Voice Activity Detection

Jahyun Goo, Ph.D student, KAIST

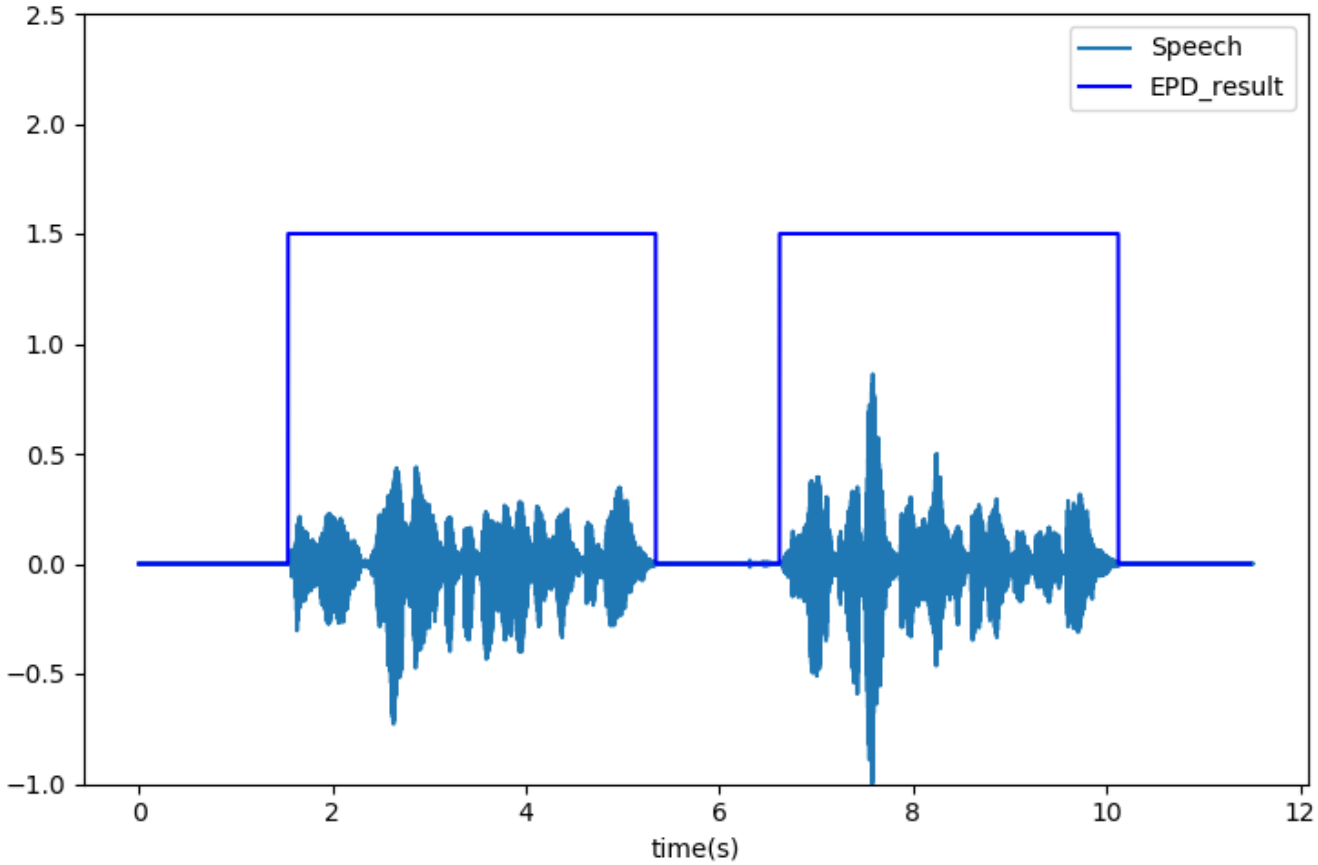
jahyun.goo@kaist.ac.kr

(based on Youngmoon's implementation)

Introduction

What is VAD?

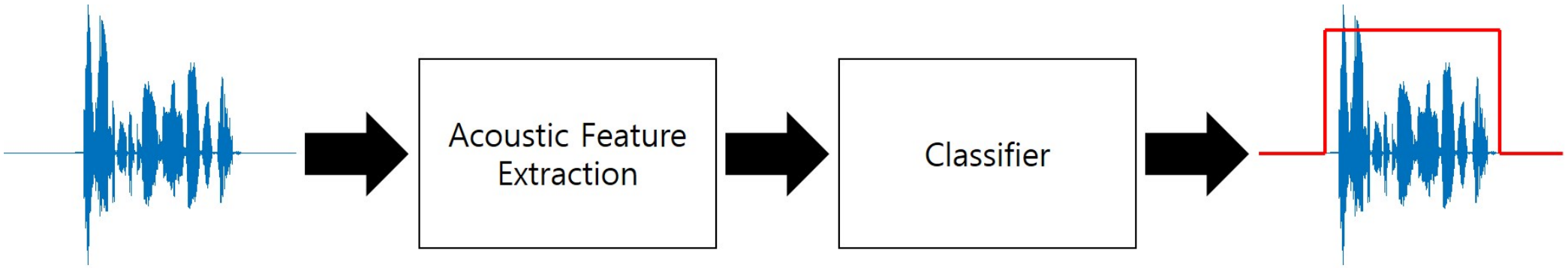
- Voice Activity Detection
  - **이 프레임**이 음성인가 아닌가



What is VAD?

