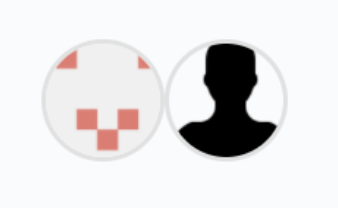
Rush4. 음악 추천 스테이션 분류

Mung 팀



김승빈, 노성훈

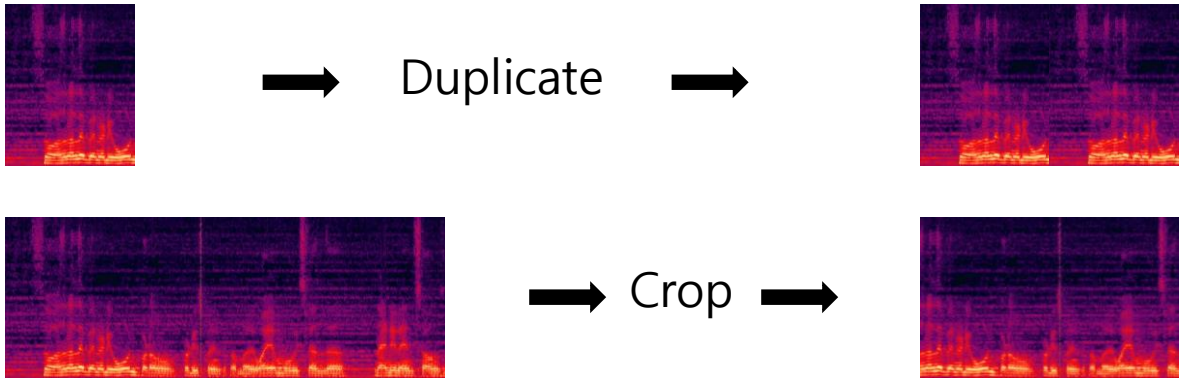
2020. 08. 27

Table of Contents

Input

❖ 입력 Mel-spectrogram의 batch 구성

- Train batch 구성: mel-spectogram의 7000 frames = 3분 44초 (length of each time frame: 32ms) 사용



Spectrogram 예시
data shape: (#frequency, #time)

data shape: (#frequency, #time=7000)

► **Frame 7000** 고정 결과 **성능 향상**

Network

❖ 1D-Conv 적용

- 기본 네트워크는 2D-CNN인 SpeakerNet(ResNetSE34L, rush8 baseline)으로 설정
- 기존 네트워크는 mel-spectrogram을 이미지로 본 후 2D-conv 적용
- 소리 데이터임을 감안하여 mel-spectrogram의 frequency 축을 channel 축으로 본 후 1D-conv를 적용하여 성능 비교
  - ► **1D-conv** 적용 결과 **성능 하락**

Network

❖ CRNN 적용

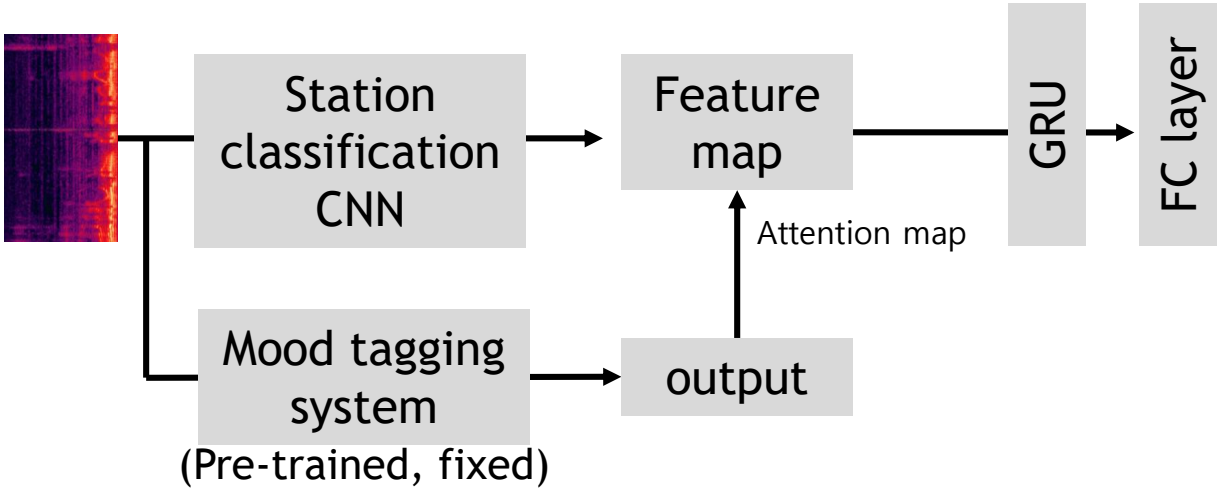
- 2016, ISMIR에 발표된 "convolutional recurrent neural networks for music classification" 논문을 참고
- 기존 2D convolutional network에서 feature를 aggregation할 때 pooling 기법 사용
- 시간적 특성을 가지고 있는 음악 데이터를 고려하여 RNN모델인 GRU를 이용하여 aggregation.

