

RNN : Recurrent Neural Network

2019. 11. 15 (목)

고려대학교 산업경영공학과

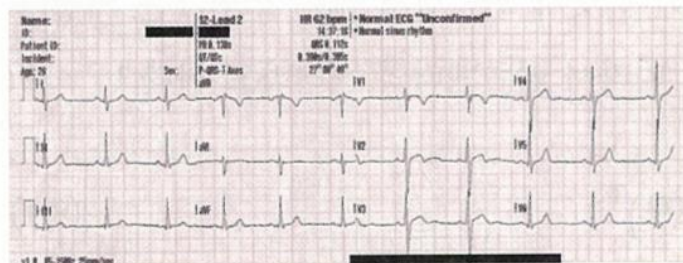
석사과정 조규원

목 차

1. RNN 등장 배경
2. RNN 기초 개념
3. RNN 응용
4. RNN 변형
5. 실습

순차형 데이터 처리

- 많은 종류의 순차형 데이터가 존재
- 순차형 데이터는 이전 상태가 다음 상태의 발생에 영향을 미침



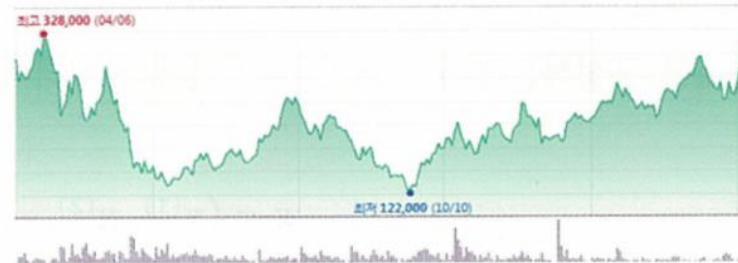
(a) 심전도 신호



(c) 음성 신호

ATGCTTCGGCAAGACTCAAAAATA

(e) 유전자 열



(b) 주식 시세

