

tensorflowTTS 설치 및 실행

(<https://github.com/TensorSpeech/TensorFlowTTS>)

(<https://enchiridion.tistory.com/69>)

Features

- High performance on Speech Synthesis.
- Be able to fine-tune on other languages.
- Fast, Scalable, and Reliable.
- Suitable for deployment.
- Easy to implement a new model, based-on abstract class.
- Mixed precision to speed-up training if possible.
- Support Single/Multi GPU gradient Accumulate.
- Support both Single/Multi GPU in base trainer class.
- TFlite conversion for all supported models.
- Android example.
- Support many languages (currently, we support Chinese, Korean, English, French and German)
- Support C++ inference.
- Support Convert weight for some models from PyTorch to TensorFlow to accelerate speed.

환경

Python 3.7+	
Cuda 10.1	
CuDNN 7.6.5	
Tensorflow 2.2/2.3/2.4/2.5/2.6	# 또는 tensorflow-gpu 2.6.0
Tensorflow Addons >= 0.10.0	

c:

cd \ai

python 3.7로 가상환경 생성

conda create -n tensorflowtts git python=3.7

소스 git

git clone <https://github.com/TensorSpeech/TensorFlowTTS.git>

소스 폴더로 이동

cd tensorflowtts

가상환경 활성화

conda activate tensorflowtts

관련 패키지 설치 (setup.py 실행)

pip install .

HDF5(계층적데이터형식, 빅데이터 처리를 위한 파일 형식)을 python에서 사용하기 위한 모듈 설치
numba 버전 업그레이드 (numba-0.53.1-py37hf11a4ad_0)

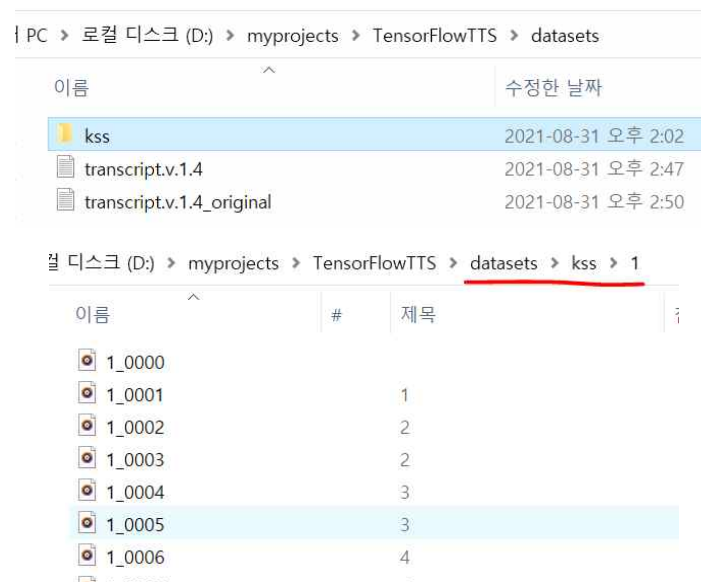
conda install h5py numba

pip install tensorflow-addons=0.8.3

#약어, 숫자, 타임 스탬프 등을 포함한 독일어 텍스트를 정리하고
음역 (예 : 정규화)하는 Python 모듈 설치

pip install git+https://github.com/repodiac/german_transliterate

데이터 구조



이름	#	제목
1_0000		
1_0001	1	
1_0002	2	
1_0003	2	
1_0004	3	
1_0005	3	
1_0006	4	

Preprocessing

The preprocessing has two steps:

1. Preprocess audio features

- Convert characters to IDs
- Compute mel spectrograms
- Normalize mel spectrograms to $[-1, 1]$ range
- Split the dataset into train and validation
- Compute the mean and standard deviation of multiple features from the **training** split

2. Standardize mel spectrogram based on computed statistics

전처리 데이터 정리- 첫 번째 줄 삭제 - 전처리 시 에러 방지

transcript.v.1.4 - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

1/1_0000.wav|그는 관창은 척하려고 애쓰는 것 같았다.|그는 관창은 척하려고
1/1_0001.wav|그녀의 사랑을 얻기 위해 애썼지만 헛수고였다.|그녀의 사랑을
1/1_0002.wav|용돈을 아껴 써라.|용돈을 아껴 써라.|
1/1_0003.wav|그는 아내를 많이 아낀다.|그는 아내를 많이 아낀다.|
1/1_0004.wav|그 애 전화번호 알아?|그 애 전화번호 알아?|
1/1_0005.wav|차에 대해 잘 아세요?|차에 대해 잘 아세요?|
1/1_0006.wav|거기 도착하면 나한테 알려 줘.|거기 도착하면 나한테 알려 줘.|
1/1_0007.wav|그들은 내가 시험에 떨어졌다고 알려 왔다.|그들은 내가 시험에
1/1_0008.wav|나는 살아오면서 감기를 앓은 적이 한 번도 없다.|나는 살아오면
1/1_0009.wav|사흘 동안 심하게 몸살을 앓았어요.|사흘 동안 심하게 몸살을 앓

