

Speech Enhancement Tutorial

Speechless

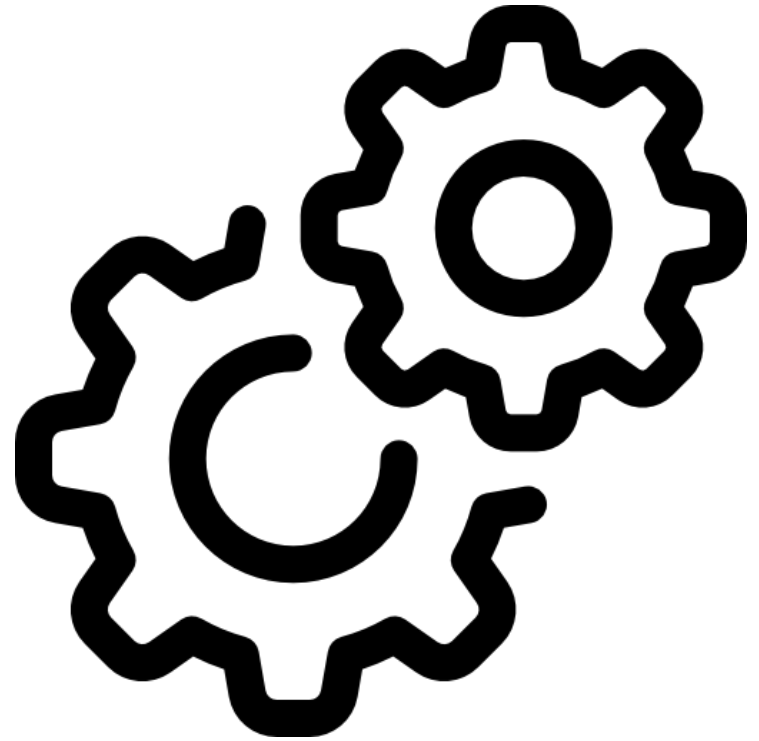
김선준, 안희영, 명수환, 최지훈

Supervised and helped by C-J Lee, S-R Hwang,
and H-U Yoon



환경 설정

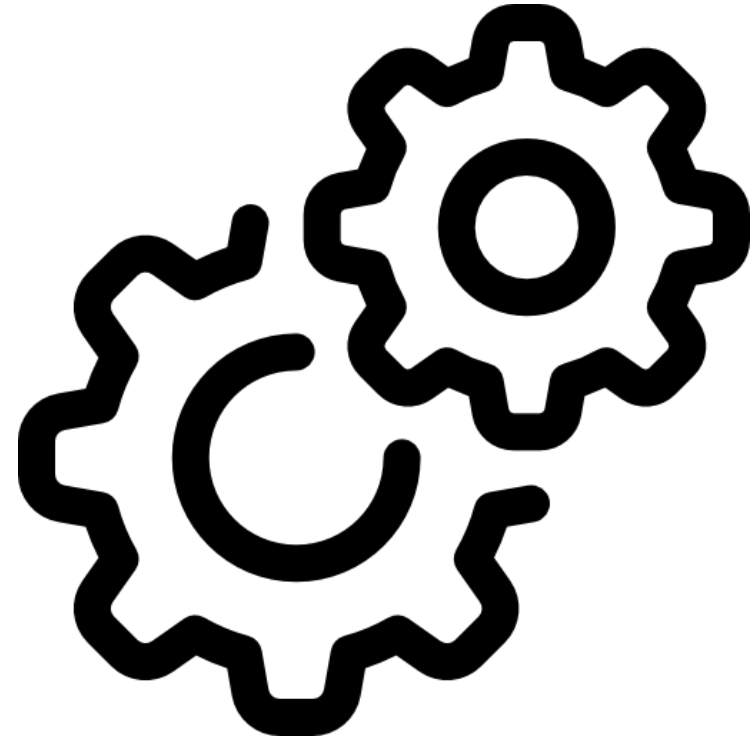
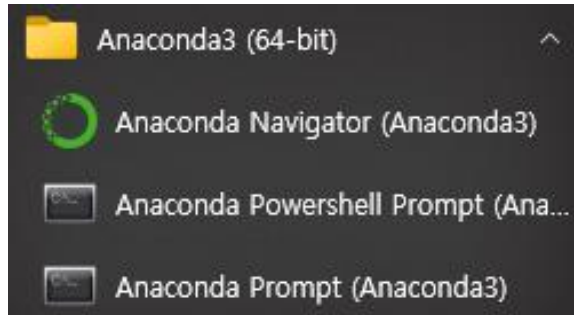
- tensorflow 2.7.0
- CUDA 11.2
- cuDNN 8.1.1
- tensorboard 2.7.0
- numpy 1.20.3
- pesq 0.0.3
- pystoi 0.3.3



