텍스트 전처리

텍스트 전처리

Robot Media Laboratory

말뭉치(코퍼스)

- 말뭉치 또는 코퍼스(영어: corpus, 복수형: corpora)
- 자연언어 연구를 위해 특정한 목적을 가지고 언어의 표본을 추출한 집합
- 자연어 처리 관련 애플리케이션은 방대한 양의 데이터
- 코퍼스 분석 뿐만 아니라 언어 분석에 사용되는 실제 언어의 체계적 디지털 모음
- 둘 이상의 코퍼스가 있으면 코포라라고 부름
- 코퍼스를 데이터세트라고도 함.

텍스트 전처리

Robot Media Laboratory

말뭉치(코퍼스)

- 1. 단일 언어 코퍼스 : 하나의 언어로 이루어짐
- 2. 이중 언어 코퍼스 : 2개의 언어로 이루어짐
- 3. 다국어 코퍼스 : 3개의 이상 언어로 이루어짐

텍스트 전처리

- 자연어 처리에서 크롤링 등으로 얻어낸 코퍼스 데이터가 필요에 맞게 전처리되지 않은 상태시 해당 데이터를 용도에 맞게 토큰화(tokenization) & 정제(cleaning) & 정규화(normalization)를 진행 해야 함

데이터 분석

- 데이터수집
- 데이터 전처리
  - 토큰화
  - 정제
  - 정규화

텍스트 전처리

Robot Media Laboratory

토큰화

- 단어 토큰화(Word Tokenization)
- 문장 토큰화(Sentence Tokenization)

단어 토큰화(word tokenization)

- 토큰의 기준을 단어(word)로 하여 토큰화 하는것
- 단어(word)는 단어 단위 외에도 단어구, 의미를 갖는 문자열로도 간주됨
- 보통 토큰화 작업은 단순히 구두점이나 특수문자를 전부 제거하는 정제(cleaning) 작업을 수행하는 것만으로 해결되지 않음
- 구두점이나 특수문자를 전부 제거하면 토큰이 의미를 잃어버리는 경우가 발생함
- 띄어쓰기 단위로 자를시 단어 토큰 구분이 망가지는 언어도 존재
  - Ex)한국어

텍스트 전처리

Robot Media Laboratory

입력: Time is an illusion. Lunchtime double so!

출력 : "Time", "is", "an", "illustion", "Lunchtime", "double", "so"

텍스트 전처리

Robot Media Laboratory

- **토큰화 진행 시, 예상하지 못한 경우가 있어서 토큰화의 기준을 생각해봐야 하는 경우가 발생함 Ex) 영어 아포스트로피(apostrophe) 토큰화 문제**

- **Don't be fooled by the dark sounding name, Mr. Jone's Orphanage is as cheery as cheery goes for a pastry shop.**
  - Don't
  - Don t
  - Dont
  - Do n't
  - Jone's
  - Jone s
  - Jone
  - Jones

텍스트 전처리

Robot Media Laboratory

토큰화 고려 사항

1) 구두점이나 특수 문자를 단순 제외해서는 안 된다.

2) 줄임말과 단어 내에 띄어쓰기가 있는 경우.

텍스트 전처리

Robot Media Laboratory

1) 구두점이나 특수 문자를 단순 제외해서는 안 된다.

- 단어들을 걸러낼 때, 구두점이나 특수 문자를 단순히 제외하는 것은 옳지 않음
- 구두점조차도 하나의 토큰으로 분류하기도 함

- **Ex)**
  - 마침표(.)
    - 문장의 경계를 통한 단어를 추출용 기준이용
  - 단어 자체에 구두점
    - m.p.h나 Ph.D나 AT&T 특수 문자의 달러($)나 슬래시(/)
    - $45.55와 같은 가격을 의미 하기도 하고, 01/02/06은 날짜를 의미
  - 숫자 사이에 컴마(,)
    - 123,456,789

2) 줄임말과 단어 내에 띄어쓰기가 있는 경우.

- 토큰화 작업에서 종종 영어권 언어의 아포스트로피(')는 압축된 단어를 다시 펼치는 역할을 하기도 함

- **Ex)**
  - what're는 what are의 줄임말
  - we're는 we are의 줄임말
  - New York이라는 단어나 rock 'n' roll
    - 하나의 단어이지만 중간에 띄어쓰기가 존재

Penn Treebank Tokenization의 규칙

– 영어 토큰화 표준으로 사용되는 규칙

규칙 1. 하이푼으로 구성된 단어는 하나로 유지
규칙 2. doesn't와 같이 아포스트로피로 '접어'가 함께하는 단어는 분리

문장 토큰화(Sentence Tokenization)

- 문장 단위로 구분하는 작업으로 때로는 문장 분류(sentence segmentation)
- 정제되지 않은 상태라면, 코퍼스는 문장 단위로 구분되어 있지 않아서 이를 사용하고자 하는 용도에 맞게 문장 토큰화가 필요
- !나 ?는 문장의 구분을 위한 꽤 명확한 구분자(boundary) 역할
- 마침표 . 는 문장의 끝이 아니더라도 등장할 수 있음

  EX1)

  IP 192.168.56.31 서버에 들어가서 로그 파일 저장해서 aaa@gmail.com로 결과 좀 보내줘. 그 후 점심 먹으러 가자.

  EX2)

  Since I'm actively looking for Ph.D. students, I get the same question a dozen times every year.

텍스트 전처리

Robot Media Laboratory

한국어 토큰화

- 영어는 New York과 같은 합성어나 he's 와 같이 줄임말에 대한 예외처리만 한다면, 띄어쓰기(whitespace)를 기준으로 하는 띄어쓰기 토큰화를 수행해도 단어 토큰화가 잘 작동
- 거의 대부분의 경우에서 단어 단위로 띄어쓰기가 이루어지기 때문에 띄어쓰기 토큰화와 단어 토큰화가 거의 같기 때문
- 한국어는 영어와는 달리 띄어쓰기만으로는 토큰화를 하기에 부족

텍스트 전처리

Robot Media Laboratory

한국어 토큰화

- 한국어의 경우에는 띄어쓰기 단위가 되는 단위를 '어절'이라고 하는데 어절 토큰화는 한국어 NLP에서 지양되고 있음
- 어절 토큰화와 단어 토큰화는 다름
- 한국어가 영어와는 다른 형태를 가지는 언어인 교착어라는 점에서 기인
- 교착어란 조사, 어미 등을 붙여서 말을 만드는 언어를 말함

텍스트 전처리

Robot Media Laboratory

한국어 토큰화

- 영어와는 달리 한국어에는 조사라는 것이 존재
  - Ex), 그라는 단어 하나에도 '그가', '그에게', '그를', '그와', '그는'과 같이 다양한 조사가 '그'라는 글자 뒤에 띄어쓰기 없이 바로 붙게됨
- 대부분의 한국어 NLP에서 조사는 분리해줄 필요가 있음

- 띄어쓰기 단위가 영어처럼 독립적인 단어가 아님
- 한국어는 어절이 독립적인 단어로 구성되는 것이 아니라 조사 등의 무언가가 붙어있는 경우가 많아서 이를 전부 분리해줘야 함

텍스트 전처리

Robot Media Laboratory

- **한국어 토큰화에서는 형태소(morpheme) 란 개념을 반드시 이해해야 함**
- **형태소(morpheme)**
  - 뜻을 가진 가장 작은 말의 단위
  - 이 형태소에는 두 가지 형태소가 있는데 자립 형태소와 의존 형태소

- **자립 형태소**
  - 접사, 어미, 조사와 상관없이 자립하여 사용할 수 있는 형태소
  - 그 자체로 단어가 됨
  - 체언(명사, 대명사, 수사), 수식언(관형사, 부사), 감탄사 등이 있음

- **의존 형태소**
  - 다른 형태소와 결합하여 사용되는 형태소.
  - 접사, 어미, 조사, 어간

텍스트 전처리

Robot Media Laboratory

- 문장 : 에디가 책을 읽었다

<띄워쓰기>

['에디가', '책을', '읽었다']

<형태소>

자립 형태소 : 에디, 책

의존 형태소 : -가, -을, 읽-, -었, -다

- '에디'라는 사람 이름과 '책'이라는 명사를 얻어낼 수 있다
- 이를 통해 유추할 수 있는 것은 한국어에서 영어에서의 단어 토큰화와 유사한 형태를 얻으려면 어절 토큰화가 아니라 형태소 토큰화를 수행해야한다

한국어는 띄어쓰기가 영어보다 잘 지켜지지 않는다.

- 뉴스 기사와 같이 띄어쓰기를 철저하게 지키려고 노력하는 글이라면 좋겠지만, 많은 경우 띄어쓰기가 틀렸거나 지켜지지 않는 코퍼스가 많다
- 한국어는 영어권 언어와 비교하여 띄어쓰기가 어렵고 잘 지켜지지 않는 경향이 있다
- 가장 기본적인 견해는 한국어의 경우 띄어쓰기가 지켜지지 않아도 글을 쉽게 이해할 수 있는 언어
- 띄어쓰기가 없던 한국어에 띄어쓰기가 보편화된 것도 근대(1933년, 한글맞춤법통일안)의 일

텍스트 전처리

Robot Media Laboratory

- EX1) 제가이렇게띄어쓰기를전혀하지않고글을썼다고하더라도글을이해할수있습니다.
- EX2) Tobeornottobethatisthequestion

텍스트 전처리

Robot Media Laboratory

- 단어는 표기는 같지만 품사에 따라서 단어의 의미가 달라지기도 함
- 단어의 의미를 제대로 파악하기 위해서는 해당 단어가 어떤 품사로 쓰였는지 보는 것이 주요 지표가 됨

- 품사 태깅(part-of-speech tagging)
  - 단어 토큰화 과정에서 각 단어가 어떤 품사로 쓰였는지를 구분해놓는 작업

텍스트 전처리

Robot Media Laboratory

- **정제(cleaning)**
  - 갖고 있는 코퍼스로부터 노이즈 데이터를 제거
  - 완벽한 정제 작업은 어려운 편
  - 대부분의 경우 이 정도면 됐다.라는 일종의 합의점을 찾음

- **정규화(normalization)**
  - 표현 방법이 다른 단어들을 통합시켜서 같은 단어로 만들기

텍스트 전처리

Robot Media Laboratory

- **규칙에 기반한 통합**
  - 정규화 규칙의 예로서 같은 의미를 갖고있음에도, 표기가 다른 단어들을 하나의 단어로 정규화하는 방법을 사용할 수 있음
- **대, 소문자 통합**
  - 대부분의 글은 소문자로 작성되기 때문에 대, 소문자 통합 작업은 대부분 대문자를 소문자로 변환하는 소문자 변환작업
- **불필요한 단어의 제거**
  - 등장 빈도가 적은 단어
  - 길이가 짧은 단어

텍스트 전처리

Robot Media Laboratory

- 정규화 기법 중 단어의 개수를 줄일 수 있는 기법
  - 표제어 추출(lemmatization)
  - 어간 추출(stemming)
- 하나의 단어로 일반화시킬 수 있다면 하나의 단어로 일반화시켜서 문서 내의 단어 수를 줄이겠다는 것
- 단어의 빈도수를 기반으로 문제를 풀고자 하는 자연어 처리 문제에서 주로 사용
- 자연어 처리에서 전처리 정규화의 지향점은 언제나 갖고 있는 복잡성을 줄이는 일

텍스트 전처리

Robot Media Laboratory

- **표제어 추출(Lemmatization)**
  - '표제어' 또는 '기본 사전형 단어' 정도의 의미
  - 표제어 추출은 단어들이 다른 형태를 가지더라도, 그 뿌리 단어를 찾아가서 단어의 개수를 줄일 수 있는지 판단
  - 표제어 추출을 하는 가장 섬세한 방법은 단어의 형태학적 파싱을 먼저 진행

- **어간(stem)**
  - 단어의 의미를 담고 있는 단어의 핵심 부분.
- **접사(affix)**
  - 단어에 추가적인 의미를 주는 부분.

- 형태학적 파싱은 이 두 가지 구성 요소를 분리하는 작업을 말함

텍스트 전처리

Robot Media Laboratory

어간 추출(Stemming)

- 어간(Stem)을 추출하는 작업
- 형태학적 분석을 단순화한 버전이
- 정해진 규칙만 보고 단어의 어미를 자르는 어림짐작의 작업
- 섬세한 작업이 아니기 때문에 어간 추출 후에 나오는 결과 단어는 사전에 존재하지 않는 단어일 수도 있음

텍스트 전처리

Robot Media Laboratory

- 한국어는 5언 9품사의 구조를 가지고 있다

1. 체언 : 명사, 대명사, 수사
2. 수식언 : 관형사, 부사
3. 관계언 : 조사
4. 독립언 : 감탄사
5. 용언 : 동사, 형용사

- 이 중 용언에 해당되는 '동사'와 '형용사'는 어간(stem)과 어미(ending)의 결합으로 구성

텍스트 전처리

Robot Media Laboratory

- **동사변화**
  - 용언의 어간(stem)이 어미(ending)를 가지는 일
  - 규칙 불규칙 형이 있음

- **어간(stem)**
  - 용언(동사, 형용사)을 활용할 때, 원칙적으로 모양이 변하지 않는 부분. 활용에서 어미에 선행하는 부분. 때론 어간의 모양도 바뀔 수 있음(예: 긋다, 긋고, 그어서, 그어라).

- **어미(ending)**
  - 용언의 어간 뒤에 붙어서 활용하면서 변하는 부분이며, 여러 문법적 기능을 수행

텍스트 전처리

Robot Media Laboratory

- **규칙 동사변화**
  - 어간이 어미를 취할 때, 어간의 모습이 일정

잡/어간 + 다/어

  - 어간이 어미가 붙기전의 모습과 어미가 붙은 후의 모습이 같으므로, 규칙 기반으로 어미를 단순히 분리해주면 어간 추출이 됨

텍스트 전처리

Robot Media Laboratory

- **불규칙 동사변화**
  - 어간이 어미를 취할 때 어간의 모습이 바뀌거나 취하는 어미가 특수한 어미일 경우

듣-, 돕-, 곱-, 잇-, 오르-, 노랗-’ 등이 ‘듣/들-,
돕/도우-, 곱/고우-, 잇/이-, 올/올-, 노랗/노라-’

- **어간이 어미가 붙는 과정에서 어간의 모습이 바뀌었으므로 단순한 분리만으로 어간 추출이 되지 않고 좀 더 복잡한 규칙을 필요**

텍스트 전처리

Robot Media Laboratory

불용어(stopword)

- 실제 의미 분석을 하는데 거의 기여하는 바가 없는 단어들
- 한국어에서 불용어를 제거하는 방법으로 간단하게는 토큰화 후에 조사, 접속사 등을 제거
- 사용자가 직접 불용어 사전을 만들게 되는 경우가 많다
- 불용어가 많은 경우에는 코드 내에서 직접 정의하지 않고 txt 파일이나 csv 파일로 정리해놓고 이를 불러와서 사용하기도 함

텍스트 전처리

Robot Media Laboratory

정규표현식

.	한 개의 임의의 문자를 나타냅니다. (줄바꿈 문자인 ₩n는 제외)
?	앞의 문자가 존재할 수도 있고, 존재하지 않을 수도 있습니다. (문자가 0개 또는 1개)
*	앞의 문자가 무한개로 존재할 수도 있고, 존재하지 않을 수도 있습니다. (문자가 0개 이상)
+	앞의 문자가 최소 한 개 이상 존재합니다. (문자가 1개 이상)
^	뒤의 문자열로 문자열이 시작됩니다.
$	앞의 문자열로 문자열이 끝납니다.
{숫자}	숫자만큼 반복합니다.
{숫자1, 숫자2}	숫자1 이상 숫자2 이하만큼 반복합니다. ?, *, +를 이것으로 대체할 수 있습니다.
{숫자,}	숫자 이상만큼 반복합니다.
[ ]	대괄호 안의 문자들 중 한 개의 문자와 매치합니다. [amk]라고 한다면 a 또는 m 또는 k 중 하나라도 존재하면 매치를 의미합니다. [a-z]와 같이 범위를 지정할 수도 있습니다. [a-zA-Z]는 알파벳 전체를 의미하는 범위이며, 문자열에 알파벳이 존재하면 매치를 의미합니다.
[^문자]	해당 문자를 제외한 문자를 매치합니다.
|	A|B와 같이 쓰이며 A 또는 B의 의미를 가집니다.

텍스트 전처리

Robot Media Laboratory

정규 표현식 문법에는 역 슬래쉬(₩)를 이용하여 자주 쓰이는 문자 규칙

₩	역 슬래쉬 문자 자체를 의미합니다
₩d	모든 숫자를 의미합니다. [0-9]와 의미가 동일합니다.
₩D	숫자를 제외한 모든 문자를 의미합니다. [^0-9]와 의미가 동일합니다.
₩s	공백을 의미합니다. [₩t₩n₩r₩f₩v]와 의미가 동일합니다.
₩S	공백을 제외한 문자를 의미합니다. [^ ₩t₩n₩r₩f₩v]와 의미가 동일합니다.
₩w	문자 또는 숫자를 의미합니다. [a-zA-Z0-9]와 의미가 동일합니다.
₩W	문자 또는 숫자가 아닌 문자를 의미합니다. [^a-zA-Z0-9]와 의미가 동일합니다.

텍스트 전처리

Robot Media Laboratory

정규표현식 모듈 함수

re.compile()	정규표현식을 컴파일하는 함수입니다. 다시 말해, 파이썬에게 전해주는 역할을 합니다. 찾고자 하는 패턴이 빈번한 경우에는 미리 컴파일해놓고 사용하면 속도와 편의성면에서 유리합니다.
re.search()	문자열 전체에 대해서 정규표현식과 매치되는지를 검색합니다.
re.match()	문자열의 처음이 정규표현식과 매치되는지를 검색합니다.
re.split()	정규 표현식을 기준으로 문자열을 분리하여 리스트로 리턴합니다.
re.findall()	문자열에서 정규 표현식과 매치되는 모든 경우의 문자열을 찾아서 리스트로 리턴합니다. 만약, 매치되는 문자열이 없다면 빈 리스트가 리턴됩니다.
re.finditer()	문자열에서 정규 표현식과 매치되는 모든 경우의 문자열에 대한 이터레이터 객체를 리턴합니다.
re.sub()	문자열에서 정규 표현식과 일치하는 부분에 대해서 다른 문자열로 대체합니다.

