AI기법과 활용

Week-08. Natural Language Processing Part 1

2022-Summer 서중원

전통적인 텍스트 분석

One-Hot Encoding Vector

- 단어를 벡터로서 표현하는 가장 간단한 방법
- 각각의 차원은 전체 단어
  - [강아지, 고양이, 토끼, 호랑이]의 단어가 존재하는 경우
    - 강아지 : [1,0,0,0]
    - 고양이: [0,1,0,0]
    - 토끼: [0,0,1,0]
    - 호랑이: [0,0,0,1]
- 만약 전체 데이터에 Unique 단어가 10,000개라면?

전통적인 텍스트 분석

Term-Document Matrix

- 각각의 문서를 벡터화 할 수 있는 방법
- 단점
  - 차원의 저주
    - 전처리를 충분히 하지 않으면 컬럼(feature)의 수가 순식간에 증가
    - 희소행렬 (Sparse Matrix)
      - 대부분의 컬럼값이 0

Term	Doc1	Doc2	Doc3	Doc4	Doc5
Hello	0	0	0	1	0
My	1	0	1	0	1
Love	0	0	1	0	0
It	1	1	1	1	1
Is	1	1	1	1	1
Very	0	1	0	0	1
Cold	0	0	0	1	0
On	1	0	1	1	1
This	0	1	1	1	1
Island	0	1	0	0	0

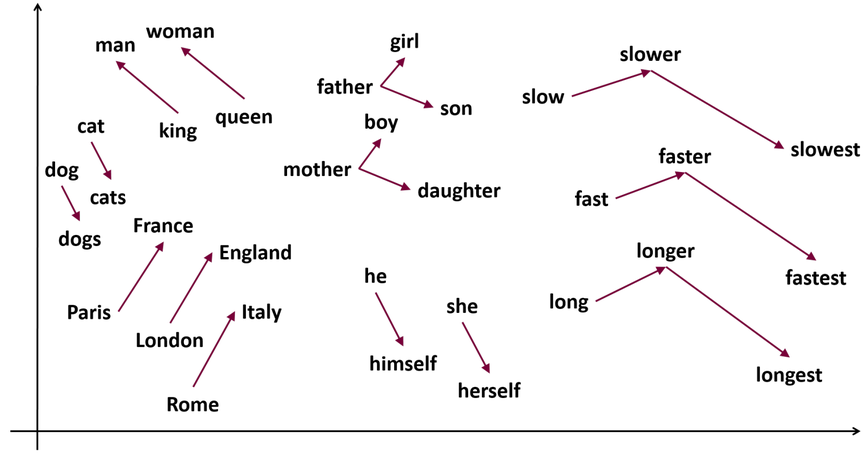
Word Embedding
Dense Representation (밀집표현)

- 기존의 Sparse Matrix의 차원을 축소하자
- How?
  - 각각의 단어를 차원으로 표현하지 말고 특정 차원 수에 각각의 단어가 어떻게 위치 하는지를 표현하자

강아지: [1, 0, 0, 0]
고양이: [0, 1, 0, 0]

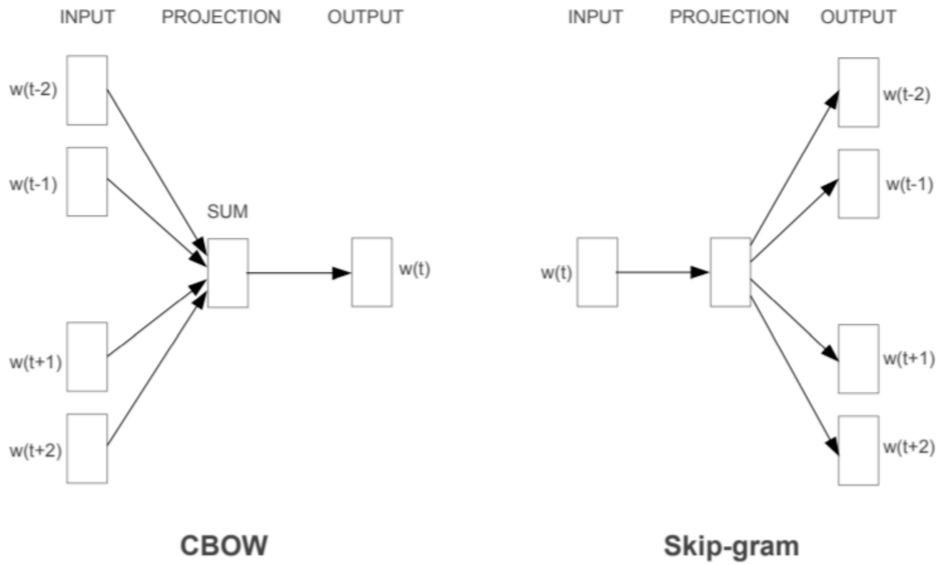
강아지: [0.6, 0.2]
고양이: [0.7, 0.3]