- VAD as system
  - **이 프레임**이 음성인가 아닌가

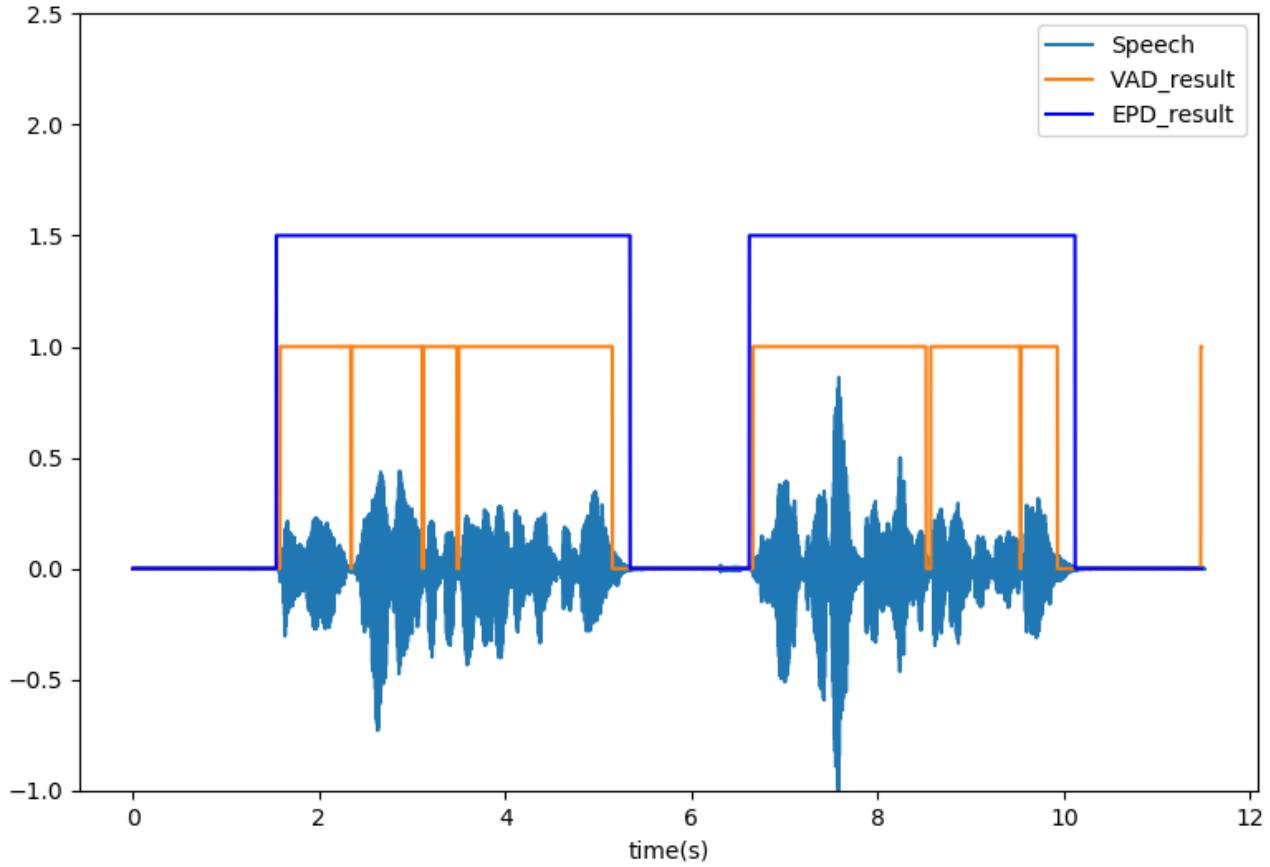
  → Frame-based binary classification (프레임 기반 이진 분류)



https://github.com/jtkim-kaist/VAD

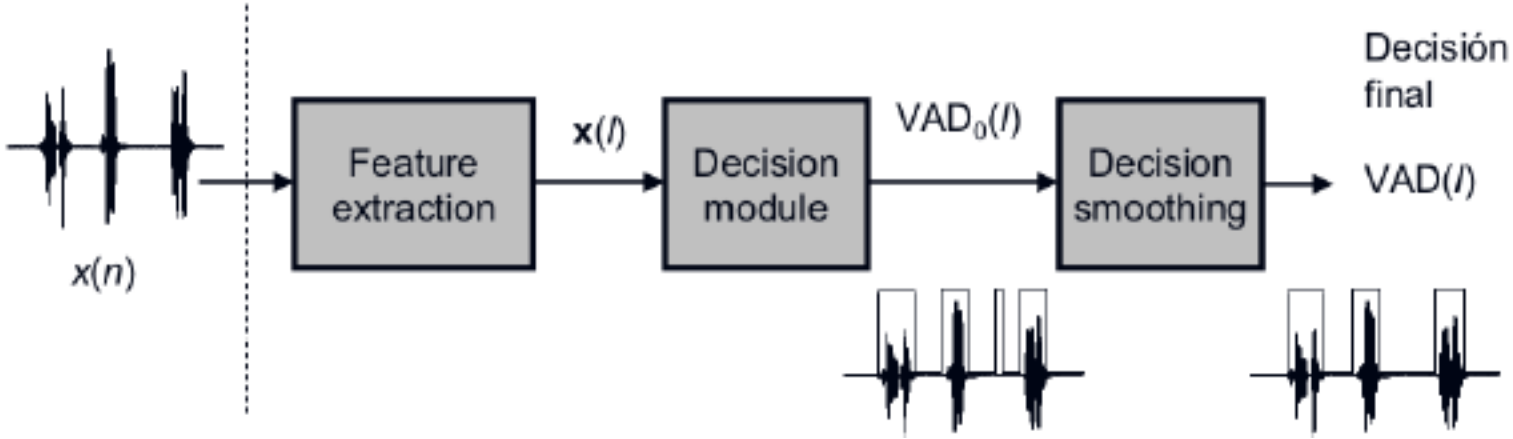
VAD? EPD?

- End-Point Detection
  - **음성구간**의 양 끝점 검출



VAD? EPD?

- EPD as system



https://www.vocal.com/dereverberation/voice-activity-detection/

Why VAD?

(고급)음성지능_화자인식 및 음성합성

과정개요

➢ 과정개요

- 목적

. 화자인식 및 음성합성의 기본사항 및 최신동향을 이해한다

. 주요알고리즘 성능을 분석하고 기능을 제어하는 방법을 이해한다

- 대상 : 음성인식 중급이상의 역량보유자

➢ 강사 : **김회린 교수**(KAIST 전기전자공학부) 외 실습조교

➢ 교육일정 : 2019년 2월 25일(월) ~ 2월 27일(수), 08:30 ~ 17:30

➢ 교육장소 : 서초R&D캠퍼스 14층 교육실3

Why VAD?