정수 인코딩(Integer Encoding)

- 각 단어를 고유한 정수에 맵핑(mapping)시키는 전처리 작업
- 단어를 빈도수 순으로 정렬한 단어 집합(vocabulary)을 만들고, 빈도수가 높은 순서대로 차례로 낮은 숫자부터 정수를 부여하는 방법
- dictionary
- Counter

텍스트 전처리

Robot Media Laboratory

패딩(Padding)

- 병렬 연산을 위해서 여러 문장의 길이를 임의로 동일하게 맞춰주는 작업

텍스트 전처리

Robot Media Laboratory

- **원-핫 인코딩(One-Hot Encoding)**
  - 단어 집합의 크기를 벡터의 차원으로 하고, 표현하고 싶은 단어의 인덱스에 1의 값을 부여하고, 다른 인덱스에는 0을 부여하는 단어의 벡터 표현 방식

- **단어집합 (vocabulary)**
  - **기본적으로 book과 books와 같이 단어의 변형 형태도 다른 단어로 간주**

- **원-핫 인코딩(One-Hot Encoding)의 한계**
  - 벡터를 저장하기 위해 필요한 공간이 계속 늘어난다는 단점
  - 다른 표현으로는 벡터의 차원이 늘어난다고 표현

언어모델

언어모델

Robot Media Laboratory

- **언어 모델이란? (LM, Language Model)**
  - 단어 시퀀스에 확률을 할당하는 모델, 이전 단어들을 이용하여 다음 단어를 예측함
  - BERT는 양쪽 단어들로부터 가운데 단어 예측

- **통계를 이용한 방법과 인공신경망을 이용한 방법**
- **최근에는 인공신경망 방법이 더 성능 좋음**

언어모델

Robot Media Laboratory

통계적 언어 모델(Statistical Language Model, SLM)

- 확률 기반의 언어모델
- m개의 단어 w1,w2,...,wm 열(word sequence)이 주어졌을 때 문장으로써 성립될 확률 P(w1,w2,...,wm) 을 출력함으로써 이 단어 열이 실제로 현실에서 사용될 수 있는 문장(sentence)인지를 판별하는 모델
- 조건부 확률
- 문장에 대한확률을 카운트 기반의 접근으로 계산

언어모델

Robot Media Laboratory

- **조건부 확률**
  - 주어진 사건이 일어났다는 가정 하에 다른 한 사건이 일어날 확률
  - 기존의 단어를 바탕으로 만들어질 다음 단어의 확률
  - 기존의 단어들의 정보=문맥정보

$$
\begin{aligned}
P(w_1, w_2, \ldots, w_m) &= P(w_1, w_2, \ldots, w_{m-1}) \cdot P(w_m \mid w_1, w_2, \ldots, w_{m-1}) \\
&= P(w_1, w_2, \ldots, w_{m-2}) \cdot P(w_{m-1} \mid w_1, w_2, \ldots, w_{m-2}) \cdot P(w_m \mid w_1, w_2, \ldots, w_{m-1}) \\
&= P(w_1) \cdot P(w_2 \mid w_1) \cdot P(w_3 \mid w_1, w_2) P(w_4 \mid w_1, w_2, w_3) \cdots P(w_m \mid w_1, w_2, \ldots, w_{m-1})
\end{aligned}
$$

- **문장의 확률**

$$
P(w_1, w_2, w_3, w_4, w_5, \ldots w_n) = \prod_{n=1}^{n} P(w_n | w_1, \ldots, w_{n-1})
$$

언어모델

Robot Media Laboratory

- **조건부 확률**
  - 유니그램 모형(Unigram Model)
  - 바이그램 모형(Bigram Model)
  - N그램 모형(N-gram Model)

언어모델

Robot Media Laboratory

- **유니그램 모델**
  - 모든 단어의 활용이 완전히 서로 독립이라면 단어 열의 확률은 다음과 같이 각 단어의 확률의 곱되는 기반의 모델

$$P(w_1, w_2, \ldots, w_m) = \prod_{i=1}^{m} P(w_i)$$

- **바이그램 모델 or 마코프 모형(Markov Model)**
  - 단어의 활용이 바로 전 단어에만 의존한다면 단어 열의 확률

$$P(w_1, w_2, \ldots, w_m) = P(w_1) \prod_{i=2}^{m} P(w_i \mid w_{i-1})$$

언어모델

Robot Media Laboratory

통계적 언어 모델한계

- SLM의 한계는 훈련 코퍼스에 확률을 계산하고 싶은 문장이나 단어가 없을 수 있다는 점
- 확률을 계산하고 싶은 문장이 길어질수록 갖고 있는 코퍼스에서 그 문장이 존재하지 않을 가능성이 높다.
- 단어의 확률을 구하고자 기준 단어의 앞 단어를 전부 포함해서 카운트하는 것이 아니라, 앞 단어 중 임의의 개수만 포함해서 카운트하여 근사하자는 것입니다. 이렇게 하면 갖고 있는 코퍼스에서 해당 단어의 시퀀스를 카운트할 확률이 높아짐

언어모델

Robot Media Laboratory

- **N-gram 언어 모델(N-gram Language Model)**
  - 임의의 개수를 정하기 위한 기준을 위해 사용하는 것
  - n개의 연속적인 단어 나열
  - n개의 단어 뭉치 단위로 끊어서 이를 하나의 토큰으로 간주
  - n-gram을 통한 언어 모델에서는 다음에 나올 단어의 예측은 오직 n-1개의 단어에만 의존

언어모델

Robot Media Laboratory

- **Ex)** An adorable little boy is spreading smiles
- **유니그램**
  - an, adorable, little, boy, is, spreading, smiles
- **바이그램**
  - an adorable, adorable little, little boy, boy is, is spreading, spreading smiles
- **4-그램(n-그램)**
  - an adorable little boy, adorable little boy is, little boy is spreading, boy is spreading smiles

언어모델

Robot Media Laboratory

- **N-gram Language Model의 한계**
  - 희소 문제(Sparsity Problem)
  - n을 선택하는 것은 trade-off 문제.

- **희소문제**
  - 충분한 데이터를 관측하지 못하여 언어를 정확히 모델링하지 못하는 문제

단어표현

단어 표현

Robot Media Laboratory

- **단어의 표현 방법**

- **국소 표현(Local Representation)**
  - 해당 단어 그 자체만 보고, 특정값을 맵핑하여 단어를 표현하는 방법
  - 이산 표현(Discrete Representation)
  - BoW, DTM TF-IDF

- **분산 표현(Distributed Representation)**
  - 그 단어를 표현하고자 주변을 참고하여 단어를 표현하는 방법
  - 연속 표현(Continuous Represnetation)
  - Word2Vec,FastText

단어 표현

Robot Media Laboratory

- **Bag of Words(BoW)**
  - 단어들의 순서는 전혀 고려하지 않고, 단어들의 출현 빈도(frequency)에만 집중하는 텍스트 데이터의 수치화 표현 방법

- **문서 단어 행렬(Document-Term Matrix, DTM)**
  - 다수의 문서에서 등장하는 각 단어들의 빈도를 행렬로 표현
  - BoW들을 결합한 표현 방법
  - BoW와 다른 표현 방법이 아니라 BoW 표현을 다수의 문서에 대해서 행렬로 표현하고 부르는 용어

단어 표현

Robot Media Laboratory

- **문서 단어 행렬(Document-Term Matrix)의 한계**
  - 희소 표현(Sparse representation)
  - 단순 빈도 수 기반 접근

단어 표현

Robot Media Laboratory

TF-IDF(Term Frequency-Inverse Document Frequency)

- 단어의 빈도와 역 문서 빈도(문서의 빈도에 특정 식을 취함)를 사용하여 DTM 내의 각 단어들마다 중요한 정도를 가중치로 주는 방법
- 주로 문서의 유사도를 구하는 작업, 검색 시스템에서 검색 결과의 중요도를 정하는 작업, 문서 내에서 특정 단어의 중요도를 구하는 작업 등 사용

$$idf(d,t) = log(\frac{n}{1+df(t)})$$

- tf(d,t) : 특정 문서 d에서의 특정 단어 t의 등장 횟수.
- df(t) : 특정 단어 t가 등장한 문서의 수.
- idf(d, t) : df(t)에 반비례하는 수.

벡터의 유사도

벡터의 유사도

Robot Media Laboratory

- 문장이나 문서의 유사도

벡터의 유사도

Robot Media Laboratory

- **문장이나 문서의 유사도**
  - 문서의 유사도는 주로 문서들 간에 동일한 단어 또는 비슷한 단어가 얼마나 공통적으로 많이 사용되었는지에 의존

벡터의 유사도

Robot Media Laboratory

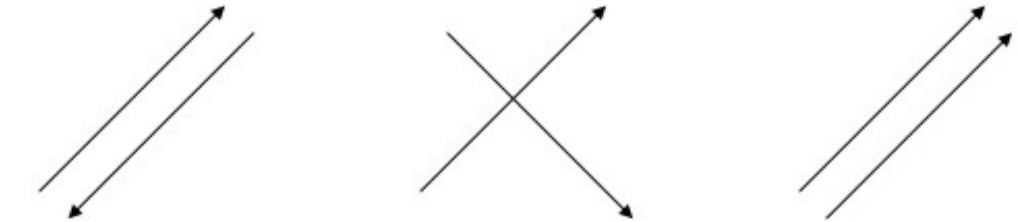
- **백터의 유사도**
  - 코사인 유사도
  - 자카드 유사도
  - 유클리디안 유사도
  - 멘하탄 유사도

벡터의 유사도

Robot Media Laboratory

코사인 유사도(Cosine Similarity)

- 두 벡터 간의 코사인 각도를 이용하여 구할 수 있는 두 벡터의 유사도



코사인 유사도 : -1　　코사인 유사도 : 0　　코사인 유사도 : 1

$$similarity = cos(\Theta) = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

벡터의 유사도

벡터의 유사도

Robot Media Laboratory

- 문장이나 문서의 유사도

벡터의 유사도

Robot Media Laboratory

- **문장이나 문서의 유사도**
  - 문서의 유사도는 주로 문서들 간에 동일한 단어 또는 비슷한 단어가 얼마나 공통적으로 많이 사용되었는지에 의존

벡터의 유사도

Robot Media Laboratory

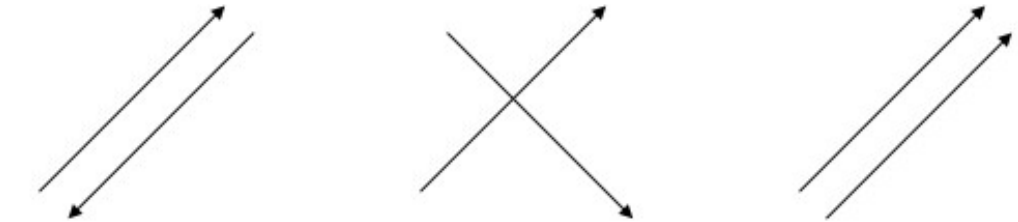
- **백터의 유사도**
  - 코사인 유사도
  - 자카드 유사도
  - 유클리디안 유사도
  - 멘하탄 유사도

벡터의 유사도

Robot Media Laboratory

코사인 유사도(Cosine Similarity)

- 두 벡터 간의 코사인 각도를 이용하여 구할 수 있는 두 벡터의 유사도



코사인 유사도 : -1　　코사인 유사도 : 0　　코사인 유사도 : 1

$$similarity = cos(\Theta) = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

벡터의 유사도

Robot Media Laboratory

자카드 유사도

- 합집합에서 교집합의 비율을 바탕으로 대상의 유사도 계산방식
- 0과 1사이의 값 동일하다면 1, 공통 원소가 없다면 0

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

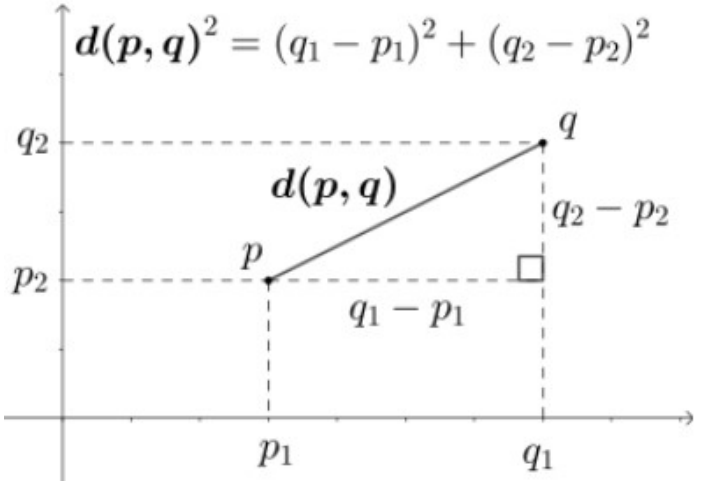
벡터의 유사도

Robot Media Laboratory

유클리디안(유클리드 거리) 유사도

- 피타고라스의 정리를 통해 두 점 사이의 거리를 구하는 것과 동일
- 점과 점사이의 거리

$$\sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \ \dots \ + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$



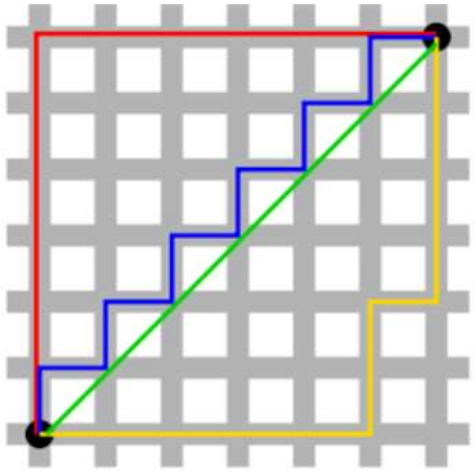
벡터의 유사도

Robot Media Laboratory

- **멘하탄 유사도**
  - 멘하탄거리를 이용하여 유사도 측정

$$MaDistance = \sum_{i=1}^{n} |a_i - b_i|$$

- **멘하탄거리**
  - 격자로 이뤄진 공간에서 출발점에서 도착점까지 길을 따라 만들어진 최단거리



벡터의 유사도

Robot Media Laboratory

방법	특징	수식
자카드 유사도	• 두 문장을 각 단어의 집합으로 만든 뒤, 집합을 통해 유사도를 측정 • 1에 가까울수록 유사도가 높음 • 0~1사이의 값을 가짐	$J(A,B) = \frac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert} = \frac{\lvert token\ in\ A\ \cap\ token\ in\ B \rvert}{\lvert token\ in\ A\ \cup\ token\ in\ B \rvert}$
유클리디언 유사도	• 거리를 측정하는 공식과 같음, L_2거리라고도 함 • n차원 공간에서 두 점 사이의 최단 거리를 구하는 접근법 • 1보다 큰 값이 나올 수 있기 때문에 값의 제한을 걸어줌(L_1정규화)	$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$
맨하탄 유사도	• 사각형 격자로 이루어진 지도에서 출발점에서 도착점까지 가로지르지 않고 갈 수 있는 최단거리, L_1거리라고도 함	$MaDistance = \sum_{i=1}^{n} \lvert a_i - b_i \rvert$
코사인 유사도	• 두 벡터 사이의 코사인 각도를 구하는 방법 • -1~1사이를 가지고 1에 가까울수록 유사함 • 가장 널리 쓰임, 다른 방법보다 유사도가 정확함 → 단순 좌표상의 거리가 아닌, 단어의 공간에서 벡터 간의 각도를 구하기 때문에	$\cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{\lvert\lvert \vec{a} \rvert\rvert \cdot \lvert\lvert \vec{b} \rvert\rvert}$

탐색적 데이터 분석

EDA

Robot Media Laboratory

EDA(탐색적 데이터 분석)

- 수집한 데이터가 들어왔을 때, 이를 다양한 각도에서 관찰하고 이해하는 과정
- 데이터를 분석하기 전에 그래프나 통계적인 방법으로 자료를 직관적으로 바라보는 과정
- 데이터의 분포 및 값을 검토함으로써 데이터가 표현하는 현상을 더 잘 이해하고, 데이터에 대한 잠재적인 문제를 발견 가능
- 이를 통해, 본격적인 분석에 들어가기에 앞서 데이터의 수집을 결정 가능
- 다양한 각도에서 살펴보는 과정을 통해 문제 정의 단계에서 미쳐 발생하지 못했을 다양한 패턴을 발견하고, 이를 바탕으로 기존의 가설을 수정하거나 새로운 가설을 세울 수 있다.

EDA

Robot Media Laboratory

- **분석의 목적과 변수가 무엇이 있는지 확인.**
  - 개별 변수의 이름이나 설명을 가지는지 확인

- **데이터를 전체적으로 살펴보기**
  - 데이터에 문제가 없는지 확인.
  - head나 tail부분을 확인, 추가적으로 다양한 탐색(이상치, 결측치 등을 확인하는 과정)

- **데이터의 개별 속성값을 관찰**
  - 각 속성 값이 예측한 범위와 분포를 갖는지 확인. 만약 그렇지 않다면, 이유가 무엇인지를 확인.
  - 속성 간의 관계에 초점을 맞추어, 개별 속성 관찰에서 찾아내지 못했던 패턴을 발견 (상관관계, 시각화 등)

성능 최적화

성능 최적화

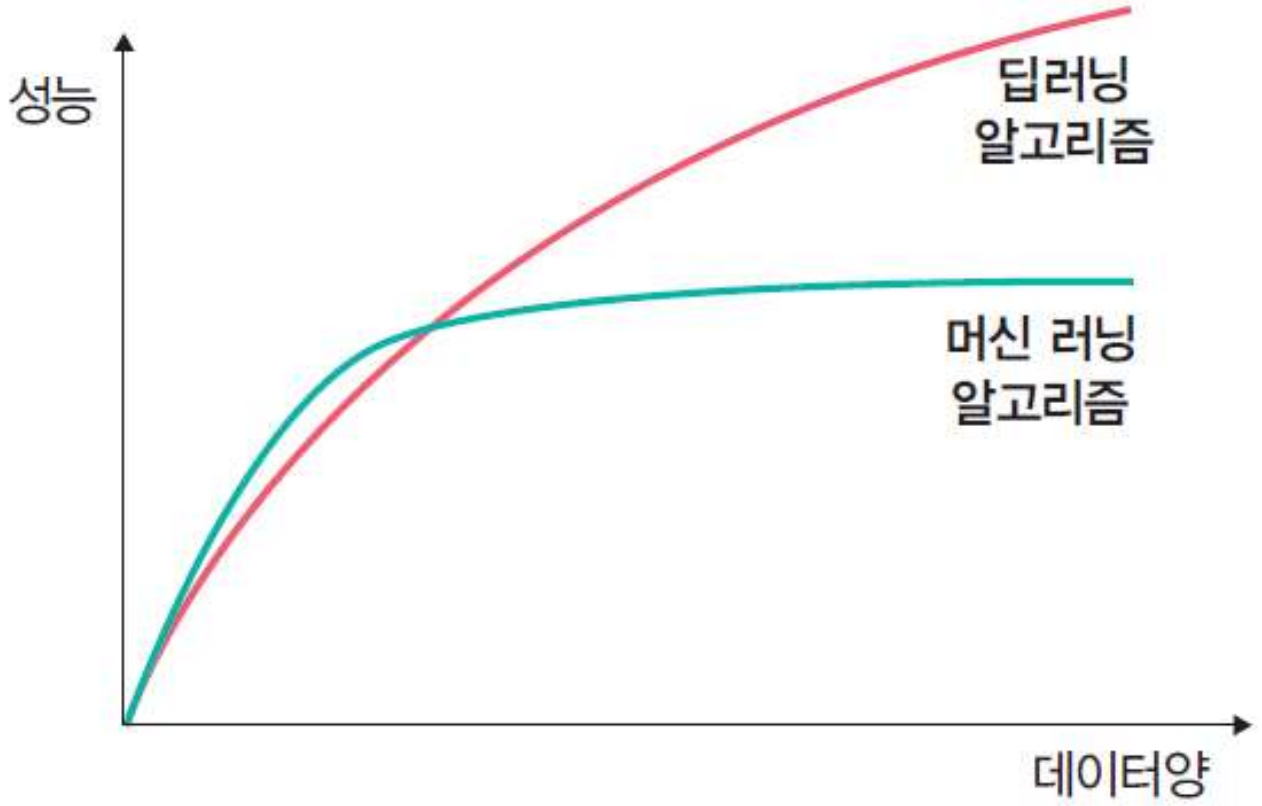
Robot Media Laboratory

- **데이터를 사용한 성능 최적화**
  - 데이터를 사용한 성능 최적화 방법은 많은 데이터를 수집하는 것
  - 데이터 수집이 여의치 않은 상황에서는 임의로 데이터를 생성하는 방법도 고려해 볼 수 있음