환경 설정

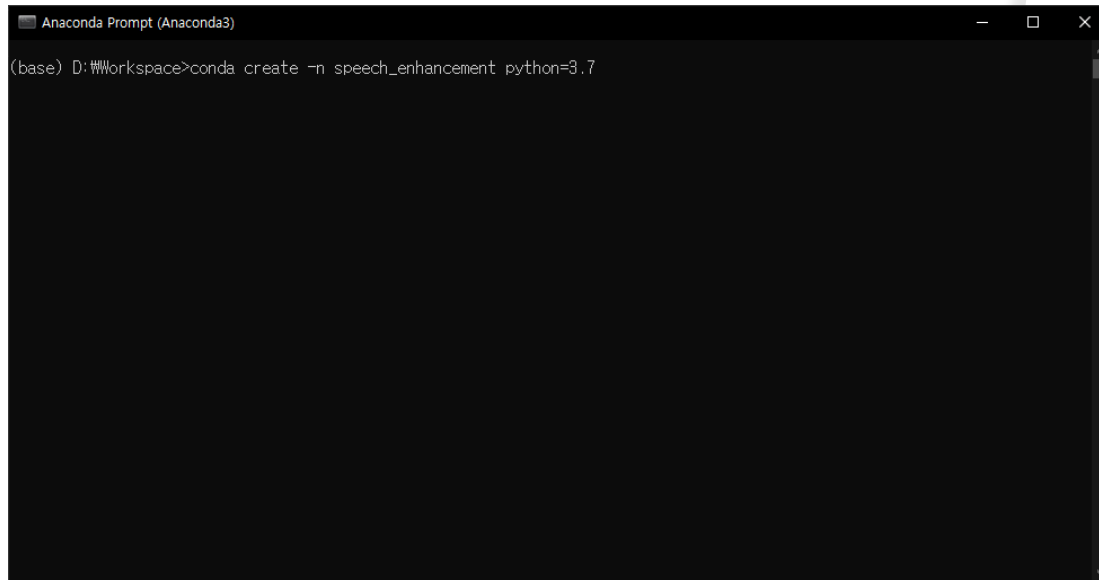
Anaconda3 설치

<https://www.anaconda.com/products/individual>

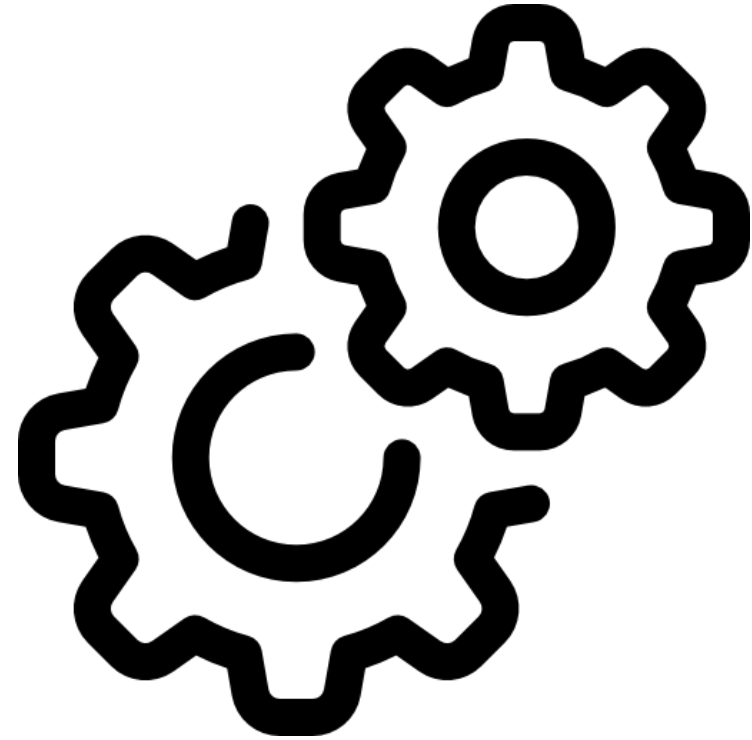


환경 설정

`conda create -n speech_enhancement python=3.7`



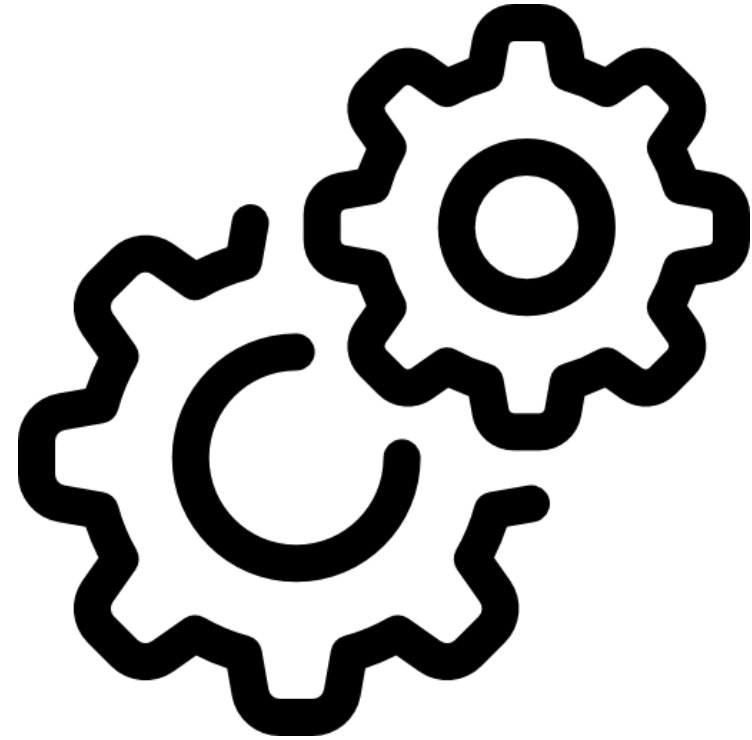
```
Anaconda Prompt (Anaconda3)
(base) D:\Workspace>conda create -n speech_enhancement python=3.7
```



환경 설정

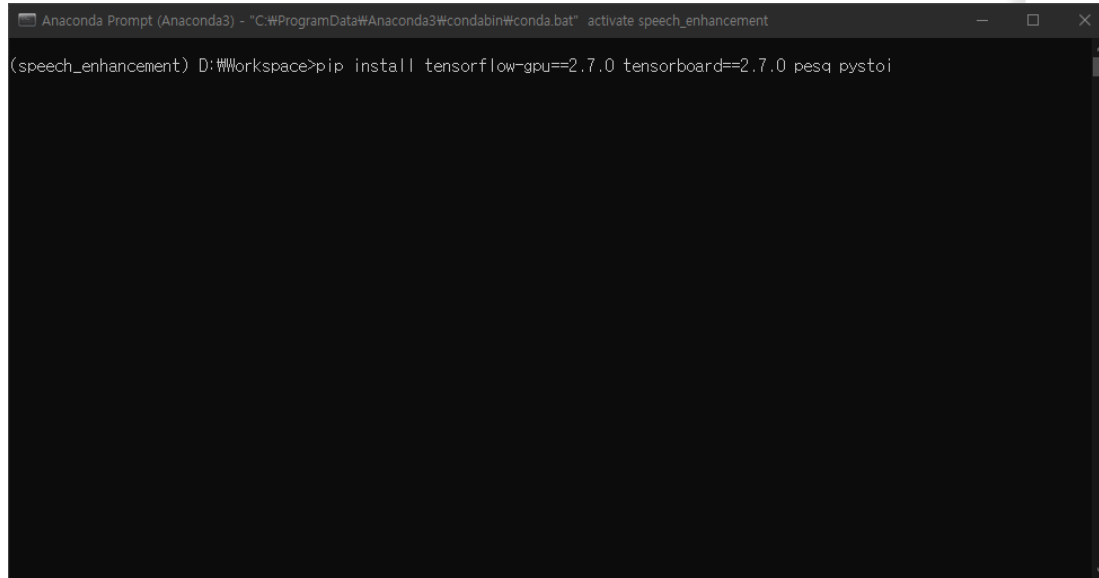
conda activate speech_enhancement

```
Anaconda Prompt (Anaconda3) - "C:\ProgramData\Anaconda3\condabin\conda.bat" activate speech_enhancement
(base) D:\Workspace>conda activate speech_enhancement
(speech_enhancement) D:\Workspace>
```

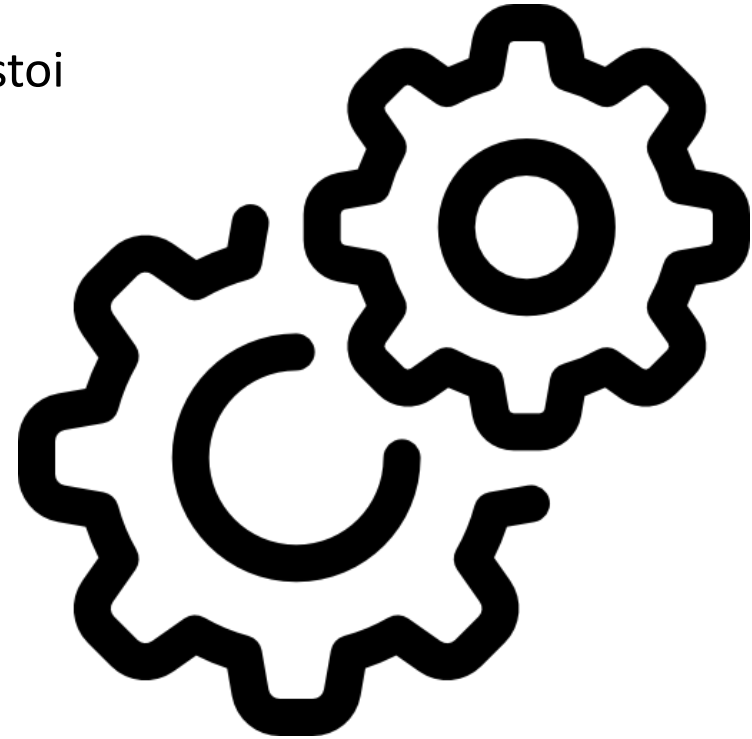


환경 설정

`pip install tensorflow-gpu=2.7.0 tensorboard=2.7.0 pesq pystoi`

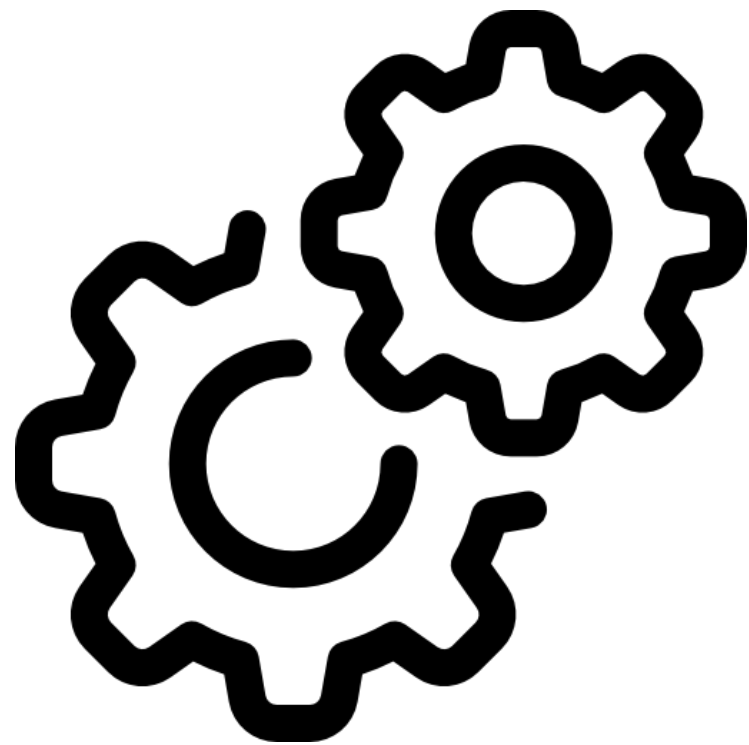
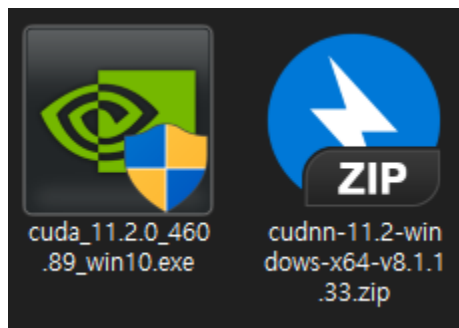


```
Anaconda Prompt (Anaconda3) - "C:\ProgramData\Anaconda3\condabin\conda.bat" activate speech_enhancement
(speech_enhancement) D:\Workspace>pip install tensorflow-gpu==2.7.0 tensorboard==2.7.0 pesq pystoi
```



