실시간 박자 추적 모델 학습 매뉴얼

작성자 카이스트 문화기술대학원 박사과정 김종수 (실무자)

작성완료일 2025. 01. 13. (월)

문의 jongsoo.kim@kaist.ac.kr

설명에 앞서, 모델 학습의 경우에는 실제 프로그램 실행 환경인 Jetson 보드에서 수행하지 않고 NVIDIA GeForce RTX 2080 Ti 11GB와 16-core i9-9900K CPU 3.60GHz의 Ubuntu OS 환경에서 수행하였다. 따라서 이와 동일하게 Linux 환경에서 모델 학습을 수행하는 것을 권장하며, 추가로 anaconda를 설치하여 가상환경에서 실험을 진행하는 것을 권장한다.

1. 학습 데이터셋 설치

학습 데이터셋으로는 Harmonix 데이터셋을 단독으로 사용하였다. 모델의 사이즈가 작아 오히려 다량의 데이터셋으로 학습을 하게 되면 모델이 과소적합(Underfitting)이 되는 경우를 확인하였으며, 오히려 박자가 명확한 Harmonix 데이터셋 단일로 학습할 경우 더 좋은 성능이 나오는 것을 알 수 있었다. Harmonix 데이터셋은 해당 매뉴얼과 함께 제공되며, 파일에 문제가 있을 경우에는 [링크]를 통해서 음악들의 youtub url을 찾아 직접 다운로드 할 수 있다. 또한, 평가 시에는 Harmonix가 아닌 unseen 데이터인 GTZAN 데이터셋을 사용하며 해당 데이터셋 또한 해당매뉴얼과 함께 제공된다. 직접 설치를 원한다면 해당 링크를 통해 직접 설치할 수 있다.

더불어, 학습 데이터셋은 다음과 같은 형태로 반드시 저장되어 있어야 한다. (segments는 없어도 된다.) 데이터셋은 제공된 압축 파일의 model training 폴더 안에 있다.

[그림 1] 데이터셋 구조

2. 학습 환경 세팅

이는 정확히 기존의 Allin1 패키지의 환경을 따르기 때문에 Allin1 패키지의 <mark>공식 홈페이지</mark>를 참고하여도 무방하다. python 버전은 3.8을 권장한다.

1) PyTorch 설치

PyTorch의 경우, 본인의 실험 환경에 알맞은 PyTorch 버전을 설치해주면 된다. 자세한 설치 방법은 <u>공식 홈페이지</u>를 참고하여 알맞은 명령어로 설치를 진행하면 된다. 여기서 중요한 것은 반드시 "Stable" 버전으로 설치해야 하며, 자신의 GPU CUDA 버전을 알맞게 설정해야 한다.

2) NATTEN 설치

NATTEN 또한 PyTorch와 동일하게 본인의 실험 환경에 알맞은 버전으로 설치해주어야한다. 자세한 설치 방법은 <u>공식 홈페이지</u>를 참고하여 알맞은 명령어로 설치를 진행하면 된다. NATTEN도 반드시 자신의 CPU CUDA 버전을 알맞게 설정해야한다.

3) Madmom 설치

Madmom 패키지의 경우에는 개발자 버전의 설치가 필요한데 현재 공개된 개발자 버전은 다른 패키지들 간의 버전 충돌이 발생하기 때문에 반드시 함께 제공되는 Madmom 패키지 소스 코드로부터 설치해야 한다. 설치 명령어는 아래와 같다.

\$ cd madmom

\$ python setup.py develop --user

중간에 설치 오류가 발생할 수 있는데 이는 보통 numpy, scipy, cython, mido와의 버전 충돌 문제이며, 이럴 경우에는 해당 패키지들을 알맞게 재설치 후 다시 실행해주면 된다.

4) Allin1 설치

PyTorch, NATTEN, Madmom가 모두 설치되면 아래 명령어를 통해 allin1 패키지의 개발자 버전을 설치하면 된다.