[1] 전처리

```
tensorflow-tts-preprocess --rootdir ./datasets --outdir ./datasets/dump_kss --config
preprocess/kss_preprocess.yaml --dataset kss
```

```
(tensorflowtts) D:\myprojects\TensorFlowTTS>tensorflow-tts-preprocess --rootdir ./datasets --outdir ./dump_kss --config
preprocess/kss_preprocess.yaml --dataset kss
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data] C:\Users\pkkim\AppData\Roaming\nltk_data...
[nltk_data] Unzipping taggers\averaged_perceptron_tagger.zip.
[nltk_data] Downloading package cmudict to
[nltk_data] C:\Users\pkkim\AppData\Roaming\nltk_data...
[nltk_data] Unzipping corpora\cmudict.zip.
2021-08-31 14:22:03,082 (preprocess:378) INFO: Selected 'kss' processor.
2021-08-31 14:22:03,183 (preprocess:417) INFO: Training items: 12211
2021-08-31 14:22:03,184 (preprocess:418) INFO: Validation items: 643
[Preprocessing train]: 3% | 355/12211 [00:30<16:37, 11.89it/s]W
ARNING:root:1_0922 causes clipping. It is better to reconsider global gain scale value.
[Preprocessing train]: 3% | 411/12211 [00:34<24:36, 7.99it/s]W
ARNING:root:1_0014 causes clipping. It is better to reconsider global gain scale value.
[Preprocessing train]: 4% | 468/12211 [00:38<10:49, 18.07it/s]W
ARNING:root:1_0554 causes clipping. It is better to reconsider global gain scale value.
[Preprocessing train]: 5% | 561/12211 [00:44<17:30, 11.09it/s]W
ARNING:root:1_0696 causes clipping. It is better to reconsider global gain scale value.
[Preprocessing train]: 5% | 627/12211 [00:48<07:57, 24.27it/s]W
ARNING:root:1_0668 causes clipping. It is better to reconsider global gain scale value.
[Preprocessing train]: 9% | 1105/12211 [01:20<06:48, 27.18it/s]
```

.....

```
[Preprocessing train]: 94% | 11448/12210 [13:03<00:37, 20.39it/s]W
ARNING:root:1_0668 causes clipping. It is better to reconsider global gain scale value.
[Preprocessing train]: 98% | 11914/12210 [13:34<00:11, 26.88it/s]W
ARNING:root:2_0609 causes clipping. It is better to reconsider global gain scale value.
[Preprocessing train]: 100% | 12210/12210 [13:54<00:00, 14.63it/s]
2021-08-31 15:01:33,056 (preprocess:478) INFO: Saving computed statistics.
[Preprocessing valid]: 7% | 48/643 [00:01<00:18, 32.92it/s]W
ARNING:root:1_0654 causes clipping. It is better to reconsider global gain scale value.
[Preprocessing valid]: 9% | 55/643 [00:02<00:50, 11.54it/s]W
ARNING:root:1_0386 causes clipping. It is better to reconsider global gain scale value.
[Preprocessing valid]: 100% | 643/643 [00:41<00:00, 15.49it/s]
```

PC > 로컬 디스크 (D:) > myprojects > TensorFlowTTS > datasets

▼

🔄

datasets 검색

이름	수정한 날짜	유형	크기
kss	2021-08-31 오후 2:02	파일 폴더	
transcript.v.1.4	2021-08-31 오후 2:47	텍스트 문서	3,225KB
transcript.v.1.4_original	2021-08-31 오후 2:50	텍스트 문서	3,225KB
<u>dump_kss</u>	2021-08-31 오후 3:01	파일 폴더	

[2] Normalize

```
tensorflow-tts-normalize --rootdir ./datasets/dump_kss --outdir ./datasets/dump_kss --config preprocess/kss_preprocess.yaml --dataset kss
```