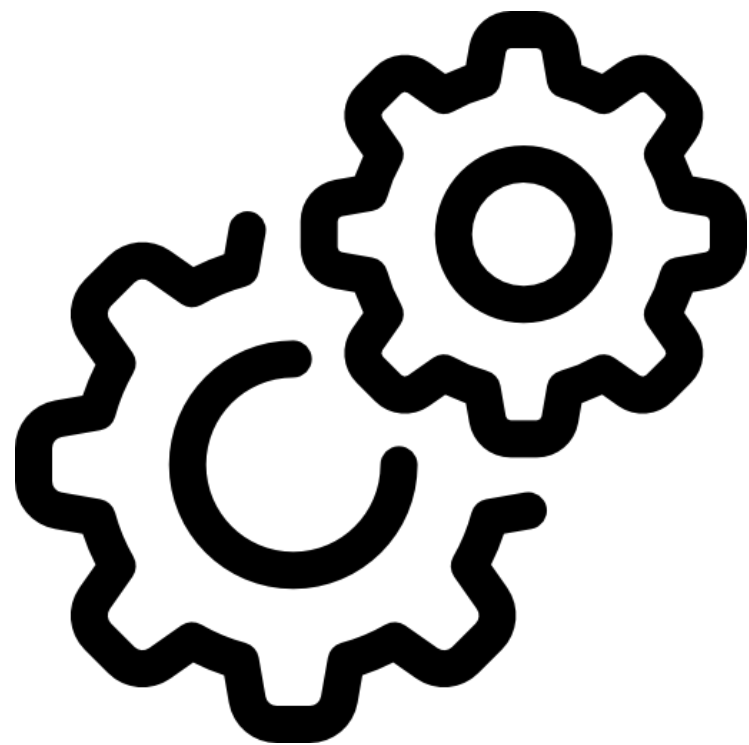
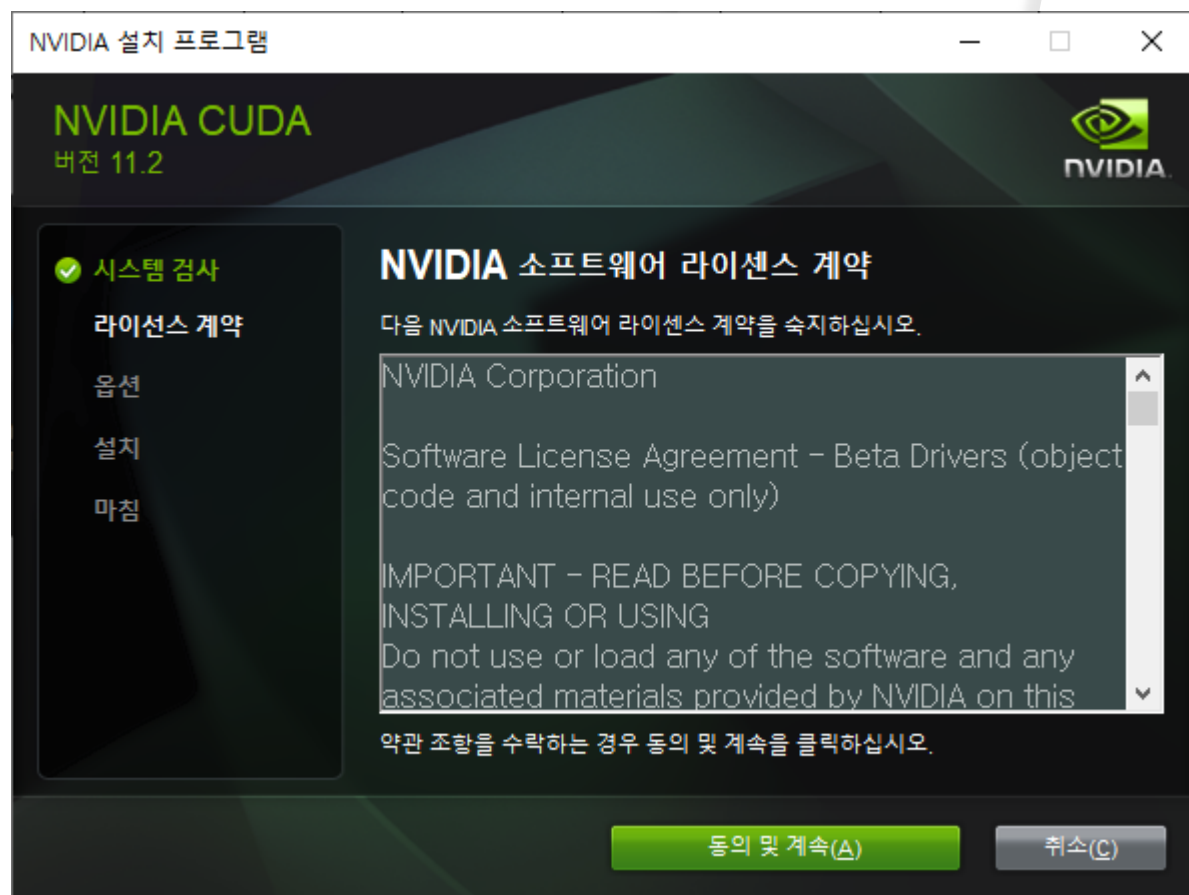
환경 설정

CUDA & cuDNN 설치



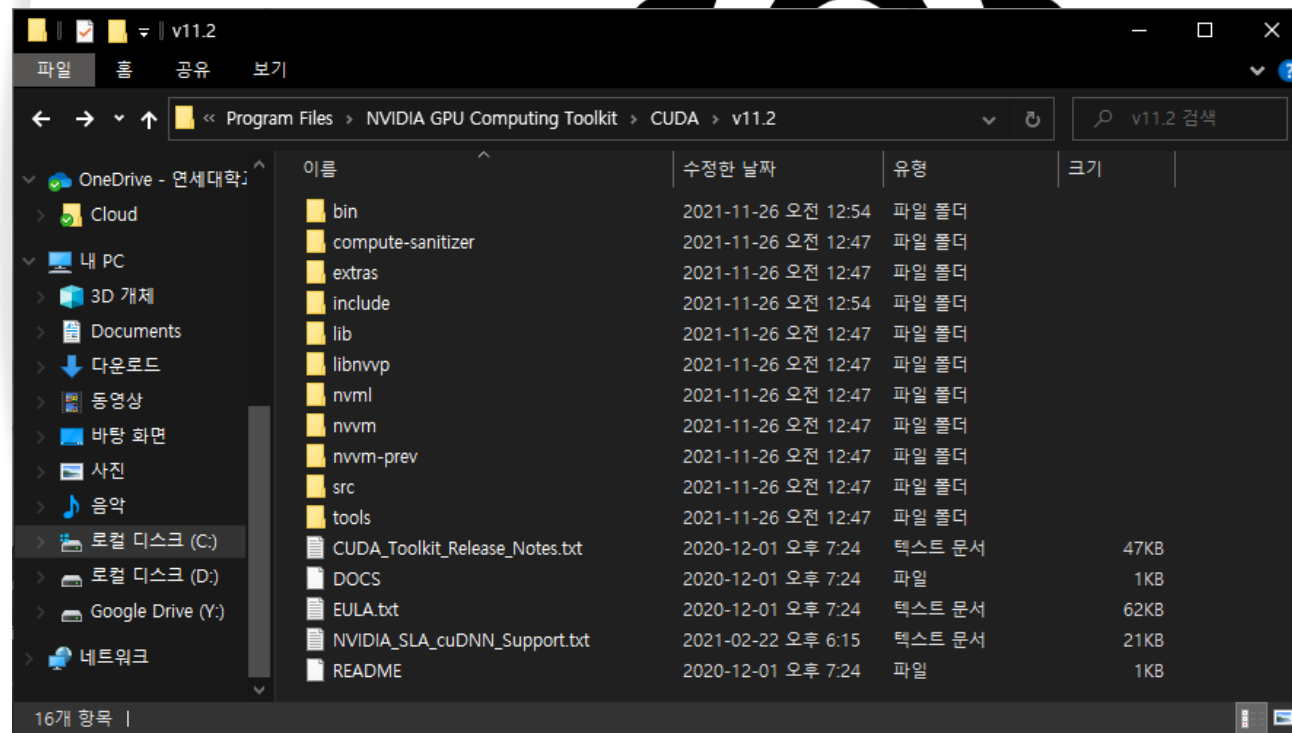
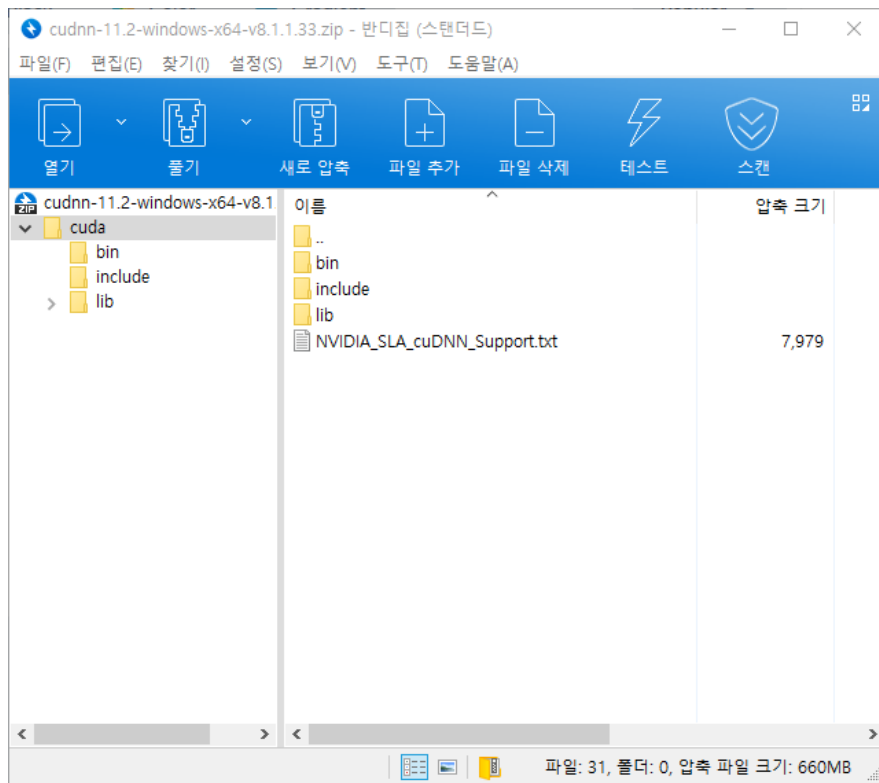
환경 설정

