

요구사항명세(SRS)

Team 11 - 열려라, 쿠(:KU)굴

201411303 이준호

201511243 김동연

201511262 박우진

201714160 권혜주

1. Introduction	3
1. 1. Purpose	3
1. 2. Scope	3
1. 3. Definition, acronyms, and abbreviations	4
1. 4. References	5
2. Overall Description	6
2. 1. Product Perspective	6
2. 2. Product Functions	6
2.2.1 Use Case	7
2. 3. Constraints	10
2. 4. Assumptions and Dependencies	10
3. Specific Requirements	11
3. 1. External interface requirements	11
3. 1. 1. User Interfaces	11
Figure-3. User Interface 예시	11
3. 1. 2. Hardware Interfaces	11
3. 1. 3. Software Interfaces	11
3. 1. 4. Communications interfaces	11
3. 2. Functional Requirements	12
3. 2. 1. Live Model System	12
3. 2. 1. 1. Voice Pre-processing	12
3. 2. 1. 2. STT Execution	12
3. 2. 1. 3. SR Execution	12
3. 2. 1. 4. Function Execution	13
3. 2. 2. Train Model System	13
3. 2. 2. 1. Dataset Pre-processing	13
3. 2. 2. 2. STT Training	13
3. 2. 2. 3. SR Training	13
3. 3. Non-Functional Requirements	14
4. Appendixes	15
4. 1. MFCC (Mel-Frequency Cepstral Coefficients)	15
4. 2. STFT (Short-Time Fourier Transform)	16
4. 3. STT 음성 인식 과정	17
4. 4. SR 화자 인식 과정	18
4. 5. NN (Neural Network)	19
4.5.1 ANN (Artificial Neural Network)	19
4.5.3 CNN (Convolutional Neural Network)	20
4.5.4 RNN (Recurrent Neural Network)	21
4.5.4.1 LSTM (Long Short-Term Memory)	21
4.5.4.2 GRU (Gate Recurrent Unit)	22

1. Introduction

1. 1. Purpose

‘알리바바와 40인의 도둑’에서 알리바바의 ‘열려라 참깨’ 라는 명령어에 대한 처리 결과로 동굴이 열리는 기능이 수행되었습니다. 동굴이 도둑 우두머리의 화자인지 확인을 하지 않고 명령어만 처리한 결과 도둑들은 보물을 알리바바에게 도둑질당했습니다.

이 동굴 시스템의 문제는 화자에 대해 인식을 하지 못한 것입니다. 이러한 문제점을 보완하고자 화자 인식에 대한 기능을 추가하려고 합니다.

이 시스템에서는 화자를 인식하고 등록된 화자에 대한 음성을 텍스트로 변환하여 등록된 기능을 수행합니다.



1. 2. Scope

기존의 음성인식 시스템 (예 - Siri, Bixby, Nugu, Giga Genie)은 IoT 기기를 활용하여 조명을 제어하거나 집안의 온도를 제어하는 등으로 활용되고 있습니다. 이처럼 음성인식 기술을 활용한다면 사용자에게 편리한 일상생활을 제공할 수 있습니다.

이러한 Speech To Text (STT) 기능에 Speaker Recognition(SR) 기능을 추가하여 보안성 있는 시스템을 구축하려고 합니다. 이 시스템을 활용하여 Home automation뿐만 아니라 보안 인가가 필요한 곳에서 보안성이 강화된 시스템을 설계할 수 있습니다.

1. 3. Definition, acronyms, and abbreviations

Term	Definition
STT (Speech To Text)	음성 파일을 통하여 어떤 문장을 말하였는지 음성을 문자로 나타내는 것
SR (Speaker Recognition)	음성 파일을 통하여 어떤 화자가 말하였는지 식별하는 것
ANN (Artificial Neural Network)	사람의 신경망 원리와 구조를 모방하여 만든 기계학습 알고리즘
DNN (Deep Neural Network)	은닉층을 2개 이상 지닌 기계학습 알고리즘
RNN (Recurrent Neural Network)	반복적이고 순차적인 데이터 학습에 특화된 인공신경망 내부의 순환구조를 가진 기계학습 알고리즘
LSTM (Long Short-Term Memory)	RNN의 장기 의존성 문제를 해결하기 위해 개선된 모델
GRU (Gate Recurrent Unit)	LSTM을 변형하여 만든 모델
CNN (Convolution Neural Network)	데이터 특징을 추출하여 특징들의 패턴을 파악하는 구조
MFCC (Mel-frequency cepstrum)	오디오 신호에서 추출할 수 있는 feature로, 소리의 고유한 특징을 나타내는 수치
STFT (Short-Time Fourier Transform)	시간에 따라 변화하는 긴 신호를 짧은 시간 단위로 분할한 다음에 푸리에 변환을 적용하는 것
FT (Fourier Transform)	시간에 대한 함수를 구성하고 있는 주파수 성분으로 분해하는 작업
FFT (Fast Fourier Transform)	신호를 주파수 성분으로 변환하는 알고리즘

1. 4. References

- IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.
- S. Lee, S. Han, S. Park, K. Lee, and J. Lee, “딥러닝 모델을 사용한 한국어 음성인식,” *응용통계연구*, vol. 32, no. 2, pp. 213-227, Apr. 2019.
- 서영주, 김회린. (2014). 최근 화자인식 기술 동향. *전자공학회지*, 41(3), 40-49.
- 강병욱, 전형배, 송화전, 한란, 박전규. (2018). 녹취 데이터 인식을 위한 LSTM RNN 기반 음성인식 시스템. *한국통신학회 학술대회논문집*, (), 1285-1286.
- Awni Hannun* , Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, Andrew Y. Ng. (2014). “Deep Speech: Scaling up end-to-end speech recognition”
- <https://blog.ncsoft.com/게임과-ai-7-화자-인식-기술/>
- <https://ebbnflow.tistory.com/119>