- 근거리 음성인식이 아닌 모든 음성분류 분야에 선행
  - 정도에 따라 다르다

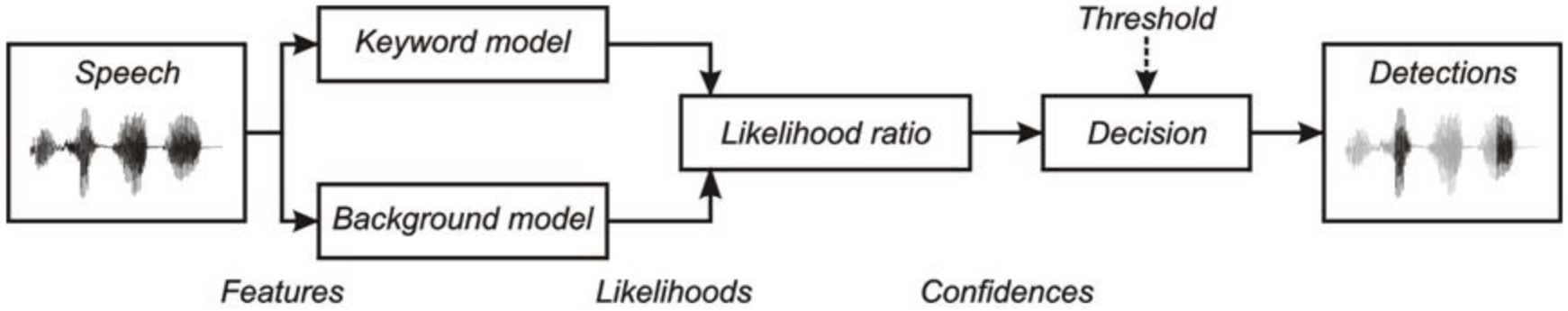
화자인식 (Speaker recog.),
감정인식 (Emotion recog.),
and etc...

키워드 인식
(Wake-up Word Detection,
Key-Word Spotting)

원거리 음성인식
(Distant Speech Recognition),
음질 향상 (Speech enhancement)

여담 1: KWS

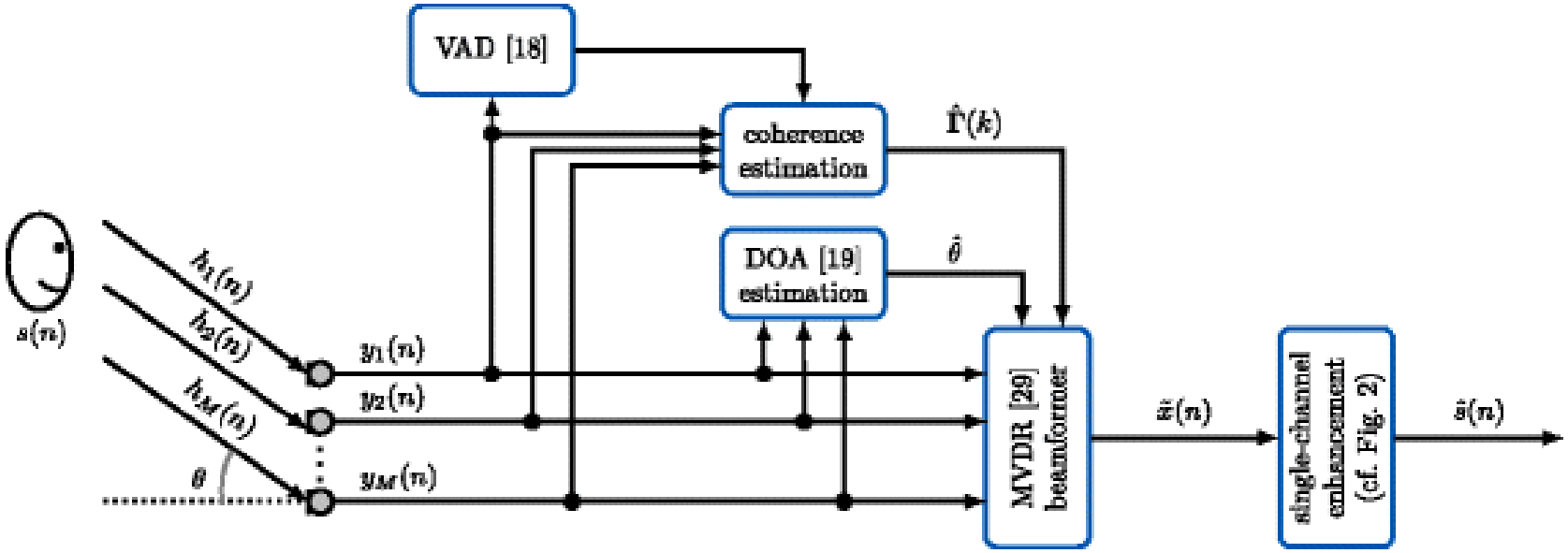
- KWS system에서의 EPD
  - **정확히** 단어 단위로 넘겨줘야 한다