  - 장점
    - 차원의 저주 탈출!
    - 단어간의 관계 표현 가능
      - 기존에는 모든 단어들은 서로 유사도가 0 (e.g., cosine similarity)
      - 강아지와 고양이의 유사도가, 강아지와 컴퓨터 보다 높아야 된다



Word Embedding
Word2Vec

- 어떻게 단어를 잘 표현 할 수 있는 Embedding을 생성할 수 있을까?
  - 랜덤 값들을 넣게되면, 그냥 랜덤 벡터
- 가설: 단어라는 것을 결국 주변에 어떤 단어가 있는지로 역할/의미가 결정된다

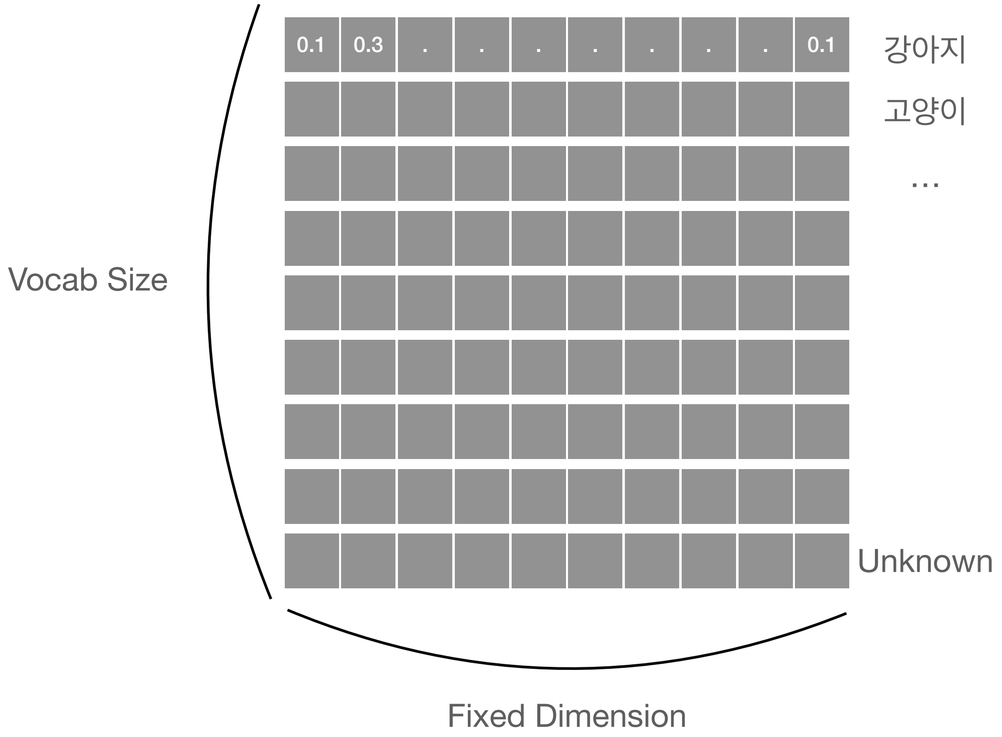


Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever. "Exploiting similarities among languages for machine translation." *arXiv preprint arXiv:1309.4168* (2013).