2. Overall Description

2. 1. Product Perspective

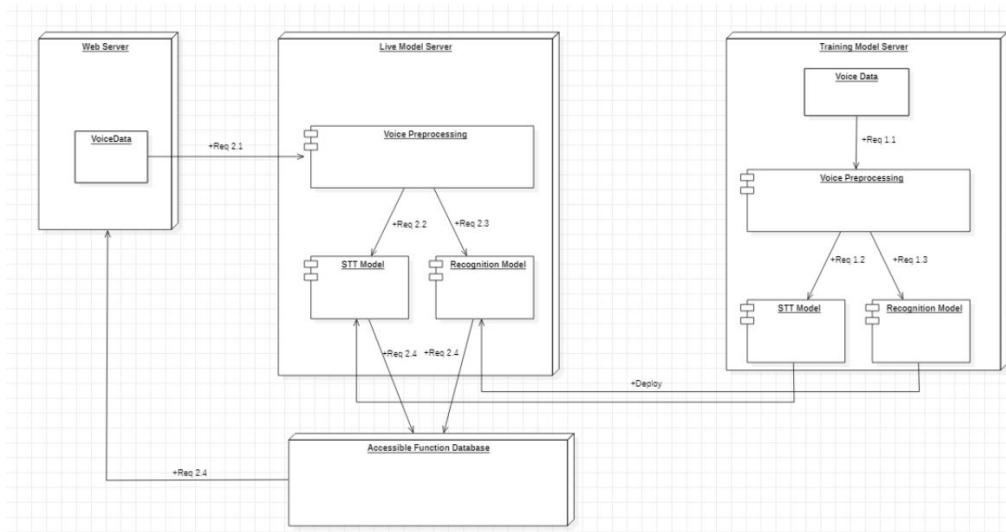


Figure-1. 아키텍처 다이어그램

첫 번째로 화자가 “열려라 참깨”라고 말을 합니다. 이 음성은 파일로써 전달이 되어 전처리 과정을 가진 후, STT Model, SR Model을 적용해 나온 결과를 바탕으로 인가된 화자의 명령 기능을 수행합니다.

2. 2. Product Functions

ID	Function	Description
1	Voice Pre-processing	마이크를 통해 입력받은 음성에 전처리를 수행
2	STT Execution	학습된 모델에 전처리된 데이터를 적용하여 STT (Speech To Text)를 수행
3	SR Execution	학습된 모델에 전처리된 데이터를 적용하여 SR(Speaker Recognition)을 수행
4	Function Execution	인식된 결과를 통해 등록된 기능을 수행
5	Dataset Pre-processing	저장된 학습용 음성 데이터에 전처리를 수행
6	STT Training	전 처리된 데이터를 이용해 STT 모델 학습을 수행
7	SR Training	전 처리된 데이터를 이용해 SR 모델 학습을 수행

2.2.1 Use Case

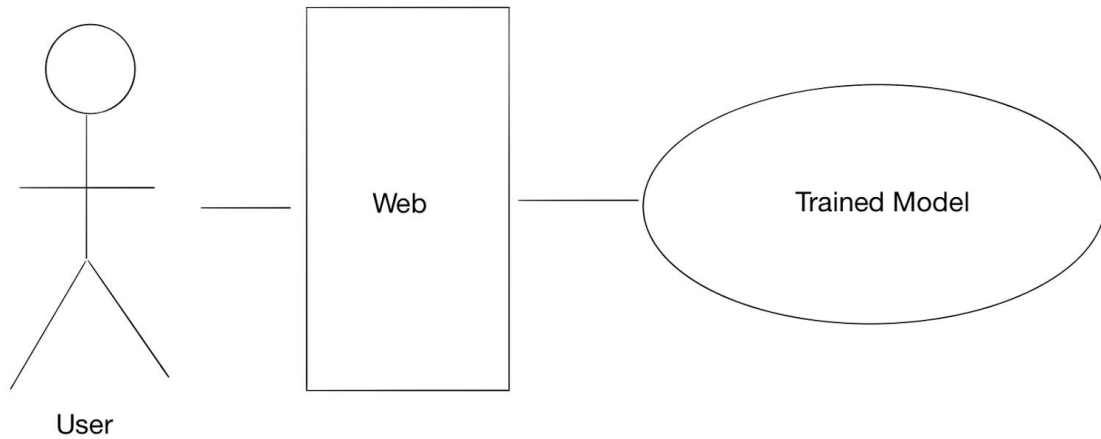


Figure-2. Live Model Server

Use Case ID	1
Use Case	Voice Pre-processing
Description	마이크를 통해 입력받은 음성에 전처리를 수행
Actor	User
Trigger	Pressed a record button
Primary Scenario	<ol style="list-style-type: none"> 1. 사용자가 마이크를 통해 음성을 녹음한다. 2. 사용자가 버튼을 눌러 녹음된 음성을 전송한다. 3. 시스템은 전송받은 음성을 전처리 수행하여 저장한다.
Alternative Scenario	None
Exceptional Scenario	None

Use Case ID	2
Use Case	STT Execution
Description	학습된 모델에 전 처리된 데이터를 적용하여 STT (Speech To Text)를 수행
Actor	System
Trigger	After pre-processing
Primary Scenario	<ol style="list-style-type: none"> 1. 시스템은 전처리 된 음성을 받아온다. 2. 시스템은 음성을 STT Model에 적용한다. 3. 시스템은 텍스트로 변환된 결과를 반환한다.
Alternative Scenario	None
Exceptional Scenario	None

Use Case ID	3
Use Case	SR Execution
Description	학습된 모델에 전 처리된 데이터를 적용하여 SR(Speaker Recognition)을 수행
Actor	System
Trigger	After pre-processing
Primary Scenario	<ol style="list-style-type: none"> 1. 시스템은 전 처리된 음성을 받아온다. 2. 시스템은 음성을 SR Model 에 적용한다. 3. 시스템은 SR의 결과를 반환한다.
Alternative Scenario	None
Exceptional Scenario	None

Use Case ID	4
Use Case	Function Execution
Description	인식된 결과를 통해 등록된 기능을 수행
Actor	System
Trigger	After SR & STT Execution
Primary Scenario	<ol style="list-style-type: none"> 1. 시스템은 SR의 결과를 받아온다. 2. 시스템은 STT의 결과를 받아온다. 3. 시스템은 SR 결과를 바탕으로 등록된 화자를 검색한다. 4. 시스템은 STT 결과를 바탕으로 인가된 화자의 명령 기능을 수행한다.
Alternative Scenario	None
Exceptional Scenario	<p>E.1) 3. 등록되지 않은 화자일 경우, 에러를 출력한다.</p> <p>E.2) 4. 등록되지 않은 기능일 경우, 에러를 출력한다.</p>

Use Case ID	5
Use Case	Dataset Pre-processing
Description	저장된 학습용 음성 데이터에 전 처리를 수행
Actor	System
Trigger	Whenever in need
Primary Scenario	1. 시스템은 음성 Dataset을 전처리 수행하여 저장한다.
Alternative Scenario	None
Exceptional Scenario	None

