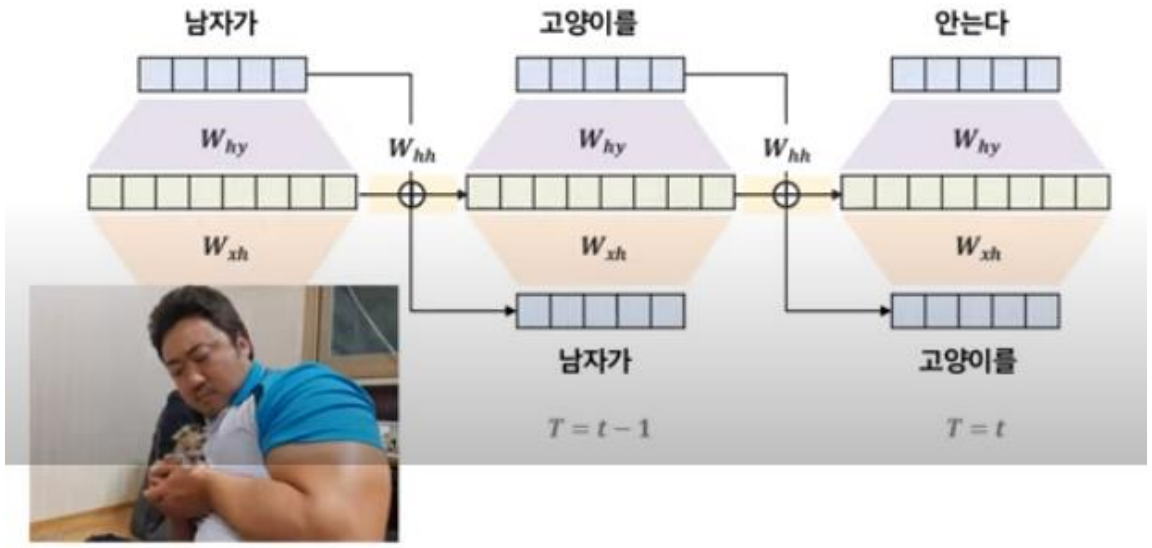
RNN의 다양한 구조

일 대 다 구조

- 단일 시점 X로 순차적인 Y를 예측
- EX) 이미지 데이터가 주어질 때, 이미지에 대한 정보를 글로 생성 (이미지 캡셔닝)



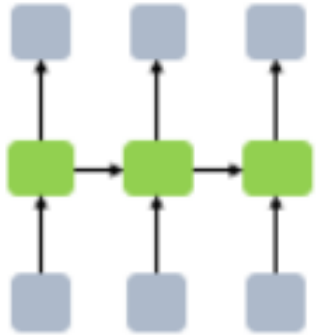
일 대 다(one-to-many)