\$ pip install allin1[train]

5) Wandb 회원가입 및 로그인

Allin1 패키지는 Wandb를 사용하여 모델의 학습 과정을 실시간으로 확인할 수 있다. Wandb에 회원가입 후 터미널에서 wandb에 로그인을 하면 wandb 홈페이지에서 모델의 학습 과정을 실시간으로 확인할 수 있다. 회원가입은 <mark>공식 홈페이지</mark>에 방문하여 진행하면 되고, 터미널에서의 로그인 명령어는 아래와 같다.

\$ wandb login

6) Allin1 코드 변경

아래의 경로로 이동하여 모든 파일과 디렉토리를 첨부된 new_allin1_code 폴더 안 내용물들로 변경해주면 된다.

/home/<user>/anaconda3/envs/<envname>/lib/<python3.x>/site-packages/allin1

3. 주요 Configuration 설명

제공된 new_allin1_code 폴더 안 파일들을 살펴보면 config.py 파일이 있는데 해당 파일이모델 학습에 필요한 세팅값과 각종 hyper parameter 값들이 저장되어 있다. 해당 파일에서 UnionDatasetConfig는 다량의 데이터셋으로 학습시키기 위해 작성된 코드이나 현재 사용하지 않으므로 무시해도 된다. 학습에 앞서 주요 변수들의 간단한 설명은 아래와 같다.

1) [dataclass] HarmonixConfig & GTZANConfig

[bpfed] 기존의 allin1에서 사용하던 음원 분리 대신 악기 별

sub-band pass filtering을 사용할 것인지에 대한 참거짓값. True이면 sub-band를 사용하고 False일 경우에는 음원 분리와 sub-band를 사용하지 않고 단일

채널로 모델에 음악이 입력된다.

[path_base_dir] Harmonix 혹은 GTZAN 데이터셋의 기본 경로

[path_track_dir] Harmonix 혹은 GTZAN 데이터셋의 오디오 파일들이

들어있는 경로

[path_bpf_feature_dir] Harmonix 혹은 GTZAN 데이터셋의 sub-band

spectrogram이 저장될 경로 (만약, 두 악기를 사용하면 [channel=2, Freqs, Times] 모양의 spectrogram이

저장된다.)

[path_no_demixed_feature_dir] Harmonix 혹은 GTZAN 데이터셋의 단일 채널

spectrogrma이 저장될 경로 (sub-band pass filtering이나 음원 분리를 수행하지 않는 경우)

[path_metadata] Harmonix 혹은 GTZAN 데이터셋의 메타데이터 파일

경로

[sources] bpfed가 True일 경우, sub-band pass filtering의

악기 채널 지정 (bass, drums, vocals, others 중

원하는 조합으로 지정)

2) [dataclass] Config

[project_name] wandb에 저장될 프로젝트 이름

[model] 학습하고자 하는 모델의 모델명을 작성하면 되는데,

현재까지 SOTA 모델의 이름은 "allinonetempo"

[bpf_band_dir] sub-band pass filtering에 사용되는 악기별 sub-band

데이터가 저장된 경로

[batch_size] 모델 학습 시 데이터셋의 배치 사이즈 (11GB GPU

기준으로는 256이 적합)

[loss_weight_beat] 학습 시에 beat 예측에 얼마나 가중치를 둘 것인지

[loss_weight_downbeat] 학습 시에 downbeat 예측에 얼마나 가중치를 둘

것인지

[loss_weight_tempo] 학습 시에 tempo 예측에 얼마나 가중치를 둘 것인지

[fold] 학습 시 Harmonix 데이터셋을 8등분하여 6조각은

train, 1조각은 validation, 1조각은 test에 사용되는데 이 때 가능한 8가지의 경우의 수 중에서 어떤 fold를 통해 학습할 것인지 정해주는 변수 (0~7 사이의 값을 가지며, 8 이상의 값을 설정하면 데이터 전체로 학습을

진행함)

4. 데이터 전처리 진행

Sub-band pass filtering를 진행하는지 여부에 따라 config.py 파일에서 bpfed 변수를 수정하고 데이터 전처리를 진행한다. Sub-band pass filtering 진행한 경우, 성능이 높게 나오며 [bass, drums] 조합으로 실험을 진행했을 때 높은 성능이 나왔다. 이 외에도, config.py 파일에서 앞서 언급한 경로들을 올바르게 수정한 후 전처리를 진행해야 한다. sub_band_data는 매뉴 얼과 함께 첨부된 압축 파일 내 동명의 폴더에서 찾을 수 있다. 전처리에 앞서 ffmpeg이 필요할수 있으므로 확인 후 설치할 수 있도록 한다. 전처리 실행 명령어는 아래와 같다.