Word Embedding

Embedding

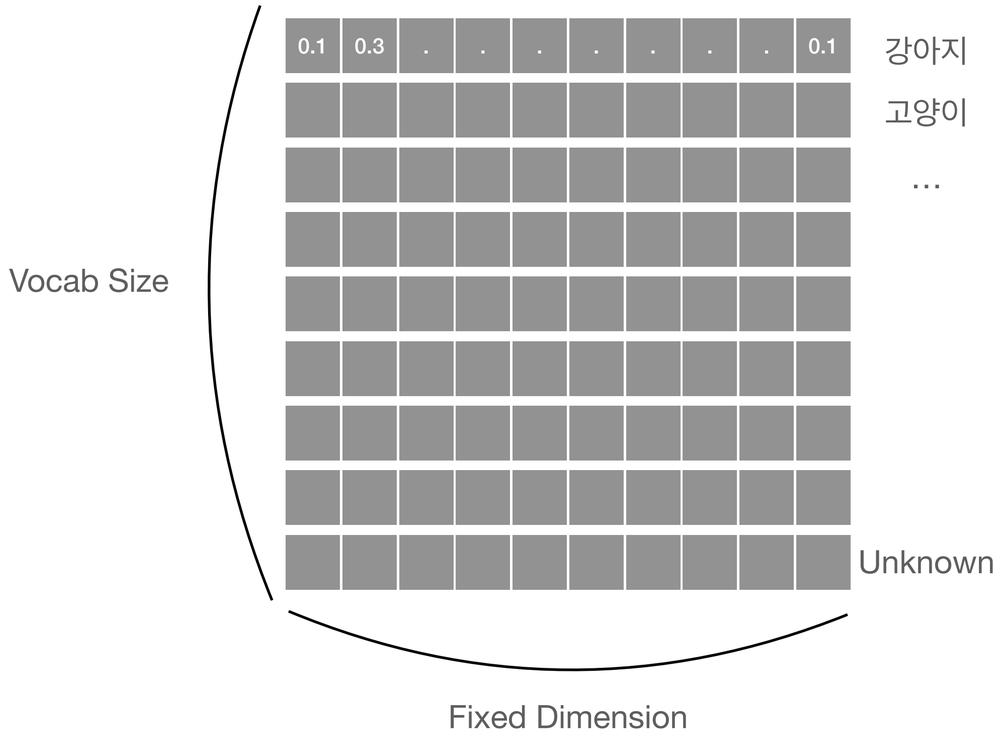
- 최종적으로 뭐가 훈련이되고, 어떤 결과물이 나오는 것인가?
- 단어 임베딩, LookUp Table