► **CRNN** 적용 결과 **성능 향상**

Network

❖ Attention map 적용

- 2020, Interspeech에 발표될 "acoustic scene classification using audio tagging" 논문을 참고
- 스테이션 분류 학습 시 tagging network의 output을 활용하여 attention map을 만들어 cnn feature map을 scaling



► **Attention map** 적용 결과 **성능 향상**

Others

- ❖ Loss
  - CCE, Focal-loss 적용 결과 **성능 변화 x**
- ❖ Augmentation
  - Mixup, frequency random masking 적용 결과 **성능 하락**
- ❖ Optimizer
  - Adam
- ❖ Scheduler
  - Cosine Annealing

Others

❖ Ensemble

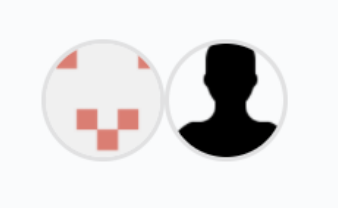
- RexNet1.0, RexNeXt50, SpeakerNet, SpeakerNet-1d

► 4개 모델 Ensemble 하여 **성능 향상**

감사합니다 ☺

Rush5. 음악 추천 무드 태그 분류

Mung 팀



김승빈, 노성훈

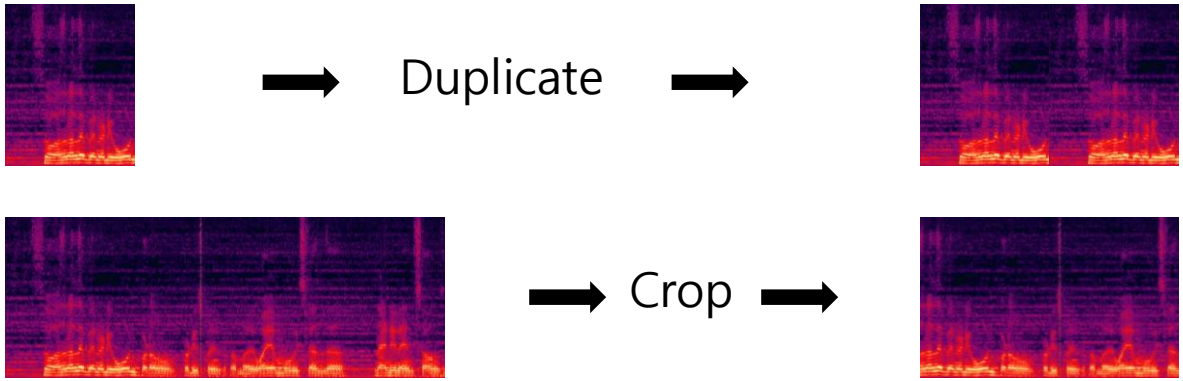
2020. 08. 27

Table of Contents

Input

❖ 입력 Mel-spectrogram의 batch 구성

- Train batch 구성: mel-spectogram의 7000 frames = 3분 44초 (length of each time frame: 32ms) 사용



Spectrogram 예시
data shape: (#frequency, #time)