\$ allin1-preprocess

5. 학습 실행 방법

config.py 파일도 올바르게 작성되어 있고 전처리가 모두 완료되면 아래 명령어를 통해 모델의 학습을 진행할 수 있으며, 모델 학습이 올바르게 시작되면 wandb에 새로운 항목이 생긴다.

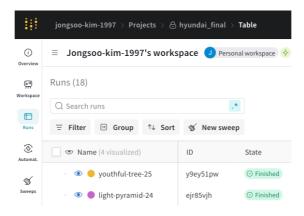
\$ allin1-train

이 때, 추가로 여러 GPU 중 특정 GPU 하나를 설정하고 싶다거나, config.py 파일을 수정하지 않고 학습 명령 시에 hyper parameter를 수정하고 싶다면 아래 예시에 같이 명령어를 실행시켜 주면 된다.

- # 특정 GPU 설정 (해당 예시는 0번 GPU로 학습할 경우)
- \$ CUDA_VISIBLE_DEVICES=0 allin1-train
- # 학습 명령 시 hyper parameter를 수정하고 싶은 경우
- \$ allin1-train fold=3 batch_size=512

6. 평가 실행 방법

학습이 모두 완료되면, 함께 제공된 별도의 test.py 파일을 통해 학습한 모델을 GTZAN 데이터 셋으로 평가할 수 있다. 평가를 실행하기에 앞서, test.py 파일에서 wandb의 project_name을 본인의 프로젝트 이름으로 변경하고 save_dir에는 평가 결과 json 파일이 저장될 경로를 작성해 준다. 그리고, RUN_ID라는 리스트에 평가를 진행하고자 하는 run_id 하나 혹은 여러개를 작성해 주면 된다. 예를 들어, 방금 학습한 모델의 run_id가 아래와 같이 확인할 수 있다.



[그림 2] wandb 프로젝트 화면 캡쳐 이미지

만약 위 그림과 같이 ID가 뜨지 않아 run_id를 확인할 수 없을 경우에는 표 우측 상단에 위치한 "Columns"를 통해서 숨겨진 "ID" 항목을 이동하여 가시화 할 수 있다. 이렇게 확인된 run_id는 RUN_ID에 작성해주면 된다. 평가 결과는 save_dir에 작성한 경로에서 확인할 수 있으며, beat와 downbeat의 threshold 값 별로 평가 결과가 저장이 된다. 후처리 방식이 max pooling이 아닌 DBN을 사용한 경우에 대한 평가 결과도 함께 저장된다.

평가 프로그램인 test.py 파일을 실행시키는 방법은 아래와 같다.

\$ python test.py

7. 모델 추출 방법

모델 학습이 완료되면 test.py 파일과 함께 제공되는 weight_download.py 파일을 통해 모델의 체크포인트와 config 파일을 저장할 수 있다. 실행하기에 앞서 먼저 해당 파일을 열어 몇 가지 변수를 수정해야 한다. artifacts_path는 학습과 평가를 실행한 위치에 생긴 artifacts라는 디렉토리의 경로를 의미한다. save_dir은 체크포인트와 config 파일을 저장할 디렉토리의 경로를 작성하면 된다. run_ids에는 다운로드 받고자 하는 모델의 run_id를 작성해주면 된다. 작성이 완료되면 아래와 같은 명령어를 통해 다운로드 프로그램을 실행하며 된다.

\$ python weight_download.py

모델의 추출이 모두 완료되면 체크포인트와 config 파일을 Jetson 보드로 옮겨 주고 pretrained_dir의 경로를 재설정해준 다음 프로그램을 실행시켜주면 된다.