Use Case ID	6
Use Case	STT Training
Description	전 처리된 데이터를 이용해 STT 모델 학습을 수행
Actor	System
Trigger	After pre-processing
Primary Scenario	1. 시스템은 전처리된 음성을 받아온다. 2. 시스템은 학습된 STT 모델을 생성한다.
Alternative Scenario	None
Exceptional Scenario	None

Use Case ID	7
Use Case	SR Training
Description	전 처리된 데이터를 이용해 SR 모델 학습을 수행
Actor	System
Trigger	After pre-processing
Primary Scenario	1. 시스템은 전 처리된 음성을 받아온다. 2. 시스템은 학습된 SR 모델을 생성한다.
Alternative Scenario	None
Exceptional Scenario	None

2. 3. Constraints

- 정확한 척도를 기반으로 선택하는 것이 아니라 확률상 높은 것을 선택하기 때문에 STT와 SR 모델 적용한 결과에 오차가 존재할 수 있음
- 우리 모델을 학습시키기 위한 Train dataset의 양적인 부족으로 인해 학습량에 제한
- 한국어의 언어적 특성(교착어)으로 인한 STT 모델 학습의 어려움 존재
- Deep learning에 대한 사전지식 부족
- 모델 학습 연산을 위한 고성능 하드웨어 필요

2. 4. Assumptions and Dependencies

- 사용자에게 음성을 전달
- 노이즈가 최소화된 음성을 사용
- wav 파일로 음성 파일 형식을 통일
- HTTP 통신사용

3. Specific Requirements

3. 1. External interface requirements

3. 1. 1. User Interfaces

UI-1: 음성 녹음을 할 수 있는 User Interface를 웹 형태로 제공

UI-2: 실행 결과를 확인할 수 있는 User Interface를 웹 형태로 제공

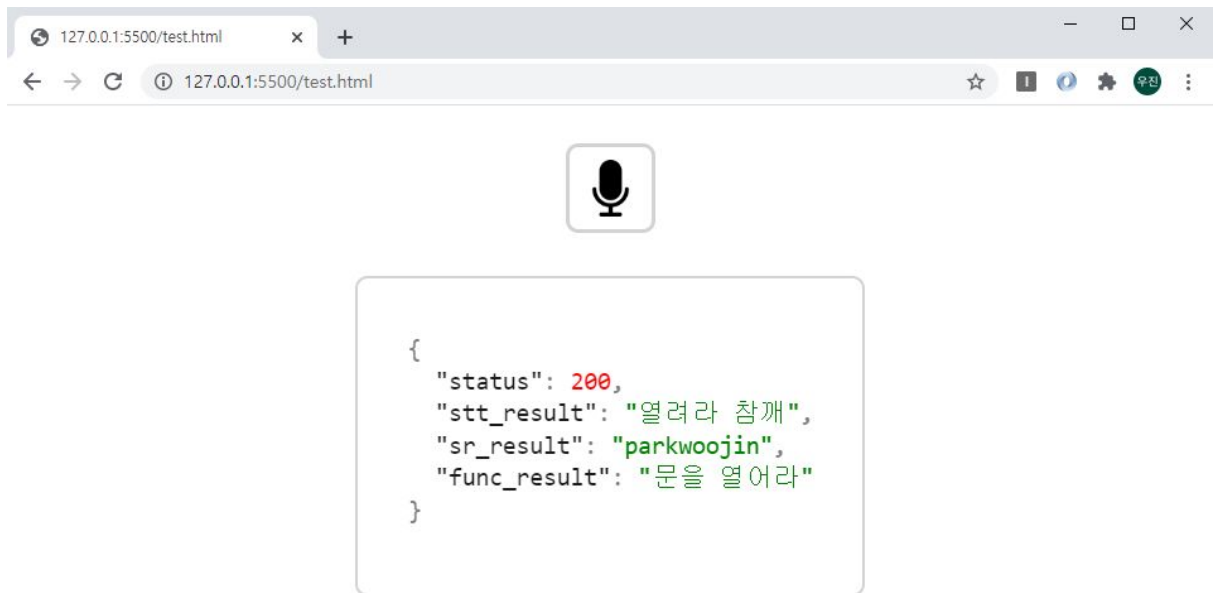


Figure-3. User Interface 예시

3. 1. 2. Hardware Interfaces

HI-1 : Microphone

HI-2 : Computing Hardware (Can install Chrome Browser)

3. 1. 3. Software Interfaces

SI-1: Chrome Browser (>= 83.0.4103.116)

3. 1. 4. Communications interfaces

CI-1: TCP/IP Protocol

CI-2: HTTP Protocol

3. 2. Functional Requirements

3. 2. 1. Live Model System

음성을 입력받고 이 음성은 파일로써 전달이 되어 전처리 과정을 가진 후, STT Model, SR Model을 적용해 나온 결과를 바탕으로 인가된 화자의 명령 기능을 수행합니다.

3. 2. 1. 1. Voice Pre-processing

기능	마이크를 통해 입력받은 음성에 전처리를 수행
설명	음성 전처리란 STT 모델과 SR 모델에서 음성의 특징정보를 이용하여 STT 기능과 SR 기능을 수행하기 위하여 사용자로부터 입력받은 음성의 특징정보를 추출하는 작업을 말한다.
파일 형식	사용자로부터 입력받은 음성은 wav 파일 포맷을 따른다
방식	경우에 따라 MFCC, STFT와 같은 음성 특징 추출방식을 사용할 예정
Input	사용자 음성
Output	사용자 음성의 특징정보

3. 2. 1. 2. STT Execution

기능	학습된 모델에 전처리된 데이터를 적용하여 STT (Speech To Text)를 수행
설명	STT란 Speech To Text로 사용자로부터 입력받은 음성의 특징정보를 이용하여 학습된 STT Model에 적용해 사용자가 말하는 한국어 음성을 컴퓨터가 해석한 다음 그 내용을 문자 데이터로 전환하는 처리 과정을 말한다.
Input	사용자 음성의 특징정보
Output	사용자 음성의 문자 데이터

3. 2. 1. 3. SR Execution

기능	학습된 모델에 전 처리된 데이터를 적용하여 SR(Speaker Recognition)을 수행
설명	SR 이란 Speaker Recognition으로 사용자로부터 입력받은 음성의 특징정보를 학습된 SR Model에 적용해 시스템에 등록된 화자를 식별하는 과정을 말한다.
Input	사용자 음성의 특징정보
Output	사용자 식별한 결과의 데이터

3. 2. 1. 4. Function Execution

기능	인식된 결과를 통해 등록된 기능을 수행
설명	사용자 음성의 문자 데이터와 사용자 식별한 결과의 데이터를 이용하여 등록된 사용자를 검색하고 사용자가 등록한 기능을 수행한다.
Input	사용자 음성의 문자 데이터, 사용자 식별한 결과의 데이터
Output	사용자 등록해 놓은 기능

3. 2. 2. Train Model System

한글 음성 데이터 Train Set을 전 처리 과정을 거쳐서 STT Model과 SR Model을 학습시킵니다.