- **최대한 많은 데이터 수집하기**
  - 일반적으로 딥러닝이나 머신 러닝 알고리즘은 데이터양이 많을 수록 성능이 좋음
  - 가능한 많은 데이터(빅데이터)를 수집해야 함

성능 최적화

Robot Media Laboratory



성능 최적화

Robot Media Laboratory

- **데이터 생성하기**
  - 많은 데이터를 수집할 수 없다면 데이터를 만들어 사용할 수 있음
- **데이터 범위(scale) 조정하기**
  - 활성화 함수로 시그모이드를 사용한다면 데이터셋 범위를 0~1의 값을 갖도록 하고,
  - 하이퍼볼릭 탄젠트를 사용한다면 데이터셋 범위를 -1~1의 값을 갖도록 조정할 수 있음
- **정규화, 규제화, 표준화도 성능 향상에 도움이 됨**

성능 최적화

Robot Media Laboratory

알고리즘을 이용한 성능 최적화

- 머신 러닝과 딥러닝을 위한 알고리즘은 상당히 많음
- 수많은 알고리즘 중 우리가 선택한 알고리즘이 최적의 알고리즘이 아닐 수도 있음
- 유사한 용도의 알고리즘들을 선택하여 모델을 훈련시켜 보고 최적의 성능을 보이는 알고리즘을 선택해야 함
- 머신 러닝에서는 데이터 분류를 위해 SVM, K-최근접 이웃 알고리즘들을 선택하여 훈련시켜 보거나,
- 시계열 데이터의 경우 RNN, LSTM, GRU 등의 알고리즘을 훈련시켜 성능이 가장 좋은 모델을 선택하여 사용

성능 최적화

Robot Media Laboratory

알고리즘 튜닝을 위한 성능 최적화

- 성능 최적화를 하는 데 가장 많은 시간이 소요되는 부분
- 모델을 하나 선택하여 훈련시키려면 다양한 하이퍼파라미터를 변경하면서 훈련시키고 최적의 성능을 도출해야 함

진단

- 성능 향상이 어느 순간 멈추었다면 원인을 분석할 필요가 있음
- 문제를 진단하는데 사용할 수 있는 것이 모델에 대한 평가

- 훈련(train) 성능이 검증(test)보다 눈에 띄게 좋다면 과적합을 의심
  - 해결하기 위해 규제화
- 훈련과 검증 결과가 모두 성능이 좋지 않다면 과소적합(under-fitting)을 의심
  - 과소적합 상황에서는 네트워크 구조를 변경하거나 훈련을 늘리기 위해 에포크 수를 조정
- 훈련 성능이 검증을 넘어서는 변곡점이 있다면 조기 종료를 고려

성능 최적화

Robot Media Laboratory

가중치

- 가중치에 대한 초깃값은 작은 난수를 사용
- 작은 난수라는 숫자가 애매하다면 오토인코더 같은 비지도 학습을 이용하여 사전 훈련(가중치 정보를 얻기 위한 사전 훈련)을 진행한 후 지도 학습을 진행하는 것도 방법

학습률

- 학습률은 모델의 네트워크 구성에 따라 다르기 때문에 초기에 매우 크거나 작은 임의의 난수를 선택하여 학습 결과를 보고 조금씩 변경해야 함
  - 네트워크의 계층이 많다면 학습률은 높아야 하며, 네트워크의 계층이 몇 개 되지 않는다면 학습률은 작게 설정해야 함

성능 최적화

Robot Media Laboratory

활성화 함수

- 활성화 함수의 변경은 신중해야 함
- 활성화 함수를 변경할 때 손실 함수도 함께 변경해야 하는 경우가 많기 때문
- 일반적으로는 활성화 함수로 시그모이드나 하이퍼볼릭 탄젠트를 사용했다면 출력층에서는 소프트맥스나 시그모이드 함수를 많이 선택

배치와 에포크

- 일반적으로 큰 에포크와 작은 배치를 사용하는 것이 최근 딥러닝의 트렌드이기는 하지만, 적절한 배치 크기를 위해 훈련 데이터셋의 크기와 동일하게 하거나 하나의 배치로 훈련을 시켜보는 등 다양한 테스트를 진행하는 것이 좋음

성능 최적화

Robot Media Laboratory

옵티마이저 및 손실 함수

- 일반적으로 옵티마이저는 확률적 경사 하강법을 많이 사용
- 네트워크 구성에 따라 차이는 있지만 아담(Adam)이나 알엠에스프롭(RMSProp) 등도 좋은 성능을 보이고 있음
- 다양한 옵티마이저와 손실 함수를 적용해 보고 성능이 최고인 것을 선택

네트워크 구성

- 네트워크 구성은 네트워크 토폴로지(topology)라고도 함
- 최적의 네트워크를 구성하는 것 역시 쉽게 알 수 있는 부분이 아니기 때문에 네트워크 구성을 변경해 가면서 성능을 테스트해야 함

- 하나의 은닉층에 뉴런을 여러 개 포함시키거나(네트워크가 넓다고 표현),
- 네트워크 계층을 늘리되 뉴런 개수는 줄여 봄(네트워크가 깊다고 표현)
- 혹은 두 가지를 결합하는 방법으로 최적의 네트워크가 무엇인지 확인한 후 사용할 네트워크를 결정해야 함

성능 최적화

Robot Media Laboratory

앙상블을 이용한 성능 최적화

- 앙상블은 간단히 모델을 두 개 이상 섞어서 사용하는 것
- 앙상블을 이용하는 것도 성능 향상에 도움이 됨
- 알고리즘 튜닝을 위한 성능 최적화 방법은 하이퍼파라미터에 대한 경우의 수를 모두 고려해야 하기 때문에 모델 훈련이 수십 번에서 수백 번 필요할 수 있음
- 성능 향상은 단시간에 해결되는 것이 아니고, 수많은 시행착오를 겪어야 함

성능 최적화

Robot Media Laboratory

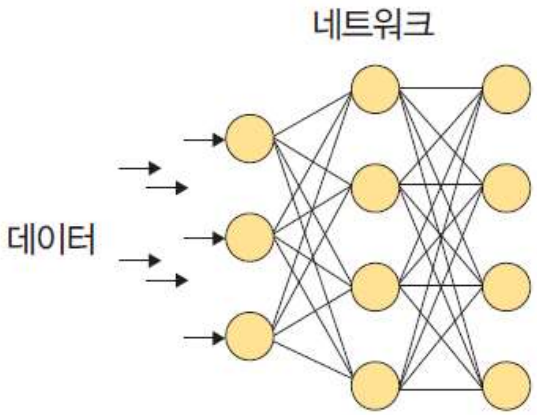
- **하이퍼파라미터를 이용한 성능 최적화**
  - 배치 정규화, 드롭아웃, 조기 종료가 있음

- **배치 정규화를 이용한 성능 최적화**
- **정규화(normalization)**
  - 데이터 범위를 사용자가 원하는 범위로 제한하는 것을 의미
  - 각 특성 범위(스케일(scale))를 조정한다는 의미로 특성 스케일링(feature scaling)

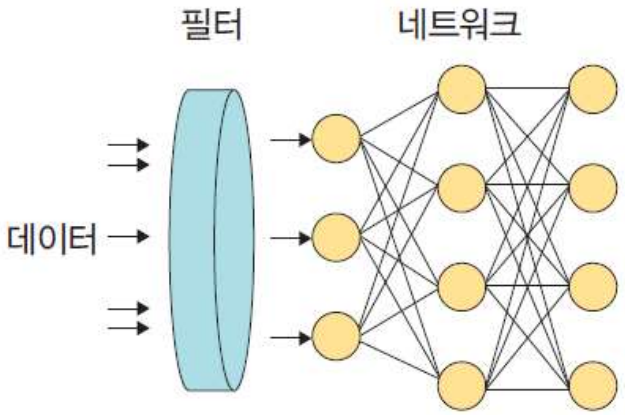
성능 최적화
Robot Media Laboratory

규제화(regularization)

- 모델 복잡도를 줄이기 위해 제약을 두는 방법
- 제약은 데이터가 네트워크에 들어가기 전에 필터를 적용한 것
- 규제를 이용하여 모델 복잡도를 줄이는 방법
- 드롭아웃, 조기 종료

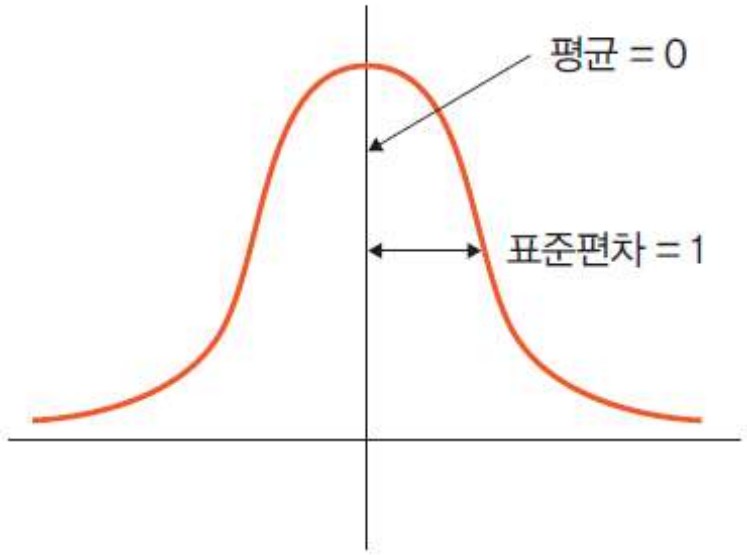


필터가 적용되지 않은 경우



필터가 적용된 경우

- **표준화(standardization)**
  - 기존 데이터를 평균은 0, 표준편차는 1인 형태의 데이터로 만드는 방법
  - 표준화 스칼라(standard scaler) 혹은 z-스코어 정규화(z-score normalization)



성능 최적화

Robot Media Laboratory

배치 정규화(batch normalization)

- 데이터 분포가 안정되어 학습 속도를 높일 수 있음
- 배치 정규화는 기울기 소멸(gradient vanishing)이나 기울기 폭발(gradient exploding) 같은 문제를 해결하기 위한 방법
- 일반적으로 기울기 소멸이나 폭발 문제를 해결하기 위해 손실 함수로 렐루(ReLU)를 사용하거나 초깃값 튜닝, 학습률(learning rate) 등을 조정
- 단계마다 활성화 함수를 거치면서 데이터셋 분포가 일정해지기 때문에 속도를 향상

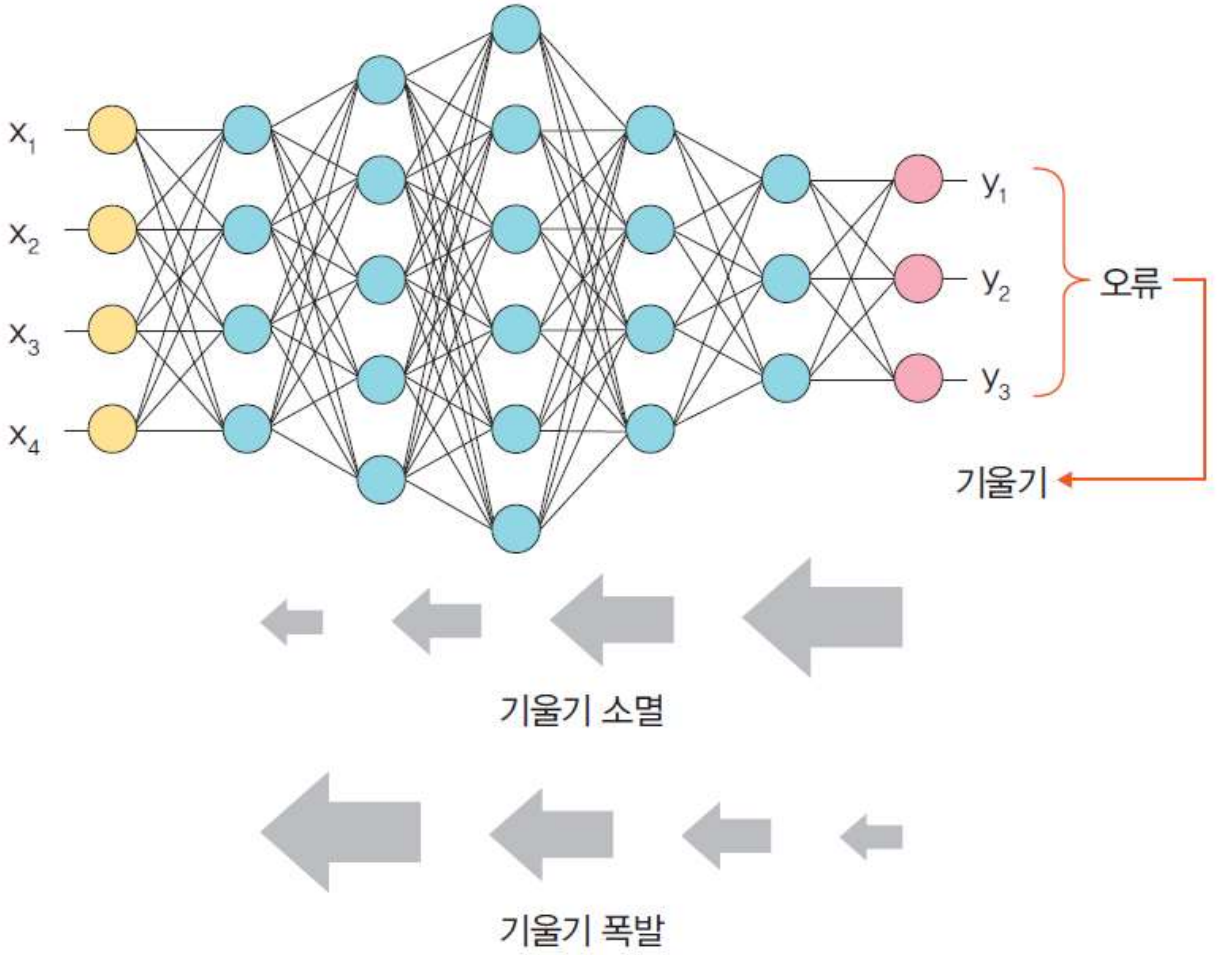
성능 최적화

Robot Media Laboratory

- **기울기 소멸**
  - 오차 정보를 역전파시키는 과정에서 기울기가 급격히 0에 가까워져 학습이 되지 않는 현상
- **기울기 폭발**
  - 학습 과정에서 기울기가 급격히 커지는 현상
- **기울기 소멸과 폭발 원인**
  - 내부 공변량 변화(internal covariance shift)
  - 네트워크의 각 층마다 활성화 함수가 적용되면서 입력 값들의 분포가 계속 바뀌는 현상

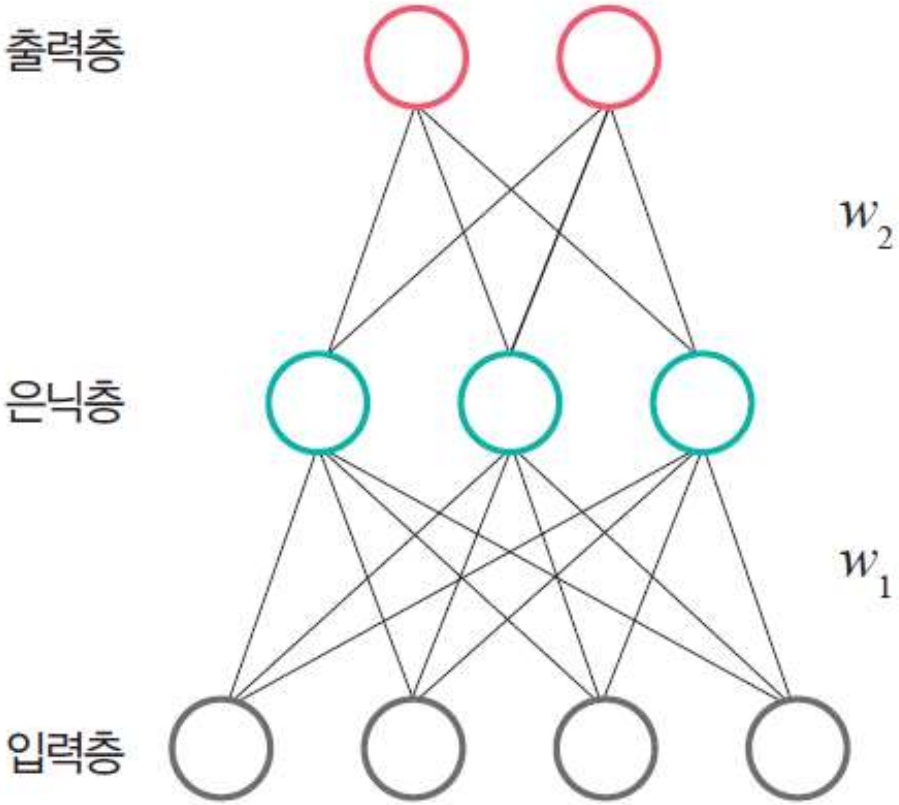
성능 최적화

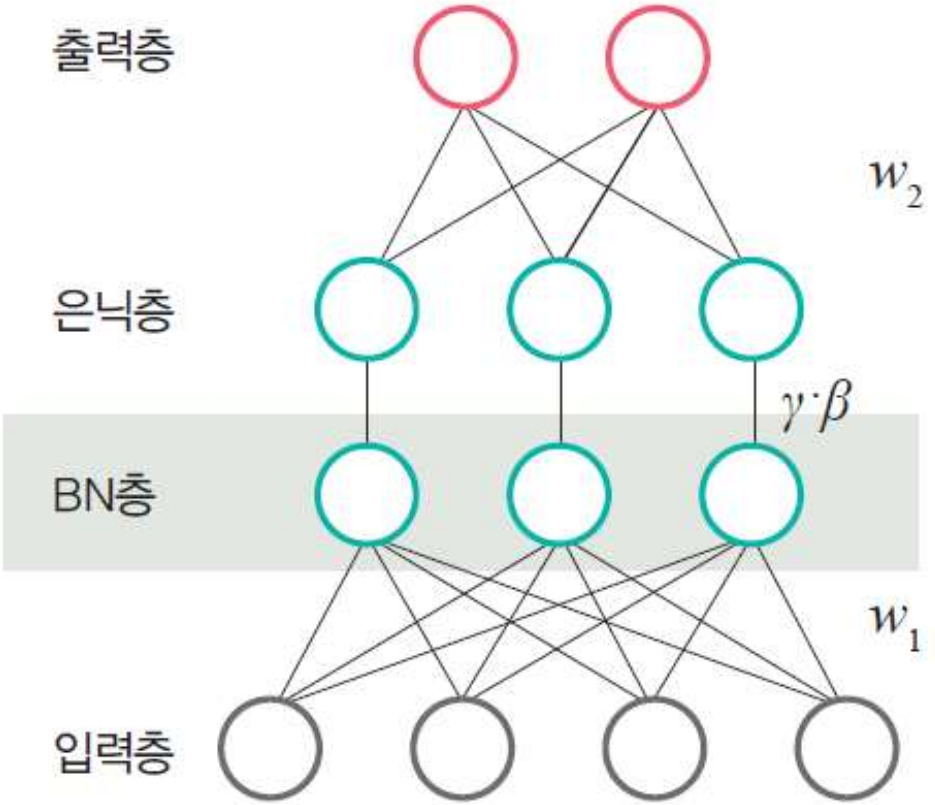
Robot Media Laboratory



성능 최적화

Robot Media Laboratory





성능 최적화

Robot Media Laboratory

배치 정규화(batch normalization)