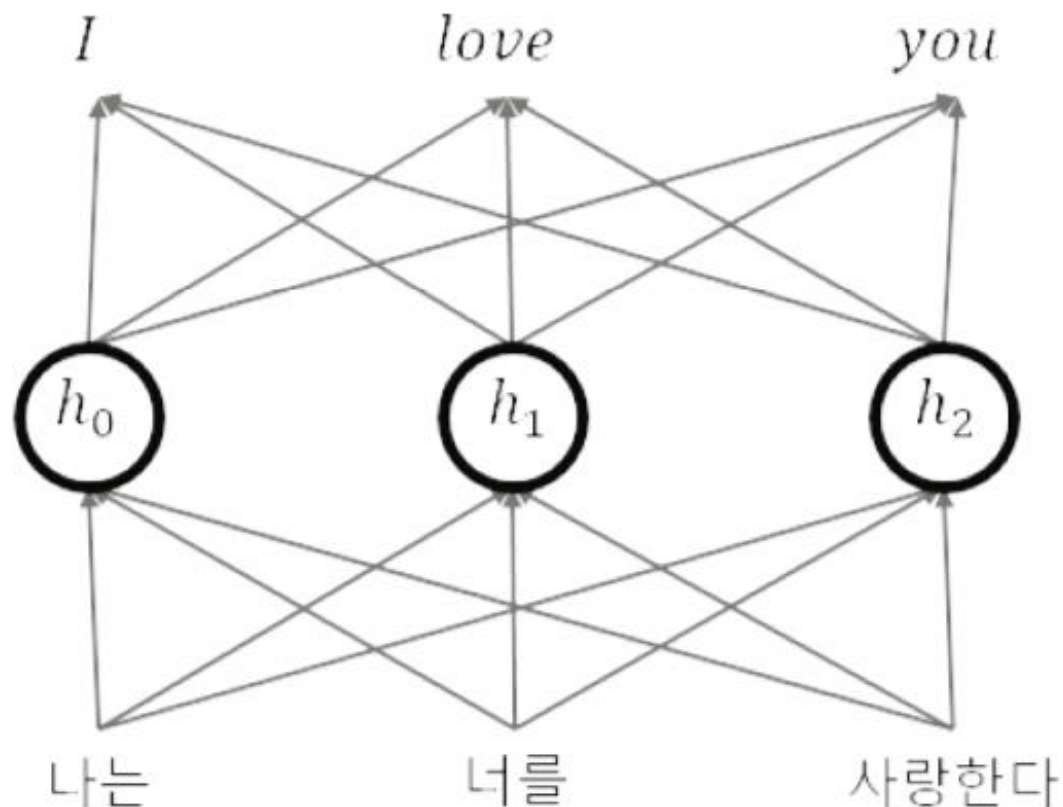
내려갈 때 보았네 올라갈 때 보지 못한 그 꽃

(d) 문장

그림 8-1 순차 데이터

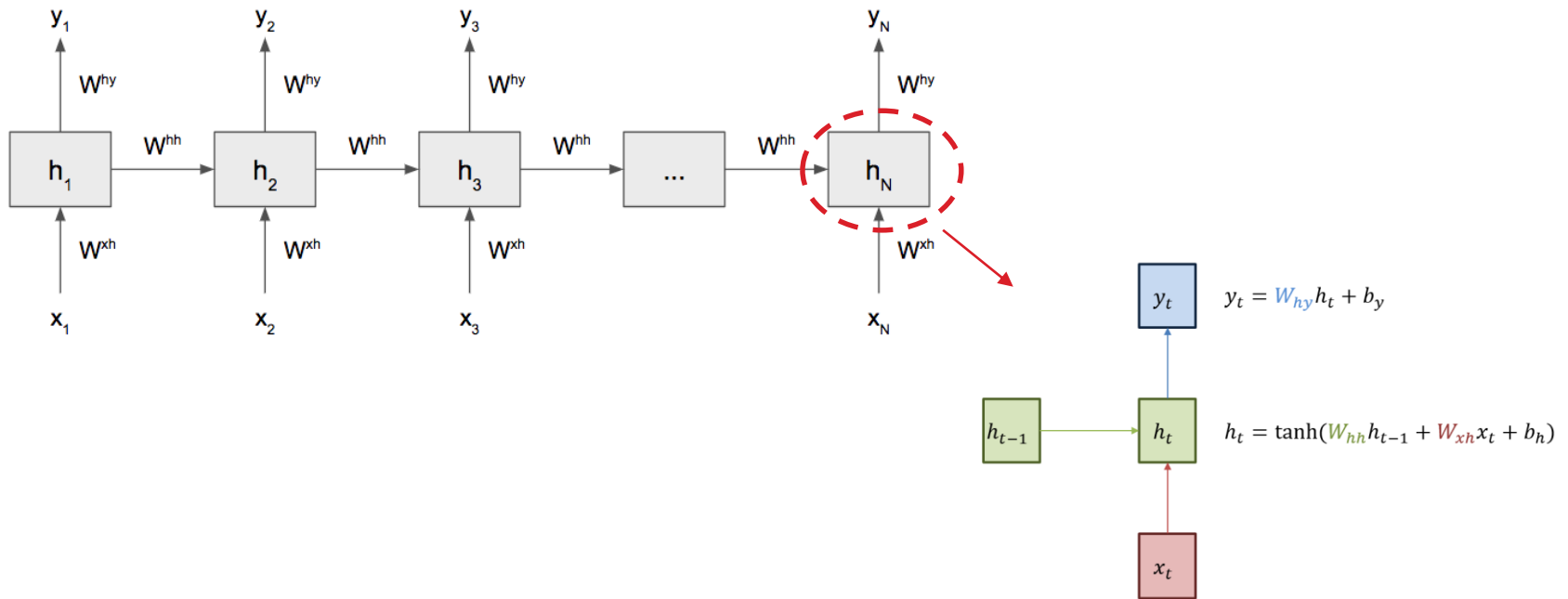
순차형 데이터 처리

- 기존의 FFNN 구조는 순차형 데이터에 적합하지 않음
 - 순서를 고려하지 않고, 한번에 데이터를 인풋으로 넣어야 하기 때문



순환 신경망 모델

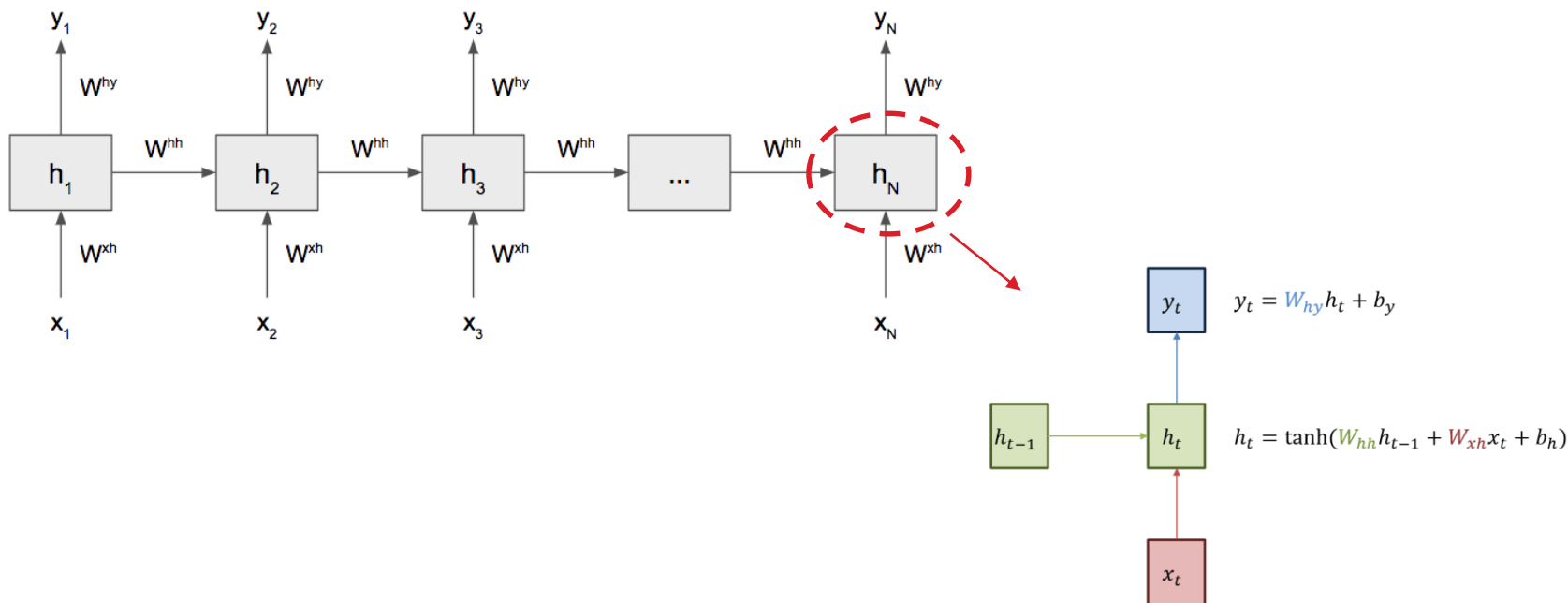
- 데이터 간 순서 관계를 고려할 수 있는 모델 구조
 - 이전 step 의 정보를, 다음 step 에서 활용할 수 있도록 구성
 - 이전 step 의 hidden state 값을 다음 step 의 인풋 정보를 합침(sum)