SRE Lectures, BUT FIT, 2010

여담 2: 빔포밍과 VAD

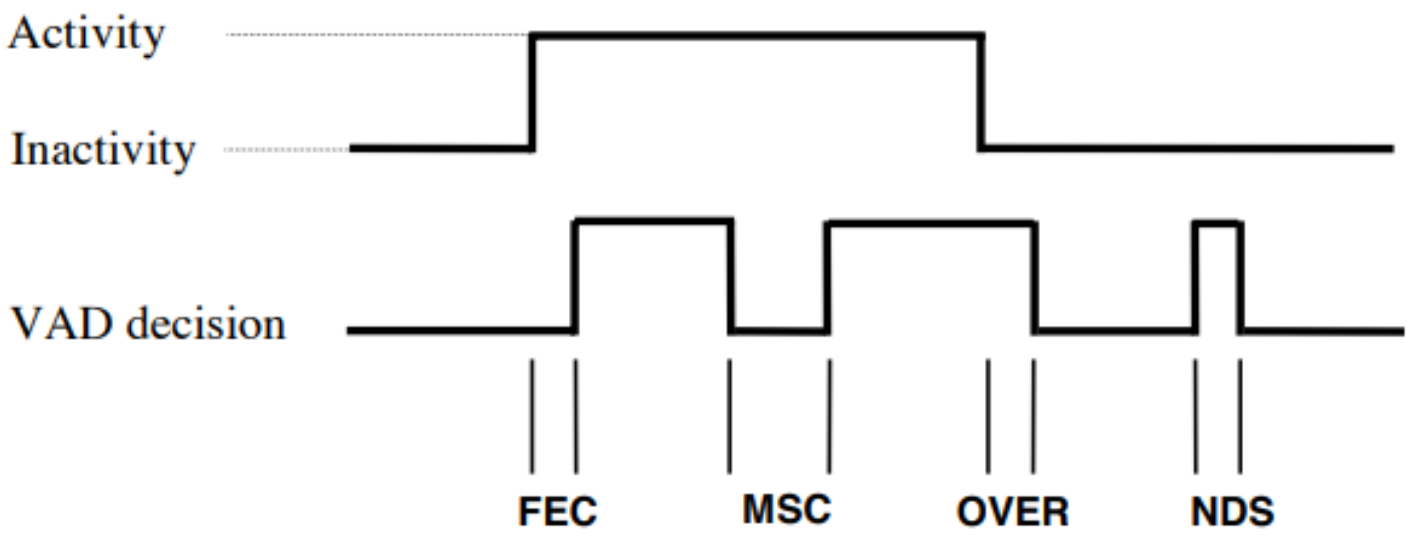
- 원거리 음성인식 시스템에서 (어레이 마이크를 이용한) 빔포밍과 VAD는 어떤 관계인가?
  - 생각 외로 연구중인, 결정/구현하기 나름인 주제



B. Cauchi and S. Goetze, EURASIP J. Adv. Signal Process., 2015(61), 2015.

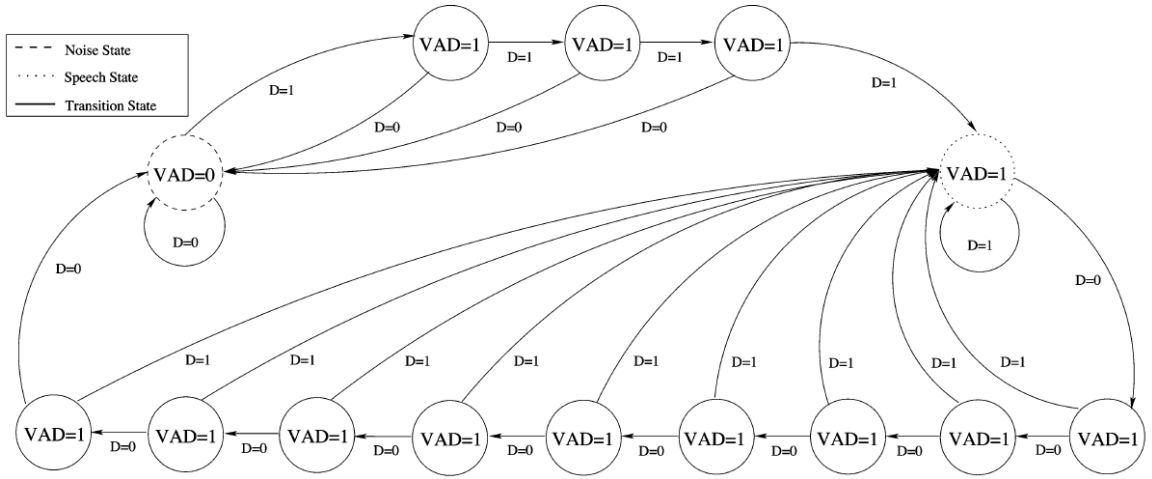
Hangover

- Decision smoothing scheme
  - 각 구간을 얼마나, 어떤 식으로 처리할지에 정답은 없다



Hangover

- State machine 기반의 hangover scheme
  - 비음성 state에서 시작
    - **4 frame (40ms)** 연속 음성이라 판명되면 음성 state로 변함
    - 비음성이라 판명되면 계속해서 비음성 state
  - 음성 state가 되었을 때
    - **25 frame (250ms)** 연속 비음성이라 판명되면 비음성 state로 변함
    - 음성이라 판명되면 계속해서 음성 state



