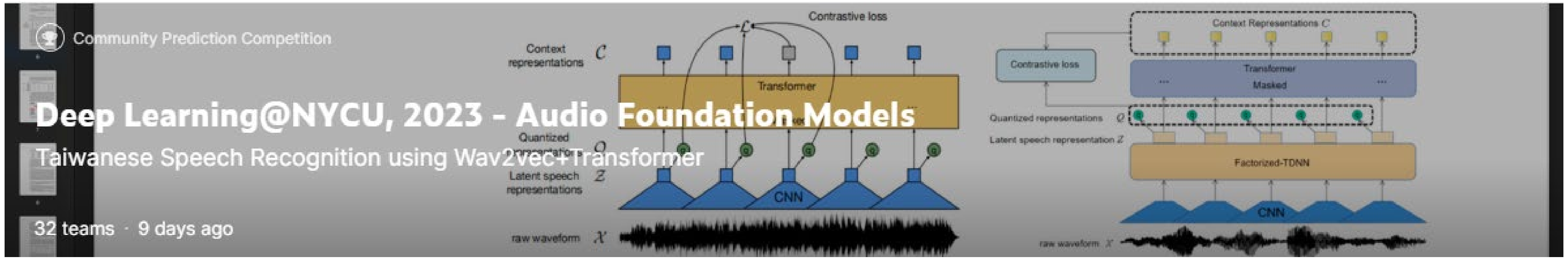

Community Prediction Competition

Deep Learning@NYCU, 2023 - Audio Foundation Models

Taiwanese Speech Recognition using Wav2vec+Transformer

32 teams · 9 days ago

Contrastive loss
Context representations $C$
Transformer
Quantized representations $Q$
Latent speech representations $Z$
CNN
raw waveform $\mathcal{X}$
Context Representations $C$
Contrastive loss
Transformer
Masked
Quantized representations $Q$
Latent speech representation $Z$
Factorized-TDNN
CNN
raw waveform $X$

# Using Self-Supervised Pre-Trained Model + Transformer

學生: 陳澤昕

學號: 311356003

# Dataset

Evaluation :

單人女聲聲音（高雄腔）

輸入：台語語音音檔（格式：wav檔, 22 kHz, mono, 32 bits）

輸出：台羅拼音（依教育部標準）

- Training set :

kaggle資料夾下的training set利用sox轉音檔格式將22kHz的音檔轉換成16kH，

另外擴增資料集使用 audiomentations & nlpaug 擴增訓練資料分別使用

min_snr_in_db = [ 5, 10, 20]

max_snr_in_db = [15, 30, 40]

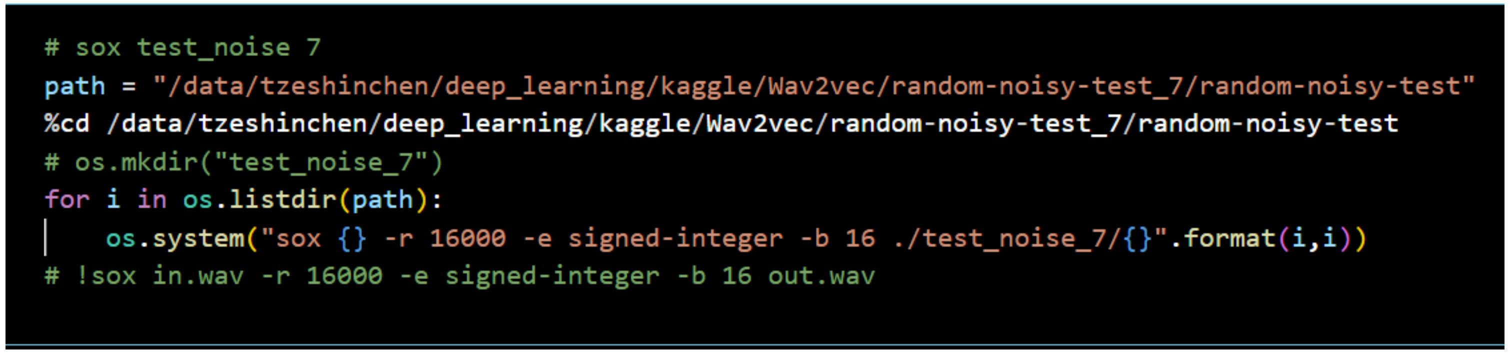
- Testing seting :