- 분산된 분포를 정규 분포로 만들기 위해 표준화와 유사한 방식을 미니 배치(mini-batch)에 적용하여 평균은 0으로, 표준편차는 1로 유지하도록함

① 미니 배치 평균을 구함
② 미니 배치의 분산과 표준편차를 구함
③ 정규화를 수행
④ 스케일(scale)을 조정(데이터 분포 조정)함

$$\mu\beta \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \quad \text{----①}$$

$$\sigma^2\beta \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu\beta)^2 \quad \text{----②}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu\beta}{\sqrt{\sigma^2\beta + \epsilon}} \quad \text{----③}$$

$$y_i \leftarrow \gamma\hat{x}_i + \beta \Leftrightarrow BN_{\gamma,\beta}(x_i) \quad \text{----④}$$

$$\left(\begin{array}{l} \text{입력: } \beta=\{x_1, x_2, \cdots, x_n\} \\ \text{학습해야 할 하이퍼파라미터: } \gamma, \beta \\ \text{출력: } y_i=BN_{\gamma,\beta}(x_i) \end{array}\right)$$

성능 최적화

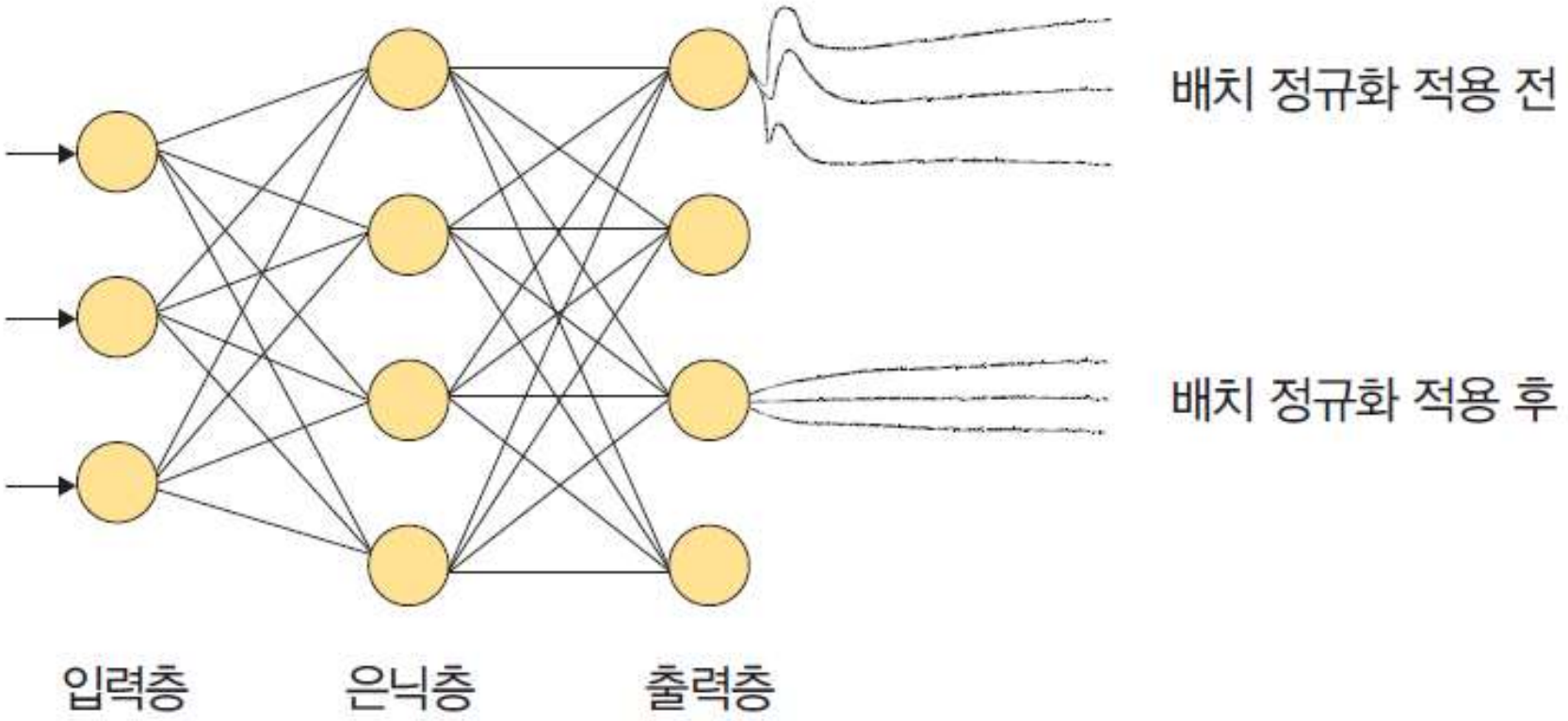
Robot Media Laboratory

배치 정규화(batch normalization) 단점

- 배치 크기가 작을 때는 정규화 값이 기존 값과 다른 방향으로 훈련될 수 있음
- RNN은 네트워크 계층별로 미니 정규화를 적용해야 하기 때문에 모델이 더 복잡해지면서 비효율적
- 문제들을 해결하기 위한 가중치 수정, 네트워크 구성 변경 등을 수행
- 배치 정규화를 적용하면 적용하지 않았을 때보다 성능이 좋아지기 때문에 많이 사용
- 신경망의 층이 깊어질수록 학습할 때 가정했던 입력 분포가 변화하여 엉뚱한 학습이 진행될 수 있음
- 배치 정규화를 적용해서 입력 분포를 고르게 맞추어 주면서 과적합을 해소

성능 최적화

Robot Media Laboratory

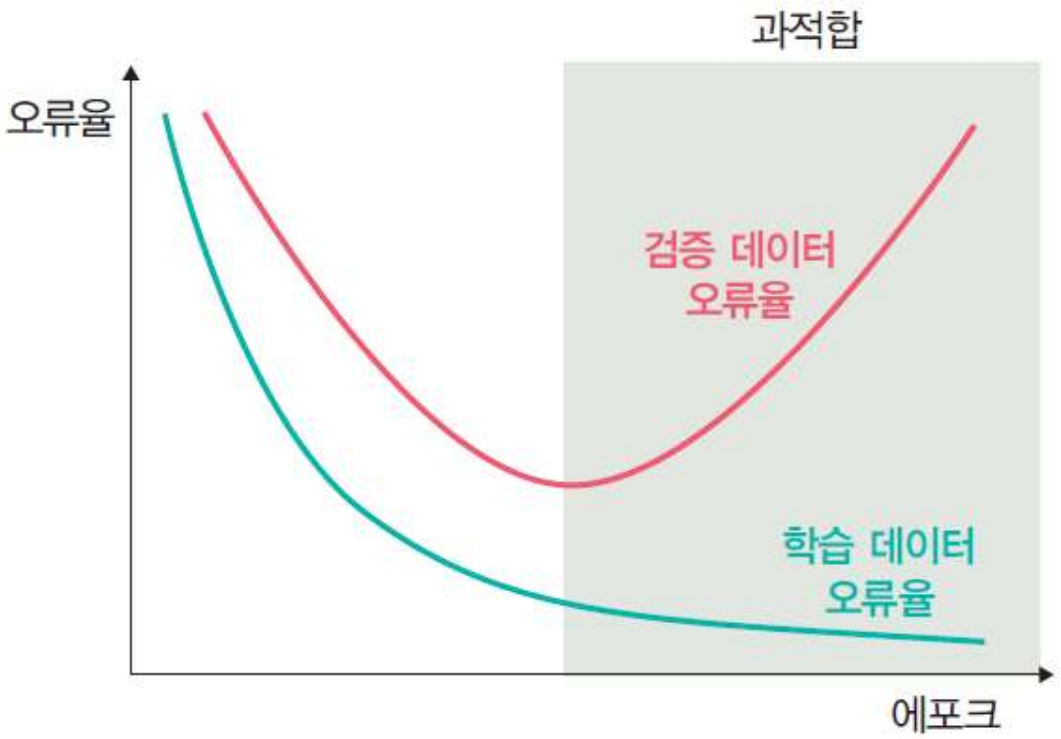


성능 최적화

Robot Media Laboratory

드롭아웃을 이용한 성능 최적화

- 훈련 데이터셋에 대해 훈련을 계속한다면 오류는 줄어들지만 검증 데이터셋에 대한 오류는 어느 순간부터 증가 -> 과적합



성능 최적화

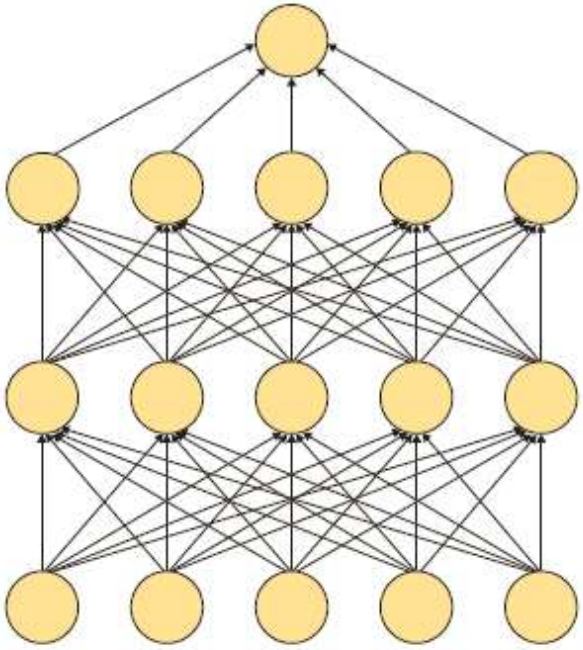
Robot Media Laboratory

드롭아웃(dropout)

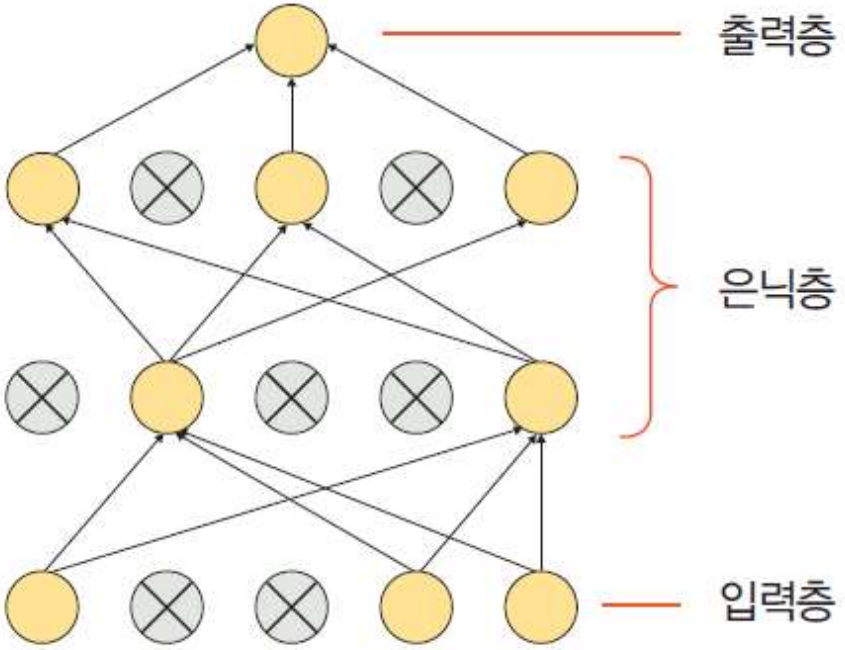
- 훈련할 때 일정 비율의 뉴런만 사용하고, 나머지 뉴런에 해당하는 가중치는 업데이트하지 않는 방법
- 매 단계마다 사용하지 않는 뉴런을 바꾸어 가며 훈련시킴
- 노드를 임의로 끄면서 학습하는 방법으로, 은닉층에 배치된 노드 중 일부를 임의로 끄면서 학습
- 꺼진 노드는 신호를 전달하지 않으므로 지나친 학습을 방지
- 적절히 사용되는 드롭아웃은 성능을 향상시키는 데 도움이 되므로 모델 최적화에 활용하기 좋음

성능 최적화

Robot Media Laboratory



일반적인 신경망



드롭아웃이
적용된 신경망

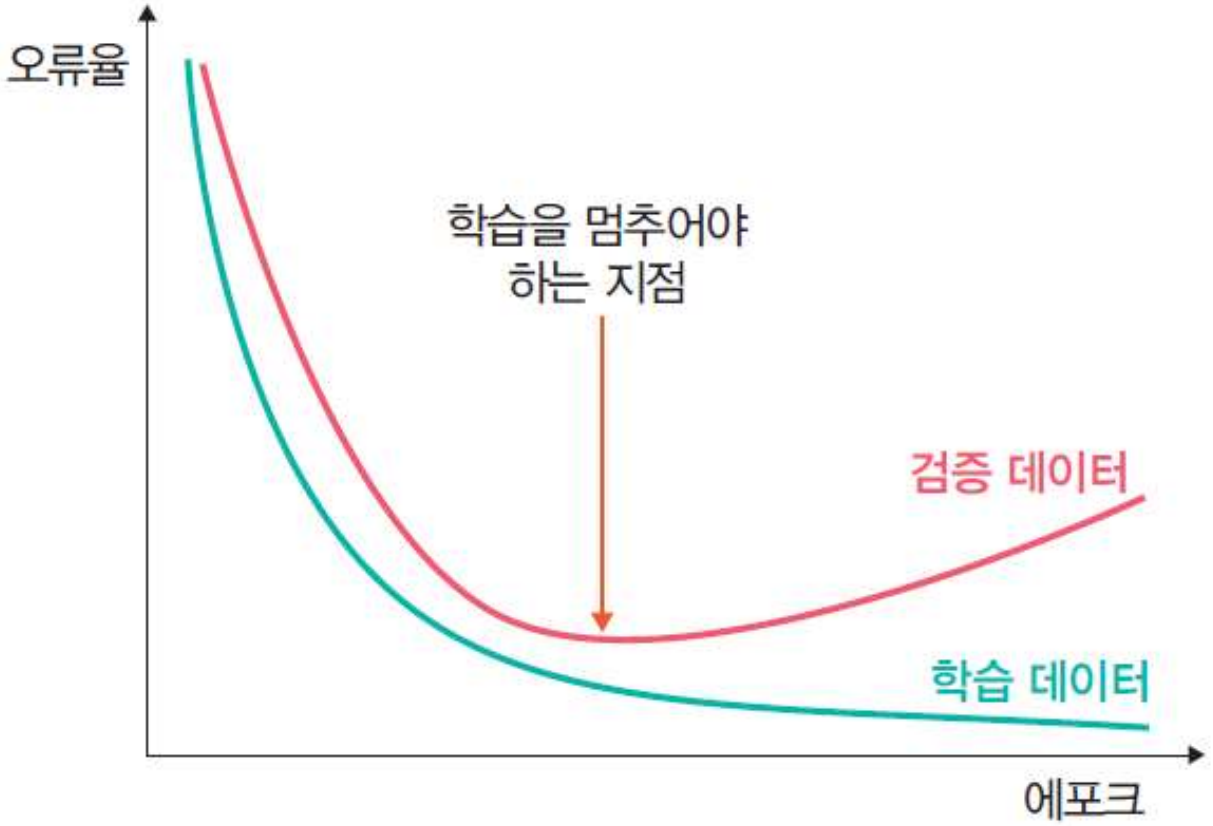
성능 최적화

Robot Media Laboratory

- **조기 종료(early stopping)**
  - 뉴럴 네트워크가 과적합을 회피하는 규제 기법
  - 훈련 데이터와 별도로 검증 데이터를 준비하고, 매 에포크마다 검증 데이터에 대한 손실(validation loss)을 측정하여 모델의 종료 시점을 제어함
  - 조기 종료는 검증에 대한 손실이 증가하는 시점에서 훈련을 멈추도록 조정함

성능 최적화

Robot Media Laboratory



순환신경망
RNN

RNN

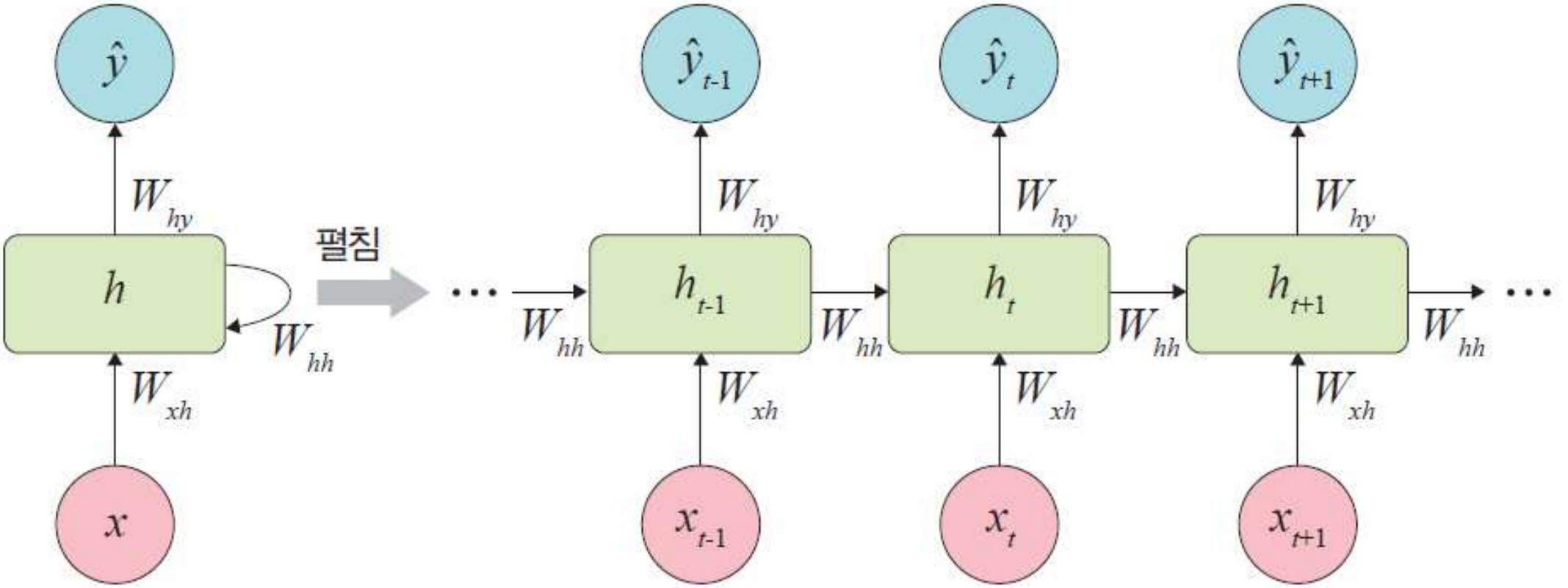
Robot Media Laboratory

RNN(Recurrent Neural Network)