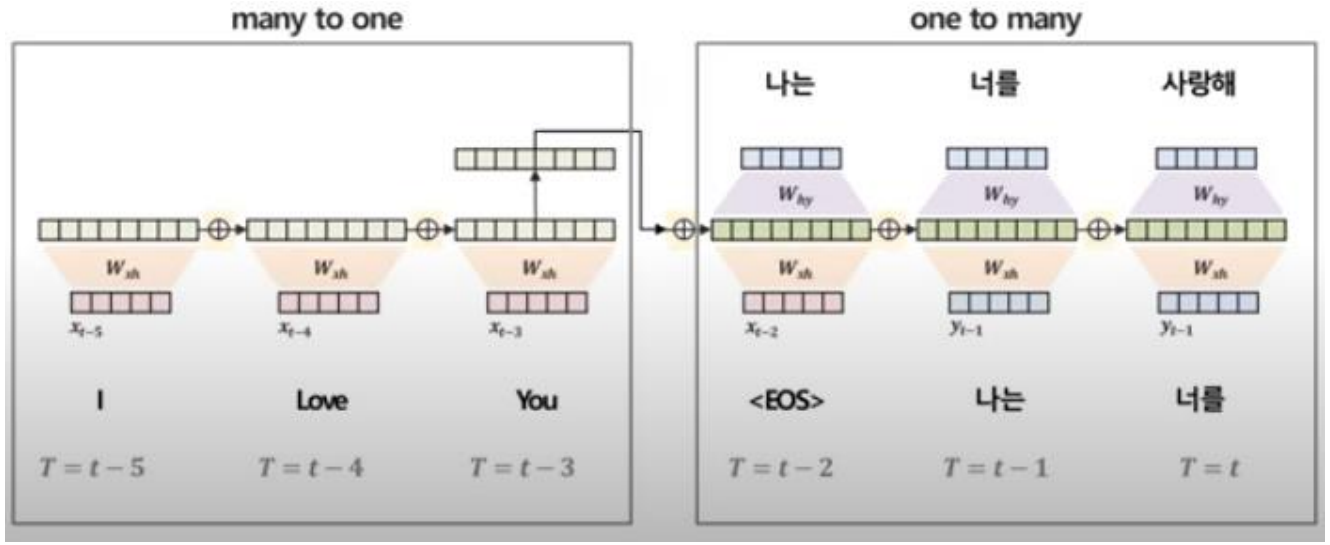
RNN의 다양한 구조

다 대 다 구조

- 순차적인 X로 순차적인 Y를 예측
- EX) 영어 문장이 주어질 때, 한글 문장으로 번역



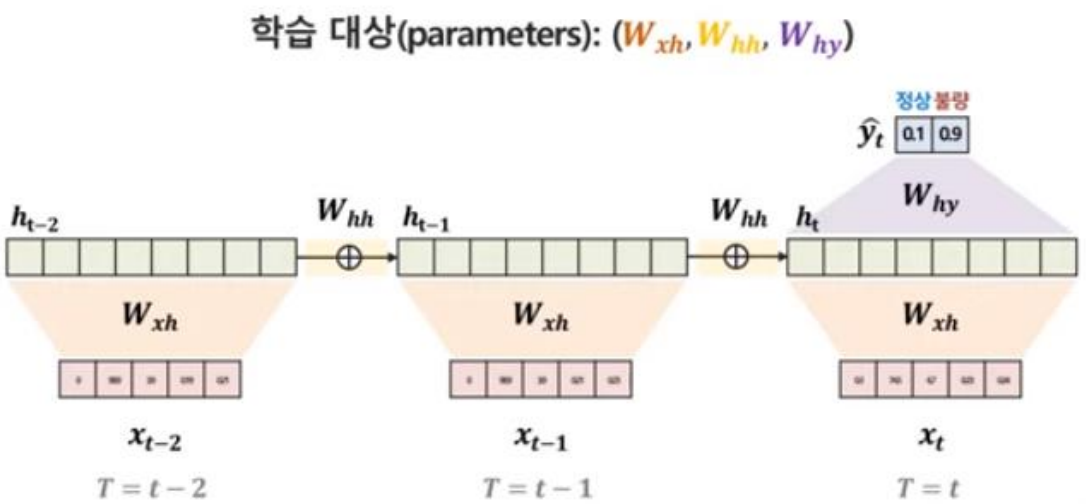
다 대 다(many-to-many)



RNN의 학습

학습 대상

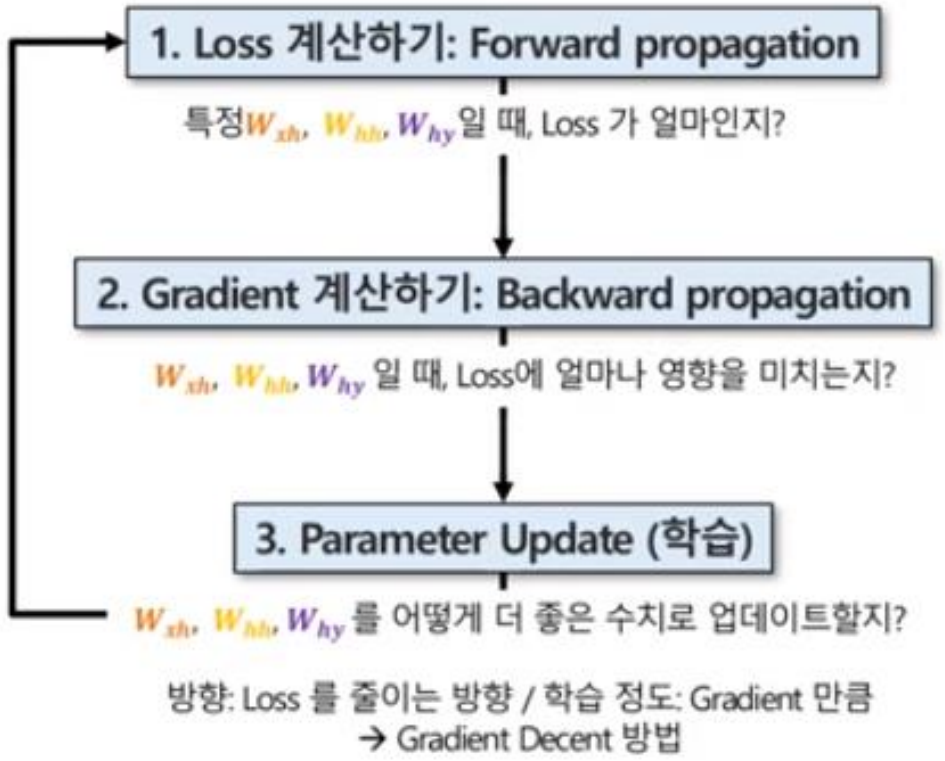
- 입력 데이터 X에 곱해지는 가중치 W_{xh}
- 은닉층의 이전 값(h_{t-1})에 곱해지는 가중치 W_{hh}
- 은닉층의 현재 값(h_t)에 곱해져 $\hat{y}$를 만드는 가중치 W_{hy}