Kaggle資料夾下的testing set同樣利用sox轉音檔格式將22kHz因檔轉換成16kHz，使用random-noisy-test 7

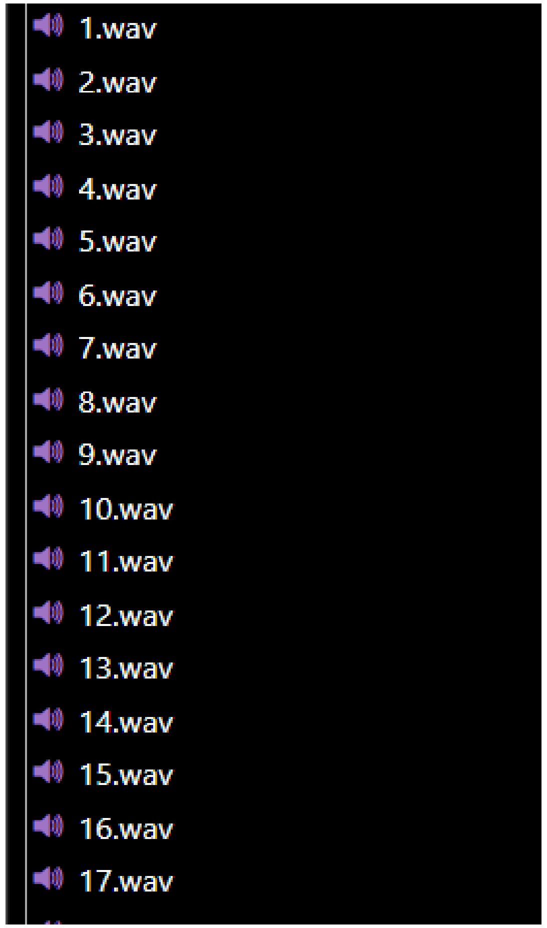
- Validation set :

從training set中隨機挑選10個資料當validation

# Sox to 16K

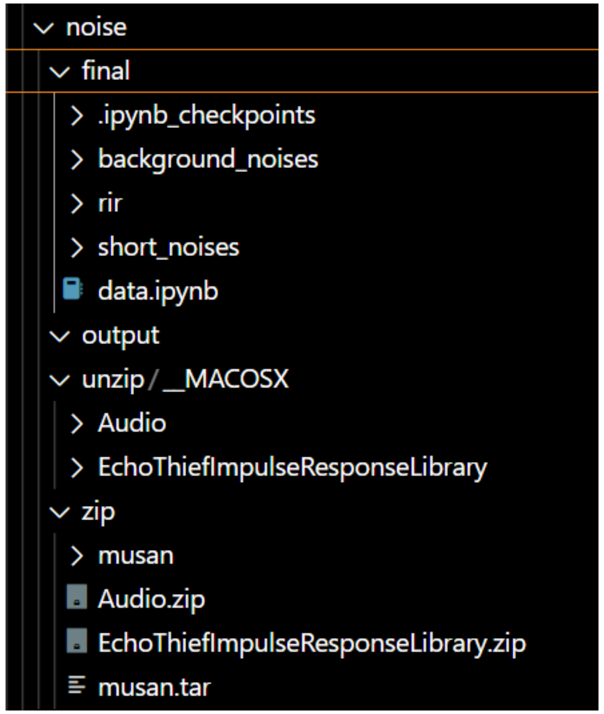


```
# sox test_noise 7
path = "/data/tzeshinchen/deep_learning/kaggle/Wav2vec/random-noisy-test_7/random-noisy-test"
%cd /data/tzeshinchen/deep_learning/kaggle/Wav2vec/random-noisy-test_7/random-noisy-test
# os.mkdir("test_noise_7")
for i in os.listdir(path):
    os.system("sox {} -r 16000 -e signed-integer -b 16 ./test_noise_7/{}".format(i,i))
# !sox in.wav -r 16000 -e signed-integer -b 16 out.wav
```