컬 디스크 (D:) > myprojects > TensorFlowTTS > datasets > dump_kss dump_kss 검색

이름	수정한 날짜	유형	크기
train	2021-08-31 오후 3:19	파일 폴더	
valid	2021-08-31 오후 3:19	파일 폴더	
kss_mapper	2021-08-31 오후 2:47	JSON 파일	3KB
stats.npy	2021-08-31 오후 3:01	NPY 파일	1KB
stats_energy.npy	2021-08-31 오후 3:01	NPY 파일	1KB
stats_f0.npy	2021-08-31 오후 3:01	NPY 파일	1KB
train_utt_ids.npy	2021-08-31 오후 2:47	NPY 파일	287KB
valid_utt_ids.npy	2021-08-31 오후 2:47	NPY 파일	16KB

```
(tensorflowtts) D:\myprojects\TensorFlowTTS>tensorflow-tts-normalize --rootdir ./datasets/dump_kss --outdir ./datasets/dump_kss --config preprocess/kss_preprocess.yaml --dataset kss
2021-08-31 15:19:35.645 (preprocess:531) INFO: Files to normalize: 12946
[Normalizing]: 100% | 12946/12946 [02:55<00:00, 73.74it/s]
```

myprojects > TensorFlowTTS > datasets > dump_kss > train train

이름	수정한 날짜	유형
ids	2021-08-31 오후 3:01	파일 폴더
norm-feats ✓	2021-08-31 오후 3:23	파일 폴더
raw-energies	2021-08-31 오후 3:01	파일 폴더
raw-f0	2021-08-31 오후 3:01	파일 폴더
raw-feats	2021-08-31 오후 3:01	파일 폴더
wavs	2021-08-31 오후 3:01	파일 폴더

cudnn64_8.dll 파일 못찾는다는 에러 방지 위해 cudnn 설치

```
conda install cudnn cudatoolkit
```

pretrained model 복사

컬 디스크 (D:) > myprojects > TensorFlowTTS > model-20210831T050255Z-001 > model > tacotron2.v1

이름	수정한 날짜	유형	크기
model-40000.h5	2020-07-17 오후 8:07	H5 파일	124,835KB
✓ model-100000.h5	2020-07-20 오전 4:34	H5 파일	124,835KB

| PC > 로컬 디스크 (D:) > myprojects > TensorFlowTTS

이름	수정된 날짜	유형	크기
tacotron-100000.h5	2020-07-20 오전 4:34	H5 파일	124,835KB
README	2021-08-30 오후 9:51	MD 파일	23KB
LICENSE	2021-08-30 오후 9:51	파일	12KB
setup	2021-08-30 오후 9:51	Python 원본 파일	4KB
.gitignore	2021-08-30 오후 9:51	Git Ignore 원본 파일	1KB
dockerfile	2021-08-30 오후 9:51	파일	1KB
3.10	2021-08-30 오후 11:16	10 파일	1KB
setup	2021-08-30 오후 9:51	CFG 파일	1KB
docker-compose	2021-08-30 오후 9:51	Yaml 원본 파일	1KB
.gitattributes	2021-08-30 오후 9:51	Git Attributes 원본 ...	1KB
.git	2021-08-30 오후 9:51	파일 폴더	
.github	2021-08-30 오후 9:51	파일 폴더	
datasets	2021-08-31 오후 3:18	파일 폴더	

GPU 메모리 할당 문제 해결을 위하여 extract_duration.py에 소스 추가

```
extract_duration.py 4 ●
D: > myprojects > TensorFlowTTS > examples > tacotron2 > extract_duration.py > ...

21 import sys
22
23 sys.path.append(".")
24
25 import matplotlib.pyplot as plt
26 import numpy as np
27 import tensorflow as tf
28 import yaml
29 from tqdm import tqdm
30
31 from examples.tacotron2.tacotron_dataset import CharacterMelDataset
32 from tensorflow_tts.configs import Tacotron2Config
33 from tensorflow_tts.models import TFTacotron2
34
35 ### added by pk
36 config = tf.compat.v1.ConfigProto()
37 config.gpu_options.allow_growth = True
38 config.log_device_placement = True
39 session = tf.compat.v1.Session(config=config)
40 ###
41
42 @jit(nopython=True)
```