- 시간적으로 연속성이 있는 데이터를 처리하려고 고안된 인공 신경망
- 은닉층 노드들이 연결되어 이전 단계 정보를 은닉층 노드에 저장할 수 있도록 구성한 신경망
- 'Recurrent(반복되는)'는 이전 은닉층이 현재 은닉층의 입력이 되면서 '반복되는 순환 구조를 갖는다'는 의미
- RNN이 기존 네트워크와 다른 점은 '기억(memory)'을 갖는다는 것
- 기억은 현재까지 입력 데이터를 요약한 정보
- 새로운 입력이 네트워크로 들어올 때마다 기억은 조금씩 수정되며, 결국 최종적으로 남겨진 기억은 모든 입력 전체를 요약한 정보

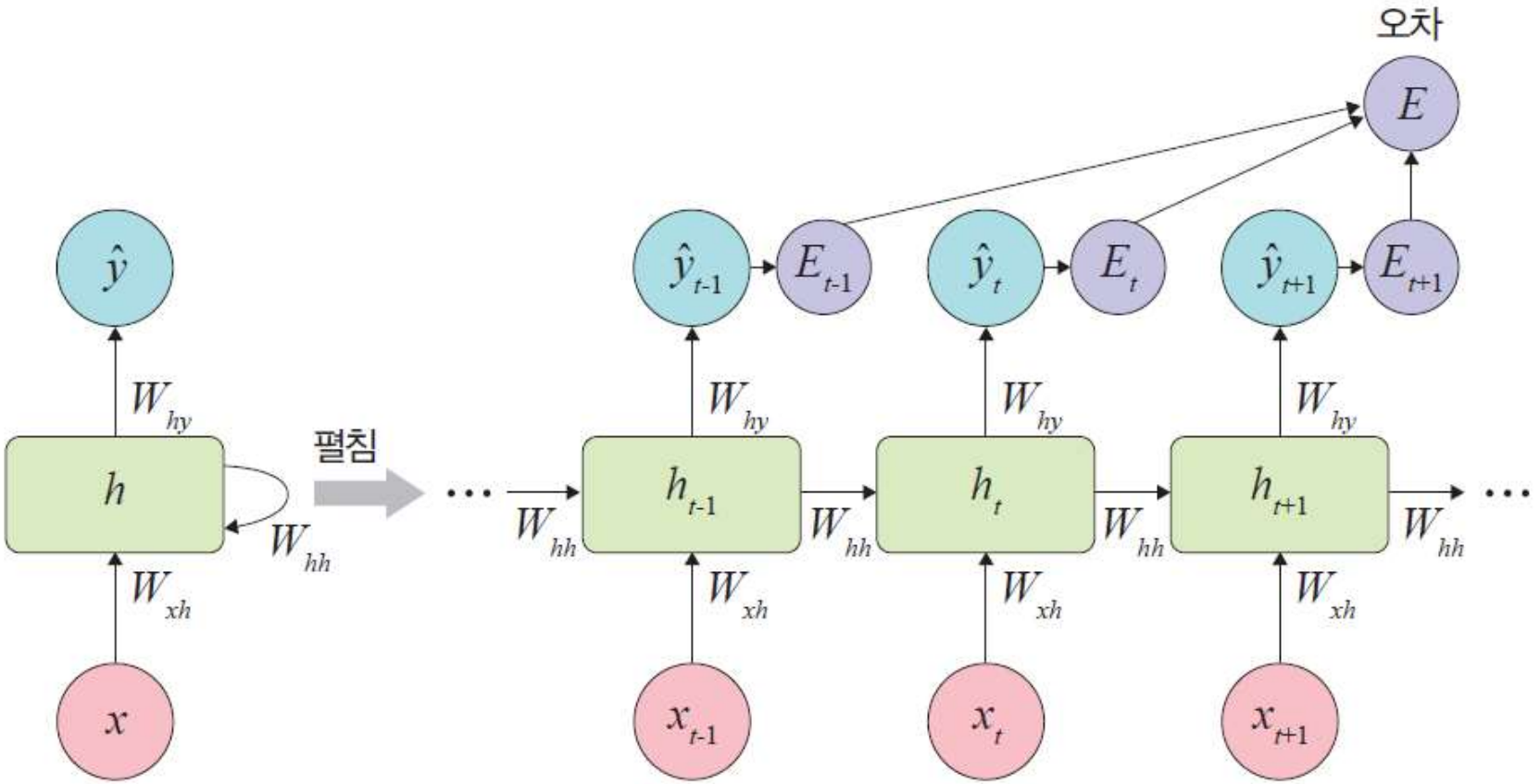
RNN

Robot Media Laboratory



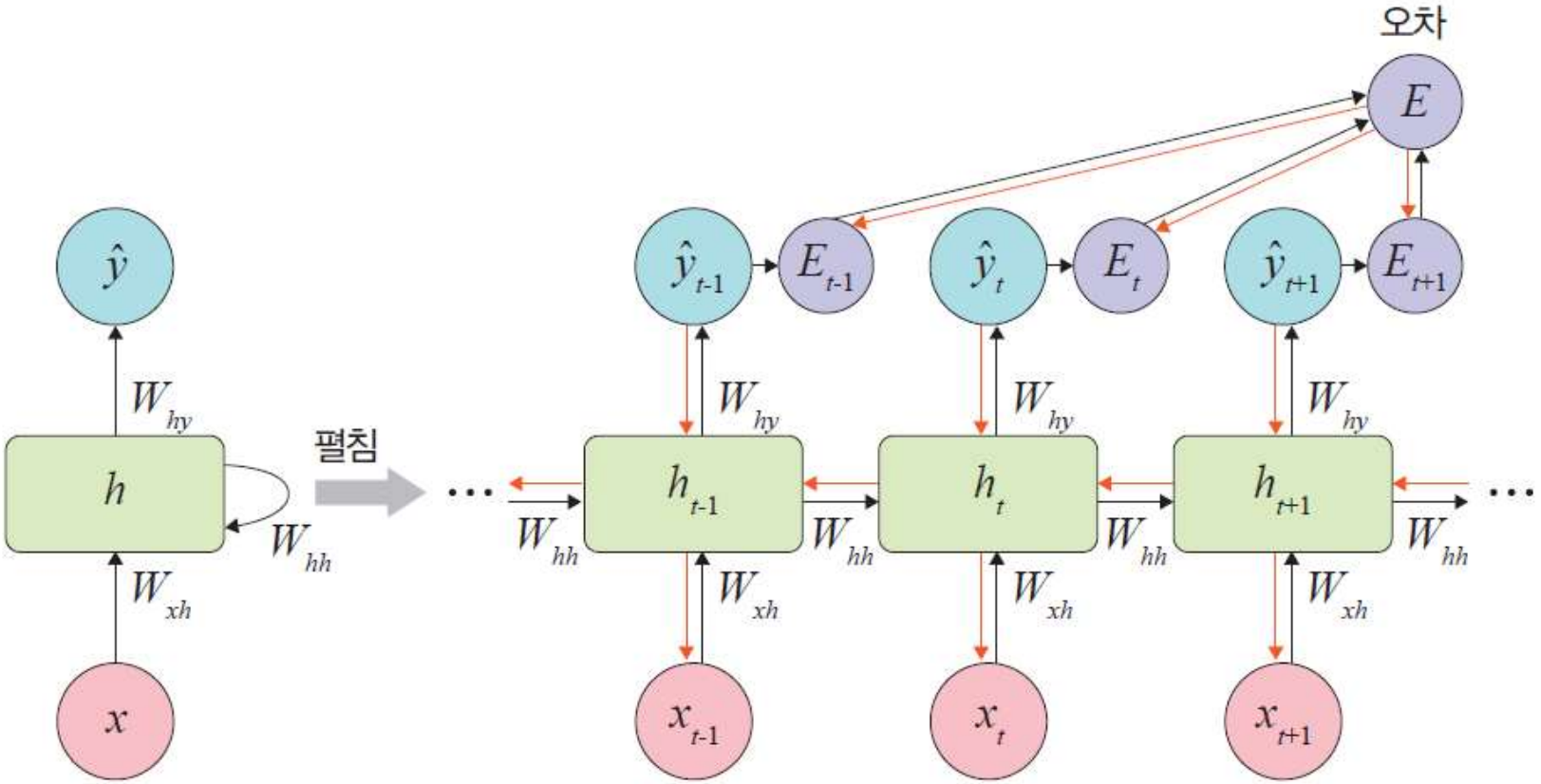
RNN

Robot Media Laboratory



RNN

Robot Media Laboratory



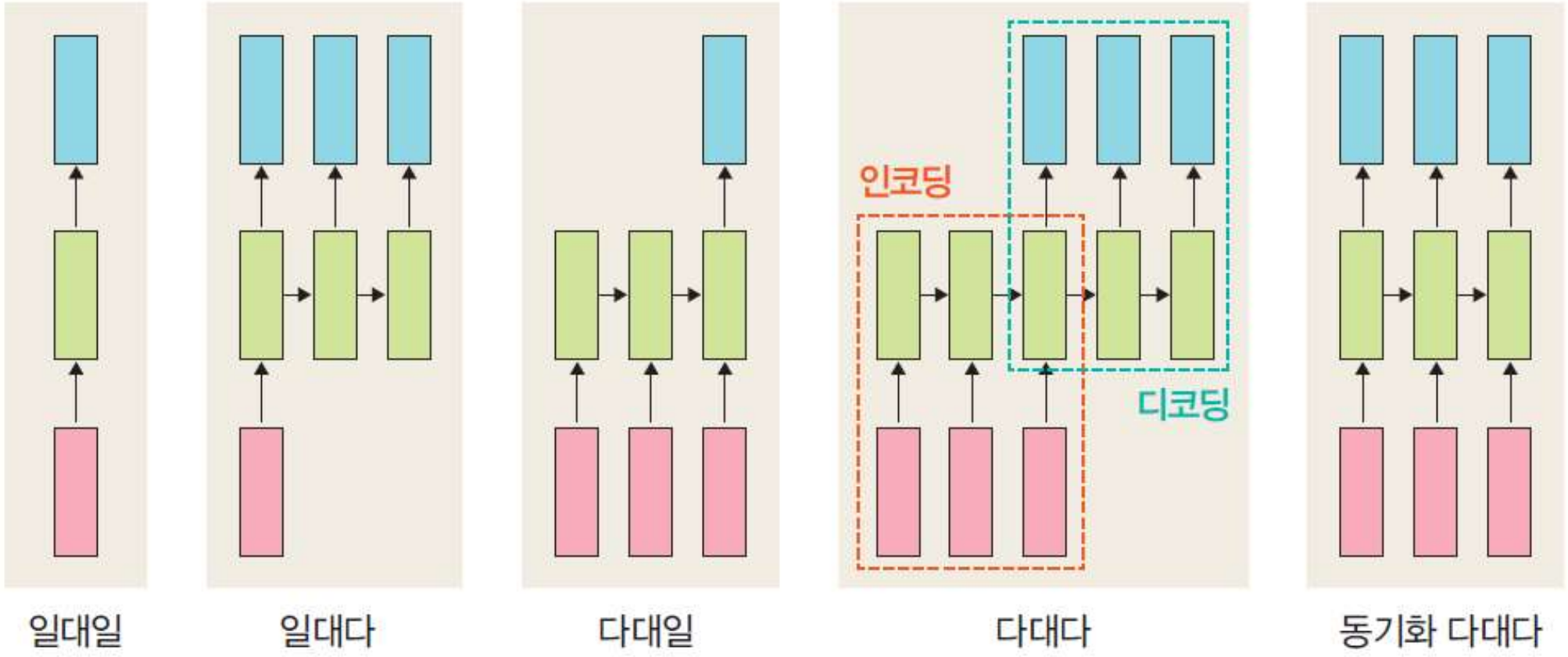
RNN

Robot Media Laboratory

- **일대일**
  - 순환이 없기 때문에 RNN이라고 말하기 어려우며, 순방향 네트워크가 대표적 사례
- **일대다**
  - 입력이 하나이고, 출력이 다수인 구조
  - 이미지를 입력해서 이미지에 대한 설명을 문장으로 출력하는 이미지 캡션(image captioning)
- **다대일**
  - 입력이 다수이고 출력이 하나인 구조로, 문장을 입력해서 긍정/부정을 출력하는 감성분석기
- **다대다**
  - 입력과 출력이 다수인 구조로, 언어를 번역하는 자동 번역기 등
- **동기화 다대다**
  - 문장에서 다음에 나올 단어를 예측하는 언어 모델, 즉 프레임 수준의 비디오 분류