RNN 동작 과정

- 주요 용어 정리

- hidden_state : 다음 단계로 넘겨주는 정보 (h_t)
- hidden_size : W_{hh} , W_{xh} 의 size



RNN 동작 과정

- 동작 예시

- Ex. 단어가 하나씩 들어갈 때 마다, 현재 문장의 긍정 부정 점수를 0 ~ 1로 예측

와! 진짜 **화날** 정도로 **재밋네요**



토큰화

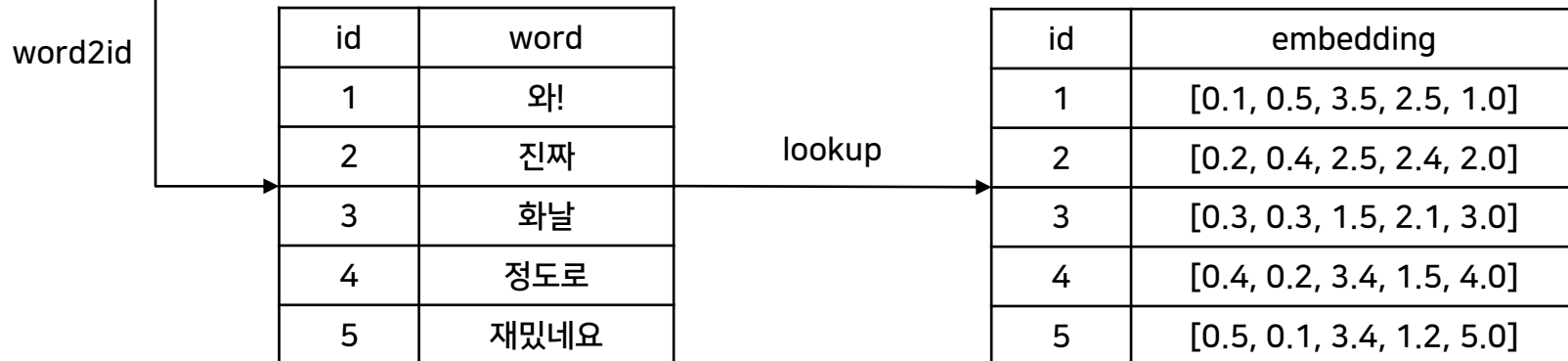
[와!, 진짜, 화날, 정도로, 재밋네요]

RNN 동작 과정

- 동작 예시

- Ex. 단어가 하나씩 들어갈 때 마다, 현재 문장의 긍정 부정 점수를 0 ~ 1로 예측

[와!, 진짜, 화날, 정도로, 재밌네요, ~, 이, 영화]