Word Embedding

Embedding

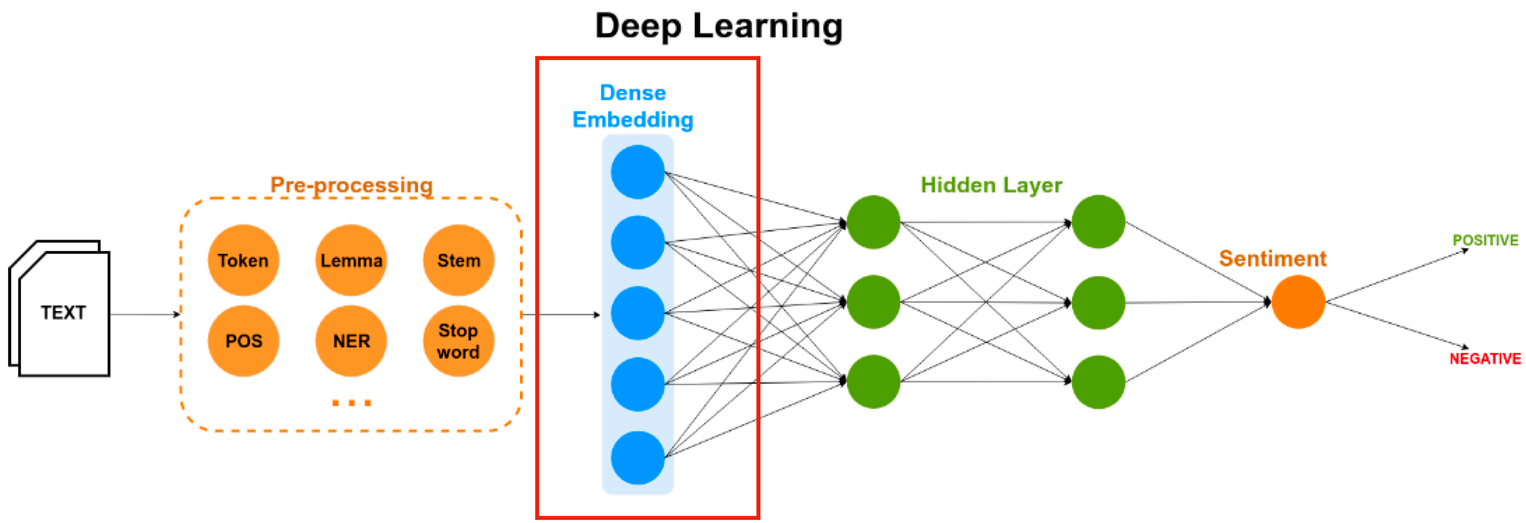
- 각각의 단어를 표현할 다른 방법이 생겼으니,
- 이제 One-hot Encoding이 아닌 Ordinal Encoding을 적용
  - 강아지: 1, 고양이: 2, 호랑이:3, 안녕: 4
  - <sos>:0, <eos>:999, <unknow>: 1000



Word Embedding

Embedding