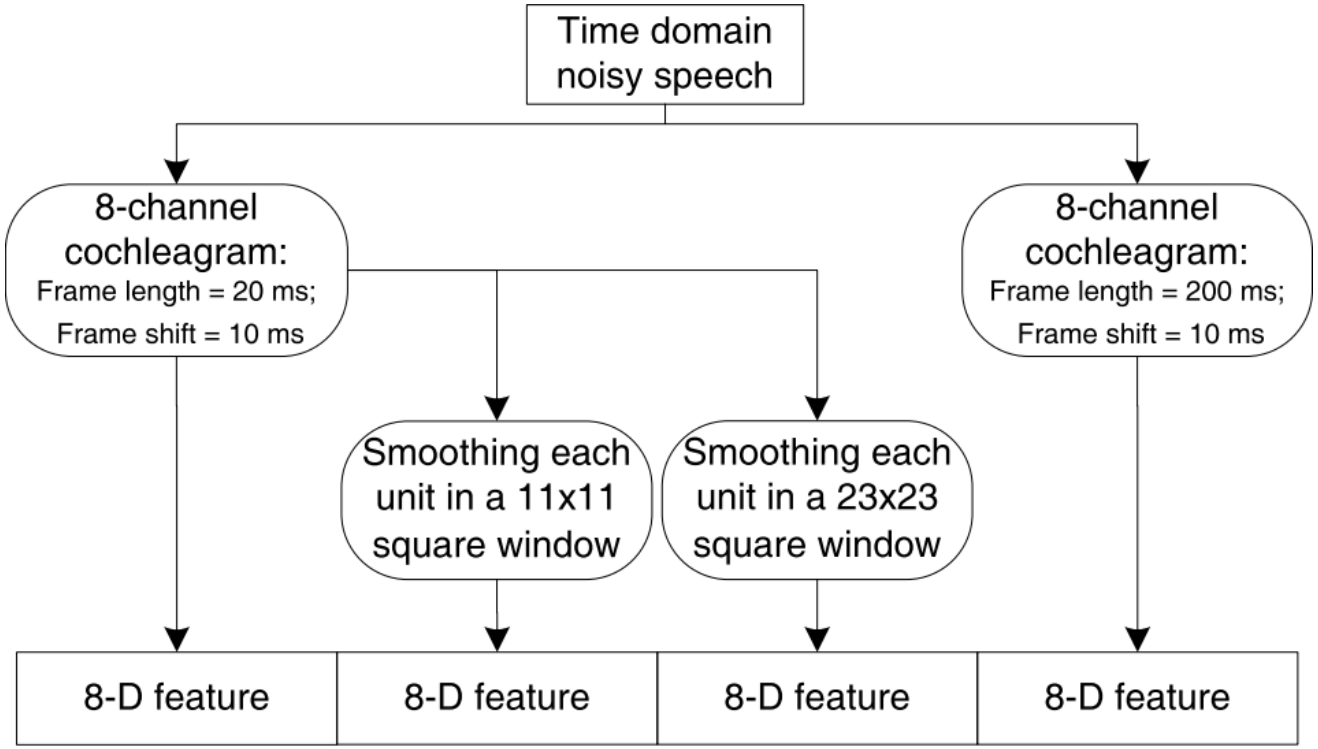
Practice

실험 환경 구성

- Databases
  - SNS단문 낭독 음성 DB
    - 산업자원부/KEIT, '실내용 음성대화 로봇을 위한 원거리 음성인식 기술 및 멀티 태스크 대화처리 기술 개발' (No. 10063424) 2차년도 수집/공개 DB
  - 1m 거리의 무잡음 음성, 0도 방향
  - 16kHz, 16bits
  - 100인, 각각 100문장
  - MRCG feature 형태로 제공

실험 환경 구성

- MRCG feature
  - 본 실습에서는 24-dim.씩 4개 해상도 사용하여 프레임당 96-dim. 특징 이용



X.-L. Zhang and D. Wang, IEEE/ACM TASLP, 24(2), 2016

기본 사용법

- 훈련 : `python train.py`
  - GPU/CPU 여부에 따라 `train.py`, `test.py`의 `main()`에서 `use_cuda=True (False)` 지정
    - GPU : 1분 30초 (Nvidia GTX 1080 1개 기준)
    - CPU : 3분 (Intel Xeon E5-2620 v4 @ 2.10GHz)
- Validation loss 기준 최적 모델 확인
  - standard output에서 확인, 혹은
  - `display loss_plot.png`
- `test.py:main()`에서 아래 지정
  - `cp_num` 선택
  - test file 선택 (0~19)
- 테스트 : `python test.py`
  - display된 VAD 및 EPD 결과 확인

Known bug fix

- TruncatedInputfromMFB -> TruncatedInputfrom**MRCG**
  - train.py:16,31
  - VAD_Dataset.py:15,174

- librosa error (test.py에서 일어난 경우)
  - test.py:98의 def load_audio_feat(filename): 아래에 다음 추가

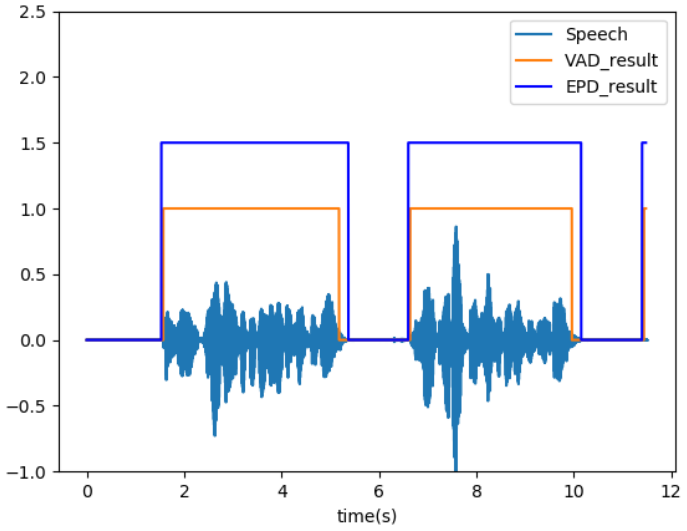
```
from scipy.io import wavfile
# audio, sr = librosa.load(wav_name, sr=c.SAMPLE_RATE, mono=True)
sr, audio = wavfile.read(wav_name)
```