Duration 추출 for train

```
python examples/tacotron2/extract_duration.py --rootdir ./datasets/dump_kss/train/ --outdir
./datasets/dump_kss/train/durations/ --checkpoint ./tacotron2-100k.h5 --use-norm 1
--config ./examples/tacotron2/conf/tacotron2.kss.v1.yaml --batch-size 32
```

```
(tensorflowtts) D:\myprojects\TensorFlowTTS>python examples/tacotron2/extract_duration.py --rootdir ./datasets/dump_kss/train/ --outdir ./datasets/dump_kss/train/durations/ --checkpoint ./tacotron-100000.h5 --use-norm 1 --config ./examples/tacotron2/conf/tacotron2.kss.v1.yaml --batch-size 8
2021-09-03 23:42:21.961401: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:
AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-09-03 23:42:22.937375: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 9627 MB memory: -> device: 0, name: GeForce GTX 1080 Ti, pci bus id: 0000:01:00.0, compute capability: 6.1
2021-09-03 23:42:22.940428: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/device:GPU:1 with 9627 MB memory: -> device: 1, name: GeForce GTX 1080 Ti, pci bus id: 0000:02:00.0, compute capability: 6.1
2021-09-03 23:42:22.942435: I tensorflow/core/common_runtime/direct_session.cc:361] Device mapping:
/job:localhost/replica:0/task:0/device:GPU:0 -> device: 0, name: GeForce GTX 1080 Ti, pci bus id: 0000:01:00.0, compute capability: 6.1
/job:localhost/replica:0/task:0/device:GPU:1 -> device: 1, name: GeForce GTX 1080 Ti, pci bus id: 0000:02:00.0, compute capability: 6.1
2021-09-03 23:42:32.745613: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 9627 MB memory: -> device: 0, name: GeForce GTX 1080 Ti, pci bus id: 0000:01:00.0, compute capability: 6.1
2021-09-03 23:42:32.746275: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/device:GPU:1 with 9627 MB memory: -> device: 1, name: GeForce GTX 1080 Ti, pci bus id: 0000:02:00.0, compute capability: 6.1
2021-09-03 23:42:33.492938: I tensorflow/stream_executor/cuda/cuda_dnn.cc:369] Loaded cuDNN version 8201
[Extract Duration]: 0it [00:00, ?it/s]2021-09-03 23:42:37.524929: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)
[Extract Duration]: 1538it [55:09, 2.15s/it]
```

← → ↕ ↗ > 내 PC > 로컬 디스크 (D:) > myprojects > TensorFlowTTS > datasets > dump_kss > train > durations

이름	수정한 날짜	유형	크기
1_0001-durations.npy	2021-09-03 오후 11:42	NPY 파일	1KB
1_0002-durations.npy	2021-09-03 오후 11:42	NPY 파일	1KB
1_0003-durations.npy	2021-09-03 오후 11:42	NPY 파일	1KB
1_0004-durations.npy	2021-09-03 오후 11:42	NPY 파일	1KB
1_0005-durations.npy	2021-09-03 오후 11:42	NPY 파일	1KB
1_0006-durations.npy	2021-09-03 오후 11:42	NPY 파일	1KB

12,303개 항목

※ numba 패키지 버전 맞지 않을 경우의 오류 메시지

```
Traceback (most recent call last):
  File "examples/tacotron2/extract_duration.py", line 244, in <module>
    main()
  File "examples/tacotron2/extract_duration.py", line 188, in main
    d = get_duration_from_alignment(alignment) # [max_char_len]
TypeError: expected dtype object, got 'numpy.dtype[int64]'
```

Duration 추출 for valid

```
python examples/tacotron2/extract_duration.py --rootdir ./datasets/dump_kss/valid/ --outdir
./datasets/dump_kss/valid/durations/ --checkpoint ./tacotron2-100k.h5 --use-norm 1
--config ./examples/tacotron2/conf/tacotron2.kss.v1.yaml --batch-size 32
```

```
[Extract Duration]: 0it [00:00, ?it/s]2021-09-04 09:43:19.592115: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)
[Extract Duration]: 81it [02:16, 1.68s/it]
```

← → ↕ ↑ > 내 PC > 로컬 디스크 (D:) > myprojects > TensorFlowTTS > datasets > dump_kss > valid > durations

	이름	수정한 날짜	유형	크기
> 동영상				
> 문서				
> 바탕 화면				
> 사진				
> 음악				
> 로컬 디스크 (C:)				
> 로컬 디스크 (D:)				
643개 항목				

이름	수정한 날짜	유형	크기
1_0009-durations.npy	2021-09-04 오전 9:43	NPY 파일	1KB
1_0048-durations.npy	2021-09-04 오전 9:43	NPY 파일	1KB
1_0051-durations.npy	2021-09-04 오전 9:43	NPY 파일	1KB
1_0064-durations.npy	2021-09-04 오전 9:43	NPY 파일	1KB
1_0084-durations.npy	2021-09-04 오전 9:43	NPY 파일	1KB
1_0100-durations.npy	2021-09-04 오전 9:43	NPY 파일	1KB
1_0102-durations.npy	2021-09-04 오전 9:43	NPY 파일	1KB