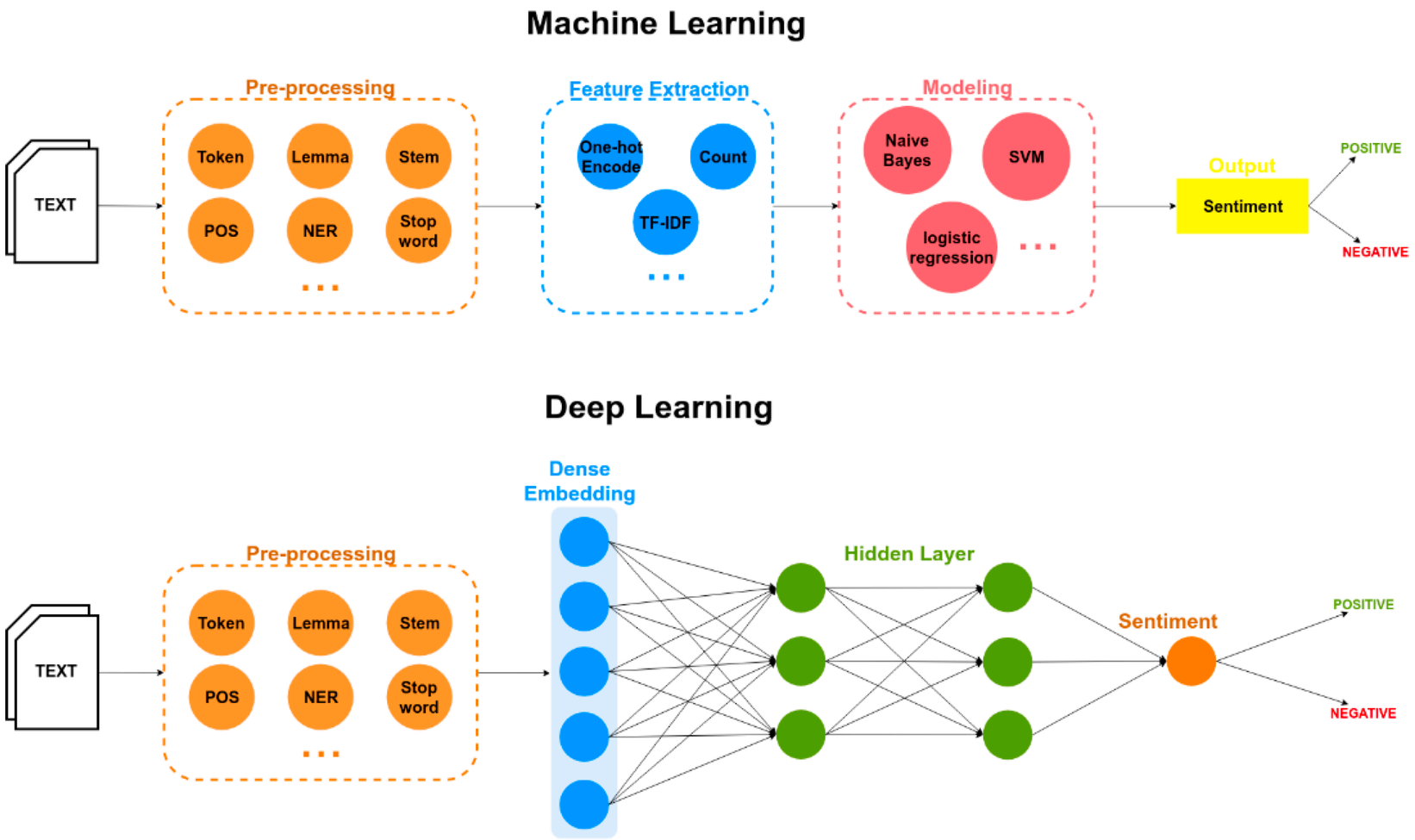
- 반드시 Word2Vec으로 훈련을 할 필요는 X
- 특정한 Task를 수행하게 끔 훈련을 하다 보면, 그 과정에서 Embedding (각각의 단어 벡터들)이 현재 Task내에서 최적에 가깝게 업데이트 됨
  - But, 이미 대량의 데이터로 잘 훈련된 Embedding을 사용하면 더욱 빠르게 원하는 성능에 도달 가능
    - Transfer Learning, Fine-tuning