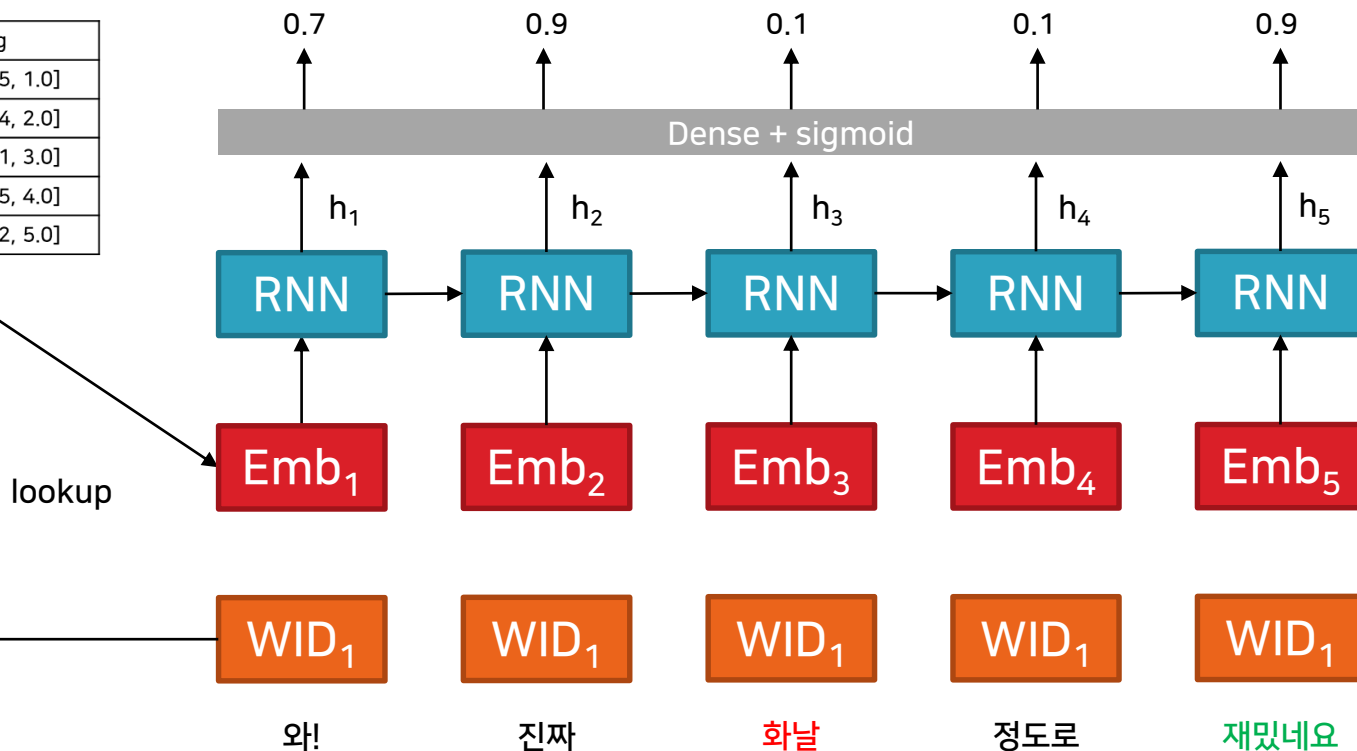


RNN 동작 과정

• 동작 예시

- Ex. 단어가 하나씩 들어갈 때 마다, 현재 문장의 긍정 부정 점수를 0 ~ 1로 예측

id	embedding
1	[0.1, 0.5, 3.5, 2.5, 1.0]
2	[0.2, 0.4, 2.5, 2.4, 2.0]
3	[0.3, 0.3, 1.5, 2.1, 3.0]
4	[0.4, 0.2, 3.4, 1.5, 4.0]
5	[0.5, 0.1, 3.4, 1.2, 5.0]