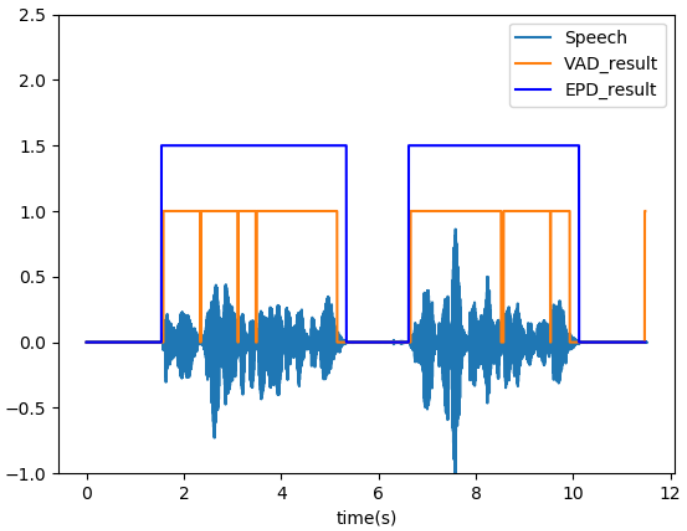
Practice

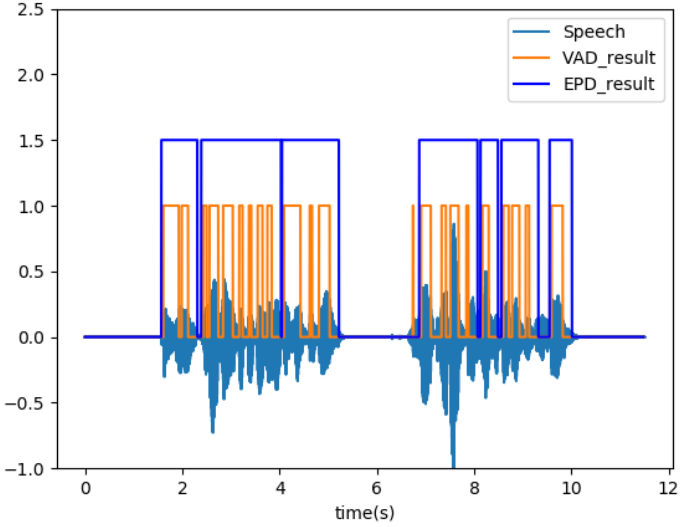
- 구현되어 있는 Hangover scheme과 관련하여, 아래 세 파라미터의 최종 EPD 성능에 대한 역할 확인
  - `thres = 0.4`
  - `min_s = 0.2`
  - `min_ns = 0.04`

Practice

- VAD threshold
  - `min_s, min_ns : default`
  - `thres = 0.01, 0.4, 0.99`

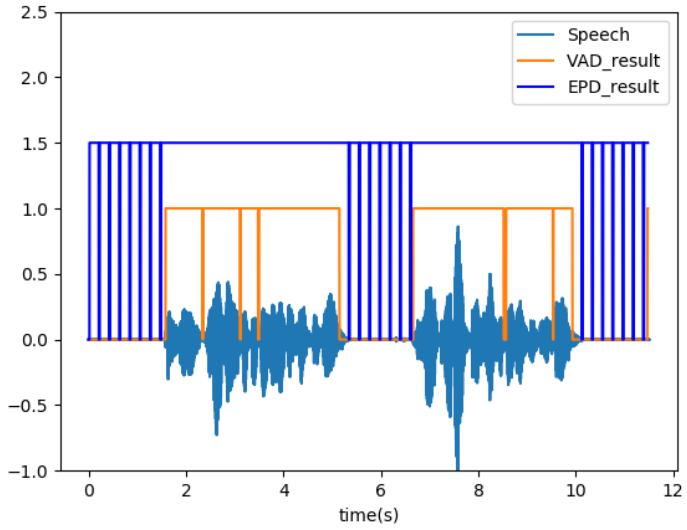


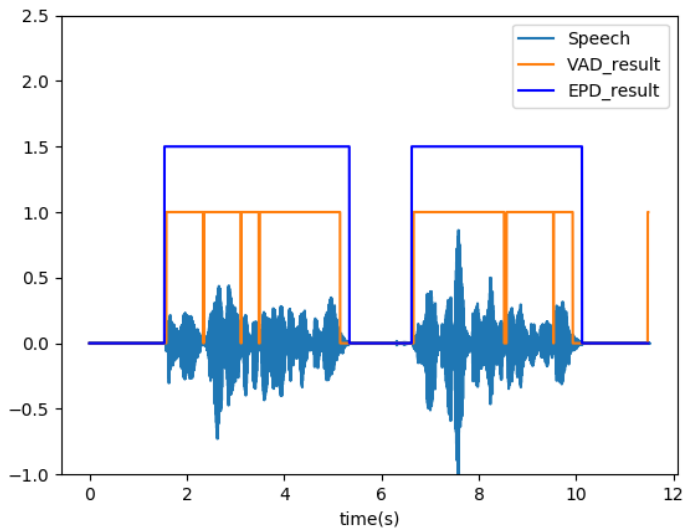


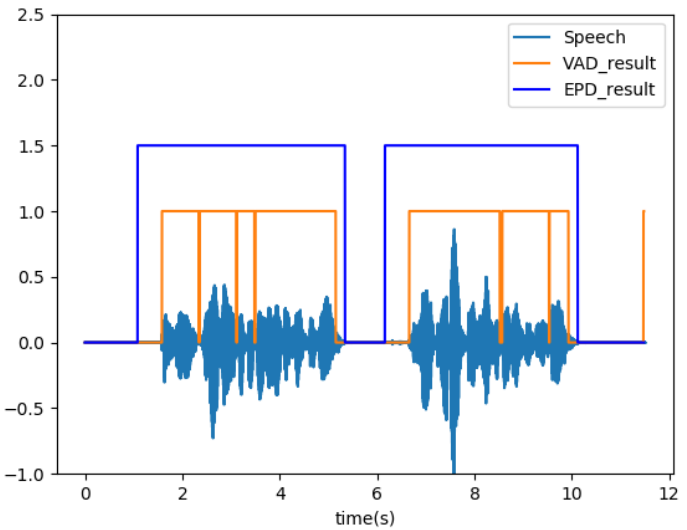


Practice

- Minimum noise-to-speech length (Hangover)
  - `thres, min_s : default`
  - `min_ns = 0.0, 0.04, 0.5`