Dang, Nhan Cach, María N. Moreno-García, and Fernando De la Prieta. "Sentiment analysis based on deep learning: A comparative study." *Electronics* 9.3 (2020): 483.

텍스트 분석 방법 비교

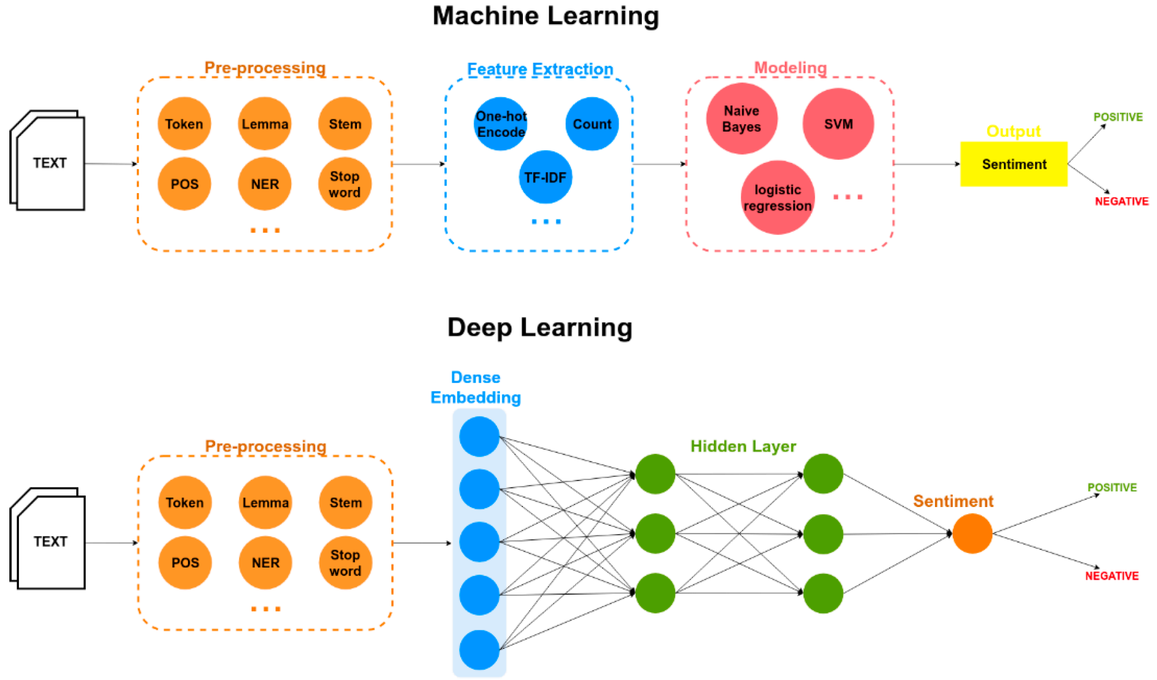
Machine Learning VS Deep Learning



Dang, Nhan Cach, María N. Moreno-García, and Fernando De la Prieta. "Sentiment analysis based on deep learning: A comparative study." *Electronics* 9.3 (2020): 483.

전통적인 텍스트 분석

텍스트 데이터 학습 과정

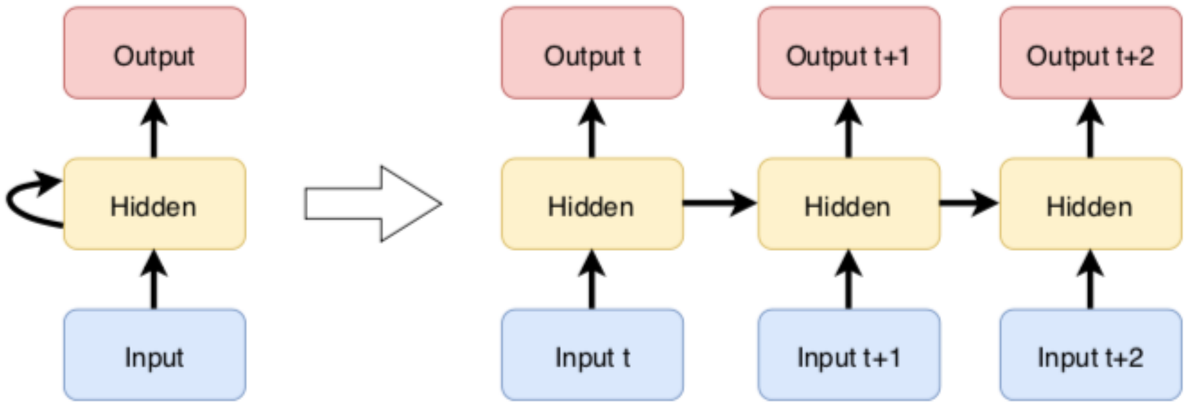


- 문서 전체를 하나의 Vector로 구성
- 문제점
  - 단어의 순서가 고려되지 않음
  - Jane killed Mike == Mike killed Jane
    - Who is dead?
- 전통적인 머신러닝 기법을 사용해서가 아닌, 일반적인 딥 러닝 모델을 사용해도 마찬가지
  - 입력값의 feature간의 관계가 고려되지 않기 때문에
- 순서를 고려할 수 있는 훈련 기법이 있을까?

Dang, Nhan Cach, María N. Moreno-García, and Fernando De la Prieta. "Sentiment analysis based on deep learning: A comparative study." *Electronics* 9.3 (2020): 483.

