



RNN(Recurrent Neural Network)

1.Introduction

RNN이란?

'순환 신경 망' 이라고 하며, 내부의 순환 구조가 포함되어 있기 때문에 **시간에 의존적이거나 순차적인 데이터(Sequential data) 학습에 활용**되며, 내부에 있는 순환 구조에 의해 현재 정보에 이전 정보가 쌓이면서 정보 표현이 가능한 알고리즘.

즉, 1회의 데이터가 아니라 과거의 데이터도 학습에 활용하는 알고리즘

Tasks predicting 'sequences'

1. Language Modeling

LM 은 텍스트의 '분포' 를 모델링하는 것, 이것은 실제 예측이 아님, 모델이 입력을 받지 않지만 출력은 단어 또는 문자 시퀀스임. (주어진 단어들로부터 아직 모르는 단어를 예측하는 것. 이전 단어들로부터 다음 단어를 예측...? 다음 언어 모델 내용을 참고ㅠ)

2. Speech - to - Text, Text to - Speech

음성을 텍스트로 변환할 때, 사람이 말하는 음성 파형을 텍스트로 출력, 텍스트를 음성으로 말할 경우 반대로 실행(stt, tts 이 친구들,,, 우리의 seesun,,,))

3. Caption Generation

이미지를 입력으로 받고, 이미지에 대한 캡션 생성(이미지를 설명하는 문장을 만드는 것)

4. Machine Translation

번역에서 특히 중요한 예시 sequence-to-sequence 예측. 즉, 영어로 된 문장을 프랑스어로 된 문장으로 생성하고 싶음(그냥, 번역을 하고 싶다~)

Neural Language Model to RNN

언어 모델(Language Model, LM) 이란?

→ 언어라는 현상을 모델링 하고자 단어 시퀀스(또는 문장)에 확률을 할당하는 모델
즉, 가장 자연스러운 단어 시퀀스를 찾아내는 모델

언어 모델링(Language Modeling):



언어 모델에 -ing를 붙인 언어 모델링은 주어진 단어들로부터 아직 모르는 단어를 예측하는 작업. 즉, 언어 모델이 이전 단어들로부터 다음 단어를 예측하는 일

단어 시퀀스의 확률 할당(LM결과를 활용할 수 있는 task를 예시들)

→ 각 단어 등장 확률의 곱

대문자 P는 확률을 의미

a. 기계 번역(Machine Translation):

$P(\text{나는 버스를 탔다}) > P(\text{나는 버스를 태운다})$

→ 언어 모델은 두 문장을 비교하여 좌측의 문장의 확률이 더 높다고 판단

b. 오타 교정(Spell Correction)

선생님이 교실로 부리나케(_____)

$P(\text{달려갔다}) > P(\text{잘려갔다})$

1. 달려갔다 2. 잘려갔다

- $P(\text{달려갔다} | \text{선생님이 교실로 부리나케}) = 0.3$

- $P(\text{잘려갔다} | \text{선생님이 교실로 부리나케}) = 0.1$

→ 언어 모델은 두 문장을 비교하여 좌측의 문장의 확률이 더 높다고 판단합니다.

c. 음성 인식(Speech Recognition)

$P(\text{나는 메롱을 먹는다}) < P(\text{나는 메론을 먹는다})$

→ 언어 모델은 두 문장을 비교하여 우측의 문장의 확률이 더 높다고 판단합니다.

단어 시퀀스의 확률

하나의 단어를 w_1 , 단어 시퀀스를 대문자 W 라고 한다면, n 개의 단어가 등장하는 단어 시퀀스 W 의 확률

$$P(W) = P(w_1, w_2, \dots, w_n)$$

다음 단어 등장 확률

$n - 1$ 개의 단어가 나열된 상태에서 n 번째 단어의 확률

$$P(w_i | w_1, \dots, w_{i-1})$$

|의 기호는 조건부 확률(conditional probability)을 의미

예시) 다섯번째 단어의 확률

$$P(w_5 | w_1, w_2, w_3, w_4)$$

전체 단어 시퀀스 W 의 확률은 모든 단어가 예측되고 나서야 알 수 있으며, 단어 시퀀스의 확률

$$P(W) = P(w_1, w_2, \dots, w_n) = \prod_{i=1}^n P(w_i | w_{i-n+1}, \dots, w_{i-1})$$

LM에서 RNN의 탄생



왼쪽 : 기존 LM 모델(fixed length context를 1로 설정), 오른쪽 : hidden 사이에 연결을 추가하여 순환 신경망으로 바뀐 것