3. 2. 2. 1. Dataset Pre-processing

기능	저장된 학습용 음성 데이터에 전처리를 수행
설명	학습용 데이터 전처리란 STT 모델 학습을 위해 학습용 음성 데이터의 특징정보 추출 및 문자 데이터를 생성하고, SR 모델 학습을 위해 사용자별 음성 데이터의 특징정보를 추출하는 과정을 말한다.
Input	학습용 음성 데이터
Output	사용자 음성의 특징정보

3. 2. 2. 2. STT Training

기능	전 처리된 데이터를 이용해 STT 모델 학습을 수행
설명	STT란 Speech To Text로 한국어 음성 데이터와 문자 데이터를 이용하여 CNN, LSTM와 같은 신경망 모델을 이용하여 학습된 STT 모델을 생성하는 과정이다.
Input	한국어 음성 데이터 및 문자 데이터
Output	학습된 STT 모델

3. 2. 2. 3. SR Training

기능	전처리된 데이터를 이용해 SR 모델 학습을 수행
설명	SR이란 Speaker Recognition으로 사용자별 음성의 특징정보를 이용하여 CNN, LSTM와 같은 신경망 모델을 통해 동일한 사용자의 음성 특징정보를 가려낼 수 있는 학습된 SR 모델을 생성하는 과정이다.
Input	사용자 음성의 특징정보
Output	학습된 SR 모델

3. 3. Non-Functional Requirements

3. 3. 1. Performance

- 3회 이내로 화자를 인식할 수 있어야 한다.
- 사용자 음성의 문자 데이터와 STT에 적용한 결과가 일치해야 한다.
- 인가된 화자의 등록된 기능이 수행되어야 한다.

3. 3. 2. Reliability

- 이 시스템은 딥러닝 방식을 이용하여 개발될 예정이다. 현재 이 시스템의 신뢰도를 측정할 수 있는 도구는 없다. 이 시스템의 신뢰도를 측정할 수 있는 방식은 사용자가 말한 음성과 우리 시스템을 적용한 그 결과를 비교하여 신뢰성을 측정할 수 있다.

3. 3. 3. Security

- 사용자의 음성 데이터가 탈취되어 악의적으로 사용될 경우를 방지하여 STT, SR 수행 후에 파기한다.

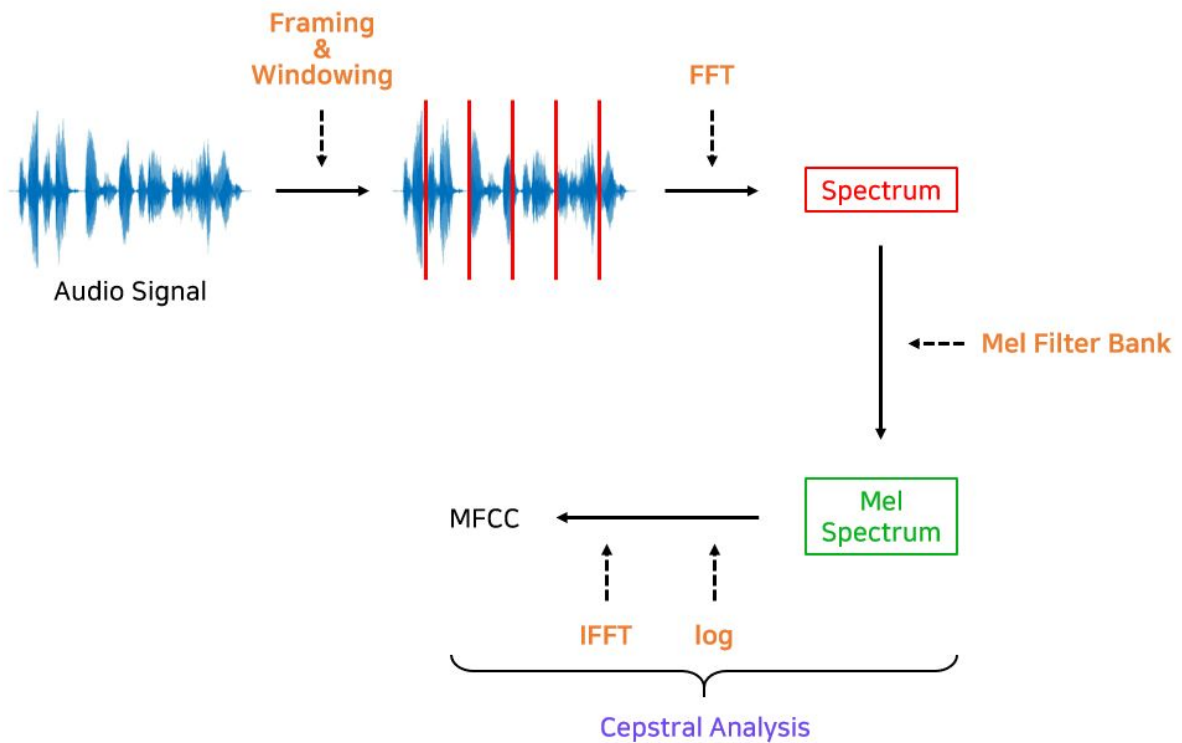
3. 3. 4. Portability

- 특정 운영체제나 하드웨어에 귀속되지 않고 Chrome Browser를 통해서 접근할 수 있다.

4. Appendixes

4. 1. MFCC (Mel-Frequency Cepstral Coefficients)

MFCC는 오디오 신호에서 추출할 수 있는 feature로, 소리의 고유한 특징을 나타내는 수치입니다. 음성 인식, 화자 인식, 음성 합성, 음악 장르 분류 등 오디오 도메인의 문제를 해결하는 데 사용됩니다.



먼저 오디오 신호를 프레임별(20ms - 40ms)로 나누어 FFT를 적용해 Spectrum을 구하고 Mel Filter Bank를 적용해 Mel Spectrum을 구한 뒤, Cepstral 분석을 적용해 MFCC를 구합니다. 오디오 신호는 시간에 따른 음압으로 시간 영역을 표현합니다.

여기에 FFT를 수행하면 주파수에 따른 음압의 표현으로 주파수 영역의 표현이 가능해집니다. 이것이 Spectrum입니다.