RNN

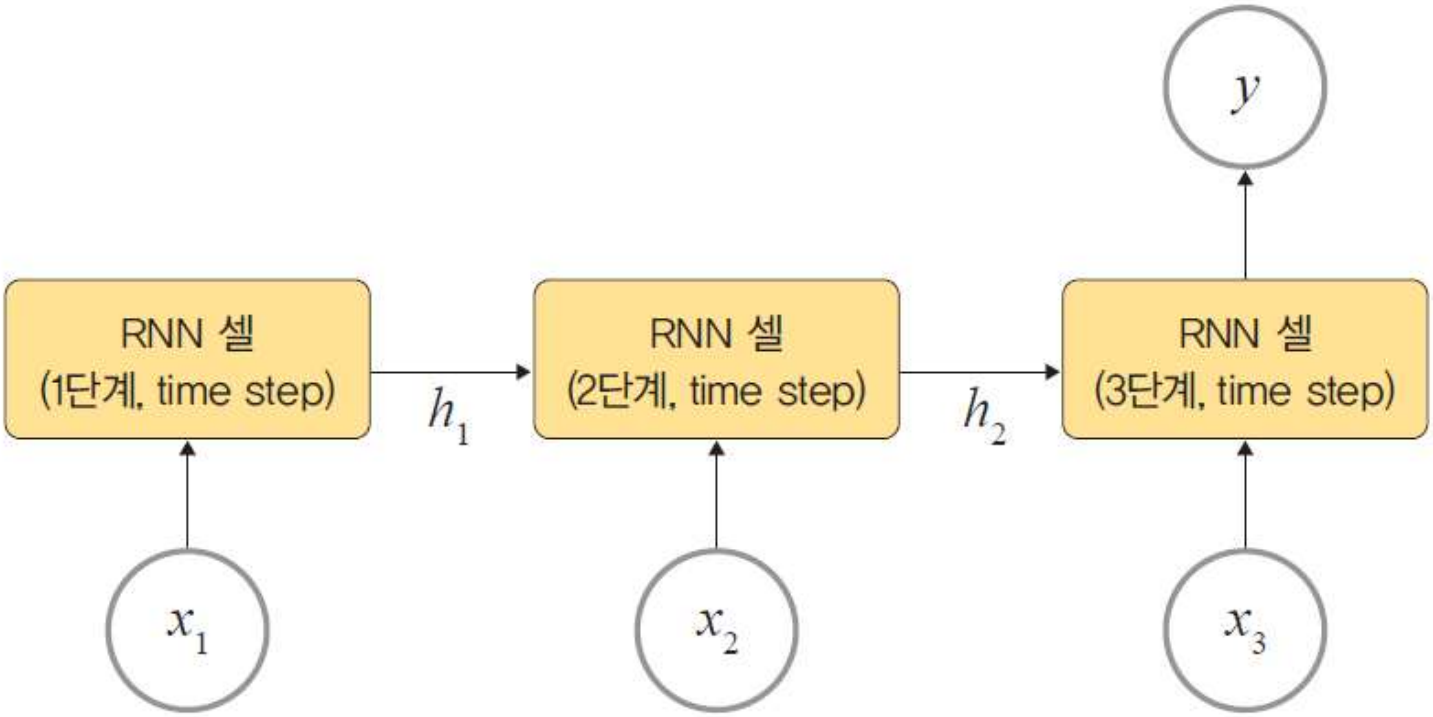
Robot Media Laboratory



RNN

Robot Media Laboratory

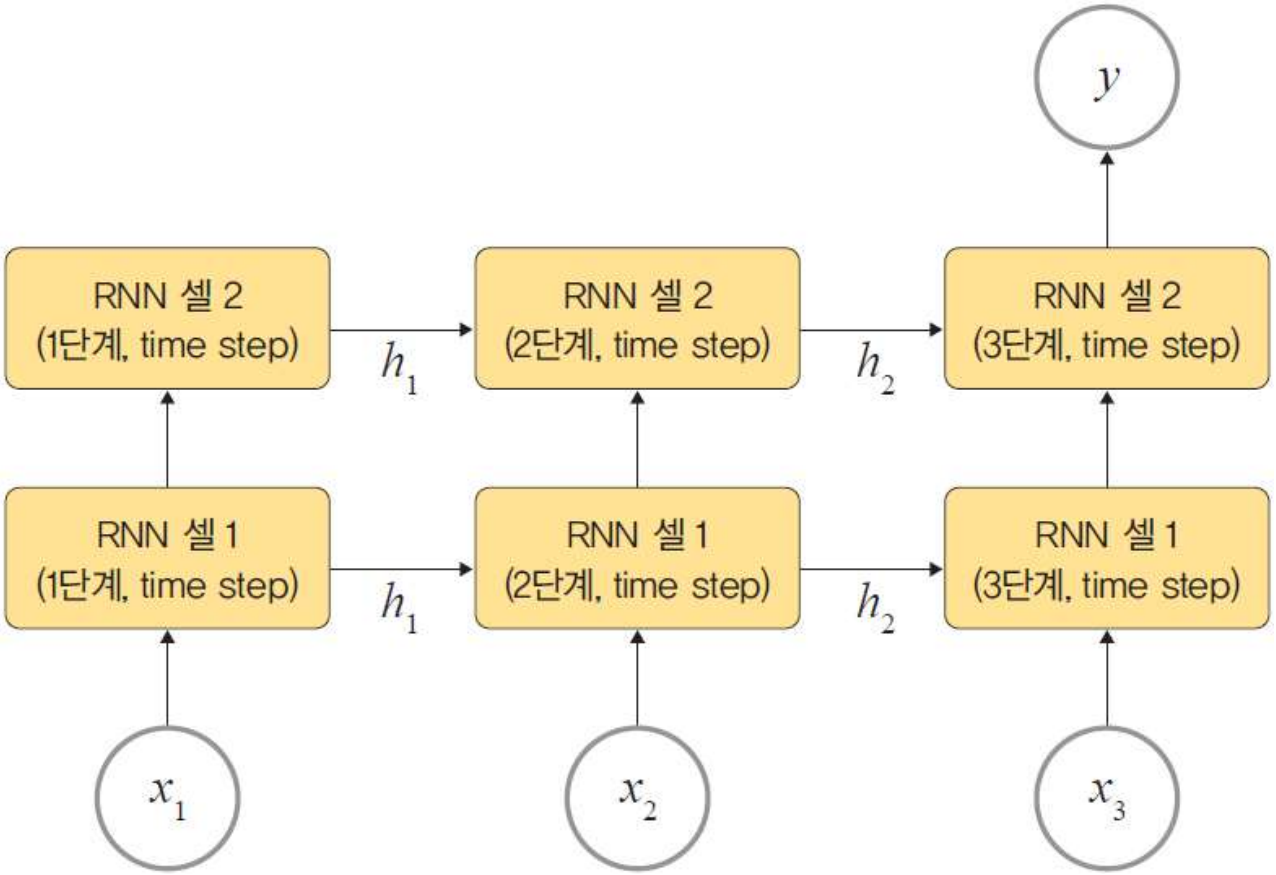
다대일



RNN

Robot Media Laboratory

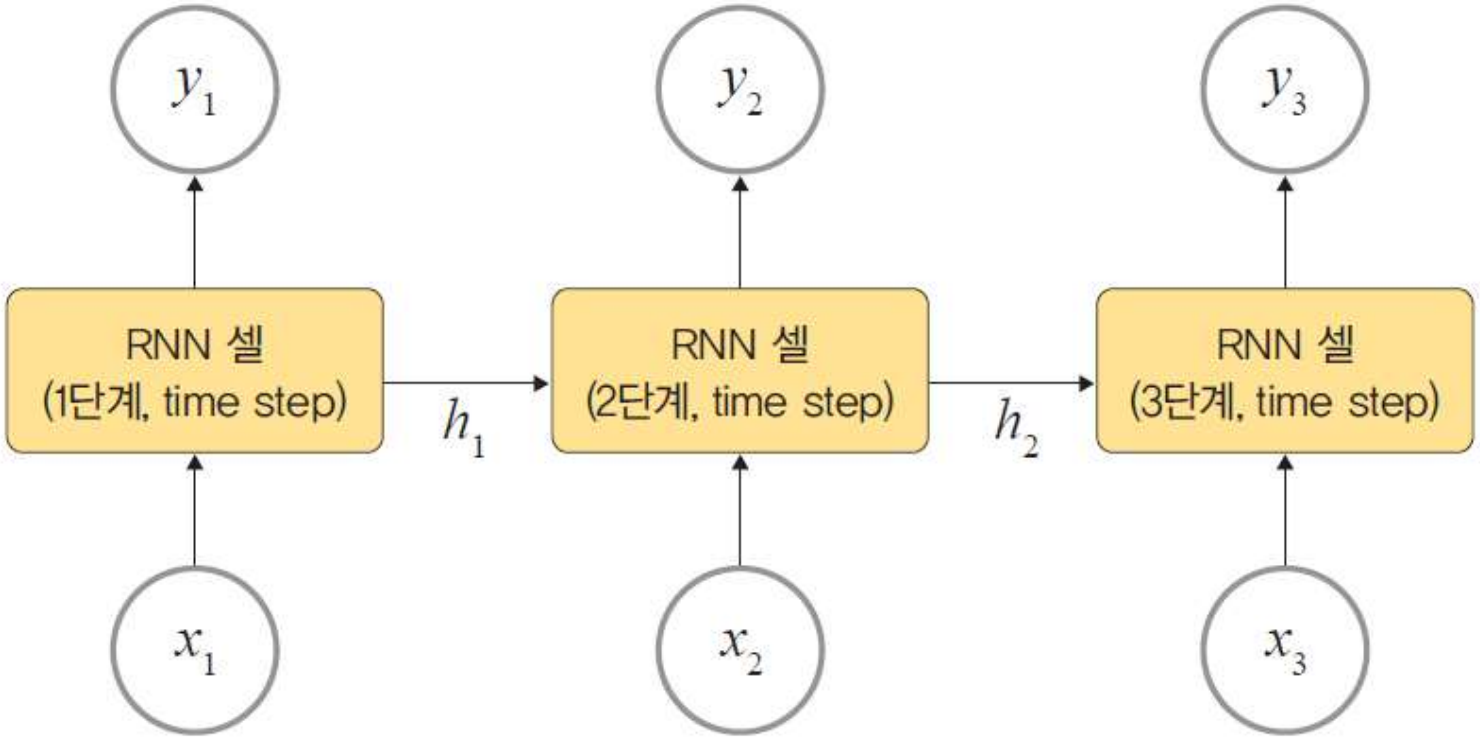
다층 다대일



RNN

Robot Media Laboratory

다대다



LSTM

- RNN은 결정적 단점이 있음

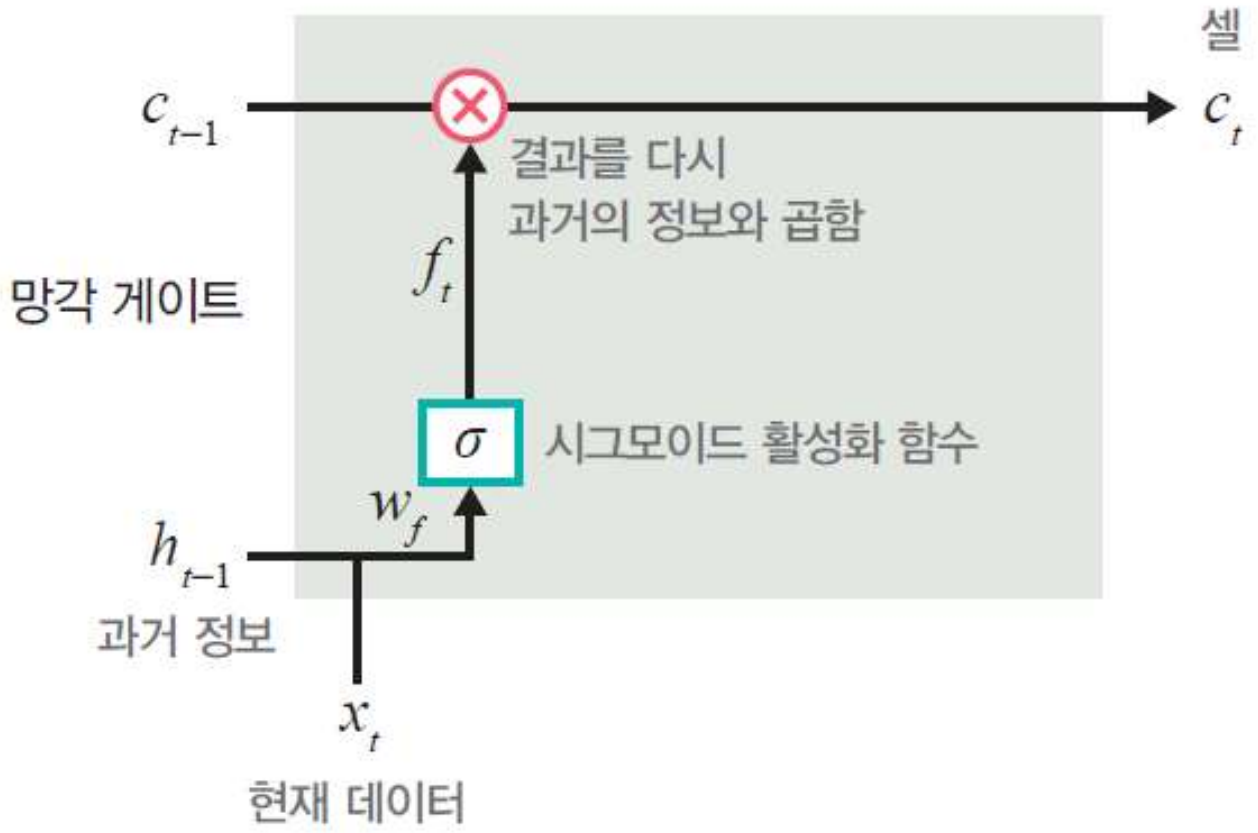
- 망각 게이트(forget gate)는 과거 정보를 어느 정도 기억할지 결정
- 과거 정보와 현재 데이터를 입력받아 시그모이드를 취한 후 그 값을 과거 정보에 곱해 줌
- 시그모이드의 출력이 0이면 과거 정보는 버리고, 1이면 과거 정보는 온전히 보존

$$f_t = \sigma(w_f[h_{t-1}, x_t])$$

$$c_t = f_t \cdot c_{t-1}$$

RNN

Robot Media Laboratory



입력 게이트(input gate)

- 현재 정보를 기억하기 위해 만들어졌음
- 과거 정보와 현재 데이터를 입력받아 시그모이드와 하이퍼볼릭 탄젠트 함수를 기반으로 현재 정보에 대한 보존량을 결정
- 즉, 현재 메모리에 새로운 정보를 반영할지 결정하는 역할을 함
- 계산한 값이 1이면 입력 xt가 들어올 수 있도록 허용(open)
- 계산한 값이 0이면 차단

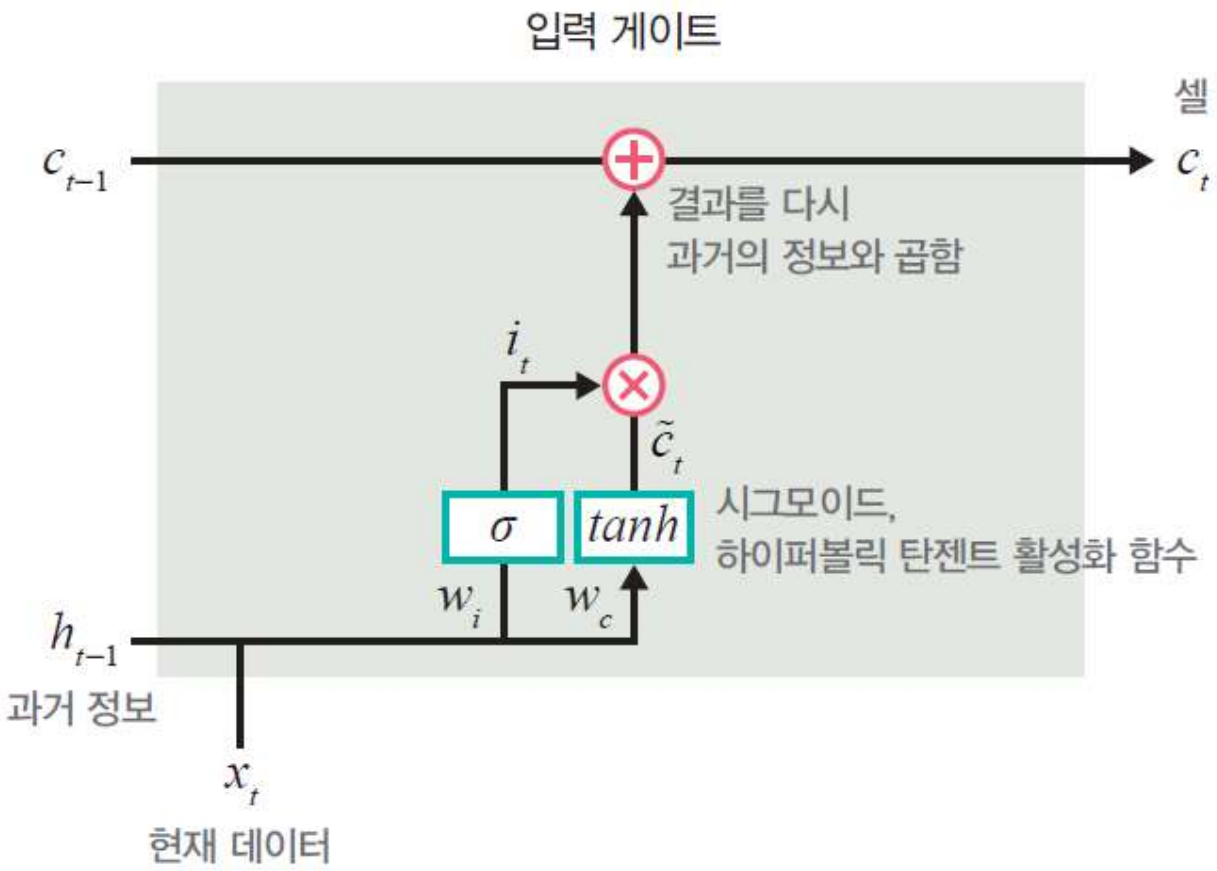
$$i_t = \sigma(w_i[h_{t-1}, x_t])$$

$$\tilde{c}_t = \tanh(w_c[h_{t-1}, x_t])$$

$$c_t = c_{t-1} + i_t \cdot \tilde{c}_t$$

RNN

Robot Media Laboratory



RNN

Robot Media Laboratory

셀

- 각 단계에 대한 은닉 노드(hidden node)를 메모리 셀이라고 함
- '총합(sum)'을 사용하여 셀 값을 반영하며, 이것으로 기울기 소멸 문제가 해결
- 망각 게이트와 입력 게이트의 이전 단계 셀 정보를 계산하여 현재 단계의 셀 상태(cell state)를 업데이트

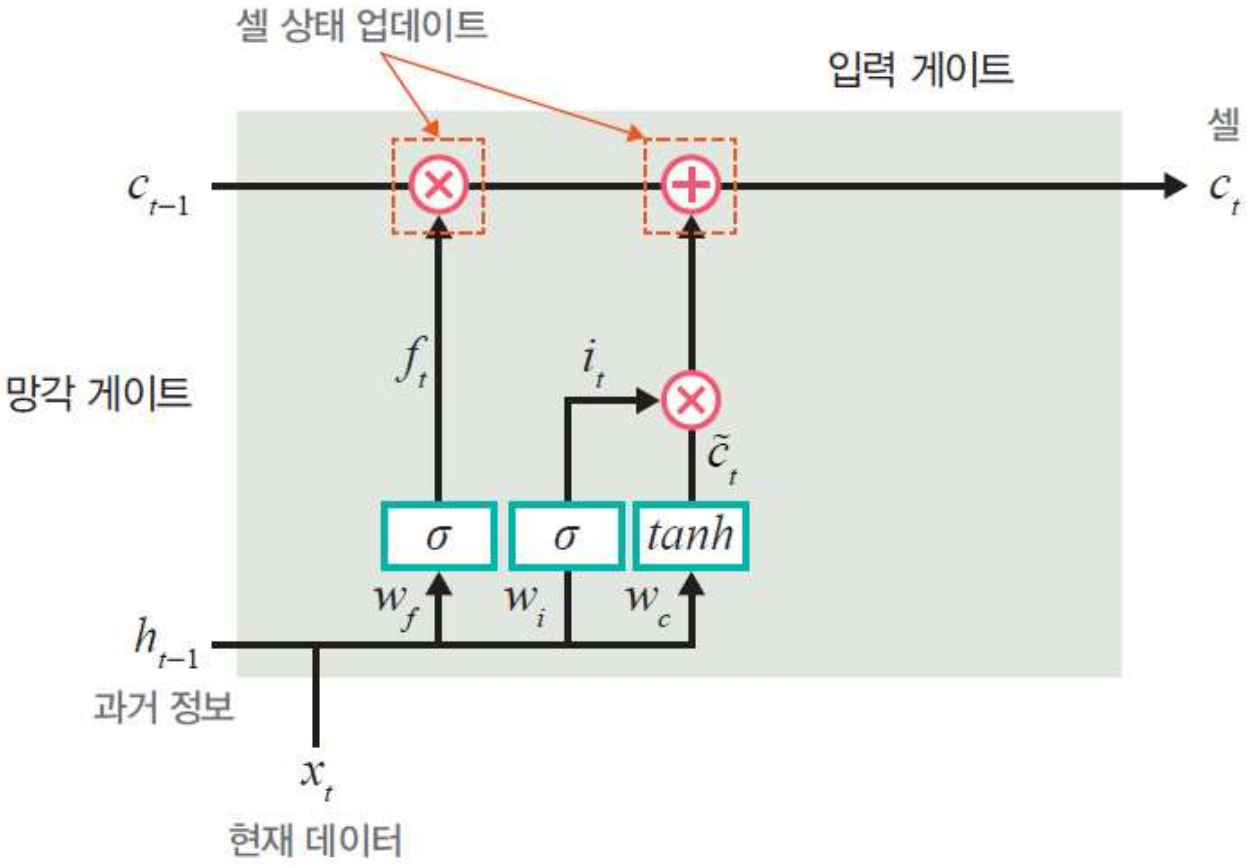
$$i_t = \sigma(w_i[h_{t-1}, x_t])$$

$$\tilde{c}_t = \tanh(w_c[h_{t-1}, x_t])$$

$$c_t = c_{t-1} + i_t \cdot \tilde{c}_t$$

RNN

Robot Media Laboratory



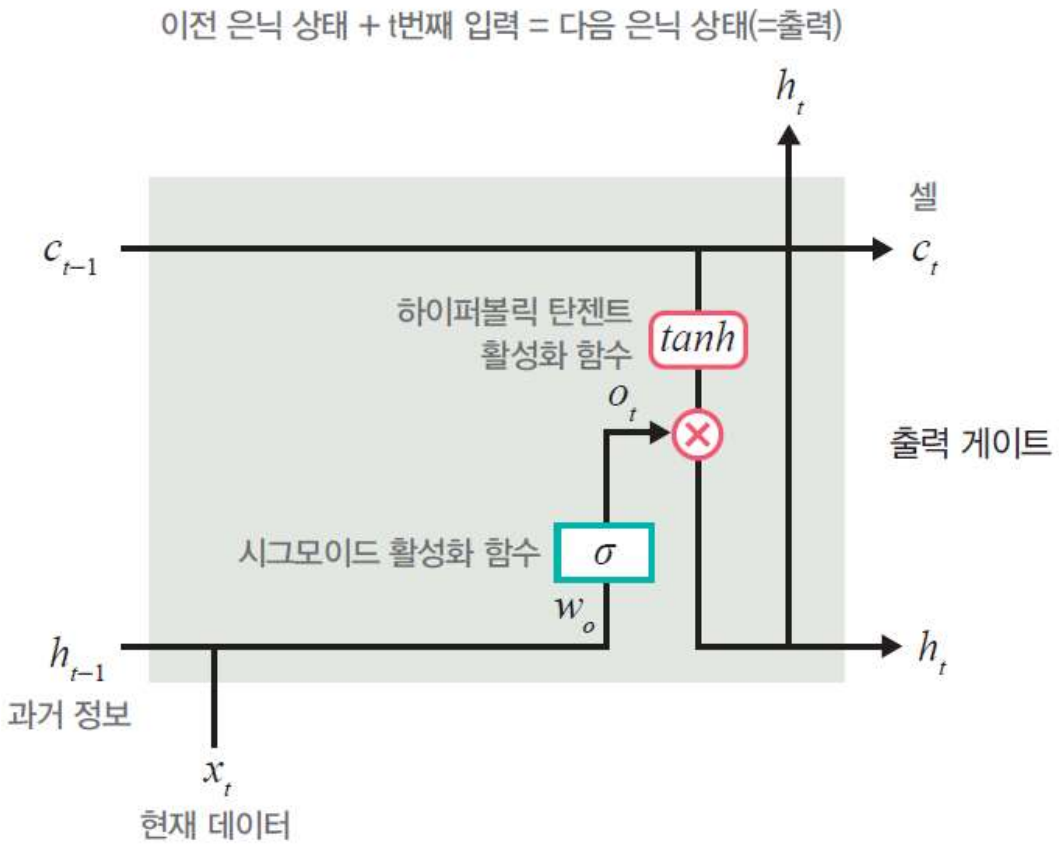
출력 게이트(output gate)