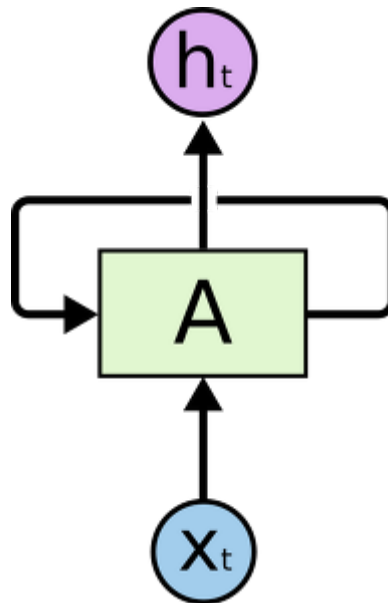
- 3-gram 예시($N=3$)

- $P(\text{나는 확률기반의 언어모형에 대하여 학습한다})$
- $= P(\text{나는, 확률기반의, 언어모형에, 대하여, 학습한다})$
- $= P(\text{나는})P(\text{확률기반의}|\text{나는})P(\text{언어모형에}|\text{나는, 확률기반의})P(\text{대하여}|\text{확률기반의, 언어모형에})P(\text{학습한다}|\text{언어모형에, 대하여})$

2. Recurrent Neural Nets

unrolling network to understand like FFNN

피드 포워드 신경망(FFNN)에서는 뉴런이라는 단위를 사용했지만, RNN에서는 뉴런이라는 단위보다는 입력층과 출력층에서는 각각 입력 벡터와 출력 벡터, 은닉층에서는 은닉 상태라는 표현을 주로 사용



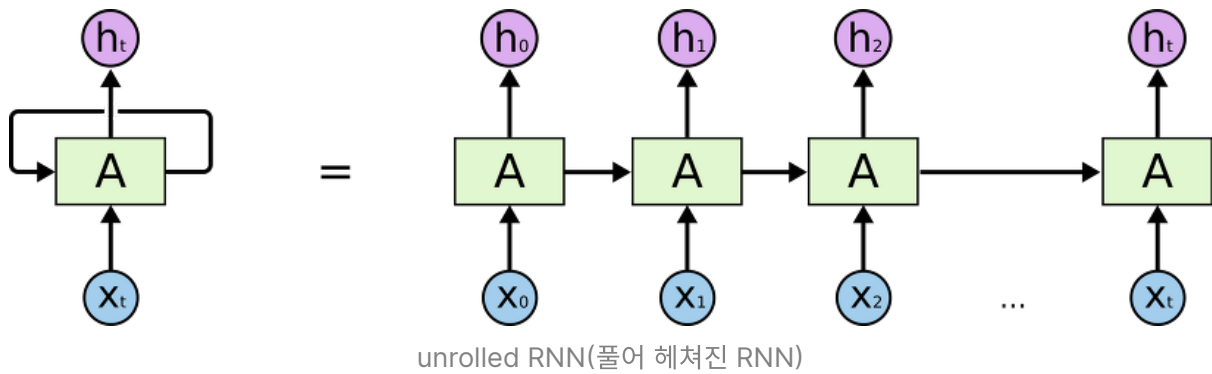
RNN have loops

A는 input X_t 를 받아서 h_t 를 내보낸다. A를 둘러싼 반복은 다음 단계에서의 network가 이전 단계의 정보를 받는다는 것을 보여줌.

즉, 하나의 network를 순서대로 정보를 전달하는 network라고 생각하는 것

A의 loop는 네트워크의 첫 step의 정보가 다음 step으로 전달되도록 하는 역할을 함

즉, 타임 스텝 t 마다 순환 뉴런을 펼쳐서 타임스텝 별 입력(x_t)과 출력(h_t)을 나타냄



결국 RNN은 전 시점의 어떤 정보를 다음 시점으로 넘겨주는 것. 다음 시점의 정보는 전 시점의 정보만이 아니라 이전까지의 정보를 모두 가지고 있으며, 이처럼 정보를 가지고 있는 것을 **cell**이라고 하며 현재 cell이 가지고 있는 정보, 즉 다음 시점으로 넘겨줄 정보를 **hidden state**라고 말함