TRAIN AND TEST

RNN의 학습

학습 순서

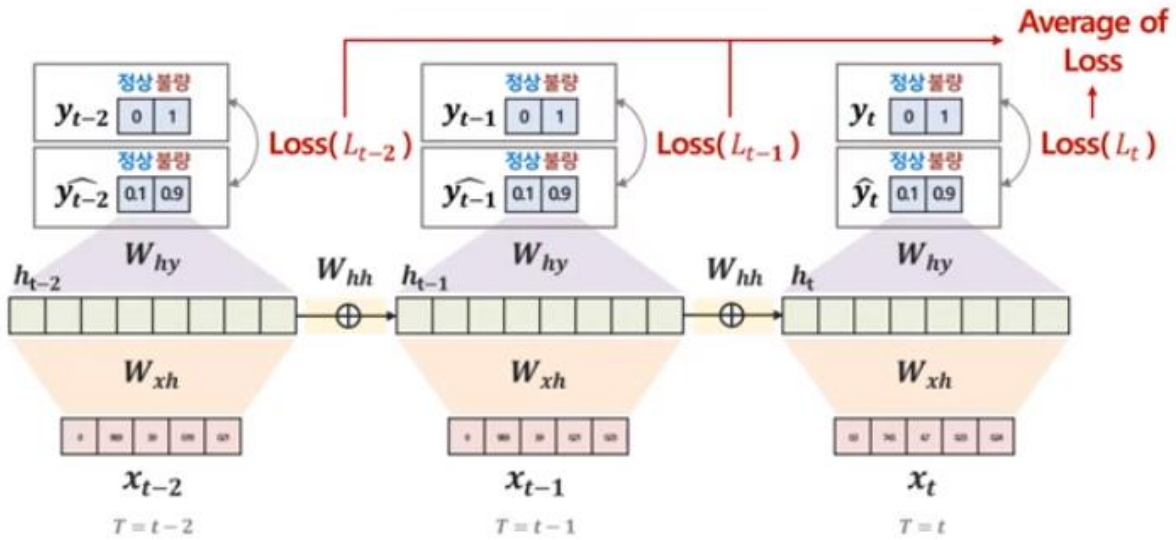


TRAIN AND TEST

RNN의 학습

1. Loss 계산하기

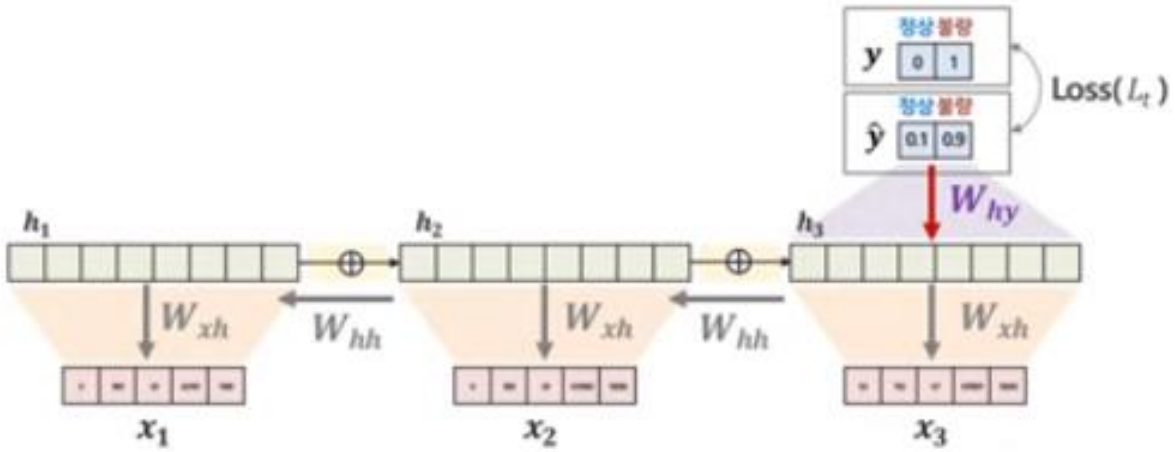
- $L_t = y_t - \hat{y_t}$ 로 계산
- 만약, 다대다 구조 또는 일대다 구조라면, 각 L_t들의 평균을 활용



TRAIN AND TEST

RNN의 학습

2. Gradient 계산하기 (W_{hy})



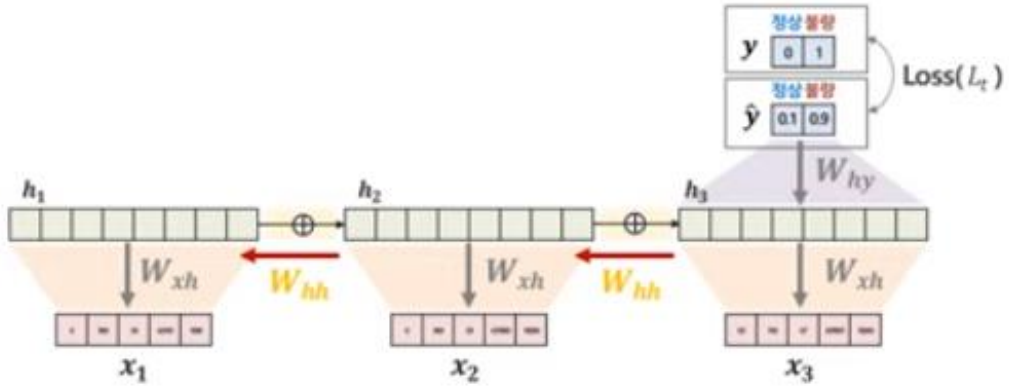
$$\frac{\partial Loss}{\partial W_{hy}} = \frac{\partial L_t}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial W_{hy} h_t} \times \frac{\partial W_{hy} h_t}{\partial W_{hy}}$$

TRAIN AND TEST

RNN의 학습

2. Gradient 계산하기 (W_{hh})

- T_3, T_2, T_1 에서 W_{hh}의 영향을 모두 고려해 Gradient 계산 후, 전부 더해줌