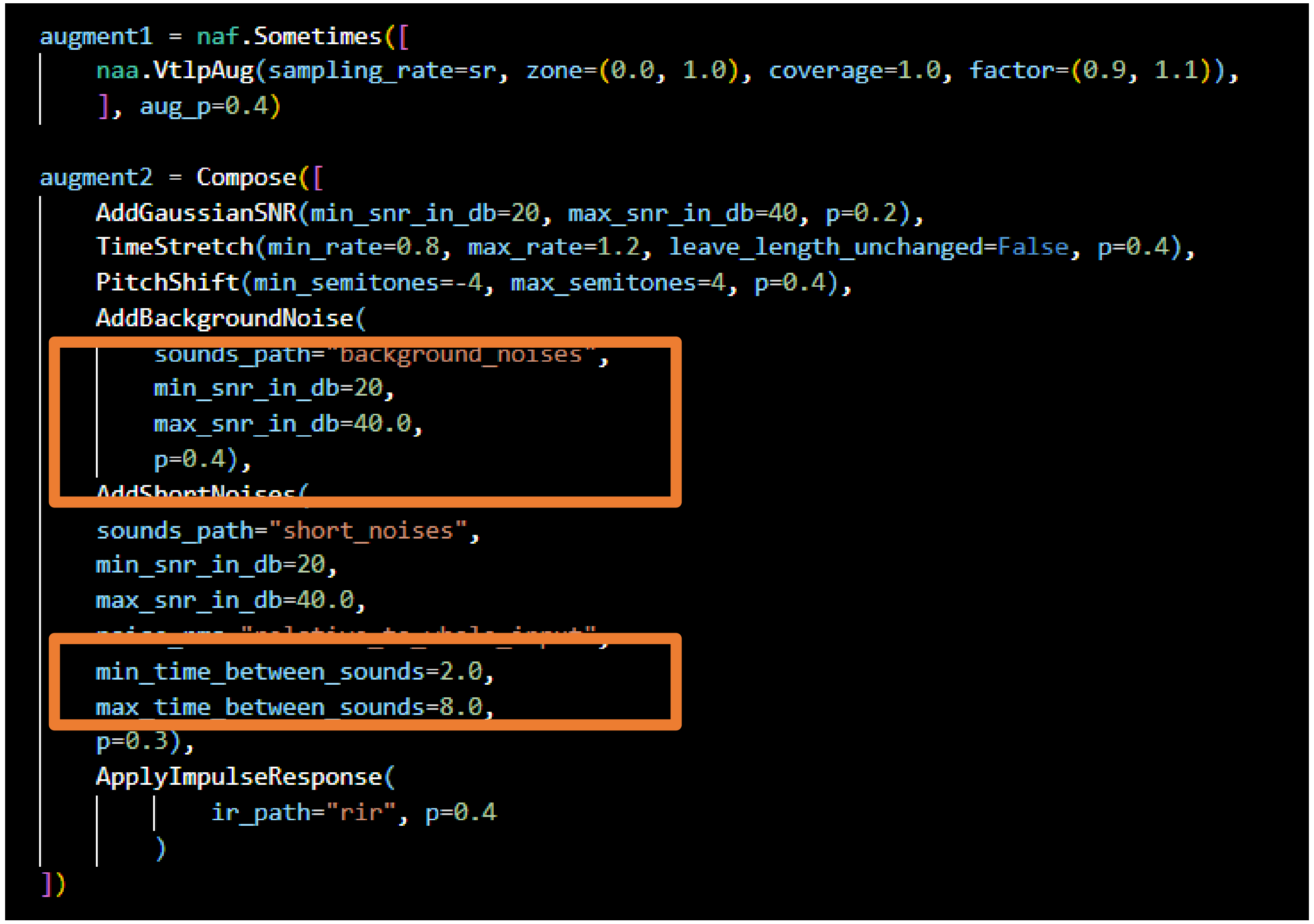


```
1.wav
2.wav
3.wav
4.wav
5.wav
6.wav
7.wav
8.wav
9.wav
10.wav
11.wav
12.wav
13.wav
14.wav
15.wav
16.wav
17.wav
```

# Noise



```
v noise
  v final
    > .ipynb_checkpoints
    > background_noises
    > rir
    > short_noises
      data.ipynb
  v output
  v unzip / __MACOSX
    > Audio
    > EchoThiefImpulseResponseLibrary
  v zip
    > musan
      Audio.zip
      EchoThiefImpulseResponseLibrary.zip
      musan.tar
```



```python
augment1 = naf.Sometimes([
    naa.VtlpAug(sampling_rate=sr, zone=(0.0, 1.0), coverage=1.0, factor=(0.9, 1.1)),
    ], aug_p=0.4)

augment2 = Compose([
    AddGaussianSNR(min_snr_in_db=20, max_snr_in_db=40, p=0.2),
    TimeStretch(min_rate=0.8, max_rate=1.2, leave_length_unchanged=False, p=0.4),
    PitchShift(min_semitones=-4, max_semitones=4, p=0.4),
    AddBackgroundNoise(
        sounds_path="background_noises",
        min_snr_in_db=20,
        max_snr_in_db=40.0,
        p=0.4),
    AddShortNoises(
    sounds_path="short_noises",
    min_snr_in_db=20,
    max_snr_in_db=40.0,
    noise_rms="relative_to_whole_input",
    min_time_between_sounds=2.0,
    max_time_between_sounds=8.0,
    p=0.3),
    ApplyImpulseResponse(
            ir_path="rir", p=0.4
        )
])
```