학습을 위하여 GPU 여부(갯수) 설정

```

extract_duration.py 4  execute.py 7  train_tacotron2.py 8 X
D: > myprojects > TensorFlowTTS > examples > tacotron2 > train_tacotron2.py > ...

17  """Train Tacotron2."""
18  import os
19  os.environ["CUDA_VISIBLE_DEVICES"]="1" # CPU only일 경우는 "-1"
20
21  import tensorflow as tf
22
23
24  physical_devices = tf.config.list_physical_devices("GPU")

D: > myprojects > TensorFlowTTS > examples > tacotron2 > train_tacotron2.py > ...

15
16
17  """Train Tacotron2."""
18  import os
19  os.environ["CUDA_VISIBLE_DEVICES"]="0,1" # CPU only일 경우는 "-1", gpu 1개는 "0"
20
21  import tensorflow as tf
22

```

원할한 학습을 위하여 tacotron2.kss.v1.yaml 파일 내 파라미터 조정

```
extract_duration.py 4  execute.py 7  train_tacotron2.py 8  ! tacotron2.kss.v1.yaml X
D: > myprojects > TensorFlowTTS > examples > tacotron2 > conf > ! tacotron2.kss.v1.yaml

47 #####
48 # DATA LOADER SETTING #
49 #####
50 #batch_size: 32 # Batch size for each GPU with assuming that gradient_accumulation_steps is 1.
51 batch_size: 8 # Batch size for each GPU with assuming that gradient_accumulation_steps is 1.
52 remove_short_samples: true # Whether to remove samples the length of which are less than batch_size.
53 allow_cache: true # Whether to allow cache in dataset. If true, it requires cpu memory.
54 mel_length_threshold: 32 # remove all targets has mel_length <= 32
55 is_shuffle: true # shuffle dataset after each epoch.
56 use_fixed_shapes: true # use fixed_shapes for training (2x speed-up)
57 # refer (https://github.com/dathudeptrai/TensorFlowTTS/issues/34#issuecomment-478111111)
58 #####
59 #####
60 # OPTIMIZER & SCHEDULER SETTING #
61 #####
62 optimizer_params:
63     initial_learning_rate: 0.001
64     end_learning_rate: 0.00001
65     decay_steps: 150000 # < train_max_steps is recommend.
66     warmup_proportion: 0.02
67     weight_decay: 0.001
68 #####
69 gradient_accumulation_steps: 1
70 var_train_expr: null # trainable variable expr (eg. 'embeddings/decoder_cell' )
71 # must separate by |. if var_train_expr is null then we
72 # training all variable
73 #####
74 # INTERVAL SETTING #
75 #####
76 #train_max_steps: 200000 # Number of training steps.
77 train_max_steps: 20 # Number of training steps.
78 save_interval_steps: 2000 # Interval steps to save checkpoint.
79 eval_interval_steps: 500 # Interval steps to evaluate the network.
80 log_interval_steps: 200 # Interval steps to record the training log.
81 start_schedule_teacher_forcing: 200001 # don't need to apply schedule teacher forcing.
82 start_ratio_value: 0.5 # start ratio of scheduled teacher forcing.
83 schedule_decay_steps: 50000 # decay step scheduled teacher forcing.
84 end_ratio_value: 0.0 # end ratio of scheduled teacher forcing.
85 #####
86 # OTHER SETTING #
87 #####
88 #num_save_intermediate_results: 1 # Number of results to be saved as intermediate results.
89 num_save_intermediate_results: 5 # Number of results to be saved as intermediate results.
90
```