RNN의 수식

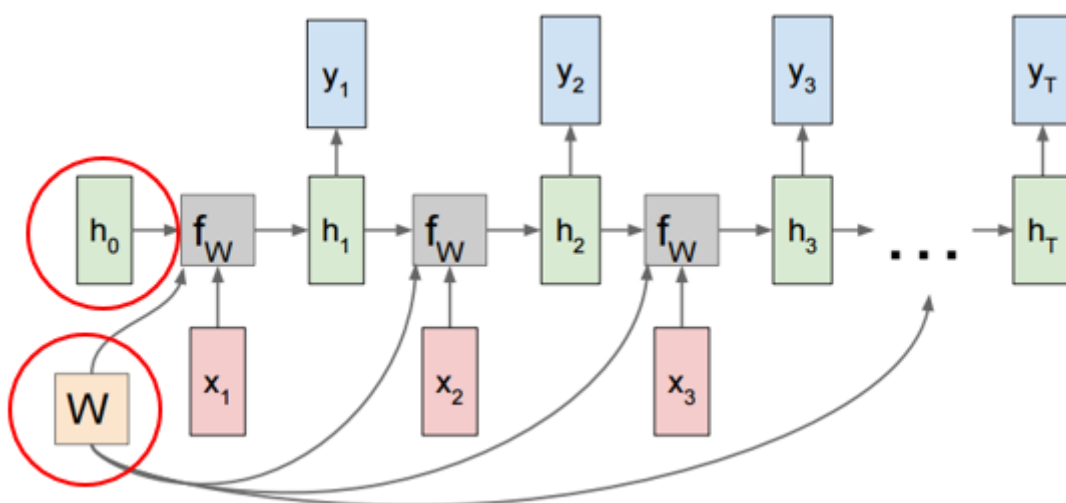
$$h_t = f(h_{t-1}, x_t)$$

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

$$y_t = W_{hy}h_t$$

시점 t에 대한 hidden state를 구하기 위해 전 시점 t-1의 h와 현 시점 x_t 로 계산

hidden에 대한 가중치, 입력 x에 대한 가중치를 곱하고 tanh를 활성화함수로 적용
최종으로 나온 hidden state를 바로 출력값 y로 사용할 수도 있음

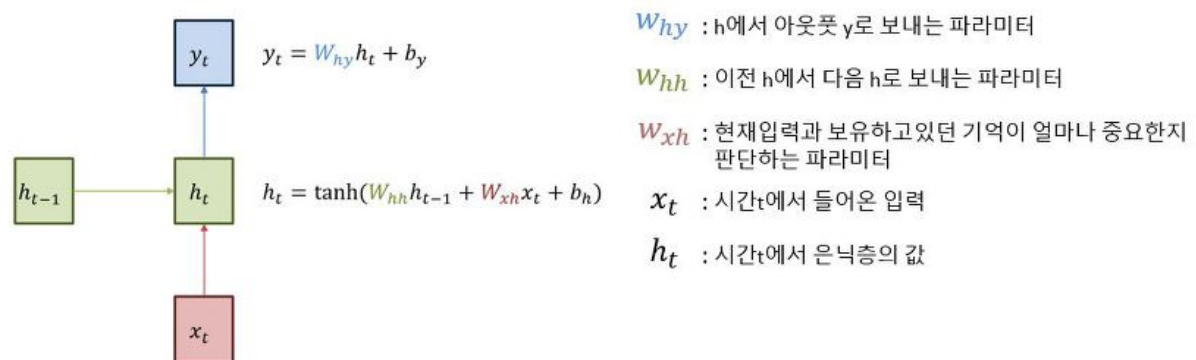


1. 전 시점의 hidden과 현 시점의 x함수 f()를 취하여 현 시점의 h를 구함

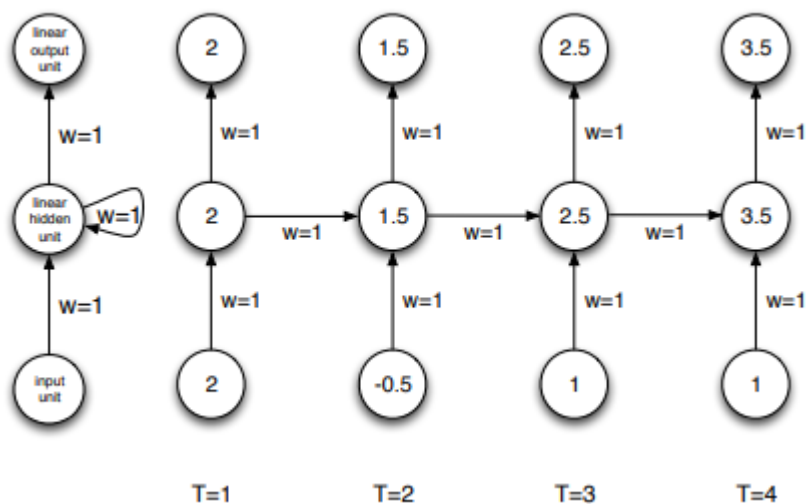
2. 현 시점의 h 로 현 시점의 y 출력값을 구함
3. 다시 h 를 넘겨주며 반복