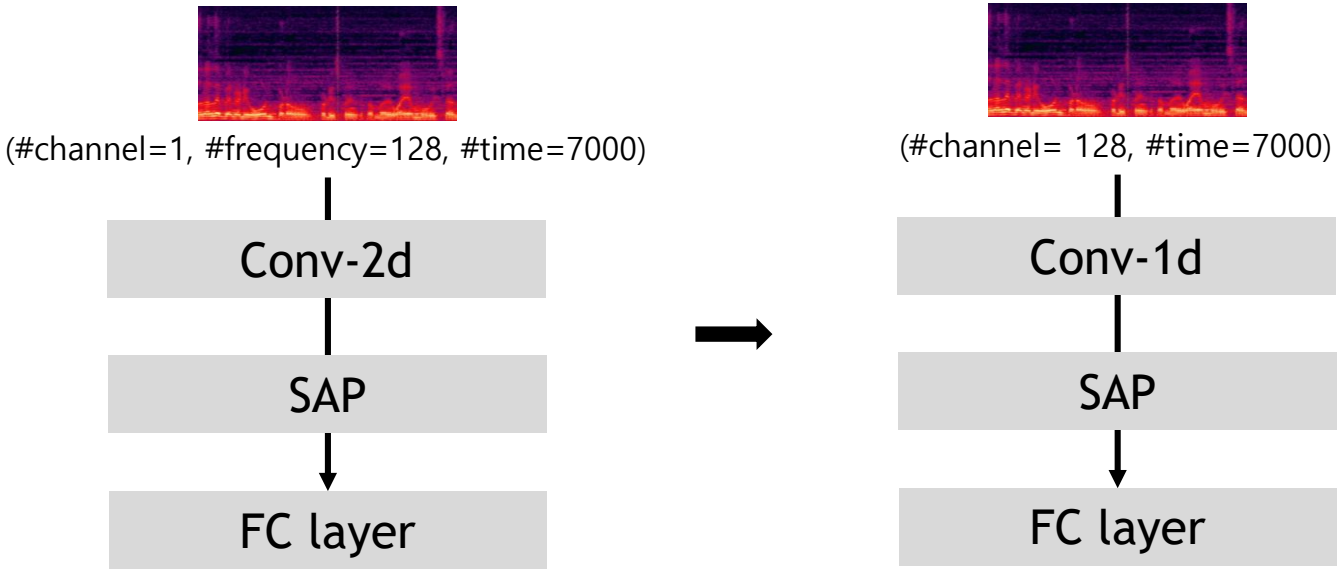
data shape: (#frequency, #time=7000)

► **Frame 7000** 고정 결과 **성능 향상**

Network

❖ 1D-Conv 적용

- 소리 데이터임을 감안하여 mel-spectrogram의 frequency 축을 channel 축으로 본 후 1D-conv를 적용하여 성능 비교

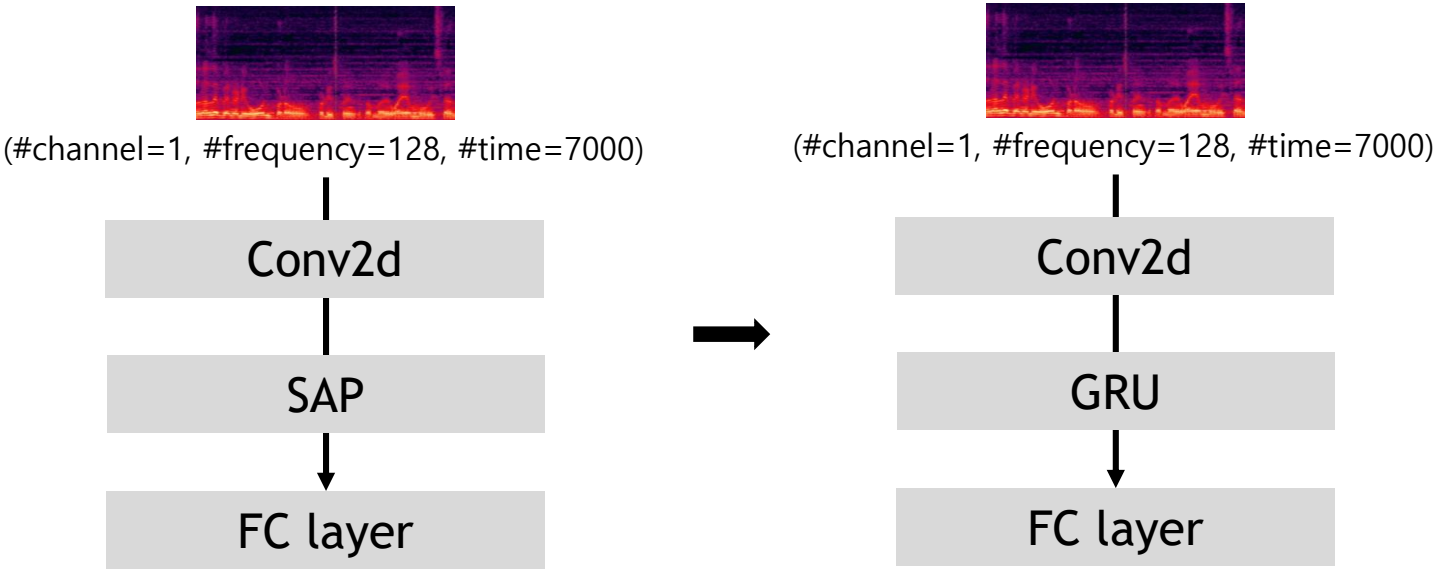


► **1D-conv** 적용 결과 **성능 하락**

Network

❖ CRNN 적용

- 시간적 특성을 가지고 있는 음악 데이터를 고려하여 RNN모델인 GRU를 이용하여 aggregation.