Spectrum을 사용하면 각 주파수의 대역별 세기를 알 수 있으니, 신호에서 어떤 주파수가 강하고 약한지를 알 수 있게 됩니다. 이렇게 주파수에 대한 정보를 가진 Spectrum에서 소리의 고유한 특징을 추출할 수 있습니다. 그리고 그 정보를 추출할 때 사용하는 방법이 Cepstral 분석입니다.

cf 1) FT (Fourier transform, 푸리에 변환)

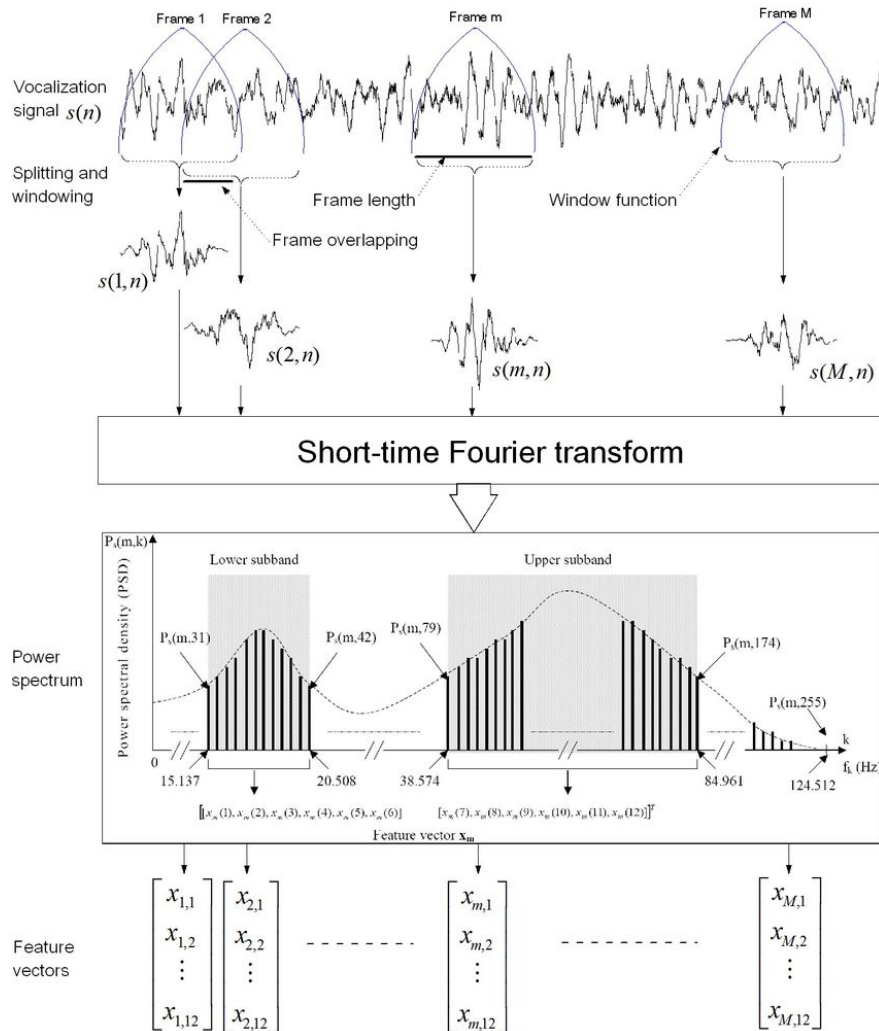
시간에 대한 함수를 구성하고 있는 주파수 성분으로 분해하는 작업으로 임의의 입력 신호를 다양한 주파수를 갖는 주기 함수들의 합으로 분해하여 표현하는 것

cf 2) FFT(Fast Fourier Transform : 고속 푸리에 변환)

신호를 주파수 성분으로 변환하는 알고리즘으로, 기존의 이산 푸리에 변환(DFT)을 더욱더 빠르게 수행할 수 있도록 최적화된 알고리즘.

4. 2. STFT (Short-Time Fourier Transform)

STFT 는 데이터에서 시간에 대해 구간을 짧게 나누고 그 구간의 데이터를 각각 푸리에 변환하는 방법입니다.

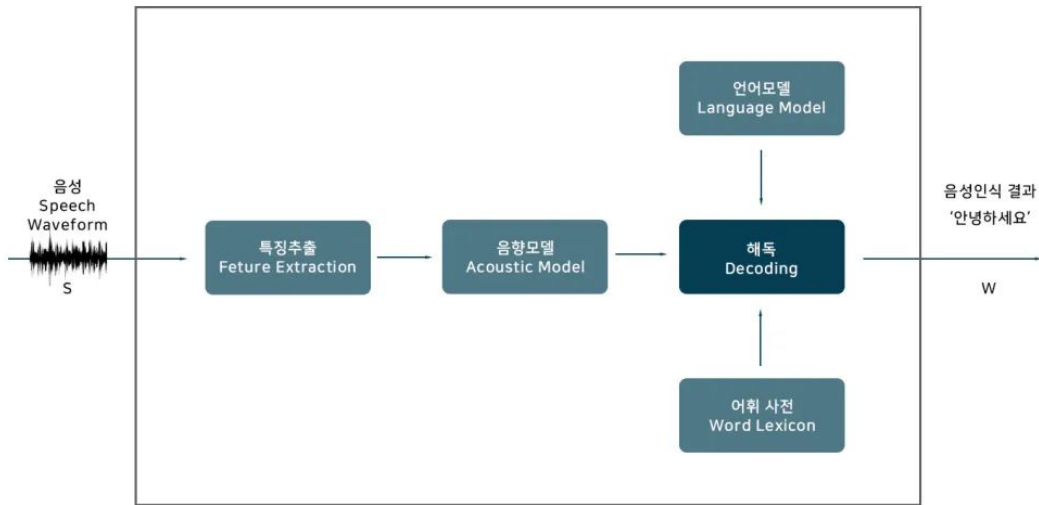


예를 들어, 9초 길이로 구성된 신호가 있다고 생각하면 FFT의 경우 9초 길이 전체에 대해 푸리에 변환을 하는 것을 말하고, STFT의 경우에는 9초 구간을 임의의 크기로 (0~3초, 3~6초, 6~9초 사이로) 나누어 나눠진 데이터를 각각 푸리에 변환하는 것을 뜻합니다.