CUDA 설치



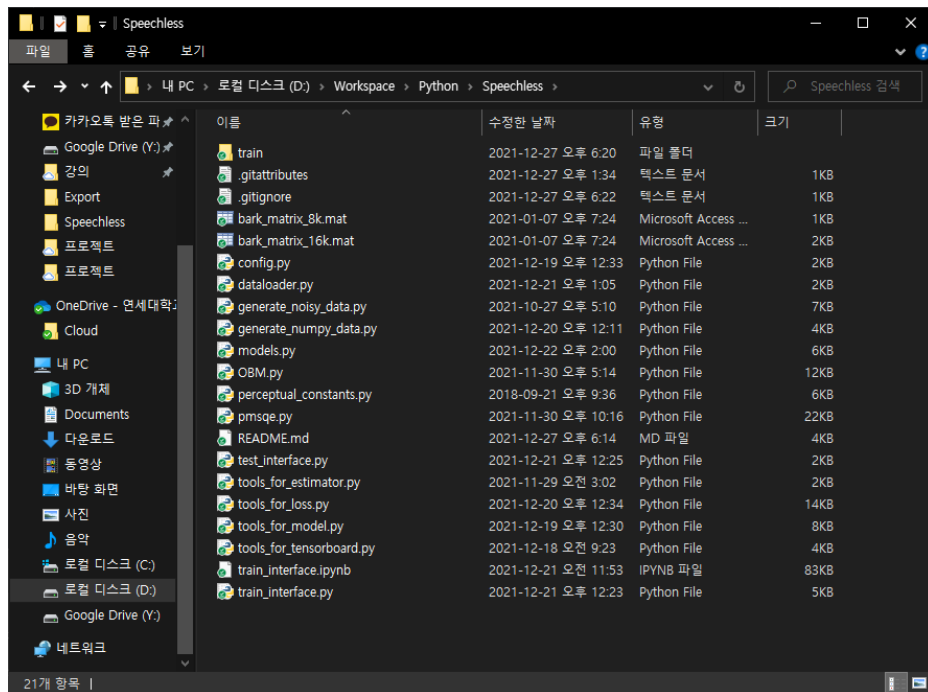
환경 설정

cuDNN: CUDA 설치 폴더에 복사



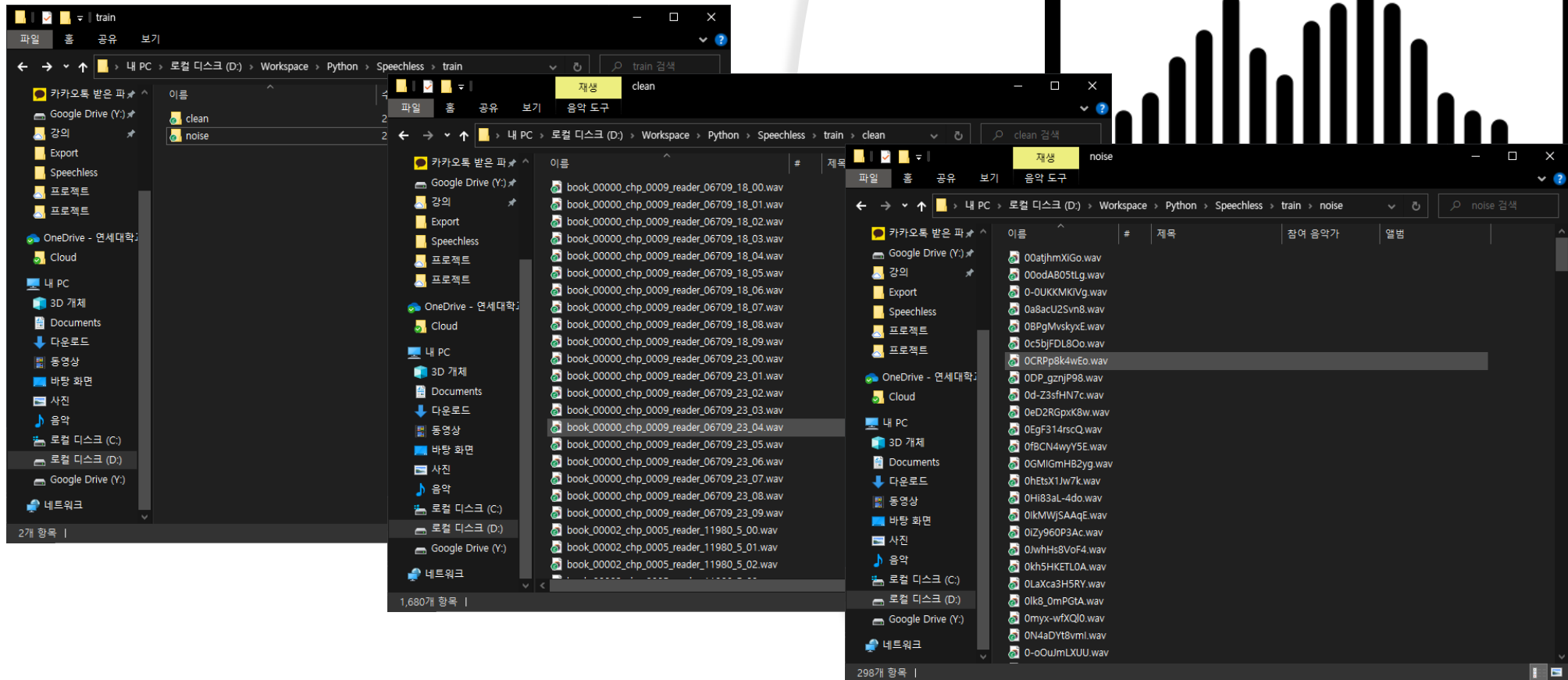
Speech Enhancement 실행

프로그램 다운로드



Speech Enhancement 실행

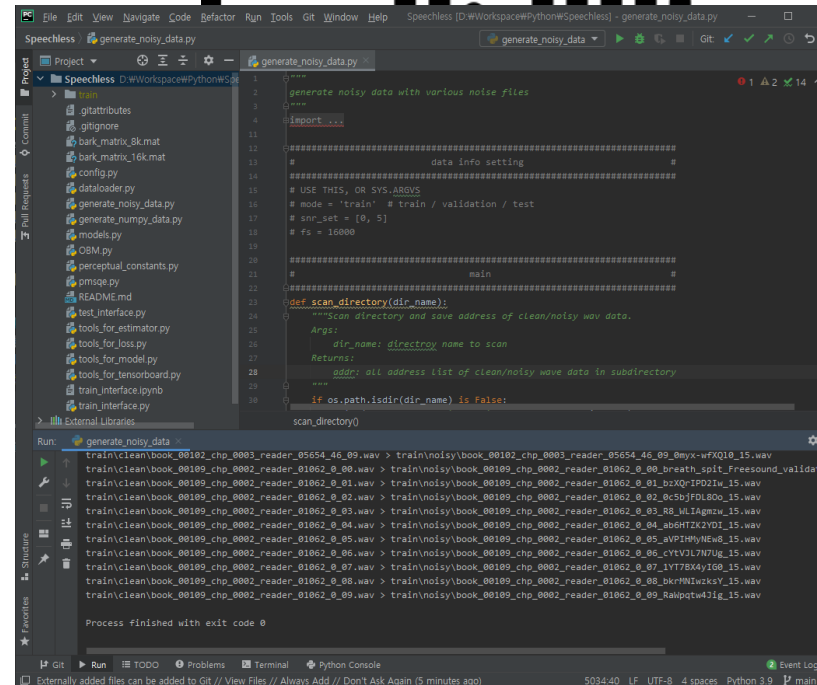
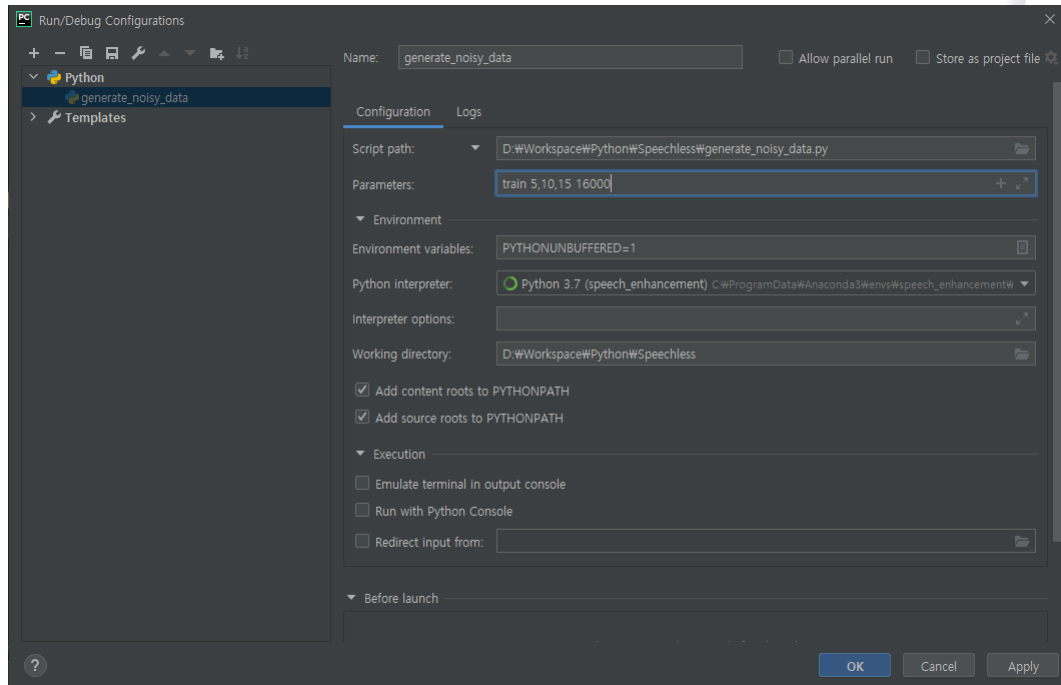
데이터셋 준비 (clean & noise 데이터셋)