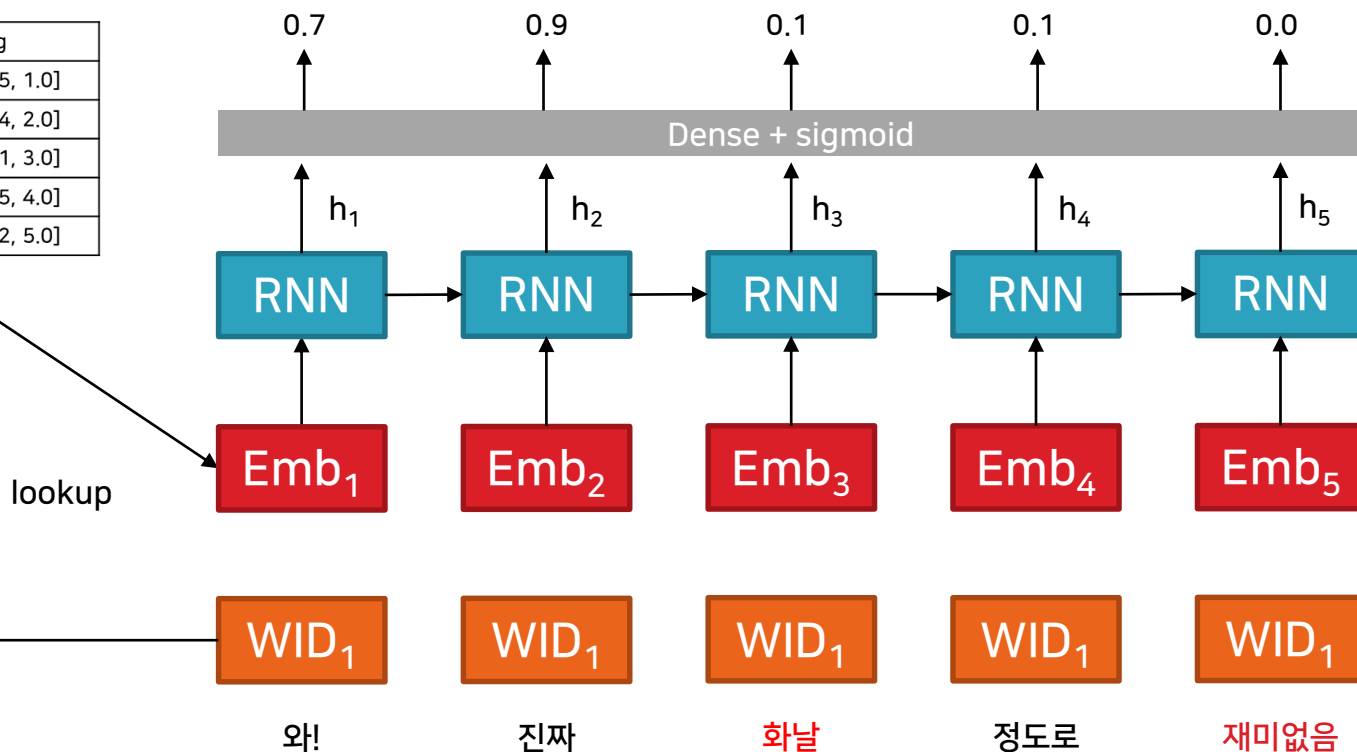


RNN 동작 과정

• 동작 예시

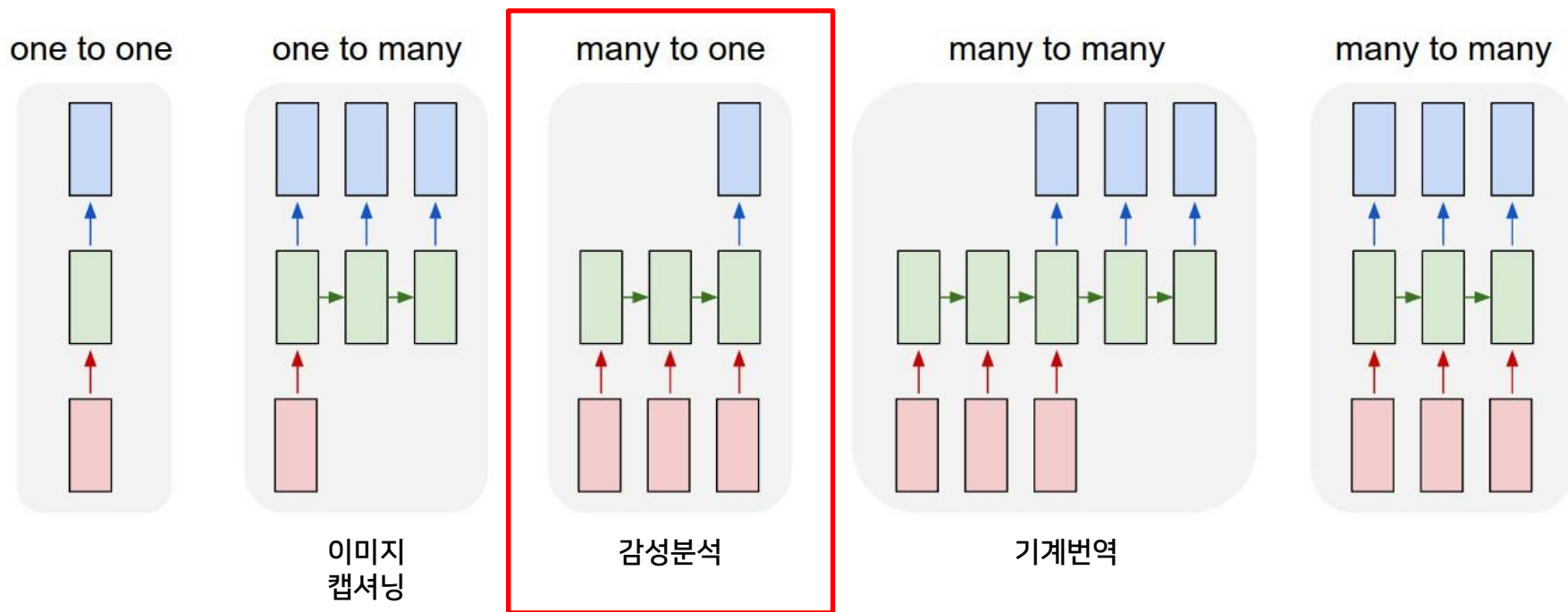
- Ex. 단어가 하나씩 들어갈 때 마다, 현재 문장의 긍정 부정 점수를 0 ~ 1로 예측

id	embedding
1	[0.1, 0.5, 3.5, 2.5, 1.0]
2	[0.2, 0.4, 2.5, 2.4, 2.0]
3	[0.3, 0.3, 1.5, 2.1, 3.0]
4	[0.4, 0.2, 3.4, 1.5, 4.0]
5	[0.5, 0.1, 3.4, 1.2, 5.0]