Recurrent Neural Networks
순환신경망

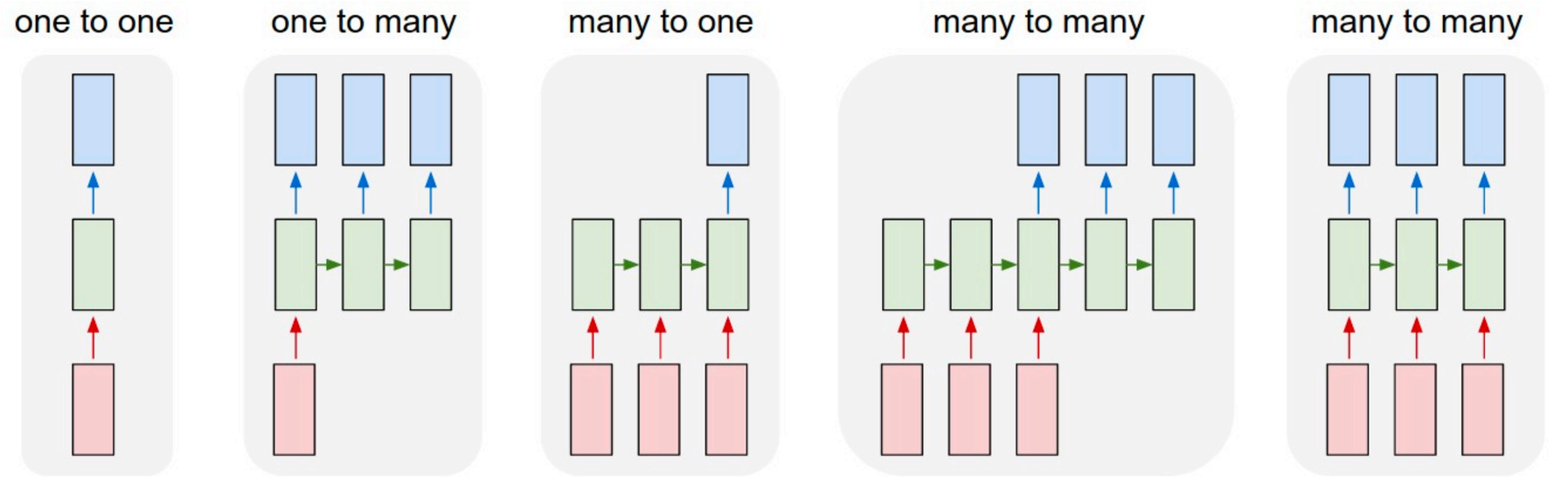
- 기존 Feed Forward Neural Networks (FNNs) 계열의 경우 시계열 또는 순서를 고려하지 못함
- 순서가 중요한 텍스트 데이터의 경우 RNN류의 모델을 쓰는게 적합
  - Vanila RNN, LSTM, GRU



(a) 간소화된 표현 (b) 확장된 표현

Recurrent Neural Networks
순환신경망

- Task에 맞게 Input-Output의 형태를 조정

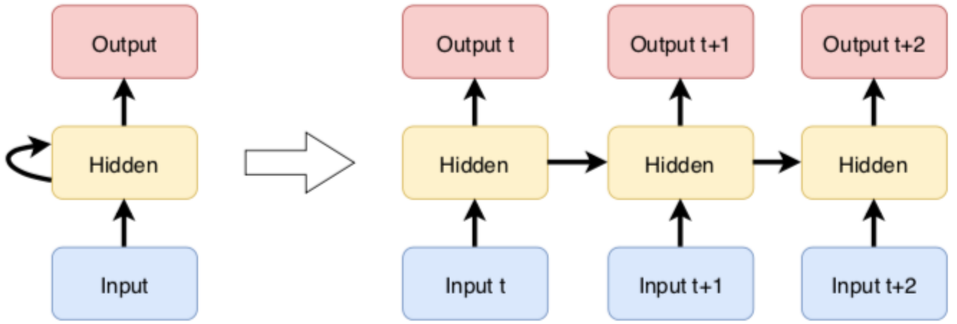


No RNN | 이미지 설명 | 감성분석 | 기계번역 | 비디오/오디오 관련 e.g, Speech Separation

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Recurrent Neural Networks

일반적인 순환신경망의 한계



- Sequence Length에 의한 영향
  - Input이 길어지면, 초반에 등장한 단어의 영향이 적어진다
  - Vanishing gradient
- 단어의 순서는 고려하나, 중요한 단어가 무엇인지는..
  - 머신러닝은 앞으로도 그리고 이전에도 매우 많은 사람들에게 흥미로운 주제지만, 딥러닝은 ..
- 연산 속도
  - Sequence가 존재하기 때문에, 이전 input이 처리(병목)되기 전에 다음 input을 처리 할 수 없다.
    - 병렬적으로 학습이 진행되지 않음!

RNN의 한계를 어떻게 극복할까?
다음 시간에..

E.O.D