Speech Enhancement 실행

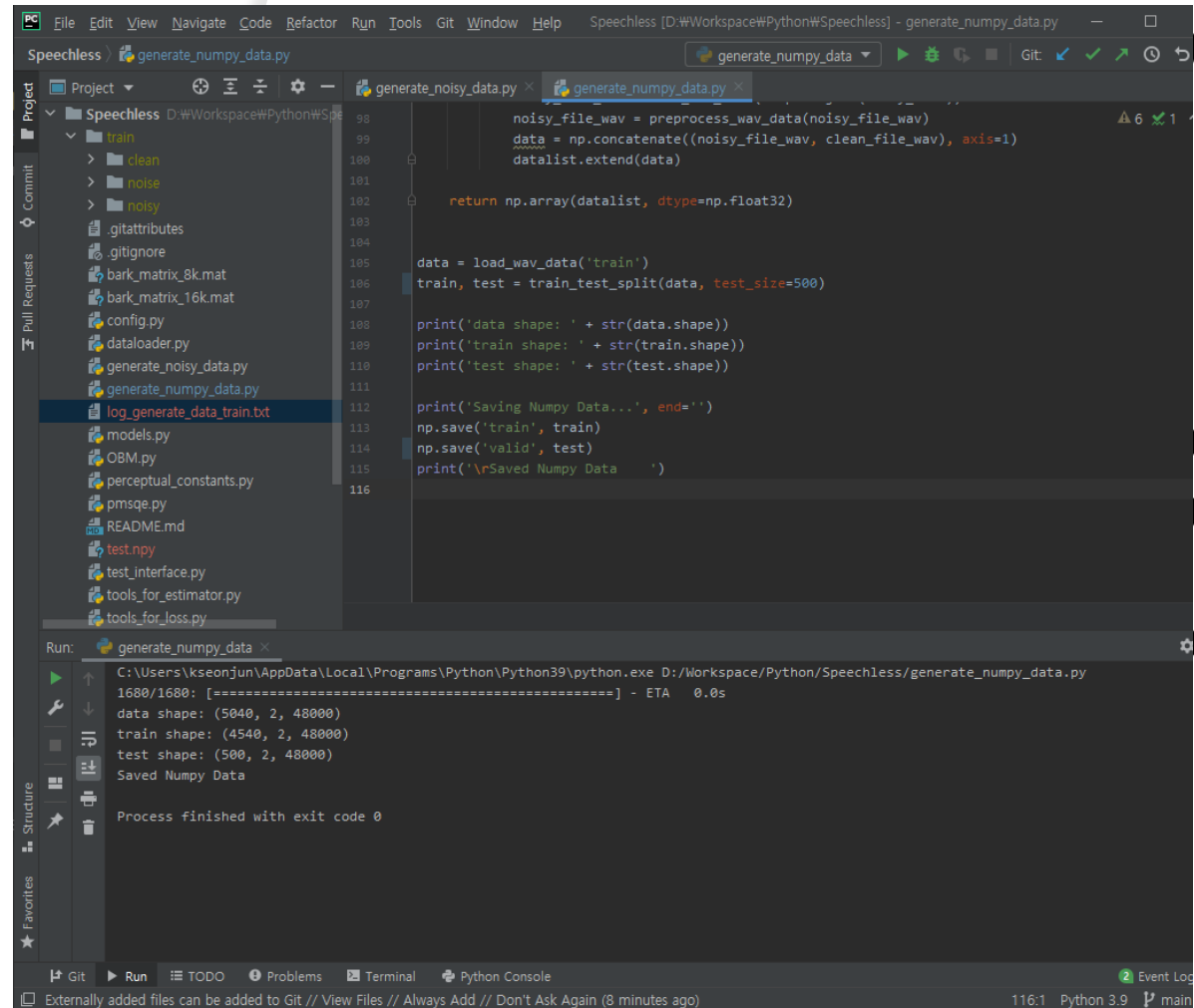
generate_noisy_data.py 실행

파라미터로 [Mode] [SNR] [fs] 설정 (아래는 예시)



Speech Enhancement 실행

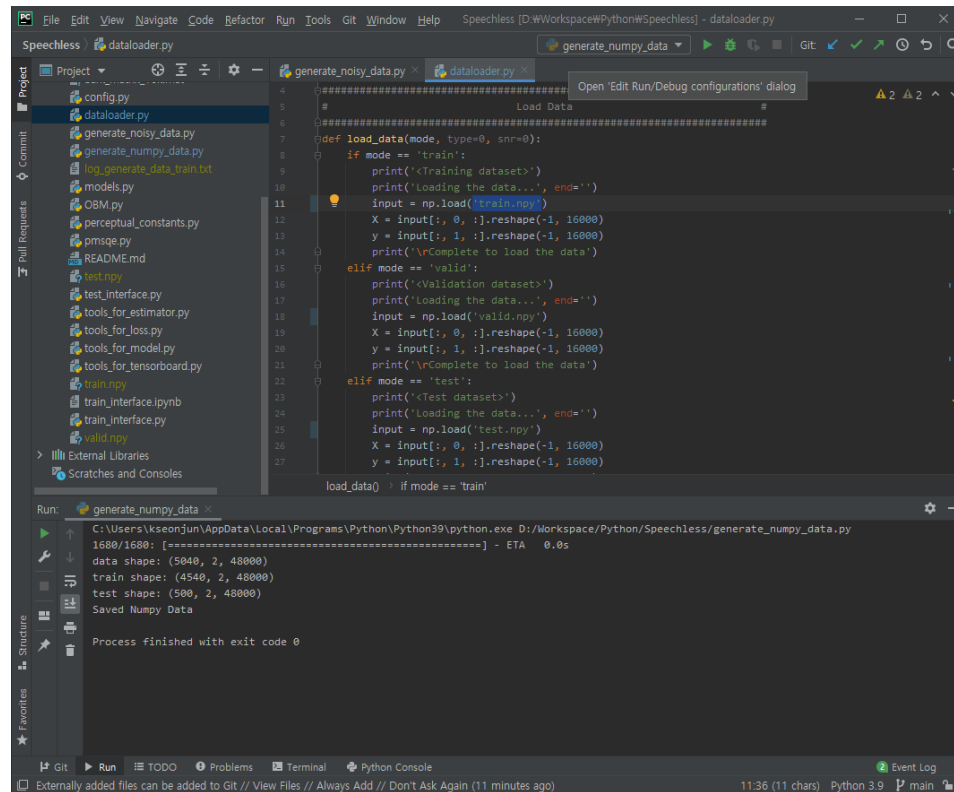
generate_numpy_data.py 실행



```
File Edit View Navigate Code Refactor Run Tools Git Window Help Speechless [D:\Workspace\Python\Speechless] - generate_numpy_data.py
Speechless generate_numpy_data.py generate_noisy_data.py generate_numpy_data.py
Project Speechless D:\Workspace\Python\Speechless
  train
    clean
    noise
    noisy
  .gitattributes
  .gitignore
  bark_matrix_8k.mat
  bark_matrix_16k.mat
  config.py
  dataloader.py
  generate_noisy_data.py
  generate_numpy_data.py
  log_generate_data_train.txt
  models.py
  OBM.py
  perceptual_constants.py
  pmsqe.py
  README.md
  test.npy
  test_interface.py
  tools_for_estimator.py
  tools_for_loss.py
Commit Pull Requests
Run: generate_numpy_data.py
C:\Users\kxeonjun\AppData\Local\Programs\Python\Python39\python.exe D:\Workspace\Python\Speechless\generate_numpy_data.py
1680/1680: [=====] - ETA 0.0s
data shape: (5040, 2, 48000)
train shape: (4540, 2, 48000)
test shape: (500, 2, 48000)
Saved Numpy Data
Process finished with exit code 0
Git Run TODO Problems Terminal Python Console Event Log
Externally added files can be added to Git // View Files // Always Add // Don't Ask Again (8 minutes ago) 116:1 Python 3.9 main
```