- 과거 정보와 현재 데이터를 사용하여 뉴런의 출력을 결정
- 이전 은닉 상태(hidden state)와 t번째 입력을 고려해서 다음 은닉 상태를 계산
- LSTM에서는 이 은닉 상태가 그 시점에서의 출력이 됨
- 출력 게이트는 갱신된 메모리의 출력 값을 제어하는 역할을 함
- 계산한 값이 1이면 의미 있는 결과로 최종 출력
- 계산한 값이 0이면 해당 연산 출력을 하지 않음

$$o_t = \sigma(w_o [h_{t-1}, x_t])$$
$$h_t = o_t \cdot \tanh(c_{t-1})$$

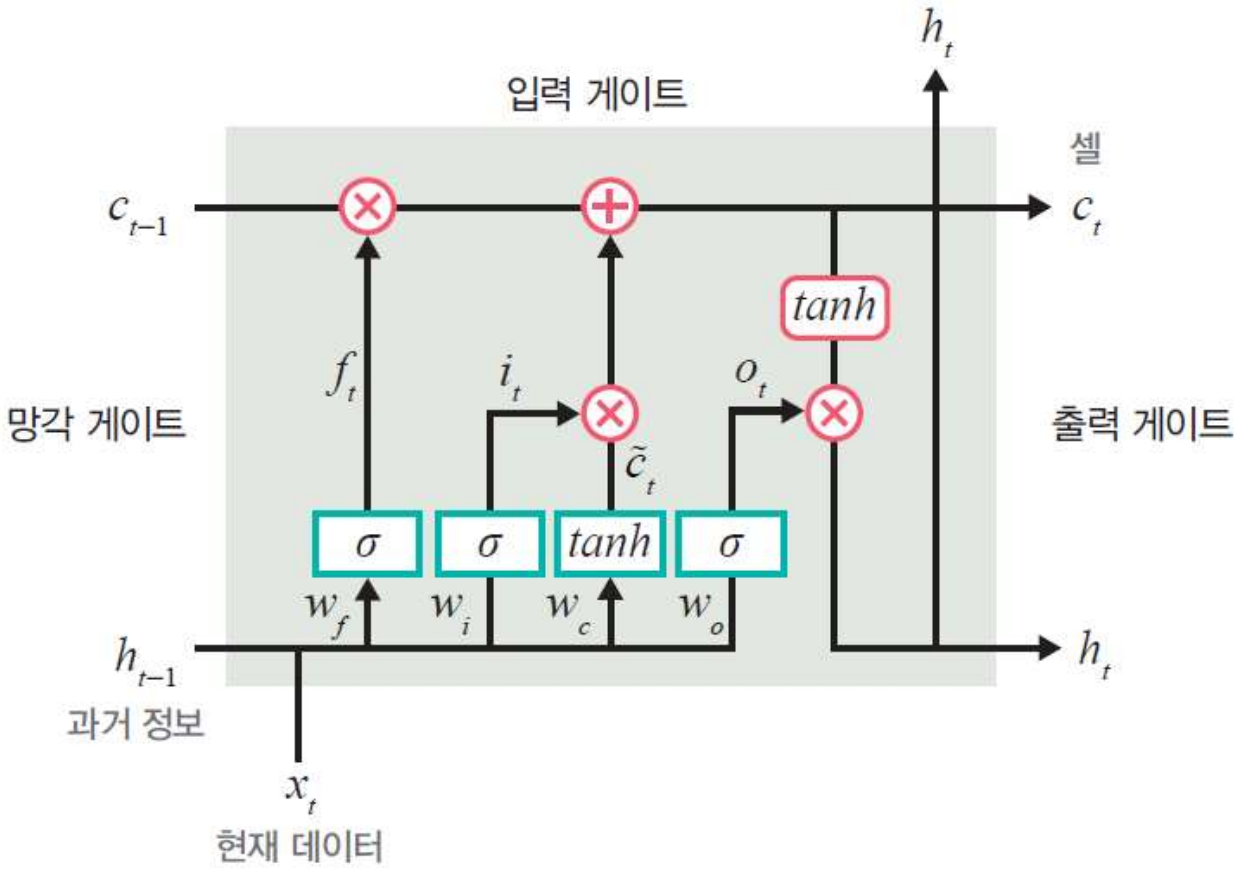
RNN

Robot Media Laboratory



RNN

Robot Media Laboratory



GRU(Gated Recurrent Unit)

- 게이트 메커니즘이 적용된 RNN 프레임워크의 한 종류이면서 LSTM보다 구조가 간단함
- GRU는 LSTM에서 사용하는 망각 게이트와 입력 게이트를 하나로 합친 것이며, 별도의 업데이트 게이트로 구성
- 하나의 게이트 컨트롤러(gate controller)가 망각 게이트와 입력 게이트를 모두 제어함
- 게이트 컨트롤러가 1을 출력하면 망각 게이트는 열리고 입력 게이트는 닫히며, 반대로 0을 출력하면 망각 게이트는 닫히고 입력 게이트는 열림
- 즉, 이전 기억이 저장될 때마다 단계별 입력은 삭제
- GRU는 출력 게이트가 없어 전체 상태 벡터가 매 단계마다 출력되며, 이전 상태의 어느 부분이 출력될지 제어하는 새로운 게이트 컨트롤러가 별도로 존재

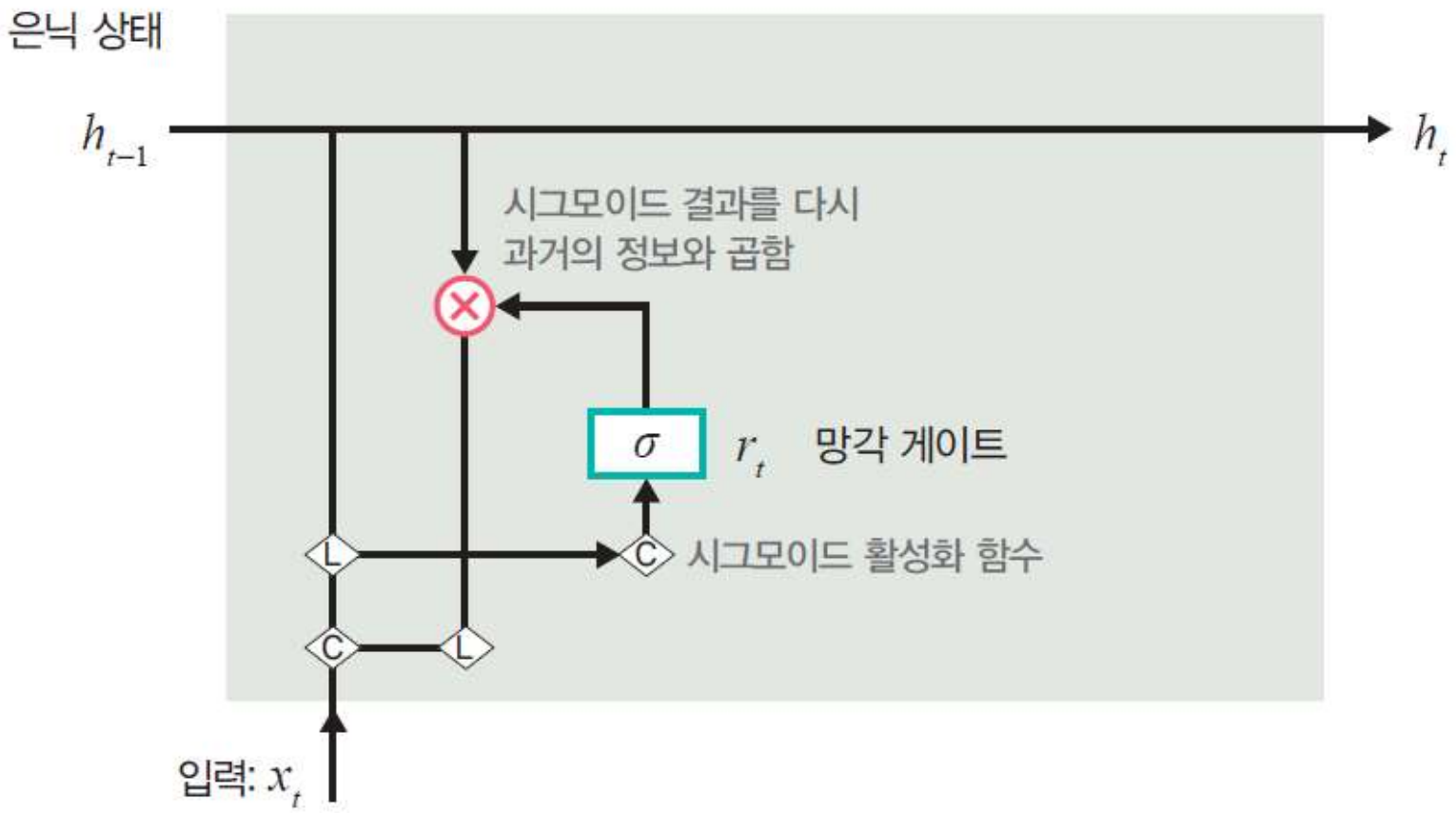
망각 게이트(reset gate)

- 과거 정보를 적당히 초기화(reset)시키려는 목적으로 시그모이드 함수를 출력으로 이용하여 (0,1) 값을 이전 은닉층에 곱함

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

RNN

Robot Media Laboratory

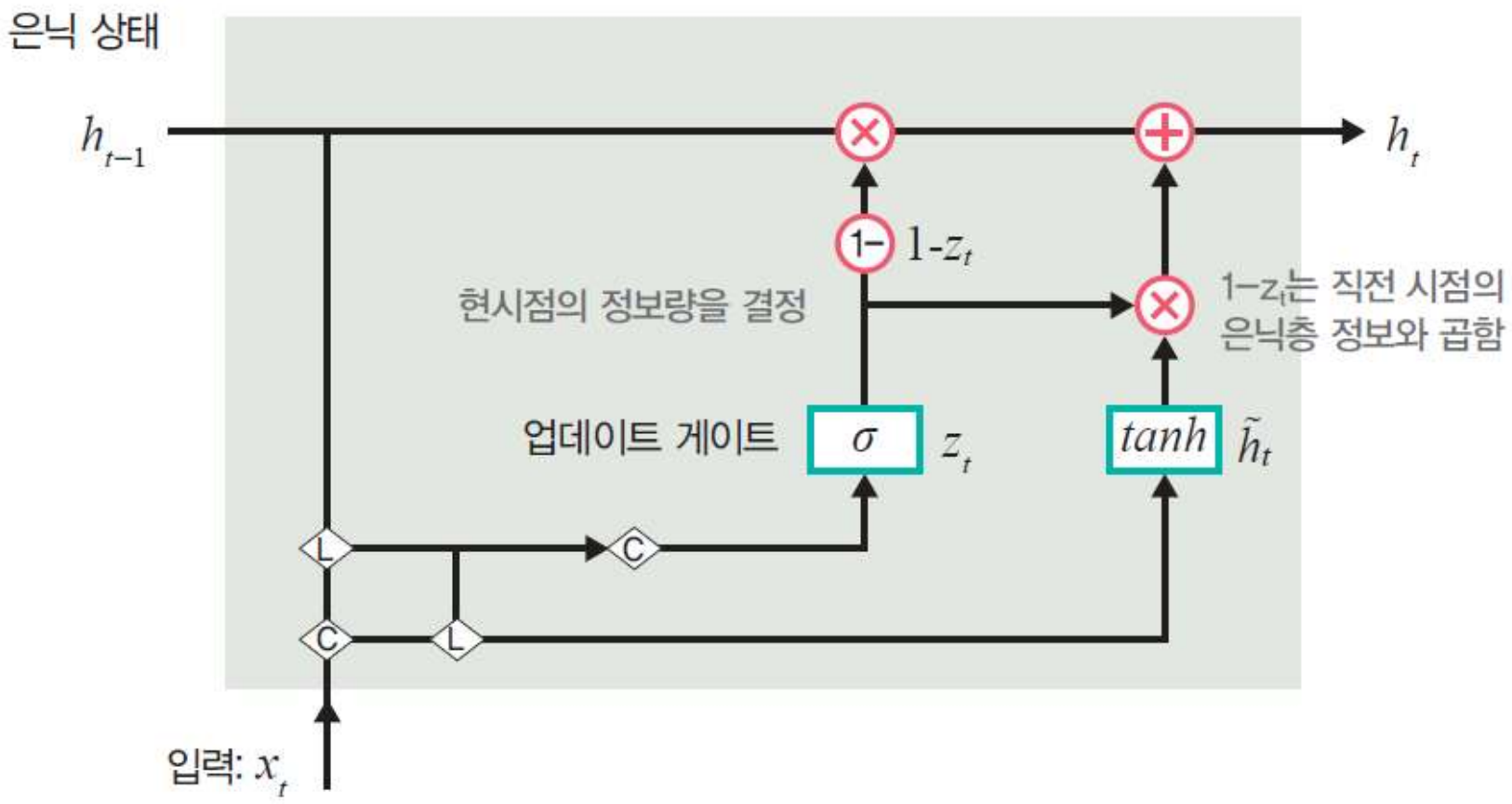


- **업데이트 게이트(update gate)**
  - 과거와 현재 정보의 최신화 비율을 결정하는 역할을 함
  - 시그모이드로 출력된 결과(zt)는 현시점의 정보량을 결정하고 1에서 뺀 값(1- zt)은 직전 시점의 은닉층 정보와 곱함

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

RNN

Robot Media Laboratory



후보군(candidate)

- 현시점의 정보에 대한 후보군을 계산
- 과거 은닉층의 정보를 그대로 이용하지 않고 망각 게이트의 결과를 이용하여 후보군을 계산

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

(*는 점 단위 연산(pointwise operation)입니다. 예를 들어 벡터를 더할 때 각각의 차원(dimension)에 맞게 곱하거나 더하는 것이 가능해집니다.)

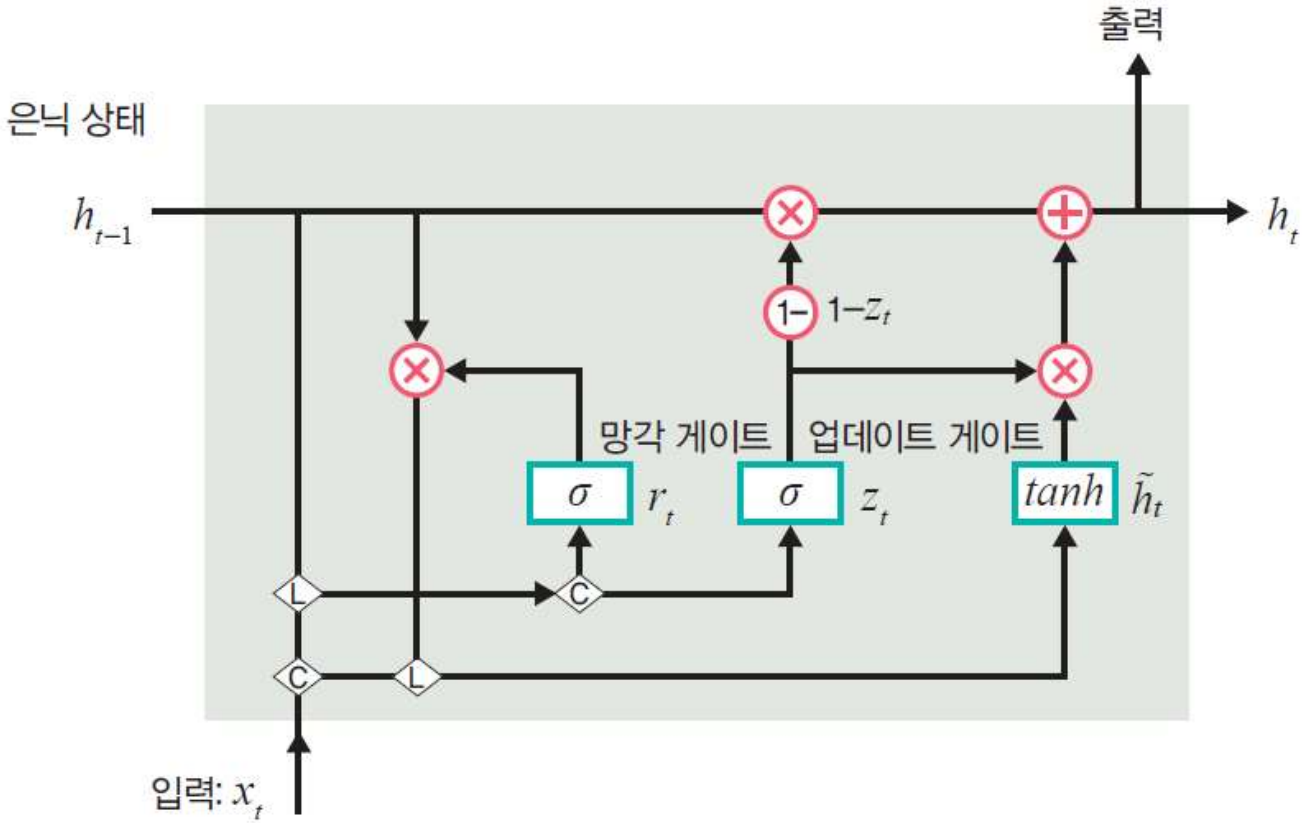
은닉층 계산

- 마지막으로 업데이트 게이트 결과와 후보군 결과를 결합하여 현시점의 은닉층을 계산
- 시그모이드 함수의 결과는 현시점에서 결과에 대한 정보량을 결정하고, 1-시그모이드 함수의 결과는 과거의 정보량을 결정

$$h_t = (1 - z_t) * h_{t-1} + z_t \times \tilde{h}_t$$

RNN

Robot Media Laboratory



양방향 RNN

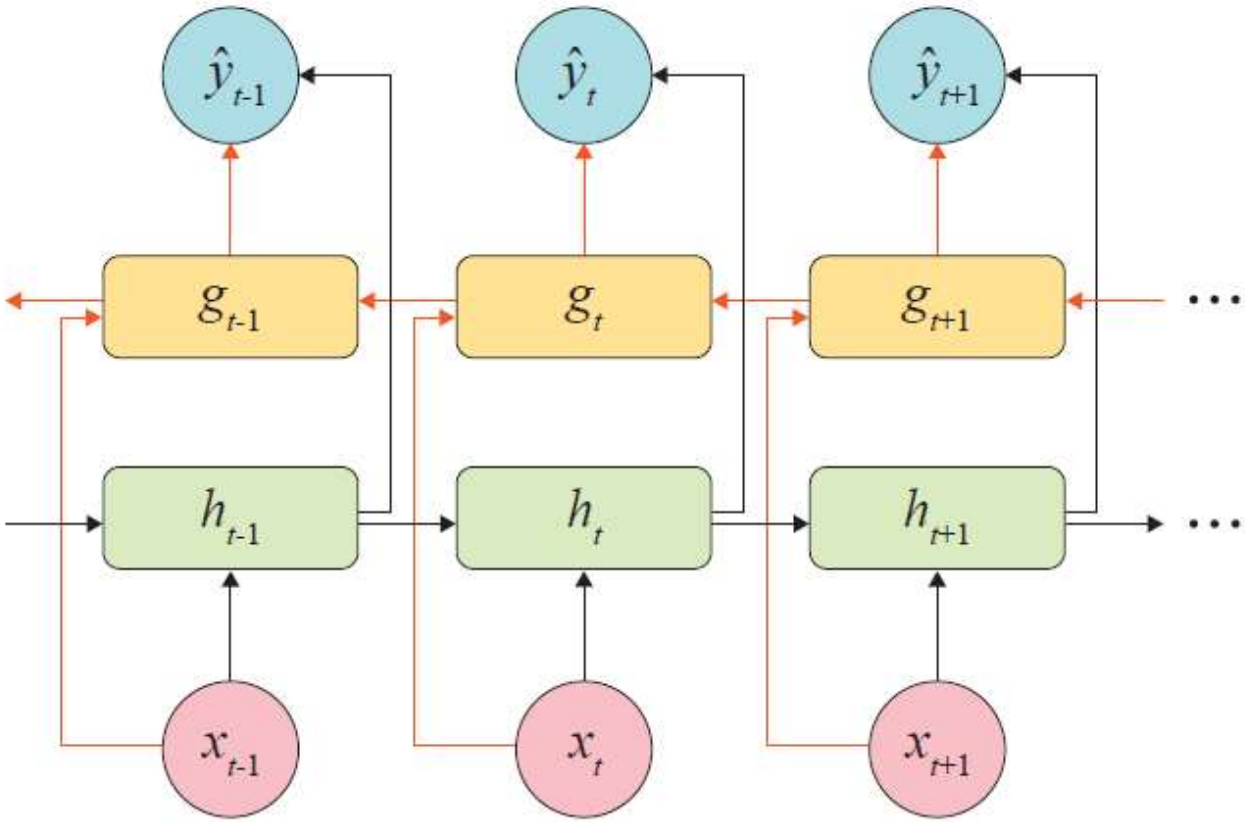
- RNN은 이전 시점의 데이터들을 참고해서 정답을 예측하지만 실제 문제에서는 과거 시점이 아닌 미래 시점의 데이터에 힌트가 있는 경우도 많음
- 이전 시점의 데이터뿐만 아니라, 이후 시점의 데이터도 함께 활용하여 출력 값을 예측하고자 하는 것이 양방향 RNN(bidirectional RNN)