RNN 활용 사례

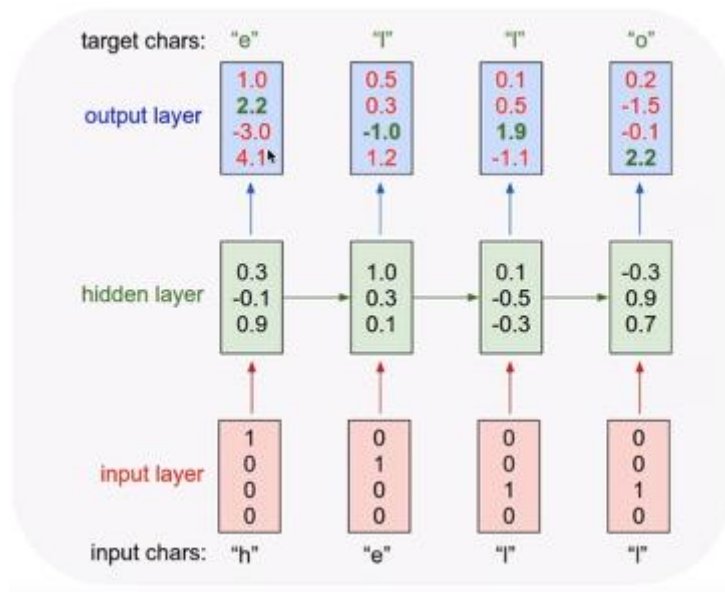
- 몇 개의 인풋, 아웃풋을 사용하는지에 따라 활용처가 달라짐



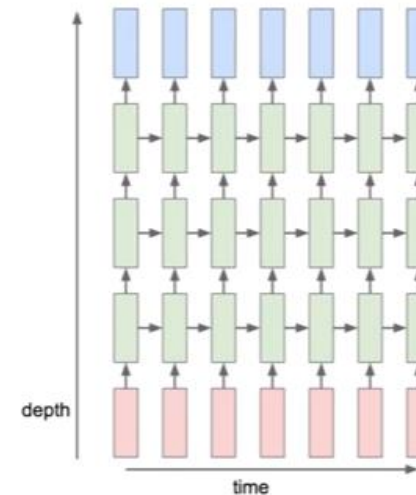
RNN 응용

- Multi-layer RNN

- RNN Cell 을 여러 층을 쌓아서 조금 더 복잡한 관계를 파악할 수 있도록 함
- 무조건 층을 많이 쌓는다고 성능이 좋아지는 것은 아님에 주의!