Speech Enhancement 실행

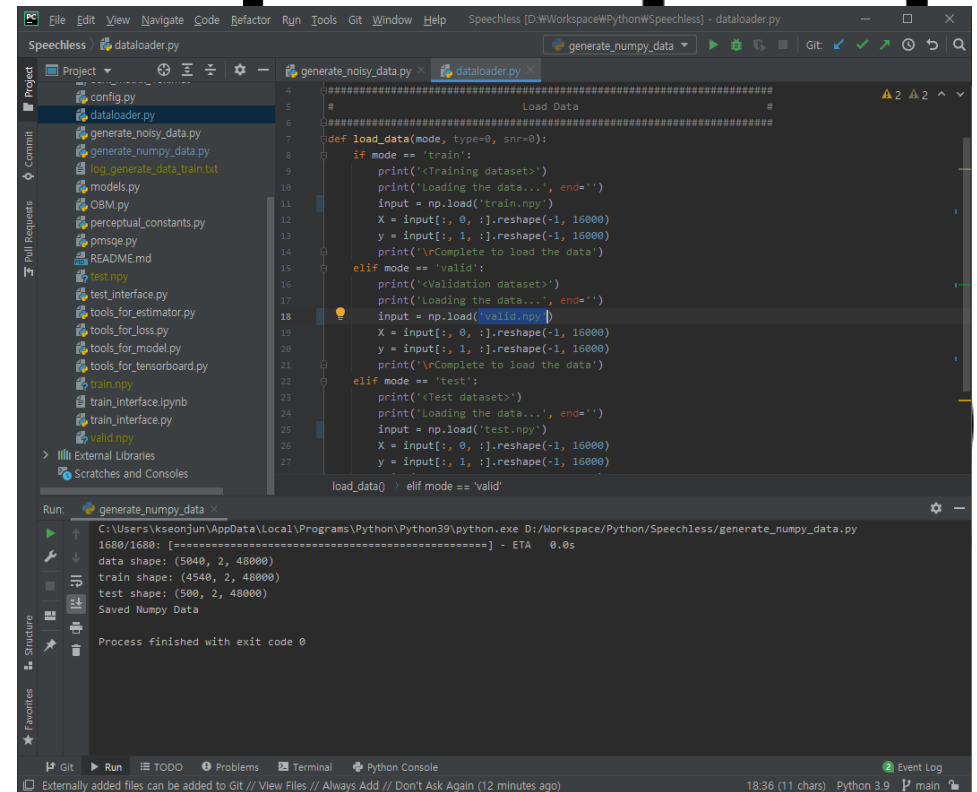
dataloader.py의 input 데이터 설정



The screenshot shows the VS Code editor with the `dataloader.py` file open. The code defines a `load_data` function that takes `mode` and `snr` as arguments. The `mode` can be 'train', 'valid', or 'test'. The function loads the corresponding data from `train.npy`, `valid.npy`, or `test.npy` and reshapes it into a 2D array of shape (500, 2, 48000). The `load_data` function is called with `mode='train'` and `snr=0`. The output of the function is displayed in the console, showing the shapes of the training, validation, and test data.

```
def load_data(mode, type=0, snr=0):  
    if mode == 'train':  
        print('<Training dataset>')  
        print('Loading the data...', end='')  
        input = np.load('train.npy')  
        X = input[:, 0, :].reshape(-1, 16000)  
        y = input[:, 1, :].reshape(-1, 16000)  
        print('\rComplete to load the data')  
    elif mode == 'valid':  
        print('<Validation dataset>')  
        print('Loading the data...', end='')  
        input = np.load('valid.npy')  
        X = input[:, 0, :].reshape(-1, 16000)  
        y = input[:, 1, :].reshape(-1, 16000)  
        print('\rComplete to load the data')  
    elif mode == 'test':  
        print('<Test dataset>')  
        print('Loading the data...', end='')  
        input = np.load('test.npy')  
        X = input[:, 0, :].reshape(-1, 16000)  
        y = input[:, 1, :].reshape(-1, 16000)
```

Run: generate_numpy_data
C:\Users\kseonjun\AppData\Local\Programs\Python\Python39\python.exe D:\Workspace\Python\Speechless\generate_numpy_data.py
1680/1680: [=====] - ETA 0.0s
data shape: (5040, 2, 48000)
train shape: (4540, 2, 48000)
test shape: (500, 2, 48000)
Saved Numpy Data
Process finished with exit code 0



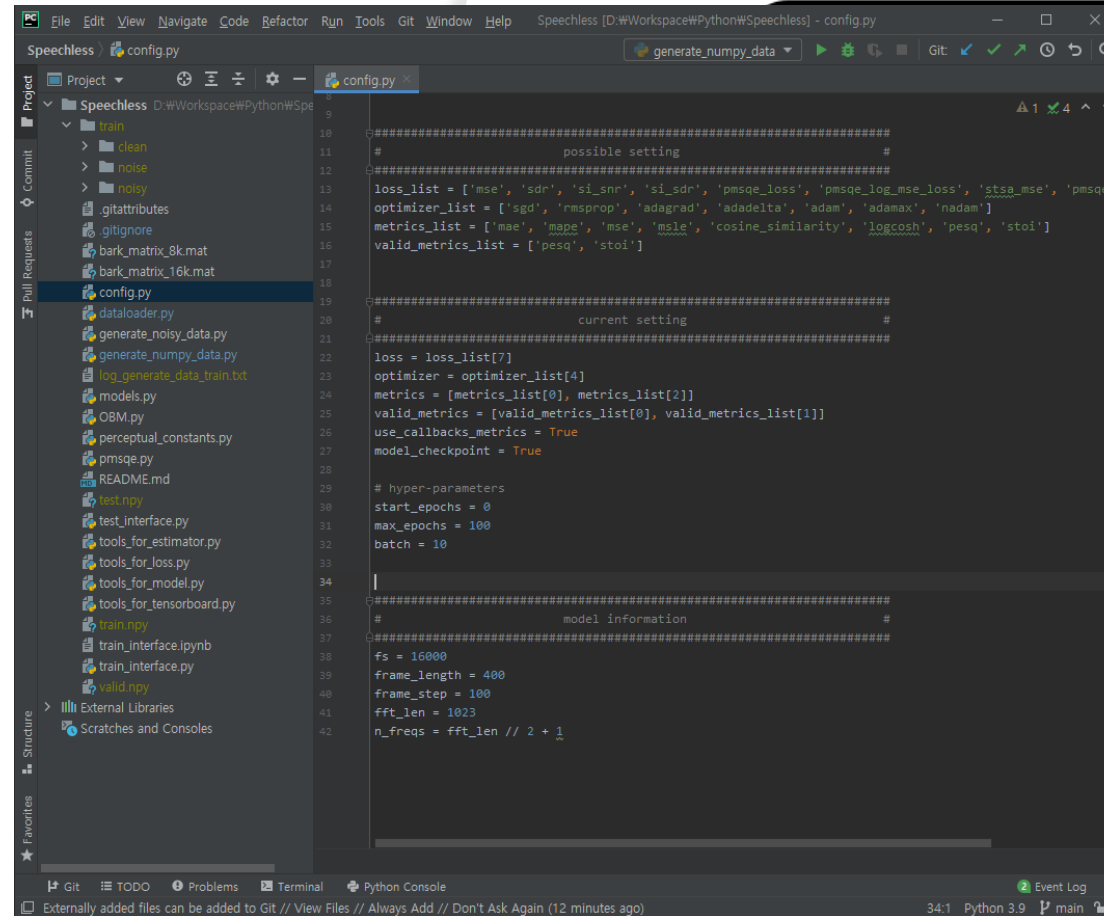
The screenshot shows the VS Code editor with the `dataloader.py` file open. The code defines a `load_data` function that takes `mode` and `snr` as arguments. The `mode` can be 'train', 'valid', or 'test'. The function loads the corresponding data from `train.npy`, `valid.npy`, or `test.npy` and reshapes it into a 2D array of shape (500, 2, 48000). The `load_data` function is called with `mode='valid'` and `snr=0`. The output of the function is displayed in the console, showing the shapes of the training, validation, and test data.

```
def load_data(mode, type=0, snr=0):  
    if mode == 'train':  
        print('<Training dataset>')  
        print('Loading the data...', end='')  
        input = np.load('train.npy')  
        X = input[:, 0, :].reshape(-1, 16000)  
        y = input[:, 1, :].reshape(-1, 16000)  
        print('\rComplete to load the data')  
    elif mode == 'valid':  
        print('<Validation dataset>')  
        print('Loading the data...', end='')  
        input = np.load('valid.npy')  
        X = input[:, 0, :].reshape(-1, 16000)  
        y = input[:, 1, :].reshape(-1, 16000)  
        print('\rComplete to load the data')  
    elif mode == 'test':  
        print('<Test dataset>')  
        print('Loading the data...', end='')  
        input = np.load('test.npy')  
        X = input[:, 0, :].reshape(-1, 16000)  
        y = input[:, 1, :].reshape(-1, 16000)
```