! 처음 시점에 대해 연산을 할 경우, 전 시점의 hidden state가 없기 때문에 초기 hidden state값은 랜덤값을 활용, h_t 를 구하기 위해 h_{t-1} 에 연산할 가중치, x_t 에 연산할 가중치 matrix는 각 타임 스텝에서 동일하게 적용

3 examples of how parameter setting result in RNN

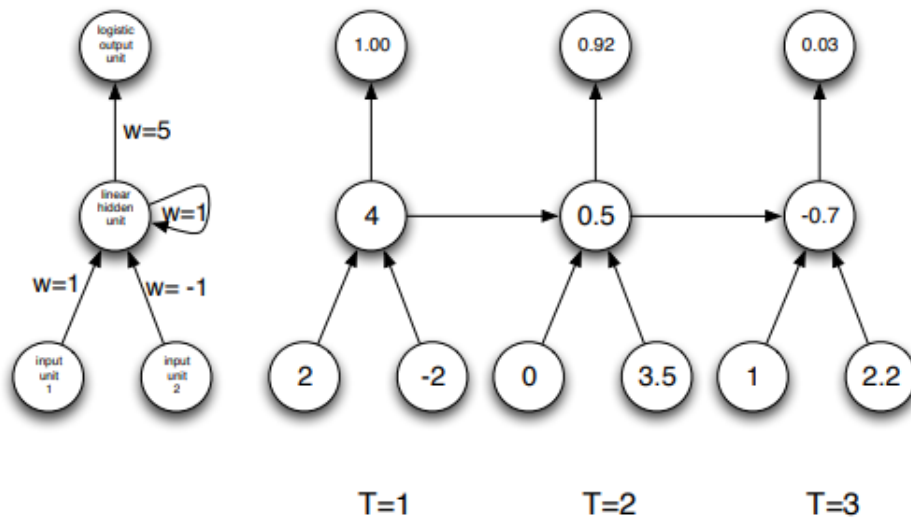


가) running sum RNN



- 누적합, 모든 가중치 1로 고정

나) 각각 타임 스텝에 두개의 input을 넣었을 경우



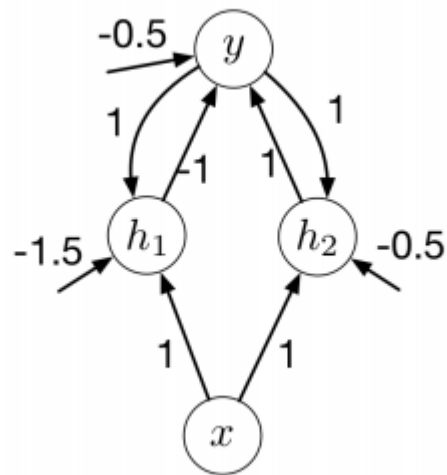
- logistic을 했을 때, hard threshold(0근처면 0 아닐경우1) 인데 가중치를 5를 주면 효과적으로 값이 나온다.
- 은닉층의 기본 가중치는 1이므로 기본적으로 이전 값을 기억
- input-to-hidden 가중치가 1, -1 이면 입력중 하나를 더하고 다른 하나는 빼줌(T =1의 경우 2는 더해주고 -2는 빼줌)

다) 패리티 비트를 통한 XOR연산 방법

Input: 0 1 0 1 1 0 1 0 1 1
Parity bits: 0 1 1 0 1 1 →

패리티 비트 예시

1의 개수가 짝수면 0 홀수면 1



들어온 패리티 비트랑 이전에 들어온 패리티랑 XOR연산
 패리티 비트 계산이 다음 연산계산에 들어가야해서 넣어줌
 XOR연산을 할 수 있는 구조

$y^{(t-1)}$	$x^{(t)}$	$y^{(t)}$
0	0	0
0	1	1
1	0	1
1	1	0

출력, 입력

$y^{(t-1)}$	$x^{(t)}$	$h_1^{(t)}$	$h_2^{(t)}$	$y^{(t)}$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

▼ 출처

<https://blog.naver.com/PostView.nhn?blogId=winddori2002&logNo=221974391796>

<https://wikidocs.net/21668>(LM)

http://bigdata.dongguk.ac.kr/lectures/TextMining/_book/언어-모델language-model.html(LM)

<https://gruuuuu.github.io/machine-learning/lstm-doc/> (rnn의 활성화 함수를 tanh를 사용하는이유)