학습

```
python examples/tacotron2/train_tacotron2.py --train-dir ./datasets/dump_kss/train/
--dev-dir ./datasets/dump_kss/valid/ --outdir
./examples/tacotron2/exp/train.tacotron2.kss.v1/ --config
./examples/tacotron2/conf/tacotron2.kss.v1.yaml --use-norm 1 --pretrained
./tacotron2-100000.h5
```

```
(tensorflow/tts) D:\myprojects\TensorFlowTTS>python examples/tacotron2/train_tacotron2.py --train-dir ./datasets/dump_kss/
/train/ --dev-dir ./datasets/dump_kss/valid/ --outdir ./examples/tacotron2/exp/train_tacotron2.kss.v1/ --config ./examples/
tacotron2/conf/tacotron2.kss.v1.yaml --use-norm 1 --pretrained ./tacotron-100000.h5
2021-09-04 10:56:42.789255: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with
oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:
AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-09-04 10:56:43.308039: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 9627 MB memory: -> device: 0, name: GeForce GTX 1080 Ti, pci bus id: 0000:02:00.0, compute capability: 6.1
2021-09-04 10:56:52.956 (train_tacotron2:431) INFO: hop_size = 256
2021-09-04 10:56:52.957 (train_tacotron2:431) INFO: format = npy
2021-09-04 10:56:52.958 (train_tacotron2:431) INFO: model_type = tacotron2
2021-09-04 10:56:52.959 (train_tacotron2:431) INFO: tacotron2_params = {'dataset': 'kss', 'embedding_hidden_size': 512, 'initializer_range': 0.02, 'embedding_dropout_prob': 0.1, 'n_speakers': 1, 'n_conv_encoder': 5, 'encoder_conv_filters': 512, 'encoder_conv_kernel_sizes': 5, 'encoder_conv_activation': 'relu', 'encoder_conv_dropout_rate': 0.5, 'encoder_lstm_units': 256, 'n_prenet_layers': 2, 'prenet_units': 256, 'prenet_activation': 'relu', 'prenet_dropout_rate': 0.5, 'n_lstm_decoder': 1, 'reduction_factor': 1, 'decoder_lstm_units': 1024, 'attention_dim': 128, 'attention_filters': 32, 'attention_kernel': 31, 'n_mels': 80, 'n_conv_postnet': 5, 'postnet_conv_filters': 512, 'postnet_conv_kernel_sizes': 5, 'postnet_dropout_rate': 0.1, 'attention_type': 'lsa'}
2021-09-04 10:56:52.961 (train_tacotron2:431) INFO: batch_size = 32
2021-09-04 10:56:52.961 (train_tacotron2:431) INFO: remove_short_samples = True
2021-09-04 10:56:52.962 (train_tacotron2:431) INFO: allow_cache = True
2021-09-04 10:56:52.962 (train_tacotron2:431) INFO: mel_length_threshold = 32
2021-09-04 10:56:52.963 (train_tacotron2:431) INFO: is_shuffle = True
2021-09-04 10:56:52.964 (train_tacotron2:431) INFO: use_fixed_shapes = True
2021-09-04 10:56:52.964 (train_tacotron2:431) INFO: optimizer_params = {'initial_learning_rate': 0.001, 'end_learning_rate': 1e-05, 'decay_steps': 150000, 'warmup_proportion': 0.02, 'weight_decay': 0.001}
2021-09-04 10:56:52.965 (train_tacotron2:431) INFO: gradient_accumulation_steps = 1
2021-09-04 10:56:52.966 (train_tacotron2:431) INFO: var_train_expr = None
2021-09-04 10:56:52.966 (train_tacotron2:431) INFO: train_max_steps = 200000
2021-09-04 10:56:52.967 (train_tacotron2:431) INFO: save_interval_steps = 2000
2021-09-04 10:56:52.967 (train_tacotron2:431) INFO: eval_interval_steps = 500
2021-09-04 10:56:52.968 (train_tacotron2:431) INFO: log_interval_steps = 200
2021-09-04 10:56:52.968 (train_tacotron2:431) INFO: start_schedule_teacher_forcing = 200001
2021-09-04 10:56:52.969 (train_tacotron2:431) INFO: start_ratio_value = 0.5
2021-09-04 10:56:52.969 (train_tacotron2:431) INFO: schedule_decay_steps = 50000
2021-09-04 10:56:52.970 (train_tacotron2:431) INFO: end_ratio_value = 0.0
2021-09-04 10:56:52.970 (train_tacotron2:431) INFO: num_save_intermediate_results = 1
2021-09-04 10:56:52.974 (train_tacotron2:431) INFO: train_dir = ./datasets/dump_kss/train/
2021-09-04 10:56:52.975 (train_tacotron2:431) INFO: dev_dir = ./datasets/dump_kss/valid/
2021-09-04 10:56:52.975 (train_tacotron2:431) INFO: use_norm = True
2021-09-04 10:56:52.976 (train_tacotron2:431) INFO: outdir = ./examples/tacotron2/exp/train_tacotron2.kss.v1/
2021-09-04 10:56:52.976 (train_tacotron2:431) INFO: config = ./examples/tacotron2/conf/tacotron2.kss.v1.yaml
2021-09-04 10:56:52.977 (train_tacotron2:431) INFO: resume =
2021-09-04 10:56:52.977 (train_tacotron2:431) INFO: verbose = 1
2021-09-04 10:56:52.978 (train_tacotron2:431) INFO: mixed_precision = False
2021-09-04 10:56:52.978 (train_tacotron2:431) INFO: pretrained = ./tacotron-100000.h5
2021-09-04 10:56:52.979 (train_tacotron2:431) INFO: version = 0.0
2021-09-04 10:56:52.980 (train_tacotron2:431) INFO: max_mel_length = 663
2021-09-04 10:56:52.980 (train_tacotron2:431) INFO: max_char_length = 105
2021-09-04 10:56:54.461822: I tensorflow/stream_executor/cuda/cuda_dnn.cc:369] Loaded cuDNN version 8201
Model: "tacotron2"
```