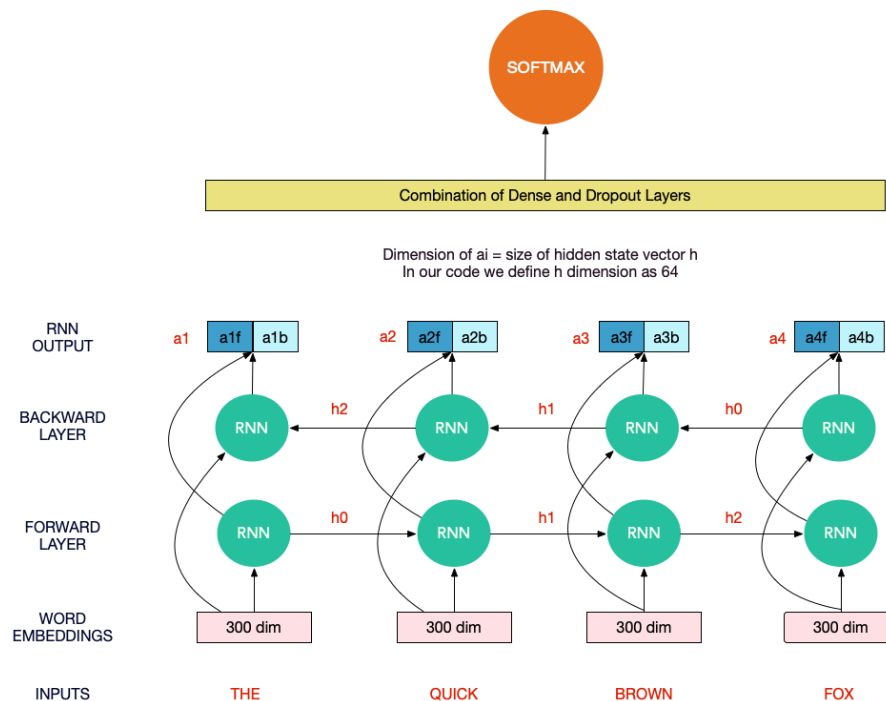
Multi-Layer RNN



RNN 응용

- Bidirectional RNN

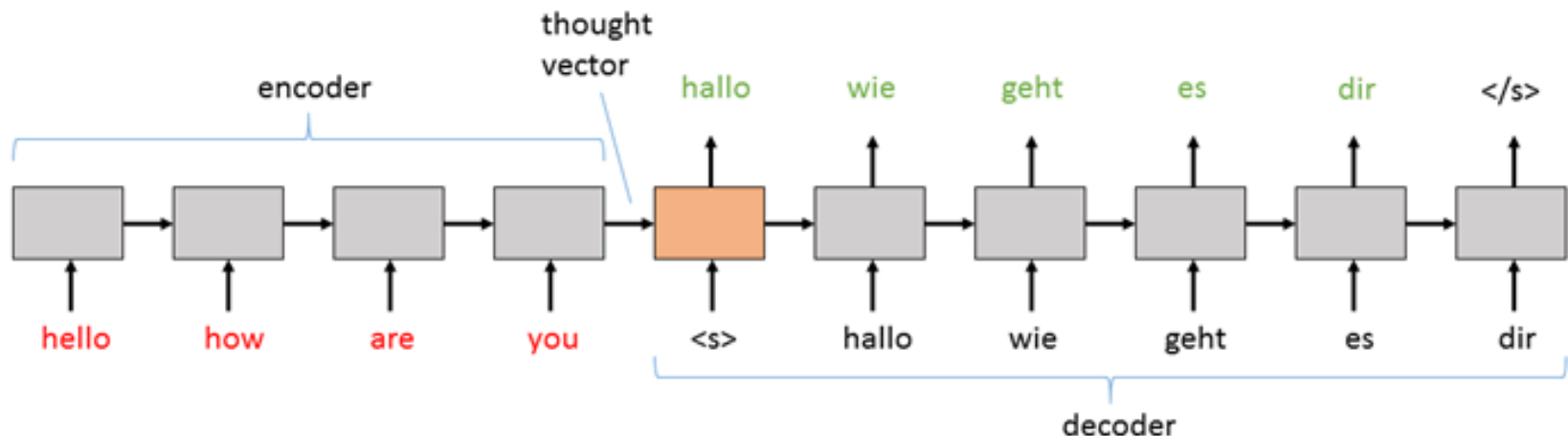
- 양방향으로 진행되는 RNN hidden state 를 결합(concatenate)하여 전달
- 등장 배경 : Machine Translation 등에서 처음 사용



RNN 응용

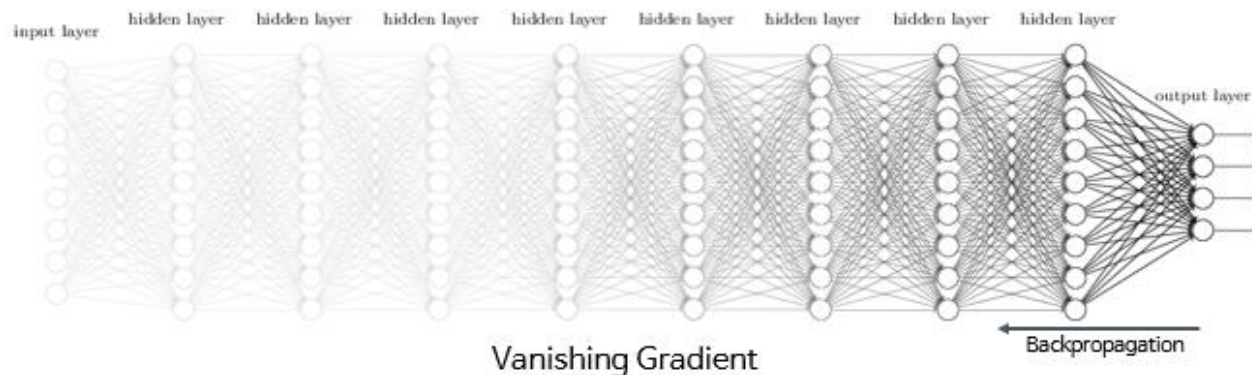
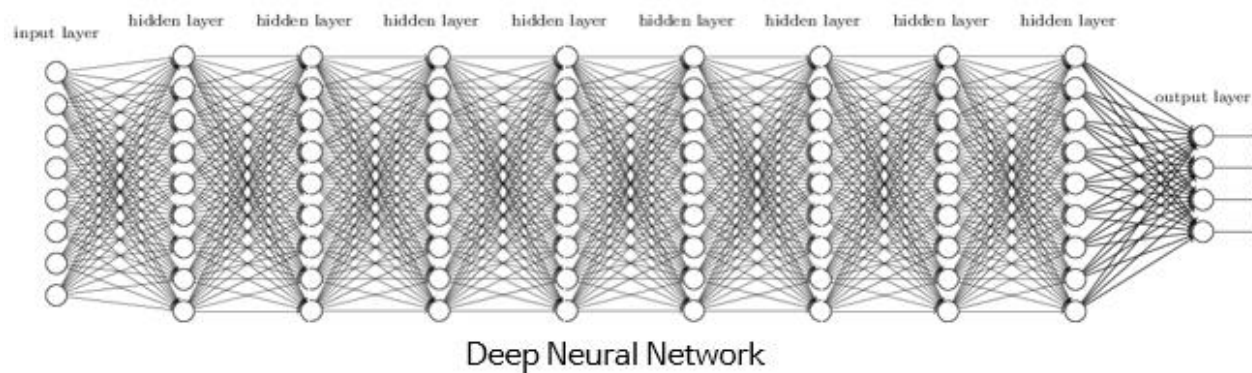
- Encoder – Decoder Model