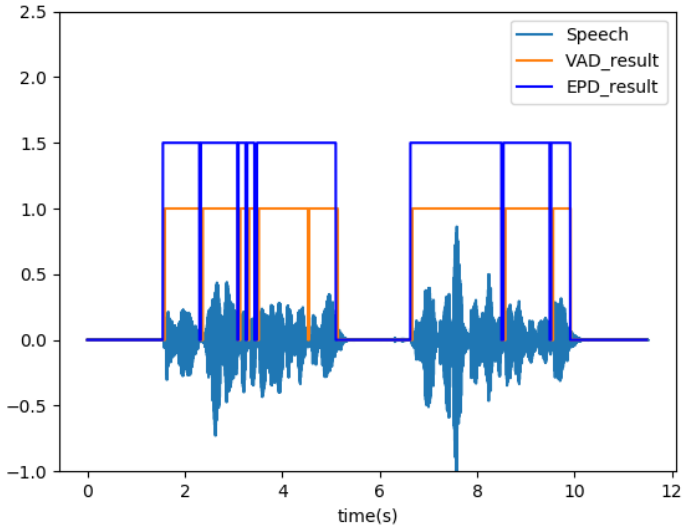


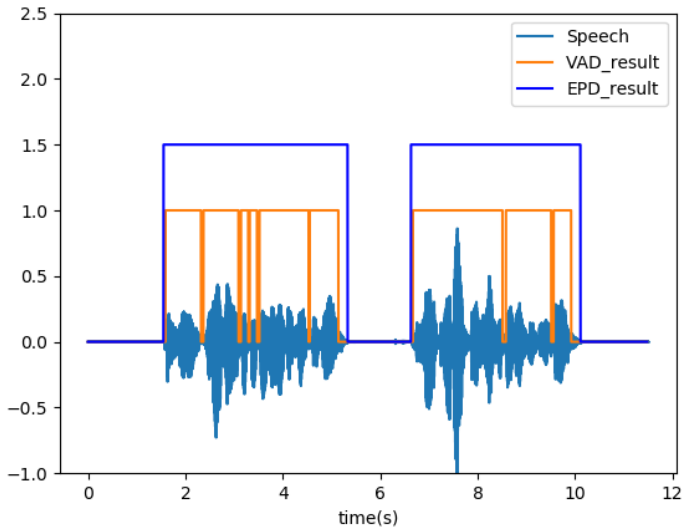


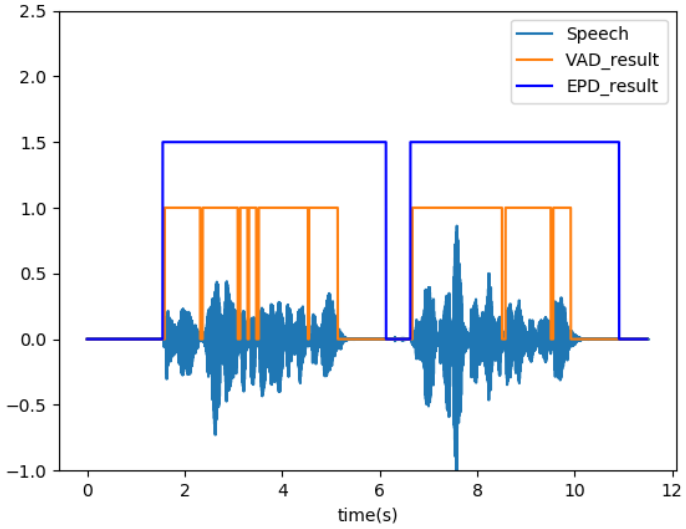


Practice

- Minimum silence length (Hangover)
  - `thres = 0.6 (high), min_ns : default`
  - `min_s = 0, 0.2, 1.0`

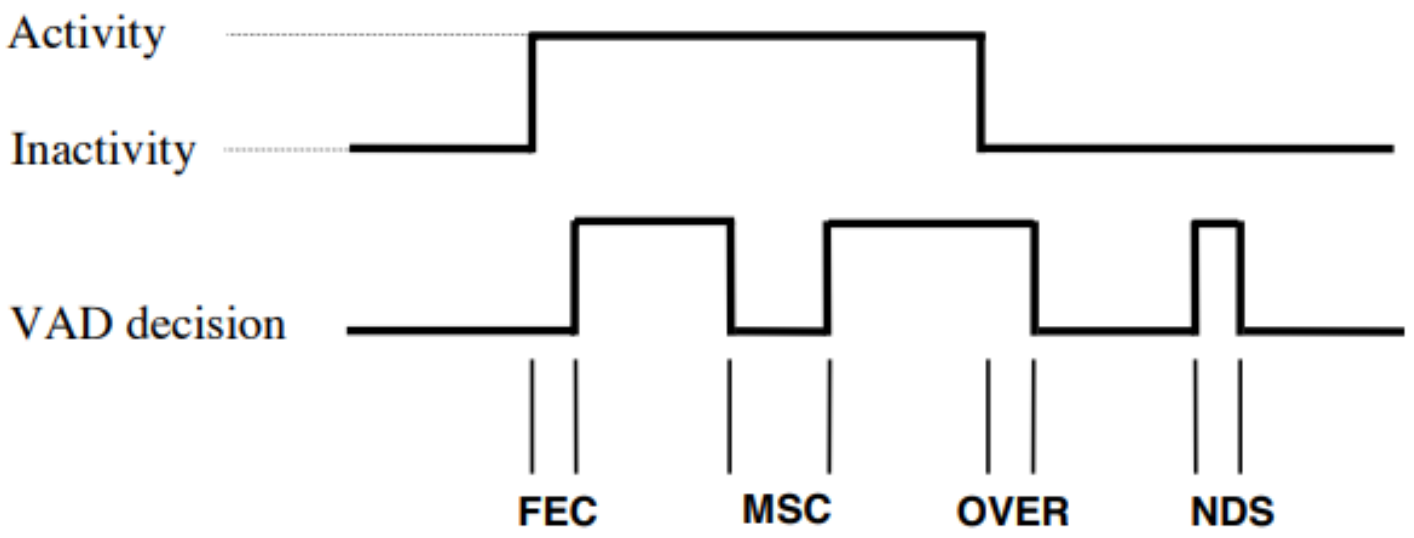






Practice

- Decision smoothing scheme
  - 각 구간을 얼마나, 어떤 식으로 처리할지에 정답은 없다



Projects & etc

Projects

- Assignments #1
  - 본 Task (test 문장 20개)에 대해서 가장 적절한 (thres, min_s, min_ns)-triplet 조합 찾기

- Assignments #2
  - 보다 최적의 모델 찾기
    - 현재는 DNN (512, 3)의 FCN 모델로 구성
    - 비교적 단순한 VAD임에도, task에 따라 훨씬 복잡한 모델 사용하기도 함

```
class DNN(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes=2):
        super(DNN, self).__init__()
        self.fc1 = nn.Linear(input_size, hidden_size)
        self.bn1 = nn.BatchNorm1d(hidden_size)
        self.fc1_drop = nn.Dropout(p=0.2)

        self.fc2 = nn.Linear(hidden_size, hidden_size)
        self.bn2 = nn.BatchNorm1d(hidden_size)
        self.fc2_drop = nn.Dropout(p=0.2)

        self.fc3 = nn.Linear(hidden_size, hidden_size)
        self.bn3 = nn.BatchNorm1d(hidden_size)
        self.fc3_drop = nn.Dropout(p=0.2)

        self.last = nn.Linear(hidden_size, num_classes)
```