FFT를 통해서 우리가 알 수 있는 정보는 주파수 성분이 각각의 Hz 로 이루어져 있다는 사실뿐이고 주파수 성분이 어느 시점에 존재하는지 여부는 알 수 없습니다. 그 이유는 복잡한 신호일수록 알기 어렵기 때문이며 이에 따라 Time-Frequency 방식을 이용하여 STFT를 사용합니다.

4. 3. STT 음성 인식 과정

음성인식은 다음과 같이 입력된 음성에 대해 여러 단계의 처리 과정을 거친 후 단어 열로 변환해 출력해 주는 것입니다. 음성인식 과정은 크게 특징추출, 음향 모델 계산, 언어 모델 계산, 디코딩의 4단계로 나눌 수 있습니다.



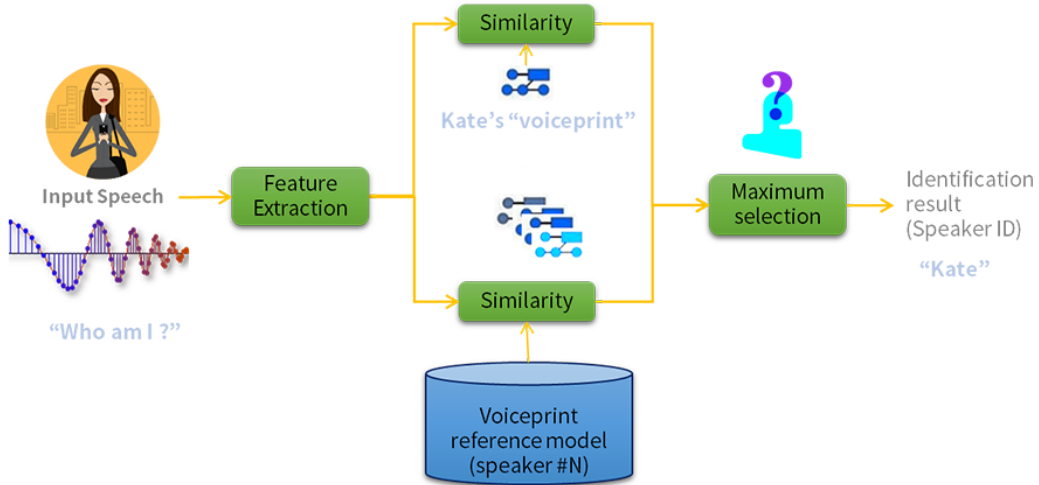
특징추출은 음성 신호에서 주파수 분석을 통해 음성의 특징되는 부분을 추출하는 과정입니다. 음향 모델은 음운 환경별 발음의 음향적 특성을 확률 모델로 하여 대표 패턴을 생성하는 모델이고, 언어 모델은 어휘 선택, 문장 단위 구문 구조 등 언어의 사용성 문제에 대해 문법 체계를 학습하는 모델입니다.

디코딩 단계에서는 음향 모델(Acoustic Model)과 언어 모델(Language Model)을 이용하여 입력된 특징 벡터를 모델과 비교하여 인식 결과를 나타냅니다.

이러한 두 모델을 거쳐서 음성을 문자로 생성하고 생성된 문자를 언어별 문법에 맞게 수정하여 최종 결과를 산출합니다.

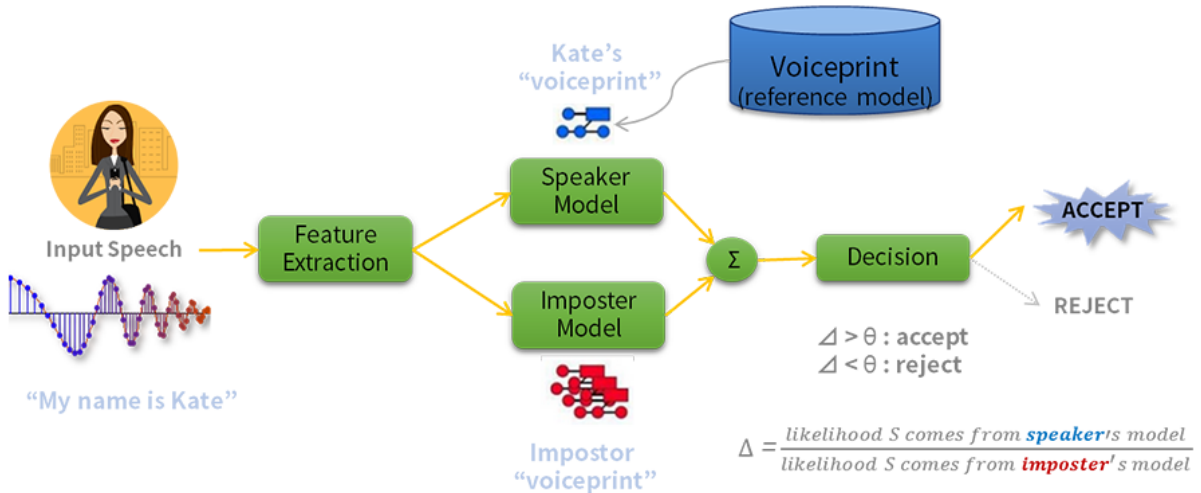
4. 4. SR 화자 인식 과정

화자 인식은 음성으로부터 화자의 정보를 찾아내는 방법을 말합니다. 일반적으로 화자식별 (Speaker identification)과 화자 검증(Speaker verification)으로 나누어집니다.



화자식별(Speaker identification)이란 주어진 음성으로부터 해당하는 화자를 찾아내는 기술입니다.

화자식별의 경우 시스템에 등록되지 않은 화자의 음성을 구분하지 못하고 등록된 음성중 가장 유사한 화자로 인식합니다. 그렇기 때문에 화자 검증과정을 진행합니다.

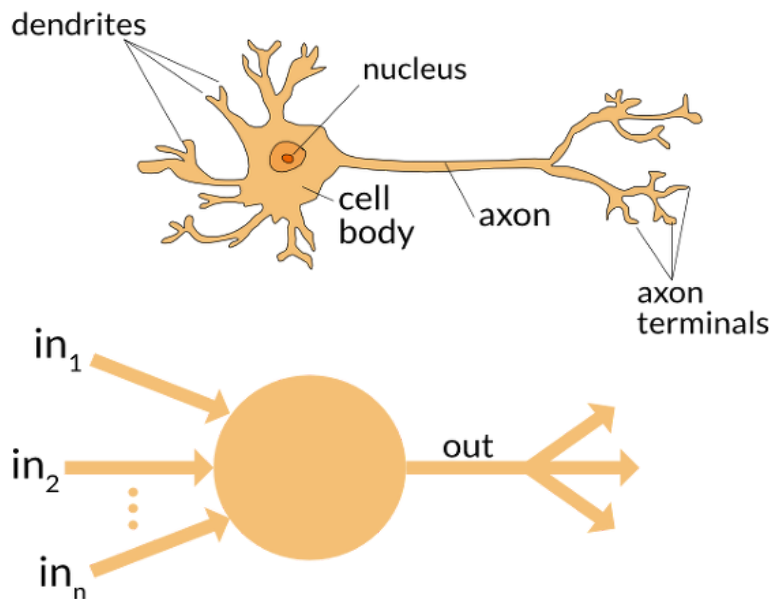


화자 검증(Speaker verification)이란 저장한 화자의 음성과 입력 음성 사이의 유사도를 구합니다.