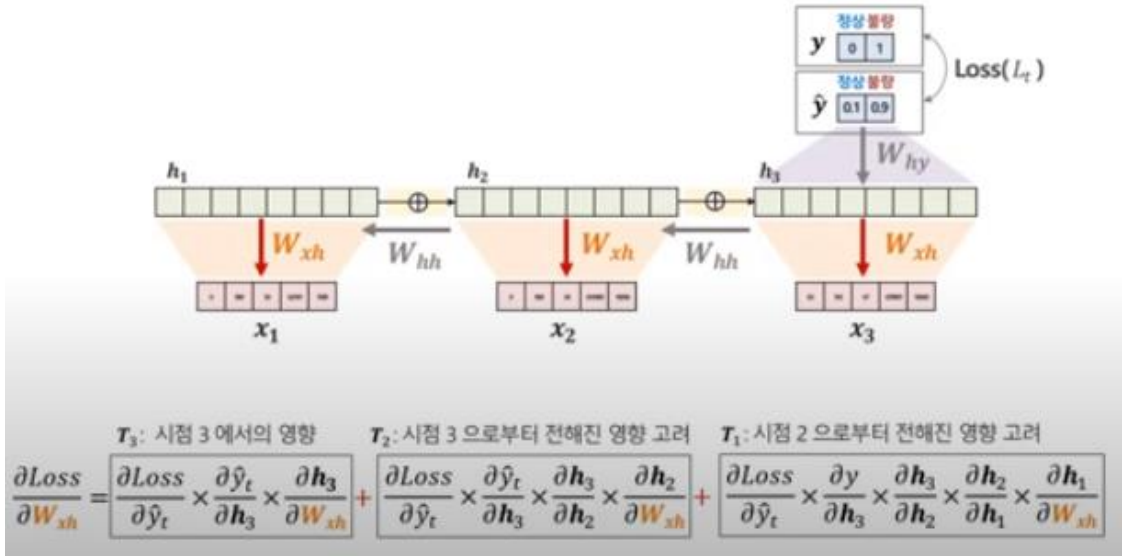
$$\frac{\partial Loss}{\partial W_{hh}} = \underbrace{\frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_3} \times \frac{\partial h_3}{\partial W_{hh}}}_{T_3\text{: 시점 3 에서의 영향}} + \underbrace{\frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_3} \times \frac{\partial h_3}{\partial h_2} \times \frac{\partial h_2}{\partial W_{hh}}}_{T_2\text{: 시점 3 으로부터 전해진 영향 고려}} + \underbrace{\frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_3} \times \frac{\partial h_3}{\partial h_2} \times \frac{\partial h_2}{\partial h_1} \times \frac{\partial h_1}{\partial W_{hh}}}_{T_1\text{: 시점 2 으로부터 전해진 영향 고려}}$$