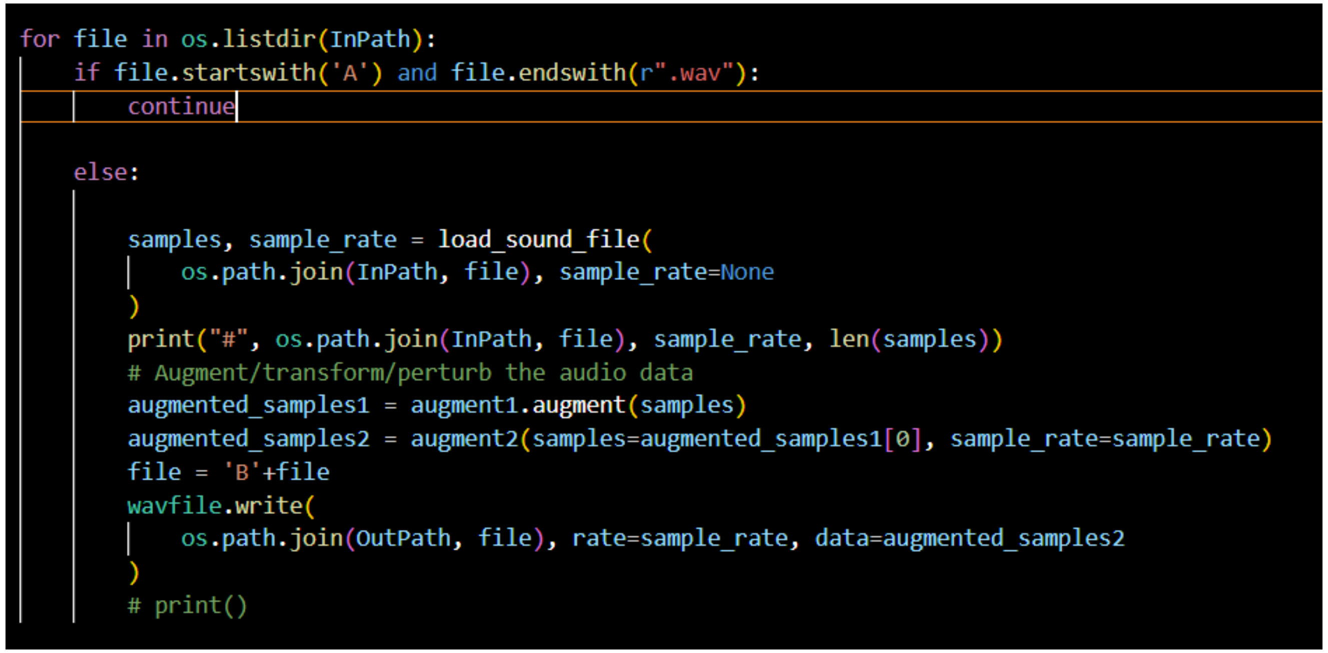
Musan : https://www.openslr.org/17

Rir :

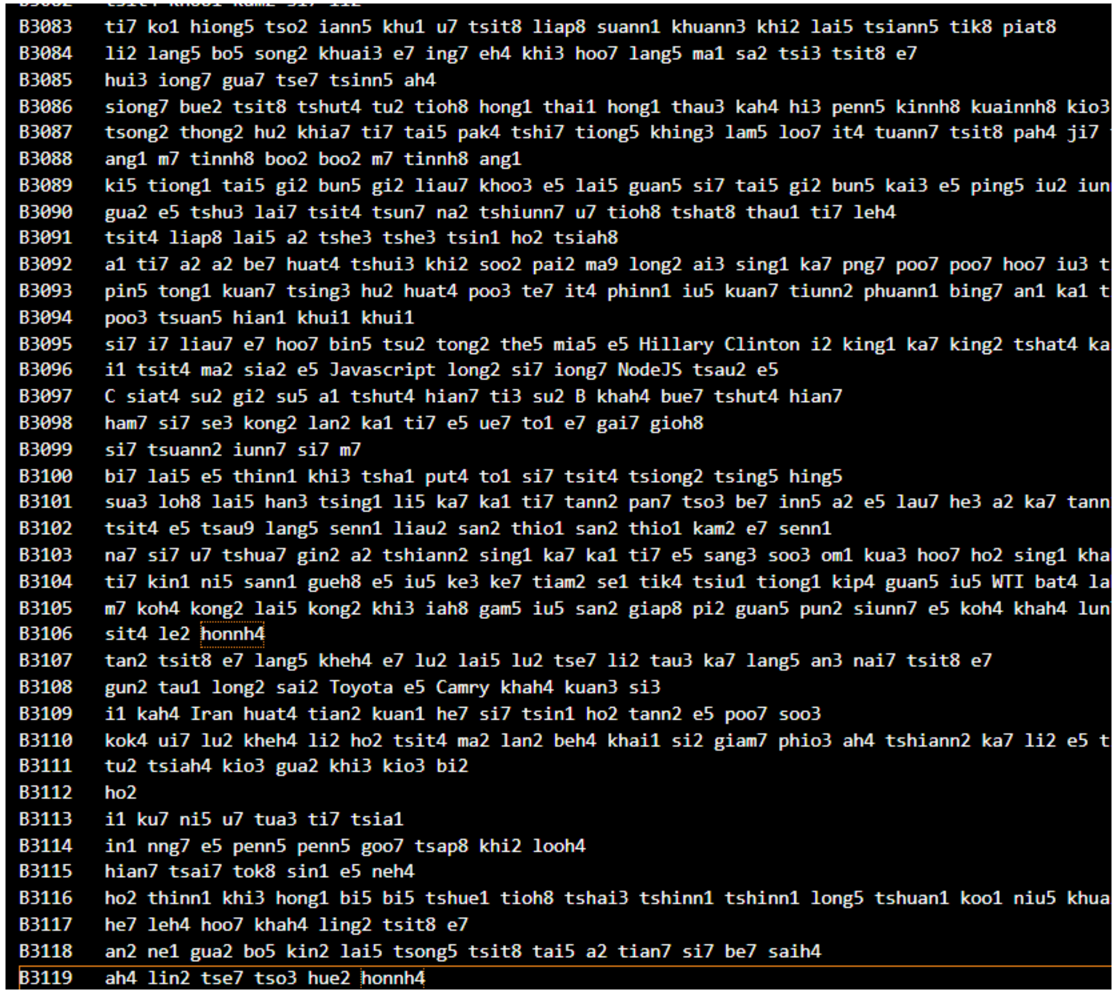
https://mcdermottlab.mit.edu/Reverb/IR_Survey.html