저장되지 않은 목소리의 화자와 유사도 간 비율을 측정하고 이를 신뢰할 수 있는 비율에 해당하는 기준값에 따라 화자 일치 여부를 검증합니다.

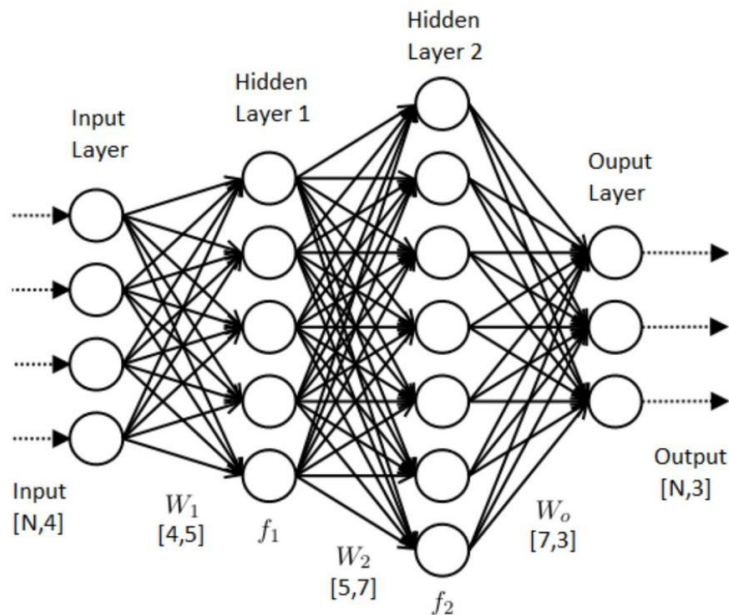
4. 5. NN (Neural Network)

NN이란 생물학에서 영감을 얻은 것으로 관측한 데이터로부터 컴퓨터가 학습을 하게 하는 프로그램 패러다임입니다.



4.5.1 ANN (Artificial Neural Network)

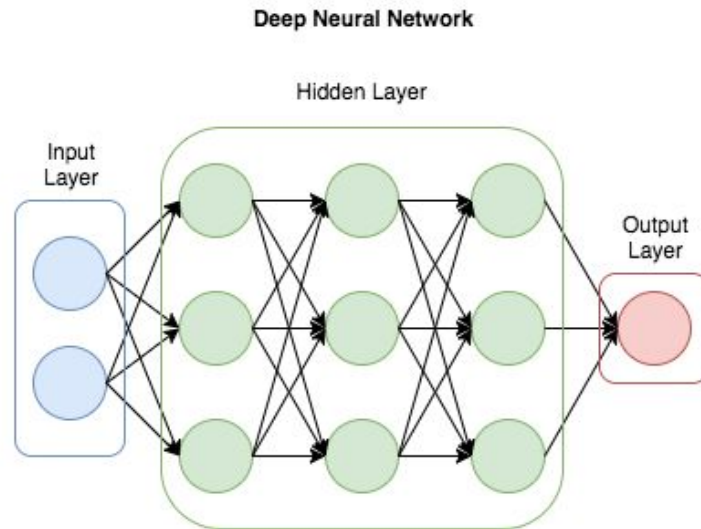
ANN은 인공신경망이라고 불립니다. 이는 사람의 신경망 원리와 구조를 모방하여 만든 기계학습 알고리즘입니다.



인간의 뇌에서 뉴런들이 어떤 신호, 자극 등을 받고, 그 자극이 임계값을 넘어서면 결과 신호를 전달하는 과정에서 착안한 것입니다. 자극, 신호 등은 인공신경망에서 Input Data이며 임계값은 가중치(weight), 자극 때문에 어떠한 행동을 하는 것은 Output Data 입니다.

4.5.2 DNN (Deep Neural Network)

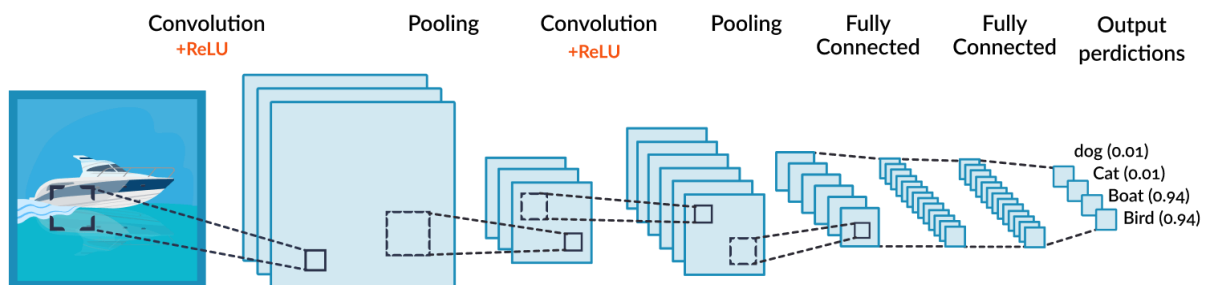
DNN은 은닉층을 2개 이상 지닌 학습 방법을 뜻합니다. 컴퓨터가 스스로 분류레이블을 만들어 내고 공간을 왜곡하고 데이터를 구분 짓는 과정을 반복하여 최적의 구분 선을 도출해냅니다.



많은 데이터와 반복 학습, 사전학습과 오류 역전파 기법을 통해 현재 널리 사용되고 있습니다.