TRAIN AND TEST

RNN의 학습

2. Gradient 계산하기 (W_{xh})

- 마찬가지로 T_3, T_2, T_1 에서 W_{hh}의 영향을 모두 고려해 Gradient 계산 후, 전부 더해줌



TRAIN AND TEST

RNN의 학습

2. Gradient 계산하기 (일반식)

$$\frac{\partial Loss}{\partial W_{hy}} = \frac{\partial L_t}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial W_{hy} h_t} \times \frac{\partial W_{hy} V h_t}{\partial W_{hy}}$$

$$\frac{\partial Loss}{\partial W_{hh}} = \sum_{k=0}^{t} \left(\frac{\partial L}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_t} \times \frac{\partial h_t}{\partial h_k} \times \frac{\partial h_k}{\partial W_{hh}} \right)$$

$$\frac{\partial Loss}{\partial W_{xh}} = \sum_{k=0}^{t} \left(\frac{\partial L}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_t} \times \frac{\partial h_t}{\partial h_k} \times \frac{\partial h_k}{\partial W_{xh}} \right)$$

시간 방향으로 펼친 신경망의
역전파를 수행

=>

BPTT (Back Propagation Through Time) 방식

TRAIN AND TEST

RNN의 학습

3. Parameter 업데이트

- 학습률(Learning rate) = η
- 앞서 구한 Gradient 대입

$$W_{hy}\text{의 기여도} = \frac{\partial Loss}{\partial W_{hy}}$$
$$\rightarrow W_{hy}^{new} = W_{hy}^{old} - \eta * \frac{\partial Loss}{\partial W_{hy}}$$

$$W_{hh}\text{의 기여도} = \frac{\partial Loss}{\partial W_{hh}}$$
$$\rightarrow W_{hh}^{new} = W_{hh}^{old} - \eta * \frac{\partial Loss}{\partial W_{hh}}$$

$$W_{xh}\text{의 기여도} = \frac{\partial Loss}{\partial W_{xh}}$$
$$\rightarrow W_{xh}^{new} = W_{xh}^{old} - \eta * \frac{\partial Loss}{\partial W_{xh}}$$

TRAIN AND TEST

RNN의 한계

1. BPTT

- 시점의 길이에 비례해서, BPTT가 소비하는 컴퓨팅 자원이 증가
- ⇒ Truncated BPTT의 사용

- Truncated BPTT : 신경망 연결을 적당한 길이로 끊고, 잘라낸 신경망에서 역전파를 수행하는 것

TRAIN AND TEST

RNN의 한계

2. 기울기 폭발

- 길이가 길어지면서, 신경망을 하나 통과할 때마다 기울기가 너무 커지는 현상
- Overflow를 일으켜 NaN값 유발 가능
- ⇒ 기울기 클리핑의 사용