► **CRNN** 적용 결과 **성능 향상**

Network

❖ Attention map 적용

- 스테이션 분류 network의 output을 활용하여 attention map을 만들어 cnn feature map을 scaling
- Pre-train된 music mood tagging network를 활용
- Tagging network 학습 시 스테이션 분류 네트워크의 output을 활용하여 attention map을 만들어 cnn feature map을 scaling
  - ► **Attention map** 적용 결과 **성능 하락**

Others

- ❖ Loss
  - BCE
- ❖ Augmentation
  - Mixup, frequency random masking 적용 결과 **성능 하락**
- ❖ Optimizer
  - Adam
- ❖ Scheduler
  - Cosine Annealing

Others

❖ Ensemble

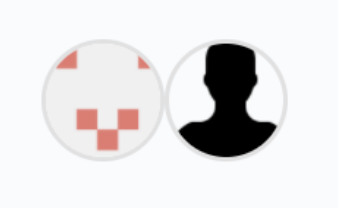
- RexNet1.0, RexNeXt50, SpeakerNet, SpeakerNet-1d

► 4개 모델 Ensemble 하여 **성능 향상**

감사합니다 ☺

Rush8. 화자 분리

Mung 팀



김승빈, 노성훈

2020. 08. 27

Table of Contents

VAD

❖ Webrtcvad 최적화

- Webrtcvad의 aggressive 정도와 end point detect를 위한 resolution과 padding-duration 값 최적화
- 공개된 test data sample을 이용해 scoring
- Webrtcvad(2), resolution=20, padding-duration=400 일때 최적화 됨

► **하이퍼파라미터 최적화** 결과 **성능 향상**

VAD

❖ Weighted Prediction Error(WPE) 적용

- 울리는 audio를 제거하기 위해 음성 인식 분야에서 주로 사용하는 반향 제거 기법
- Python 기반 WPE 오픈소스를 audio에 적용한 후 end point detector 실행
  - ► **WPE** 적용 결과 **성능 하락**

Speaker embedding extractor

❖ 짧은 발성 보상 방법 적용

- Audio를 segment하는 length가 작을수록 단일 segment에 2명 이상의 speaker가 있을 확률이 낮아질 거라 가정
- 하지만 extractor에서 speaker-specific한 특징을 추출하기 힘듬
- 짧은 segment에서 성능을 향상시키는 방법 적용
- 2020, Interspeech에 발표될 "segment aggregation for short utterances speaker verification using raw waveforms" 참고
- Test utt가 1s 일때 baseline성능 약 10% 향상

► **짧은 발성 보상 방법** 적용 결과 **성능 향상**

Clustering

❖ 다양한 Agglomerative Hierarchical Clustering(AHC) 방법 비교 적용

- 3가지 AHC clustering 방법 비교

– Single linkage 방법 : 최소 거리를 이용

– Complete linkage 방법 : 최대 거리를 이용

– Average linkage 방법 : 평균 거리를 이용

► **Average linkage** 적용 결과 **성능 향상**

Clustering

❖ Supervised clustering 방법 적용

- 2019, ICASSP에 발표된 "fully supervised speaker diarization" 논문 참고
- Bayesian non-parametric 방법인 거리 기반 차이나 레스토랑 프로세스를 이용하여 unknown number의 화자를 clustering
- Trainset을 만들기 위해 기존 Voxceleb2 데이터에서 무작위의 화자의 음성을 택하여 데이터 구성
- 하지만 predict 시간이 오래걸리는 단점을 본 논문에선 batch단위로 multiprocessing 했지만 nsml 에선 불가

  ► **Test 수행하지 못함**

Resegmentation

❖ Segments의 overlap 길이 조정

- Segment의 length를 길게 가져가서 speaker-specific한 embedding을 추출하고
- 단일 segment에 여러 명의 speaker가 있을 경우를 위해 segments의 overlap 길이를 늘려서 resegmentation 수행
  - ► **Overlap 길이** 조정 결과 **성능 향상**

감사합니다 ☺