Layer (type)	Output Shape	Param #
encoder (TFTacotronEncoder)	multiple	8183808
decoder_cell (TFTacotronDeco)	multiple	18246402
post_net (TFTacotronPostnet)	multiple	5460480
residual_projection (Dense)	multiple	41040
Total params: 31,931,730		
Trainable params: 31,921,490		
Non-trainable params: 10,240		

```
2021-09-04 10:56:58.533 (train_tacotron2:476) INFO: Successfully loaded pretrained weight from ./tacotron-100000.h5.
[train]: 0% | 0/20 [00:00<?, ?it/s]
2021-09-04 11:39:31.522079: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimization
Passes are enabled (registered 2)
[train]: 20% | 4/20 [00:46<02:21, 8.85s/it]
```

내 PC > 로컬 디스크 (D:) > myprojects > TensorFlowTTS > examples > tacotron2 > exp > train.tacotron2.kss.v1

이름	수정한 날짜	유형	크기
checkpoints	2021-09-04 오전 9:54	파일 폴더	
config	2021-09-04 오전 10:56	Yaml 원본 파일	2KB
events.out.tfevents.1630716828.AILAB.41144.0.v2	2021-09-04 오전 9:53	V2 파일	1KB
events.out.tfevents.1630716879.AILAB.16212.0.v2	2021-09-04 오전 9:54	V2 파일	1KB
events.out.tfevents.1630717961.AILAB.25316.0.v2	2021-09-04 오전 10:12	V2 파일	1KB
events.out.tfevents.1630718565.AILAB.23724.0.v2	2021-09-04 오전 10:22	V2 파일	1KB
events.out.tfevents.1630719717.AILAB.26660.0.v2	2021-09-04 오전 10:41	V2 파일	1KB
events.out.tfevents.1630720484.AILAB.26372.0.v2	2021-09-04 오전 10:54	V2 파일	1KB
events.out.tfevents.1630720613.AILAB.32064.0.v2	2021-09-04 오전 10:56	V2 파일	1KB

음성 합성

(1/2)synth.py 수정

D: > myprojects > TensorFlowTTS > synth.py > ...

```
15
16 class VoiceSynthesis:
17     # 모델 초기화
18     def __init__(self):
19         # gpu memory의 1/3 만큼 할당하기로 제한
20         gpu_options = tf.compat.v1.GPUOptions(per_process_gpu_memory_fraction=0.800)
21         conf = tf.compat.v1.ConfigProto(gpu_options=gpu_options)
22         # 탄력적인 메모리 할당
23         # conf.gpu_options.allow_growth = True
24         session = tf.compat.v1.Session(config=conf)
25
26         # tacotron 설정, 학습된 모델 가져오기
27         module_path = os.path.dirname(os.path.abspath(__file__))
28         tacotron2_config = AutoConfig.from_pretrained(os.path.join(module_path, './examples/tacotron2/conf/tacotron2.kss.v1.yaml'))
29         self.tacotron2 = TFAutoModel.from_pretrained(
30             config=tacotron2_config,
31             pretrained_path=os.path.join(module_path, "tacotron2-100k.h5"),
32             name="tacotron2"
33         )
34
35         # fast speech 설정, 학습된 모델 가져오기
36         fastspeech2_config = AutoConfig.from_pretrained(os.path.join(module_path, './examples/fastspeech2/conf/fastspeech2.kss.v1.yaml'))
37         self.fastspeech2 = TFAutoModel.from_pretrained(
38             config=fastspeech2_config,
39             pretrained_path=os.path.join(module_path, "fastspeech2-200k.h5"),
40             name="fastspeech2"
41         )
42
43         # mel gan 설정, 학습된 모델 가져오기
44         mb_melgan_config = AutoConfig.from_pretrained(os.path.join(module_path, './examples/multiband_melgan/conf/multiband_melgan.v1.yaml'))
45         self.mb_melgan = TFAutoModel.from_pretrained(
46             config=mb_melgan_config,
47             pretrained_path=os.path.join(module_path, "mb_melgan-1000k.h5"),
48             name="mb_melgan"
49         )
50
51         #processor - 글자 별 상응하는 숫자의 mapper 설정 가져오기
52         self.processor = AutoProcessor.from_pretrained(pretrained_path=os.path.join(module_path, "kss_mapper.json"))
53
```