- 기울기 클리핑 : 임계값(Threshold)를 정하고, 기울기가 이 값을 초과하면 값을 작게 수정해주는 방법

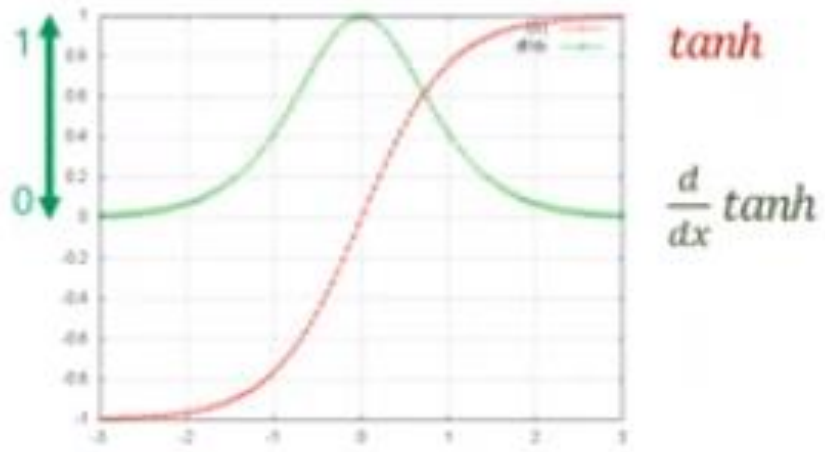
$$if\ \|\hat{g}\| \geq threshold :$$

$$\hat{g} = \frac{threshold}{\|\hat{g}\|}\hat{g}$$

TRAIN AND TEST

RNN의 한계

3. 기울기 소실 문제



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

→ 기울기가 0 ~ 1 사이의 값을 가짐

TRAIN AND TEST

RNN의 한계

3. 기울기 소실 문제

T_{100}: 시점 100 에서의 영향 T_{99}: 시점 100 으로부터 전해진 영향 고려 T_{98}: 시점 99 으로부터 전해진 영향 고려

$$\frac{\partial Loss}{\partial W_{hh}} = \boxed{\frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_{100}} \times \frac{\partial h_{100}}{\partial W_{hh}}} + \boxed{\frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_{100}} \times \frac{\partial h_{100}}{\partial h_{99}} \times \frac{\partial h_{99}}{\partial W_{hh}}} + \boxed{\frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial y}{\partial h_{100}} \times \frac{\partial h_{100}}{\partial h_{99}} \times \frac{\partial h_{99}}{\partial h_{98}} \times \frac{\partial h_{98}}{\partial W_{hh}}}$$

T_1: 시점 2 으로부터 전해진 영향 고려

$$+ \boxed{\frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial y}{\partial h_{100}} \times \frac{\partial h_{100}}{\partial h_{99}} \times \cdots \times \frac{\partial h_6}{\partial h_5} \times \frac{\partial h_5}{\partial h_4} \times \frac{\partial h_4}{\partial h_3} \times \frac{\partial h_3}{\partial h_2} \times \frac{\partial h_2}{\partial h_1} \times \frac{\partial h_1}{\partial W_{hh}}}$$

W_{hh}의 기여도 $= \frac{\partial Loss}{\partial W_{hh}}$

$$\rightarrow W_{hh}^{new} = W_{hh}^{old} - \eta * \frac{\partial Loss}{\partial W_{hh}}$$

$$\rightarrow W_{hh}^{new} = W_{hh}^{old} - \eta * \left(\frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial \hat{y}_t}{\partial h_{100}} \times \frac{\partial h_{100}}{\partial W_{hh}} + \cdots + \frac{\partial Loss}{\partial \hat{y}_t} \times \frac{\partial y}{\partial h_{100}} \times \frac{\partial h_{100}}{\partial h_{99}} \times \cdots \times \frac{\partial h_6}{\partial h_5} \times \frac{\partial h_5}{\partial h_4} \times \frac{\partial h_4}{\partial h_3} \times \frac{\partial h_3}{\partial h_2} \times \frac{\partial h_2}{\partial h_1} \times \frac{\partial h_1}{\partial W_{hh}}\right)$$

- Gradient의 뒷부분은 거의 0에 수렴하게 됨
- 앞 시점의 영향은 거의 반영되지 X
- Sequence가 길어질수록, 앞쪽의 정보 학습이 어려워짐

TRAIN AND TEST

RNN의 한계

3. 기울기 소실 문제

- 은닉층에 게이트를 추가하는 방식으로 해결 가능
- ⇒ LSTM, GRU의 사용

TRAIN AND TEST

TRAIN AND TEST