http://www.echothief.com/

# Data argument



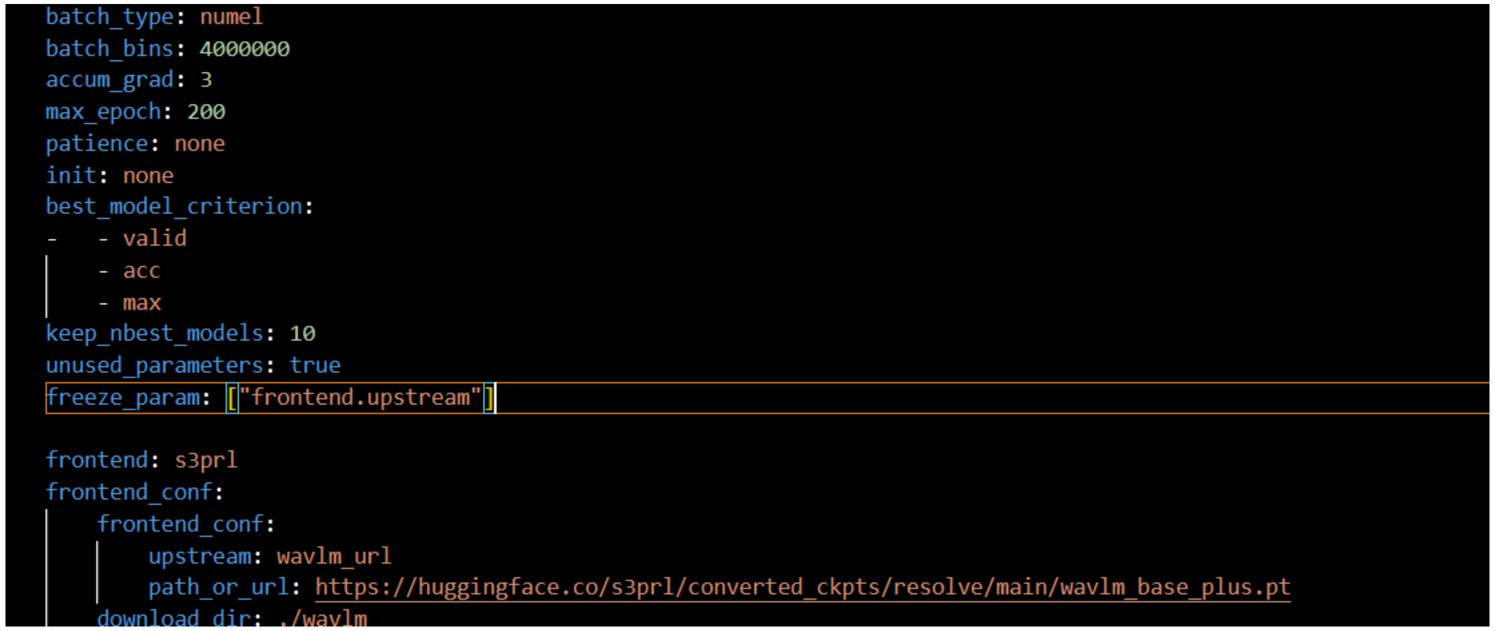
```python
for file in os.listdir(InPath):
    if file.startswith('A') and file.endswith(r".wav"):
        continue

    else:

        samples, sample_rate = load_sound_file(
            os.path.join(InPath, file), sample_rate=None
        )
        print("#", os.path.join(InPath, file), sample_rate, len(samples))
        # Augment/transform/perturb the audio data
        augmented_samples1 = augment1.augment(samples)
        augmented_samples2 = augment2(samples=augmented_samples1[0], sample_rate=sample_rate)
        file = 'B'+file
        wavfile.write(
            os.path.join(OutPath, file), rate=sample_rate, data=augmented_samples2
        )
        # print()
```



```
B3083   ti7 ko1 hiong5 tso2 iann5 khu1 u7 tsit8 liap8 suann1 khuann3 khi2 lai5 tsiann5 tik8 piat8
B3084   li2 lang5 bo5 song2 khuai3 e7 ing7 eh4 khi3 hoo7 lang5 ma1 sa2 tsi3 tsit8 e7
B3085   hui3 iong7 gua7 tse7 tsinn5 ah4
B3086   siong7 bue2 tsit8 tshut4 tu2 tioh8 hong1 thai1 hong1 thau3 kah4 hi3 penn5 kinnh8 kuainnh8 kio3
B3087   tsong2 thong2 hu2 khia7 ti7 tai5 pak4 tshi7 tiong5 khing3 lam5 loo7 it4 tuann7 tsit8 pah4 ji7
B3088   ang1 m7 tinnh8 boo2 boo2 m7 tinnh8 ang1
B3089   ki5 tiong1 tai5 gi2 bun5 gi2 liau7 khoo3 e5 lai5 guan5 si7 tai5 gi2 bun5 kai3 e5 ping5 iu2 iun
B3090   gua2 e5 tshu3 lai7 tsit4 tsun7 na2 tshiunn7 u7 tioh8 tshat8 thau1 ti7 leh4
B3091   tsit4 liap8 lai5 a2 tshe3 tshe3 tsin1 ho2 tsiah8
B3092   a1 ti7 a2 a2 be7 huat4 tshui3 khi2 soo2 pai2 ma9 long2 ai3 sing1 ka7 png7 poo7 poo7 hoo7 iu3 t
B3093   pin5 tong1 kuan7 tsing3 hu2 huat4 poo3 te7 it4 phinn1 iu5 kuan7 tiunn2 phuann1 bing7 an1 ka1 t
B3094   poo3 tsuan5 hian1 khui1 khui1