양방향 RNN 구조

- 양방향 RNN은 하나의 출력 값을 예측하는 데 메모리 셀 두 개를 사용
- 첫 번째 메모리 셀은 이전 시점의 은닉 상태(forward states)를 전달받아 현재의 은닉 상태를 계산
- 다음 그림에서는 초록색 메모리 셀에 해당
- 두 번째 메모리 셀은 다음 시점의 은닉 상태(backward states)를 전달받아 현재의 은닉 상태를 계산
- 다음 그림의 주황색 메모리 셀에 해당
- 이 값 두 개를 모두 출력층에서 출력 값을 예측하는 데 사용

RNN

Robot Media Laboratory

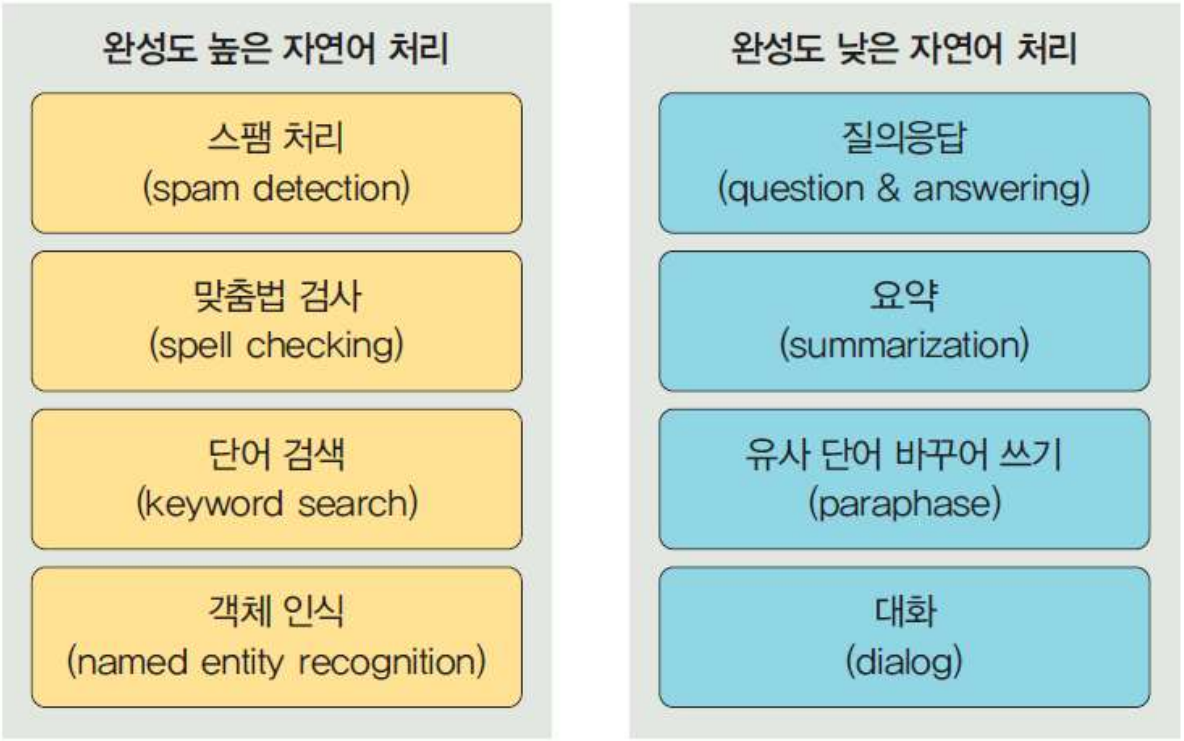


자연어 처리

자연어 처리

Robot Media Laboratory

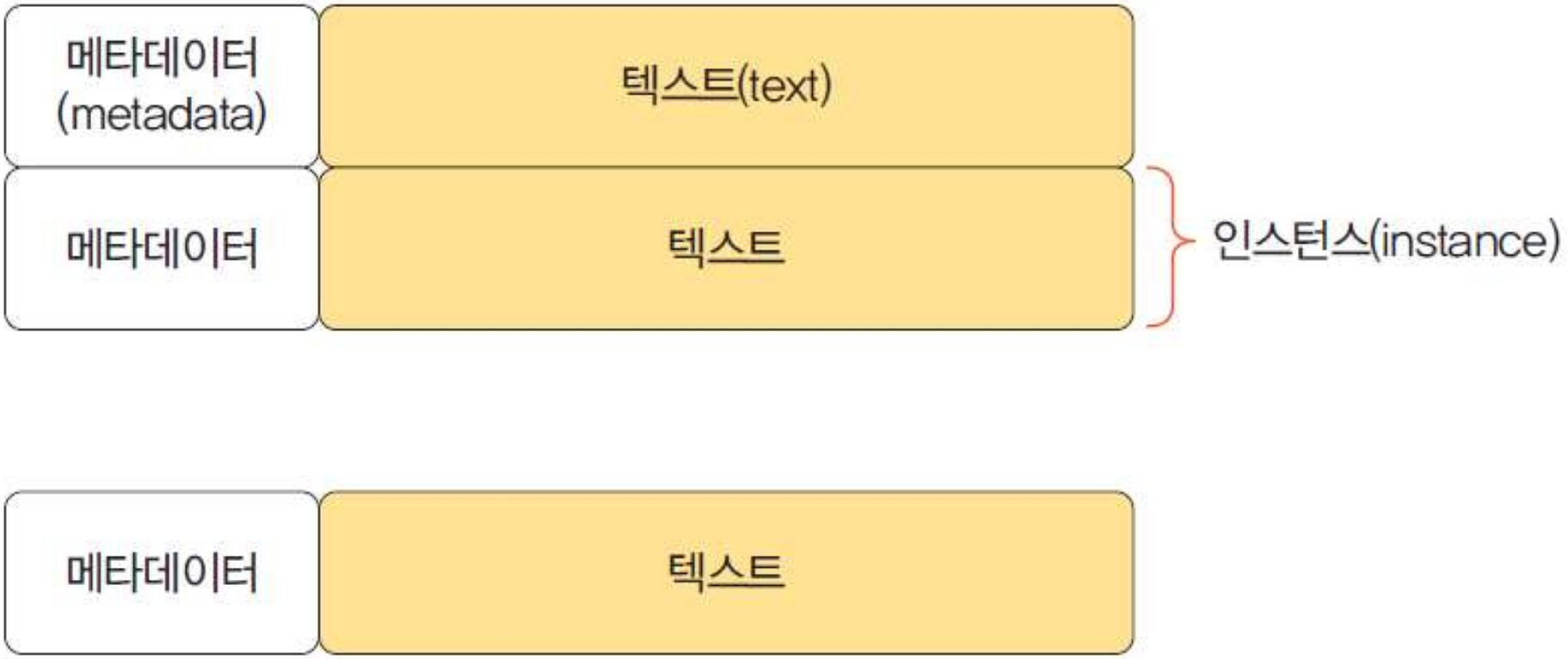
- 우리가 일상생활에서 사용하는 언어의 의미를 분석하여 컴퓨터가 처리할 수 있도록 하는 과정
- 언어 종류가 다르고 그 형태가 다양하기 때문에 처리가 매우 어려움



자연어 처리

Robot Media Laboratory

- 말뭉치(corpus(코퍼스)): 자연어 처리에서 모델을 학습시키기 위한 데이터이며, 자연어



자연어 처리

Robot Media Laboratory

- **토큰(token)**
  - 자연어 처리를 위한 문서는 작은 단위로 나누어야 하는데, 이때 문서를 나누는 단위가 토큰
  - 문자열을 토큰으로 나누는 작업을 토큰 생성(tokenizing)이라고 하며, 문자열을 토큰으로
  - 분리하는 함수를 토큰 생성 함수라고 함

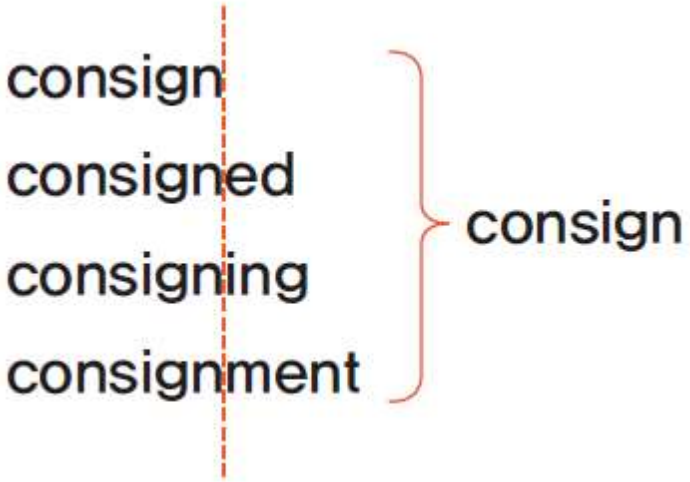
- **토큰화(tokenization)**
  - 텍스트를 문장이나 단어로 분리하는 것을 의미
  - 토큰화 단계를 마치면 텍스트가 단어 단위로 분리

자연어 처리

Robot Media Laboratory

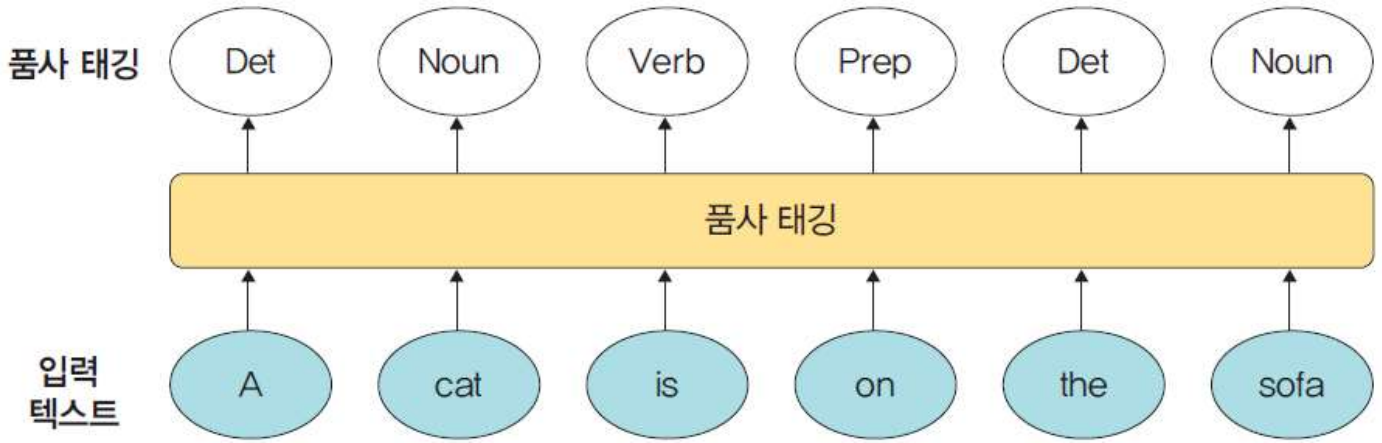
- **불용어(stop words)**
  - 문장 내에서 많이 등장하는 단어
  - 분석과 관계없으며, 자주 등장하는 빈도 때문에 성능에 영향을 미치므로 사전에 제거해 주어야 함

- **어간 추출(stemming)**
  - 단어를 기본 형태로 만드는 작업



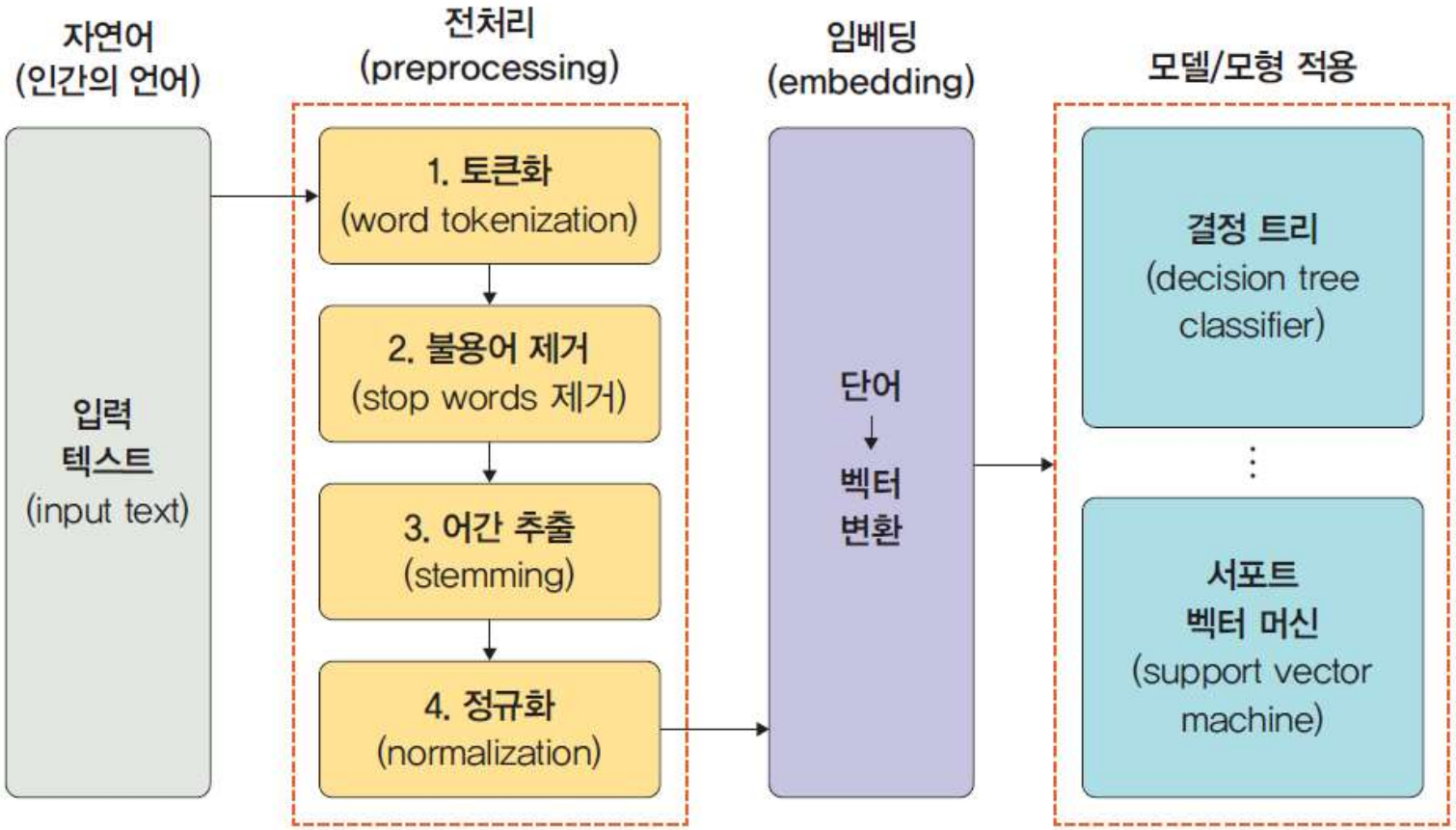
품사 태깅(part-of-speech tagging)

- 주어진 문장에서 품사를 식별하기 위해 붙여 주는 태그(식별 정보)를 의미



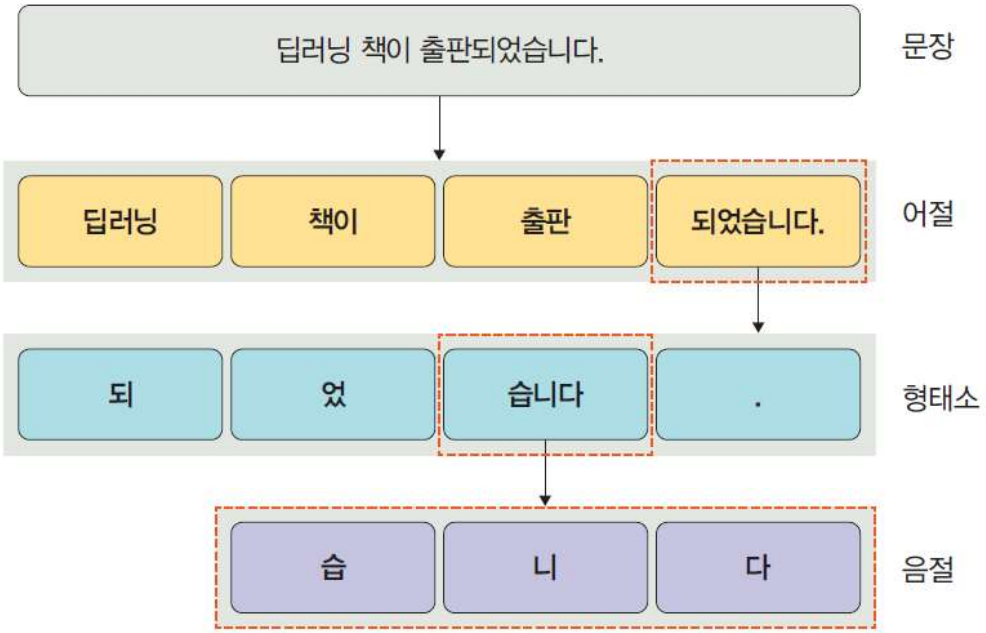
자연어 처리

Robot Media Laboratory



형태소

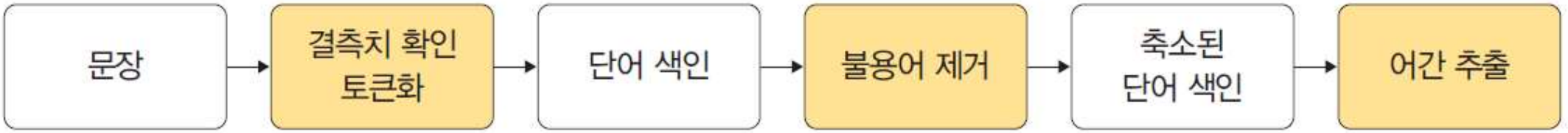
- 형태소는 언어를 쪼갤 때 의미를 가지는 최소 단위
- 다음 그림은 형태소 분석을 위한 단계를 도식화한 것



자연어 처리

Robot Media Laboratory

전처리



자연어 처리

Robot Media Laboratory

정규화(normalization)

- 표현 방법이 다른 단어들을 통합시켜서 같은 단어로 만들어 주는 것
- 머신 러닝/딥러닝은 데이터 특성들을 비교하여 패턴을 분석
- 값이 더 크면 큰영향을 가짐
- 중요한 것은 값이 크다고 해서 분석에 더 중요한 요소라고 간주 할 수 없기 때문에 정규화가 필요한 것