- Sequence-to-sequence 모델이라고도 불림, RNN 모델 2개를 붙인 형태
- 기계 번역시 문장 전체에 대한 문맥을 고려한 번역을 위해 처음 고안
- Encoder 의 요약 정보를 Decoder 에 넘겨주어 전체 문장의 문맥을 고려하게 함
- 각 RNN 은 하나의 언어에 대한 학습만 하면 되기 때문에 학습 진행이 잘 되는 특징



Vanilla RNN 의 문제점

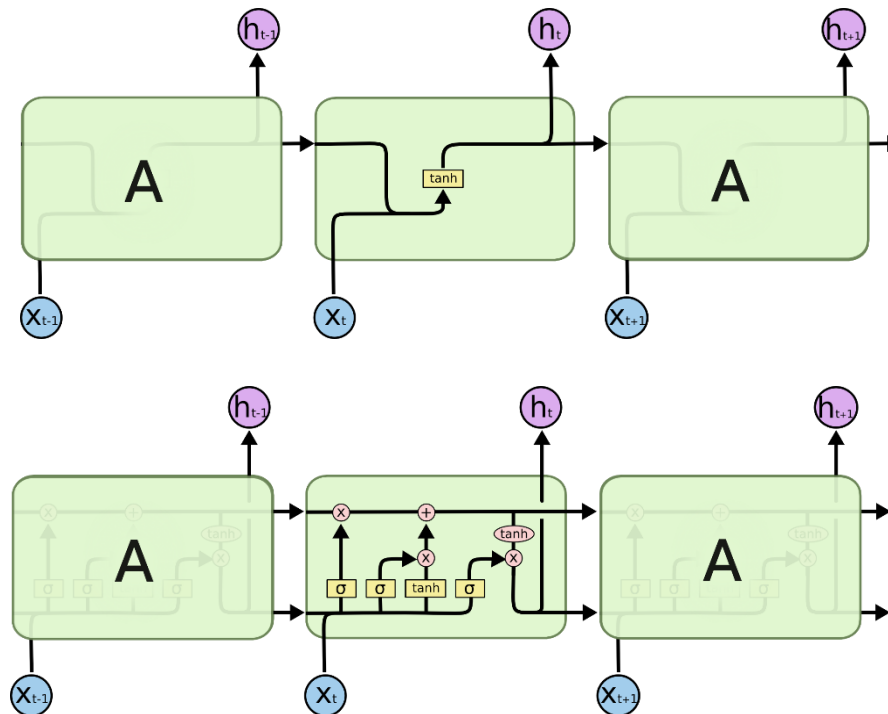
- Vanishing gradient problem
 - Gradient 역전파 과정에서 손실이 많이 발생함
 - 문장의 길이가 길어질수록 이 문제가 더욱 심각해짐



LSTM

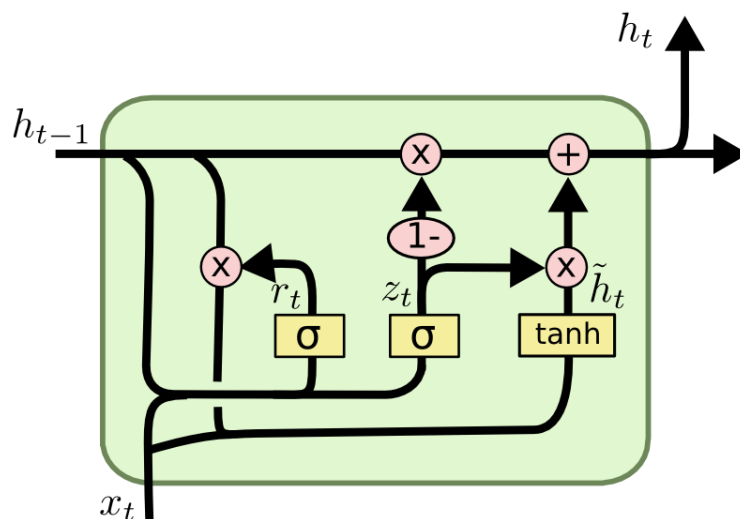
- Long Short-Term Memory

- 기존의 RNN에 여러 개의 gate 를 추가하여 hidden state 효과적으로 컨트롤
- [Forget gate, Input gate] \rightarrow update \rightarrow [Output gate]
- 시작부터 끝까지 이어지는 cell state 를 통해 장기 기억을 보관 및 전달



GRU

- Gated Recurrent Unit
 - 기존 LSTM 의 구조를 변형, 기본적인 컨셉은 동일
 - Forget gate + Input gate \rightarrow update gate 로 통합
 - Hidden state + cell state \rightarrow hidden state 로 통합
 - LSTM 과 같은 성능을 내면서도, 효율적인 구조



Q & A

실습

Tensorflow + RNN 을 활용한 영화 리뷰 데이터 감성 분석

데이터 링크 : <https://github.com/e9t/nsmc>