B3095   si7 i7 liau7 e7 hoo7 bin5 tsu2 tong2 the5 mia5 e5 Hillary Clinton i2 king1 ka7 king2 tshat4 ka
B3096   i1 tsit4 ma2 sia2 e5 Javascript long2 si7 iong7 NodeJS tsau2 e5
B3097   C siat4 su2 gi2 su5 a1 tshut4 hian7 ti3 su2 B khah4 bue7 tshut4 hian7
B3098   ham7 si7 se3 kong2 lan2 ka1 ti7 e5 ue7 to1 e7 gai7 gioh8
B3099   si7 tsuann2 iunn7 si7 m7
B3100   bi7 lai5 e5 thinn1 khi3 tsha1 put4 to1 si7 tsit4 tsiong2 tsing5 hing5
B3101   sua3 loh8 lai5 han3 tsing1 li5 ka7 ka1 ti7 tann2 pan7 tso3 be7 inn5 a2 e5 lau7 he3 a2 ka7 tann
B3102   tsit4 e5 tsau9 lang5 senn1 liau2 san2 thio1 san2 thio1 kam2 e7 senn1
B3103   na7 si7 u7 tshua7 gin2 a2 tshiann2 sing1 ka7 ka1 ti7 e5 sang3 soo3 om1 kua3 hoo7 ho2 sing1 kha
B3104   ti7 kin1 ni5 sann1 gueh8 e5 iu5 ke3 ke7 tiam2 se1 tik4 tsiu1 tiong1 kip4 guan5 iu5 WTI bat4 la
B3105   m7 koh4 kong2 lai5 kong2 khi3 iah8 gam5 iu5 san2 giap8 pi2 guan5 pun2 siunn7 e5 koh4 khah4 lun
B3106   sit4 le2 honnh4
B3107   tan2 tsit8 e7 lang5 kheh4 e7 lu2 lai5 lu2 tse7 li2 tau3 ka7 lang5 an3 nai7 tsit8 e7
B3108   gun2 tau1 long2 sai2 Toyota e5 Camry khah4 kuan3 si3
B3109   i1 kah4 Iran huat4 tian2 kuan1 he7 si7 tsin1 ho2 tann2 e5 poo7 soo3
B3110   kok4 ui7 lu2 kheh4 li2 ho2 tsit4 ma2 lan2 beh4 khai1 si2 giam7 phio3 ah4 tshiann2 ka7 li2 e5 t
B3111   tu2 tsiah4 kio3 gua2 khi3 kio3 bi2
B3112   ho2
B3113   i1 ku7 ni5 u7 tua3 ti7 tsia1
B3114   in1 nng7 e5 penn5 penn5 goo7 tsap8 khi2 looh4
B3115   hian7 tsai7 tok8 sin1 e5 neh4
B3116   ho2 thinn1 khi3 hong1 bi5 bi5 tshue1 tioh8 tshai3 tshinn1 tshinn1 long5 tshuan1 koo1 niu5 khua
B3117   he7 leh4 hoo7 khah4 ling2 tsit8 e7
B3118   an2 ne1 gua2 bo5 kin2 lai5 tsong5 tsit8 tai5 a2 tian7 si7 be7 saih4
B3119   ah4 lin2 tse7 tso3 hue2 honnh4
```