⌂ 주의 요함 | ailab.silla.ac.kr/lec/pipaai/

opencv-데이터 사... opencv강좌 컨볼루션 신경망 레... 영상처리

- [tensorflowTTS 설치 및 실행 실습강의동영상\(5-16\)](#)
- [kss 한국어 음성 데이터\(egg 압축\)](#)
- [tacotron2 모델파일 \(tacotron2-100000.h5 zip 압축\)](#)
- [synth.py 실행관련파일들 \(zip 압축\)](#)
- [Audio Deepfake](#)

PC > 로컬 디스크 (D:) > myprojects > TensorFlowTTS

이름	수정한 날짜	유형	크기
.gitignore	2021-08-31 오후 4:40	파일 폴더	
datasets	2021-08-30 오후 9:51	파일 폴더	
examples	2021-08-31 오후 2:10	파일 폴더	
model-20210831T050255Z-001	2021-08-30 오후 9:51	파일 폴더	
notebooks	2021-08-31 오후 2:11	파일 폴더	
output	2021-09-05 오후 7:47	파일 폴더	
preprocess	2021-08-31 오후 2:11	파일 폴더	
samples-20210831T050251Z-001	2021-09-05 오후 6:53	파일 폴더	
tensorflow_tts	2021-08-30 오후 9:51	파일 폴더	
test	2021-08-30 오후 9:51	파일	
.gitattributes	2021-08-30 오후 9:51	Git Attributes 원본 ...	1KB
.gitignore	2021-08-30 오후 9:51	Git Ignore 원본 파일	1KB
3.10	2021-08-30 오후 11:16	10 파일	1KB
docker-compose.yml	2021-08-30 오후 9:51	Yaml 원본 파일	1KB
dockerfile	2021-08-30 오후 9:51	파일	1KB
fastspeech2-200k.h5	2020-07-26 오후 9:02	H5 파일	121,769KB
kss_mapper.json	2021-09-05 오후 7:25	JSON 파일	3KB
LICENSE	2021-08-30 오후 9:51	파일	12KB
mb.melgan-1000k.h5	2020-07-16 오전 1:58	H5 파일	9,995KB
README.md	2021-08-30 오후 9:51	MD 파일	23KB
setup.cfg	2021-08-30 오후 9:51	CFG 파일	1KB
setup.py	2021-09-05 오후 9:25	Python 원본 파일	4KB
synth.py	2021-09-05 오후 7:46	Python 원본 파일	7KB
tacotron2-100k.h5	2020-07-20 오전 4:34	H5 파일	124,835KB
tacotron-100000.h5	2020-07-20 오전 4:34	H5 파일	124,835KB

문장출력용 폴더 생성

파일명변경 및 복사

기존 학습 모델 파일 저장

```

if __name__ == "__main__":

    tts = VoiceSynthesis()
    start = time.time() # 시작 시간 저장
    input_text = "신은 우리의 수학 문제에는 관심이 없다. 신은 다만 경험적으로 통합할 뿐이다."
    input_text = "간장 공장 공장장은 강 공장장이고, 된장공장 공장장은 공 공장장이다."
    input_text = "고객님, 총 금액은 7528원 입니다. 더 주문하시겠습니까?"
    print(tts.text_to_voice(input_text))
    print("time :", time.time() - start) # 현재시각 - 시작시간 = 실행 시간

```

한국어 음성 합성 실행
결과는 output 폴더에 저장

python synth.py

| PC > 로컬 디스크 (D:) > myprojects > TensorFlowTTS > output

이름	#	제목
 20210905_192758_215...		
 20210905_192931_849...		
 20210905_193008_860...		
 20210905_193203_844...		
 20210905_194205_657...		
 20210905_194415_731...		
 20210905_194511_454...		
 20210905_194656_925...		