Run: generate_numpy_data
C:\Users\kseonjun\AppData\Local\Programs\Python\Python39\python.exe D:\Workspace\Python\Speechless\generate_numpy_data.py
1680/1680: [=====] - ETA 0.0s
data shape: (5040, 2, 48000)
train shape: (4540, 2, 48000)
test shape: (500, 2, 48000)
Saved Numpy Data
Process finished with exit code 0

Speech Enhancement 실행

config.py에서 세부설정



```
##### possible setting #####
#
loss_list = ['mse', 'sdr', 'si_snr', 'si_sdr', 'pmsqe_loss', 'pmsqe_log_mse_loss', 'stsa_mse', 'pmsqe_
optimizer_list = ['sgd', 'rmsprop', 'adagrad', 'adadelta', 'adam', 'adamax', 'nadam']
metrics_list = ['mae', 'mape', 'mse', 'mle', 'cosine_similarity', 'logcosh', 'pesq', 'stoi']
valid_metrics_list = ['pesq', 'stoi']

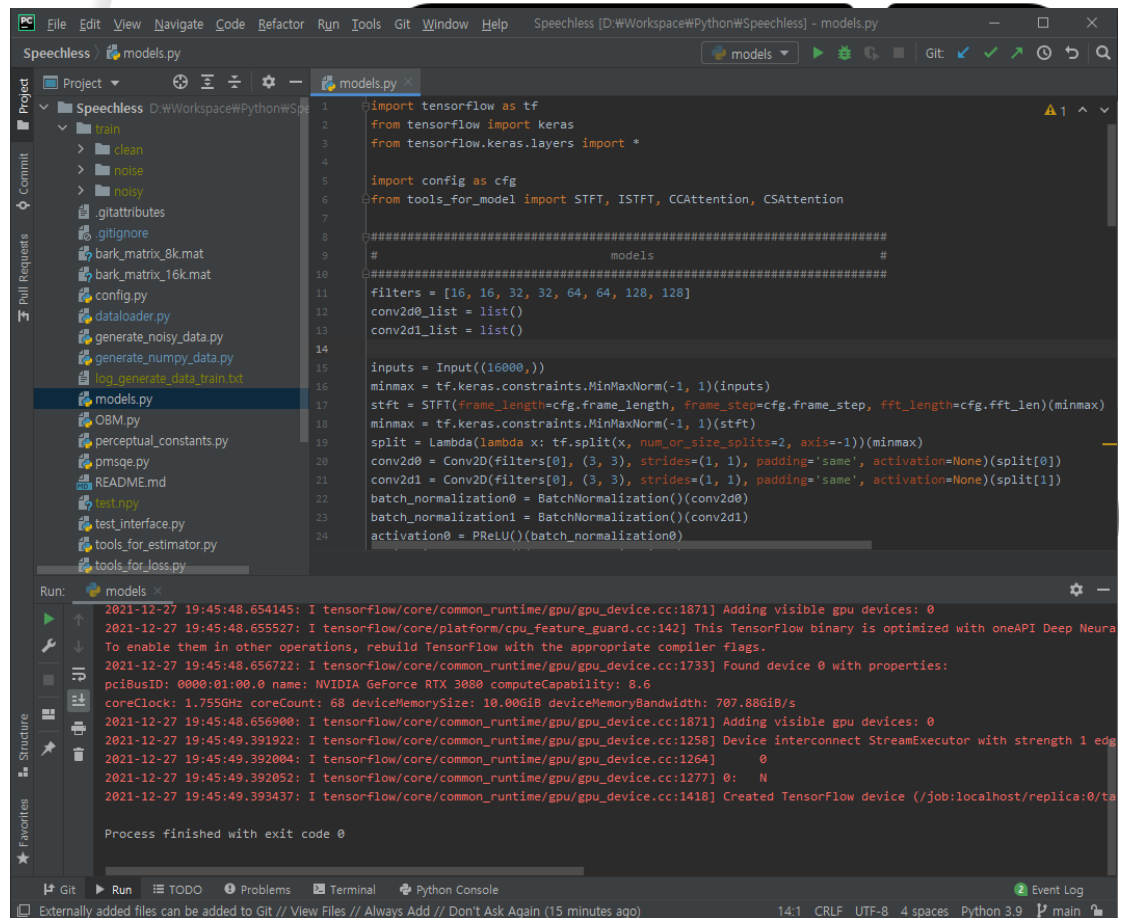
##### current setting #####
#
loss = loss_list[7]
optimizer = optimizer_list[4]
metrics = [metrics_list[0], metrics_list[2]]
valid_metrics = [valid_metrics_list[0], valid_metrics_list[1]]
use_callbacks_metrics = True
model_checkpoint = True

# hyper-parameters
start_epochs = 0
max_epochs = 100
batch = 10

##### model information #####
#
fs = 16000
frame_length = 400
frame_step = 100
fft_len = 1023
n_freqs = fft_len // 2 + 1
```

Speech Enhancement 실행

models.py에서 Keras 모델을 만들고 확인
(모델이 정상 생성된 것을 확인)



The screenshot shows an IDE window titled "Speechless [D:\Workspace\Python\Speechless] - models.py". The left sidebar displays a project tree with files like train, clean, noise, noisy, gitattributes, gitignore, bark_matrix_8k.mat, bark_matrix_16k.mat, config.py, dataloader.py, generate_noisy_data.py, generate_numpy_data.py, log_generate_data_train.txt, models.py, OBM.py, perceptual_constants.py, pmsqe.py, README.md, test.npy, test_interface.py, tools_for_estimator.py, and tools_for_loss.py. The main editor shows the contents of models.py, which includes imports for tensorflow, keras, and tensorflow.keras.layers, and a Keras model definition with layers like Conv2D, BatchNormalization, and PReLU. The bottom Run console shows the execution output, indicating that TensorFlow was successfully initialized on a GPU (NVIDIA GeForce RTX 3080) and the process finished with exit code 0.

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import *

import config as cfg
from tools_for_model import STFT, ISTFT, CCAttention, CSAttention

#####
# models
#####
filters = [16, 16, 32, 32, 64, 64, 128, 128]
conv2d0_list = list()
conv2d1_list = list()

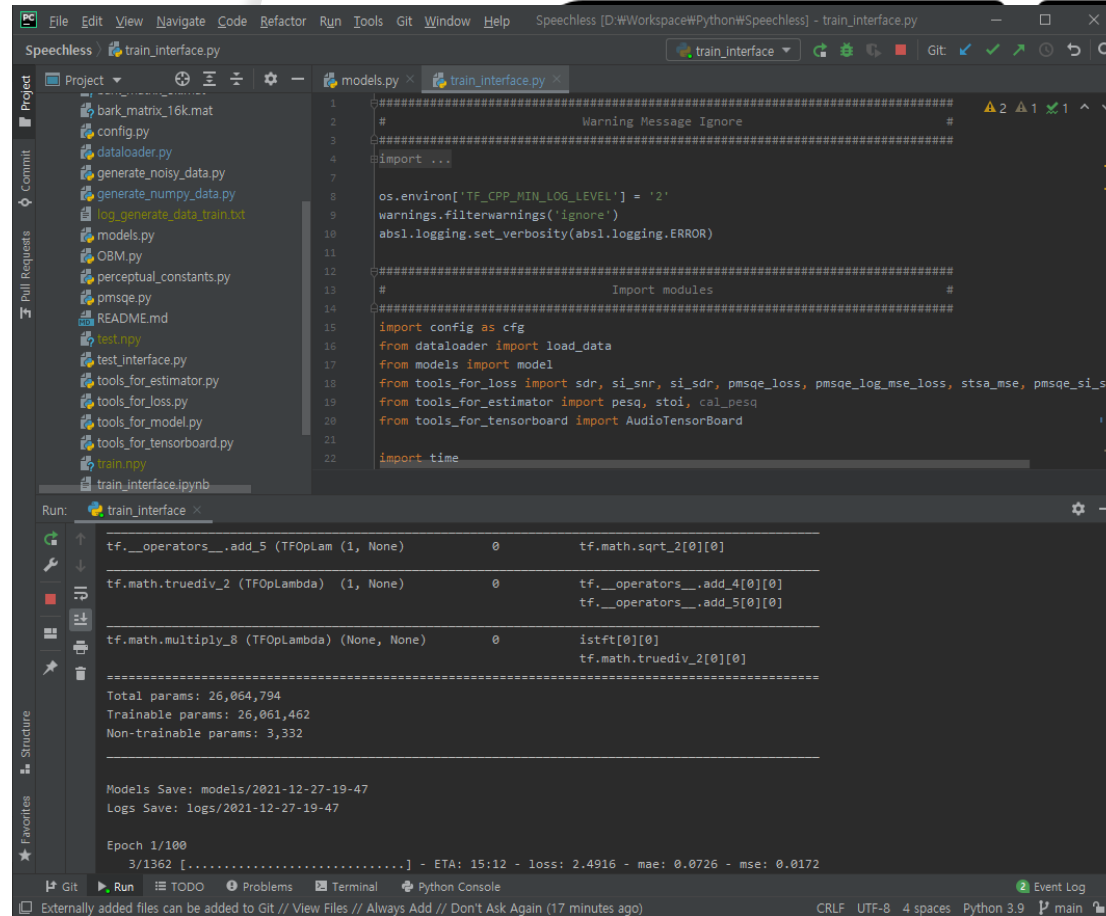
inputs = Input((16000,))
minmax = tf.keras.constraints.MinMaxNorm(-1, 1)(inputs)
stft = STFT(frame_length=cfg.frame_length, frame_step=cfg.frame_step, fft_length=cfg.fft_len)(minmax)
minmax = tf.keras.constraints.MinMaxNorm(-1, 1)(stft)
split = Lambda(lambda x: tf.split(x, num_or_size_splits=2, axis=-1))(minmax)
conv2d0 = Conv2D(filters[0], (3, 3), strides=(1, 1), padding='same', activation=None)(split[0])
conv2d1 = Conv2D(filters[0], (3, 3), strides=(1, 1), padding='same', activation=None)(split[1])
batch_normalization0 = BatchNormalization()(conv2d0)
batch_normalization1 = BatchNormalization()(conv2d1)
activation0 = PReLU()(batch_normalization0)
```

Run: models x