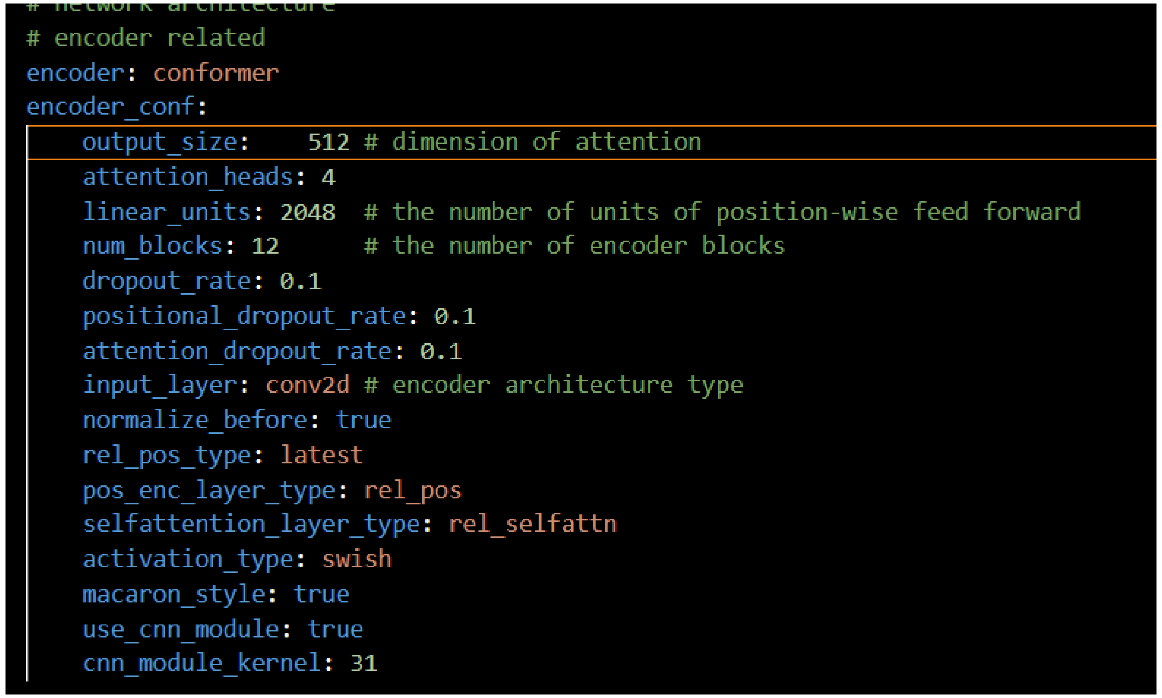
# Model

- 本次作業採用aishell, walvm來實作，使用的pretraining 經過許多測試有 HuBERT_base_robust_mgr_best_loss_2.7821.pt, vq_wav2vec_kmeans_roberta.pt, wavlm_base_plus.pt, wavlm_base.pt 經過測試， wavlm_base_plus.pt 還是最好的，另外調整了batch 將他調整到4000000 來加快訓練速度。

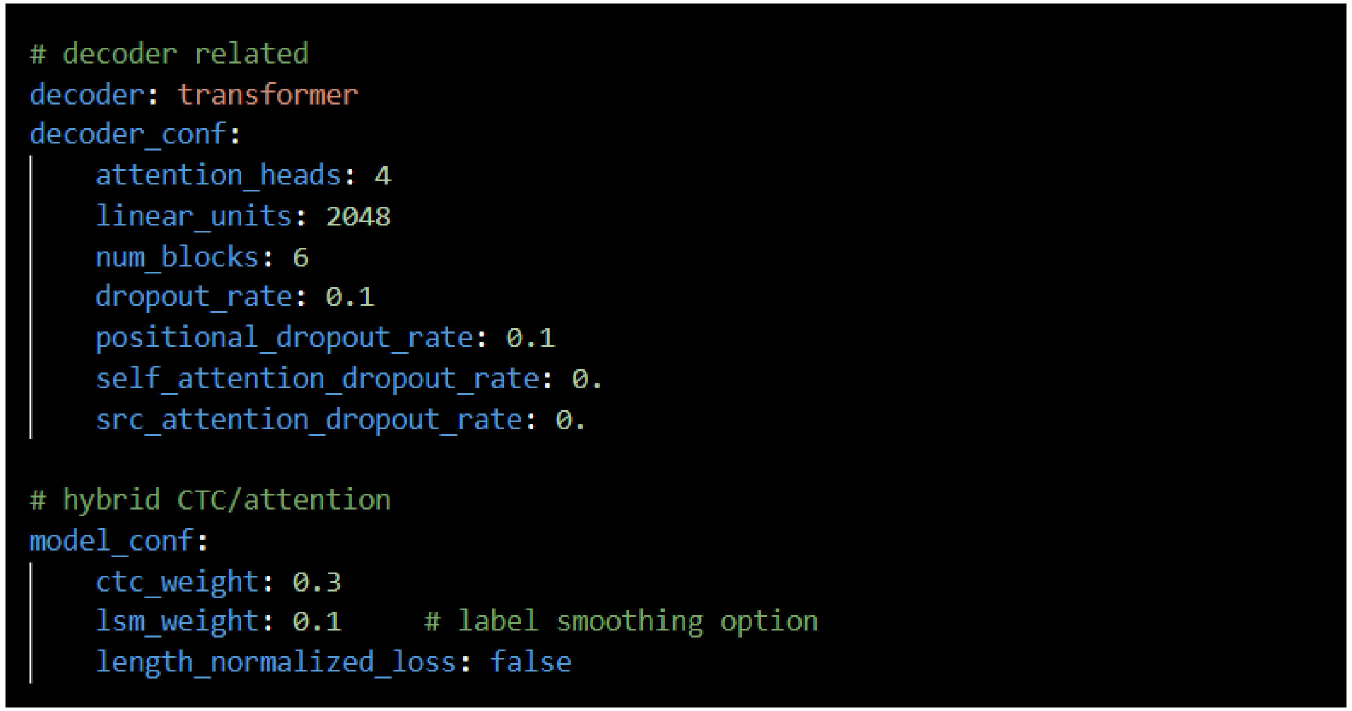


```
batch_type: numel
batch_bins: 4000000
accum_grad: 3
max_epoch: 200
patience: none
init: none
best_model_criterion:
-   - valid
    - acc
    - max
keep_nbest_models: 10
unused_parameters: true
freeze_param: ["frontend.upstream"]

frontend: s3prl
frontend_conf:
    frontend_conf:
        upstream: wavlm_url
        path_or_url: https://huggingface.co/s3prl/converted_ckpts/resolve/main/wavlm_base_plus.pt
    download_dir: ./wavlm
```