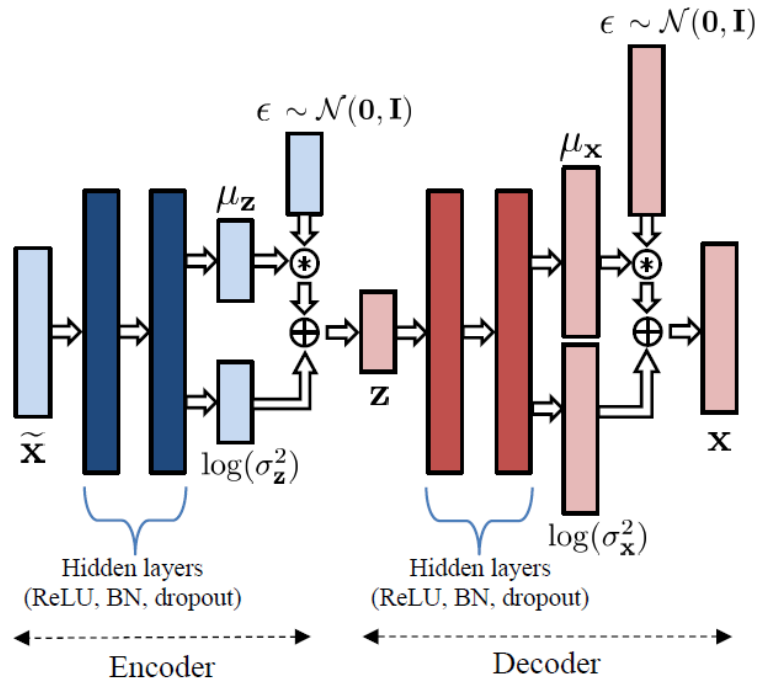
Projects

- Take-home assignment #1
  - Optimizing for more challenging/realistic domain
- Take-home assignments #2
  - Integrated system with Speaker recognition

Further theoretics

- Joint Learning using Denoising Variational Autoencoders
  (Y. Jung et al., Interspeech 2018)

The denoising variational autoencoder architecture for speech enhancement (SE-DVAE)



✓ **SE-DVAE**

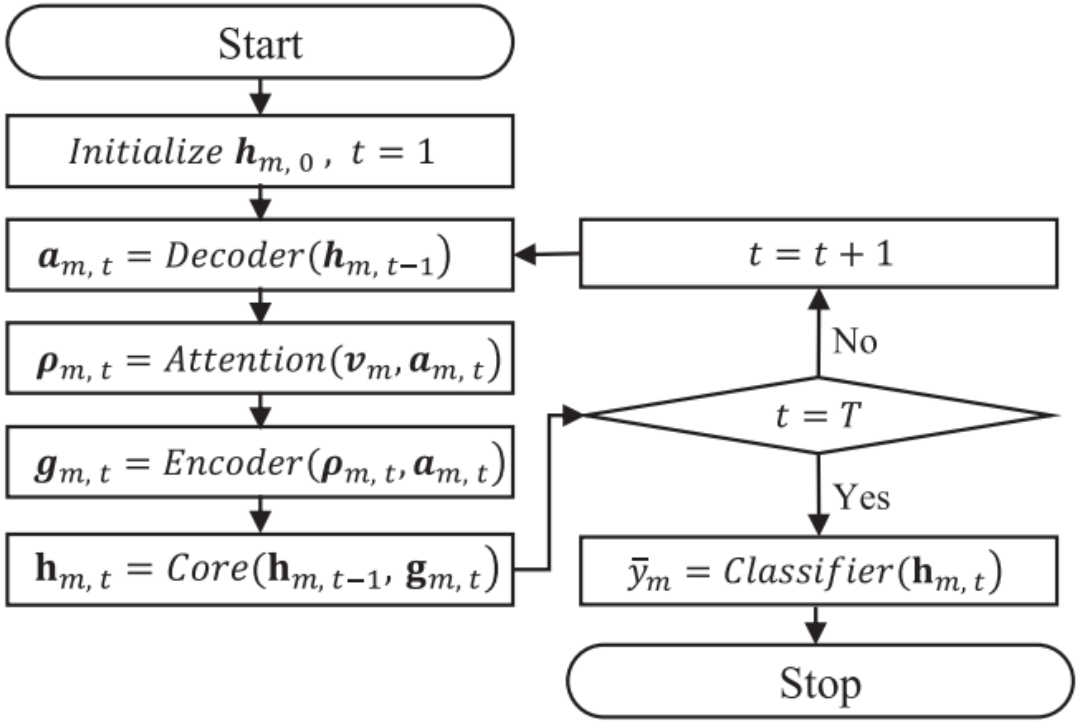
- Encoder :
  - $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^2\mathbf{I})$
  - DNN with 2 hidden layers (2048 nodes)
  - Input : noisy feature $\widetilde{\mathbf{x}}$
  - Output : 64-dim mean $\mu_{\mathbf{z}}$ + log-var $\log(\sigma_{\mathbf{z}}^2)$ => $\mathbf{Z}$
- Decoder :
  - $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mu_{\mathbf{x}}, \sigma_{\mathbf{x}}^2\mathbf{I})$
  - DNN with 2 hidden layers (2048 nodes)
  - Input : $\mathbf{Z}$
  - Output : mean $\mu_{\mathbf{x}}$ + log-var $\log(\sigma_{\mathbf{x}}^2)$ =>$\mathbf{X}$

✓ **VAD-DNN**

- DNN with 2 hidden layers (2048 nodes)
- Input : enhanced feature from SE-DVAE
- Output : probability of speech class

Further theoretics

- VAD using an Adaptive Context Attention Model
  (J. Kim and M. Han, IEEE SPL 25(8), 2018)



Resources

- https://github.com/jymsuper/VAD_tutorial
  - Source and databases for today
  - This ppt to be uploaded
  - Other open corpus

- https://github.com/jtkim-kaist/VAD
  - TensorFlow-based VAD and SE solutions