```
2021-12-27 19:45:48.654145: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1871] Adding visible gpu devices: 0
2021-12-27 19:45:48.655527: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-12-27 19:45:48.656722: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1733] Found device 0 with properties:
pciBusID: 0000:01:00.0 name: NVIDIA GeForce RTX 3080 computeCapability: 8.6
coreClock: 1.755GHz coreCount: 68 deviceMemorySize: 10.00GiB deviceMemoryBandwidth: 707.88GiB/s
2021-12-27 19:45:48.656900: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1871] Adding visible gpu devices: 0
2021-12-27 19:45:49.391922: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1258] Device interconnect StreamExecutor with strength 1 edge
2021-12-27 19:45:49.392004: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1264] 0
2021-12-27 19:45:49.392052: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1277] 0: N
2021-12-27 19:45:49.393437: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1418] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0) with 10.00GiB memory and 707.88GiB/s bandwidth
Process finished with exit code 0
```


Speech Enhancement 실행

train_interface.py를 통해서 훈련

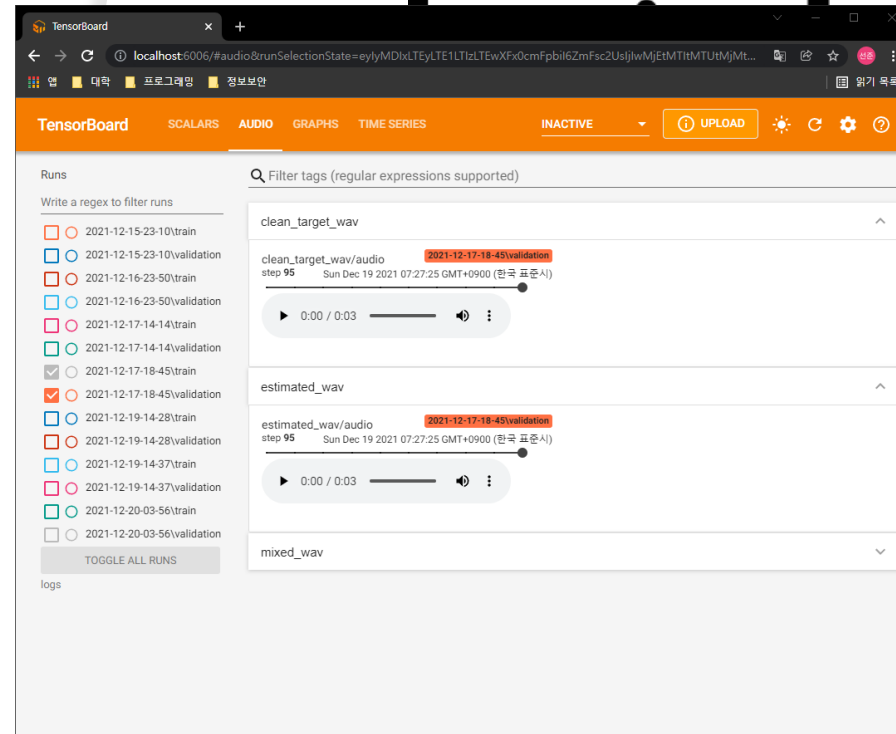
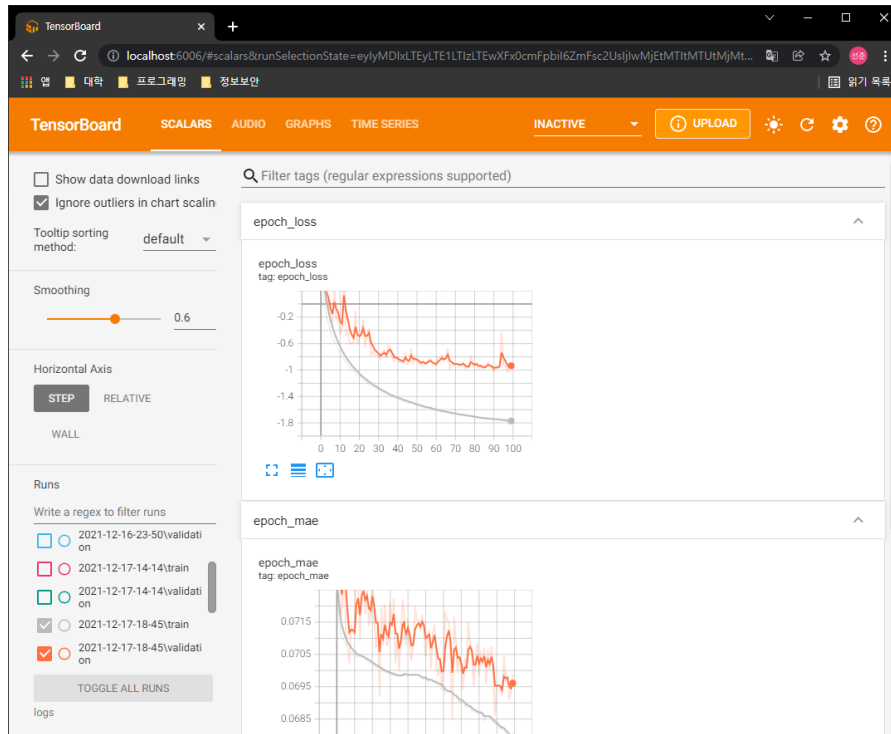


The screenshot shows an IDE window titled "Speechless [D:\Workspace\Python\Speechless] - train_interface.py". The left sidebar displays a project structure with files like bark_matrix_16k.mat, config.py, dataloader.py, generate_noisy_data.py, generate_numpy_data.py, log_generate_data_train.txt, models.py, OBM.py, perceptual_constants.py, pmsqe.py, README.md, test.npy, test_interface.py, tools_for_estimator.py, tools_for_loss.py, tools_for_model.py, tools_for_tensorboard.py, train.npy, and train_interface.ipynb. The main editor shows the code in train_interface.py, which includes imports for config, dataloader, models, tools_for_loss, tools_for_estimator, tools_for_tensorboard, and time. The bottom panel shows the execution output for the "train_interface" run.

```
tf.__operators__.add_5 (TFOpLam (1, None))      0      tf.math.sqrt_2[0][0]
tf.math.truediv_2 (TFOpLambda) (1, None)      0      tf.__operators__.add_4[0][0]
tf.math.truediv_2 (TFOpLambda) (1, None)      0      tf.__operators__.add_5[0][0]
tf.math.multiply_8 (TFOpLambda) (None, None)    0      istft[0][0]
tf.math.multiply_8 (TFOpLambda) (None, None)    0      tf.math.truediv_2[0][0]
-----
Total params: 26,064,794
Trainable params: 26,061,462
Non-trainable params: 3,332
-----
Models Save: models/2021-12-27-19-47
Logs Save: logs/2021-12-27-19-47
Epoch 1/100
3/1362 [.....] - ETA: 15:12 - loss: 2.4916 - mae: 0.0726 - mse: 0.0172
```

Speech Enhancement 확인

Tensorboard를 통해서 확인



참고자료

Anaconda3

<https://www.anaconda.com/products/individual>

CUDA 11.2

<https://developer.nvidia.com/cuda-11.2.0-download-archive>

cuDNN

<https://developer.nvidia.com/rdp/cudnn-archive>

Speechless

<https://github.com/Neurumaru/Speechless>