4.5.3 CNN (Convolutional Neural Network)

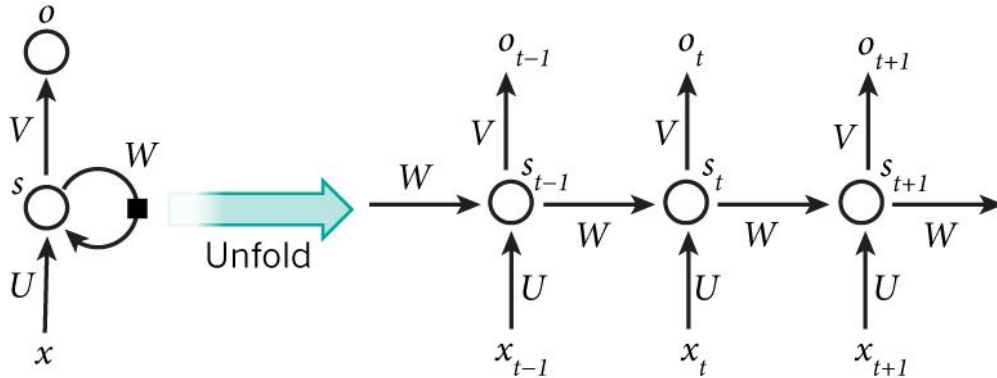
CNN 알고리즘은 데이터의 특징을 추출하여 특징들의 패턴을 파악하는 구조입니다.



이 CNN 알고리즘은 Convolution 과정과 Pooling 과정을 통해 진행됩니다. Convolution Layer와 Pooling Layer를 복합적으로 구성하여 알고리즘을 만듭니다.

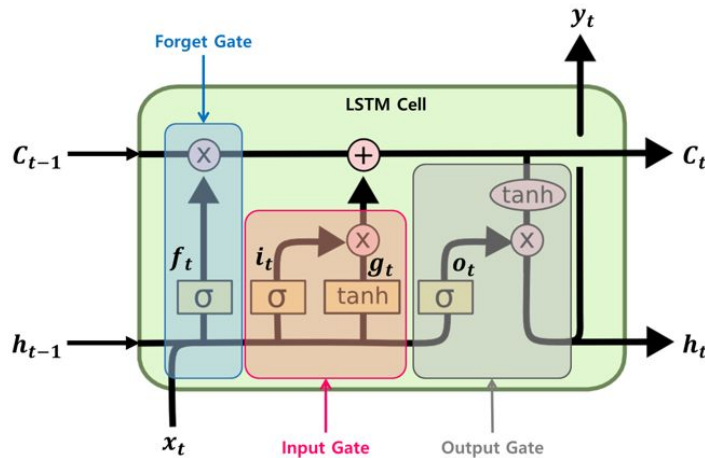
4.5.4 RNN (Recurrent Neural Network)

RNN 알고리즘은 반복적이고 순차적인 데이터(Sequential data) 학습에 특화된 인공지능망의 한 종류입니다.

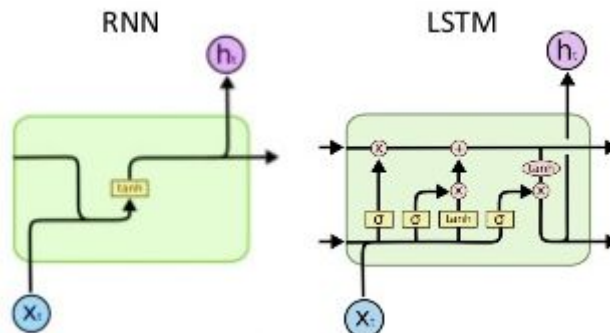


이 알고리즘은 내부의 순환구조가 들어있다는 특징을 가지고 있습니다. 순환구조를 이용하여 과거의 학습을 가중치(W)를 통해 현재 학습에 반영합니다. 기존의 지속적, 반복적, 순차적인 데이터 학습의 한계를 해결하는 알고리즘입니다. 현재와 과거 학습의 연결을 가능하게 하고 시간에 종속되는 특징도 가지고 있습니다.

4.5.4.1 LSTM (Long Short-Term Memory)



RNN은 시간에 연속적인 데이터를 학습시킬 수 있지만 시점 간의 간격이 크지 않을 경우에만 가능하기 때문에 RNN의 장기 의존성 문제를 해결하기 위해 1997년 등장했습니다.

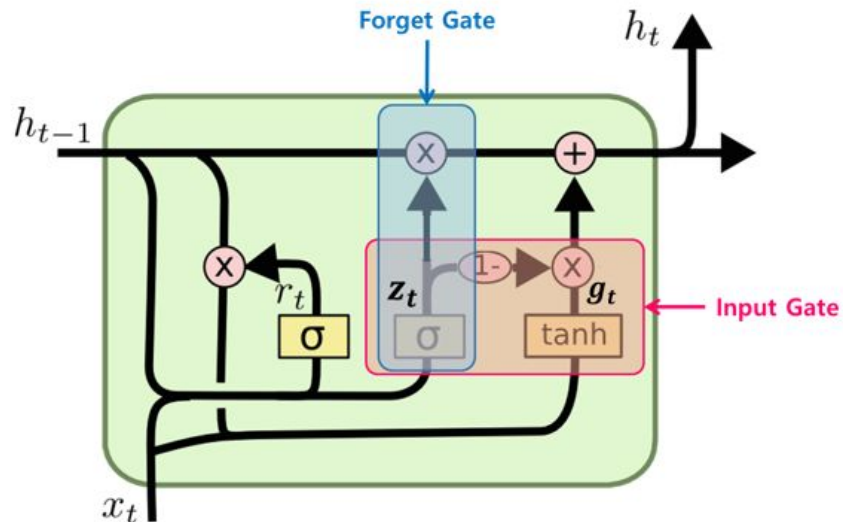


RNN과 LSTM의 차이점은 RNN은 단순한 계산을 수행하지만, LSTM은 Hidden Layer에서 총 4가지의 계산 과정이 존재합니다. Forget Gate Layer, Input Gate Layer, Update Cell State, Output Gate Layer를

통해 RNN의 장기 의존성 문제를 해결합니다. 직전 데이터뿐만 아니라, 과거 데이터를 고려하여 미래의 데이터를 예측하기 위해 사용됩니다.

4.5.4.2 GRU (Gate Recurrent Unit)

GRU는 LSTM의 간소화된 버전으로 LSTM보다 학습할 가중치가 적은 것이 장점입니다.



LSTM 게이트 3개를 2개로 줄이고, 출력값을 계산할 때 추가적인 비선형 함수를 적용하지 않습니다. Reset Gate r 과 Update Gate z 를 가지고 있습니다.

Reset Gate는 과거의 정보를 적당히 리셋시키려는 목적으로 사용됩니다. Update Gate는 LSTM의 forget gate와 input gate를 합쳐놓은 느낌으로 과거와 현재의 정보의 최신화 비율을 결정합니다.