# conf

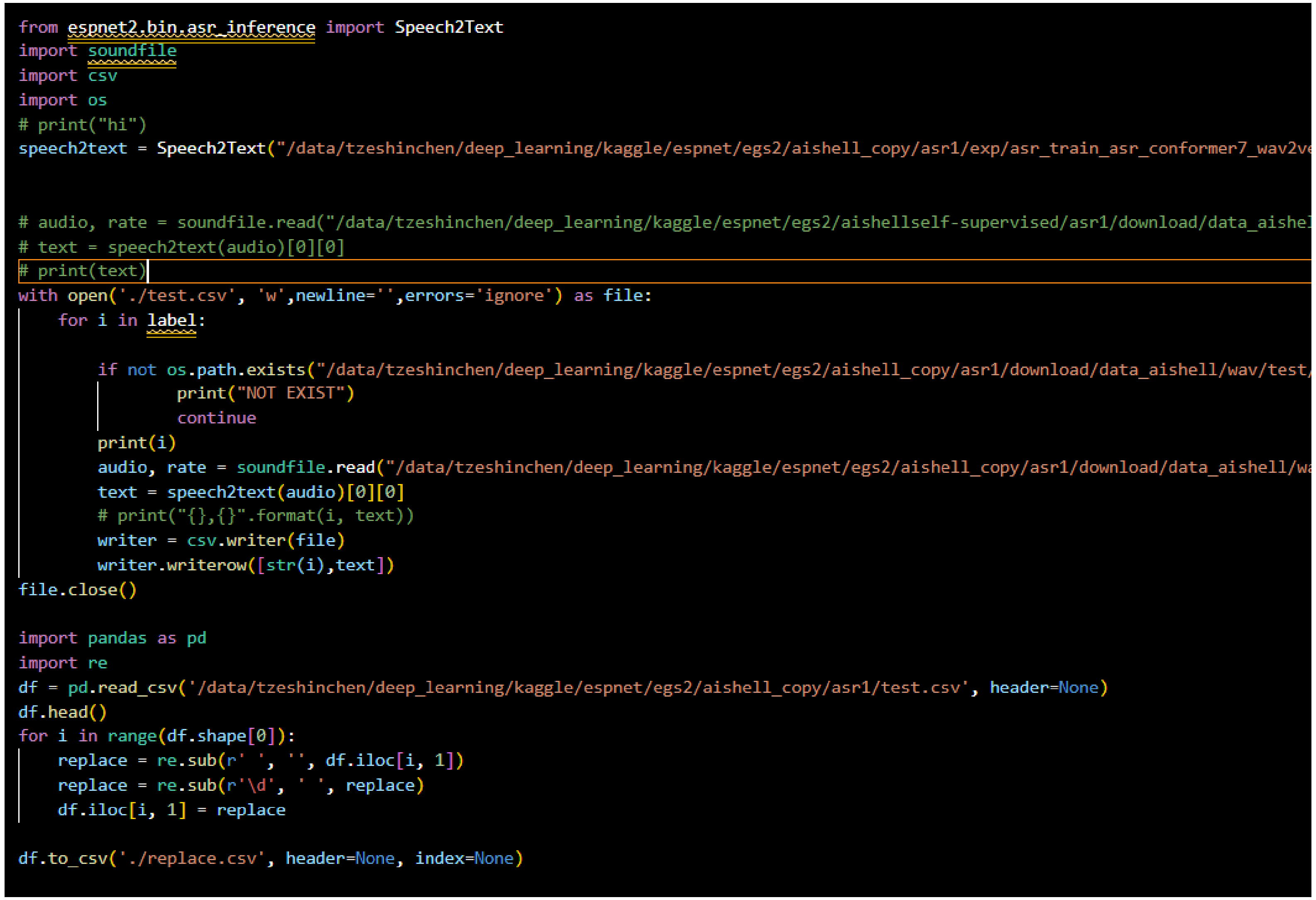


```
# network architecture
# encoder related
encoder: conformer
encoder_conf:
    output_size:    512 # dimension of attention
    attention_heads: 4
    linear_units: 2048  # the number of units of position-wise feed forward
    num_blocks: 12      # the number of encoder blocks
    dropout_rate: 0.1
    positional_dropout_rate: 0.1
    attention_dropout_rate: 0.1
    input_layer: conv2d # encoder architecture type
    normalize_before: true
    rel_pos_type: latest
    pos_enc_layer_type: rel_pos
    selfattention_layer_type: rel_selfattn
    activation_type: swish
    macaron_style: true
    use_cnn_module: true
    cnn_module_kernel: 31
```



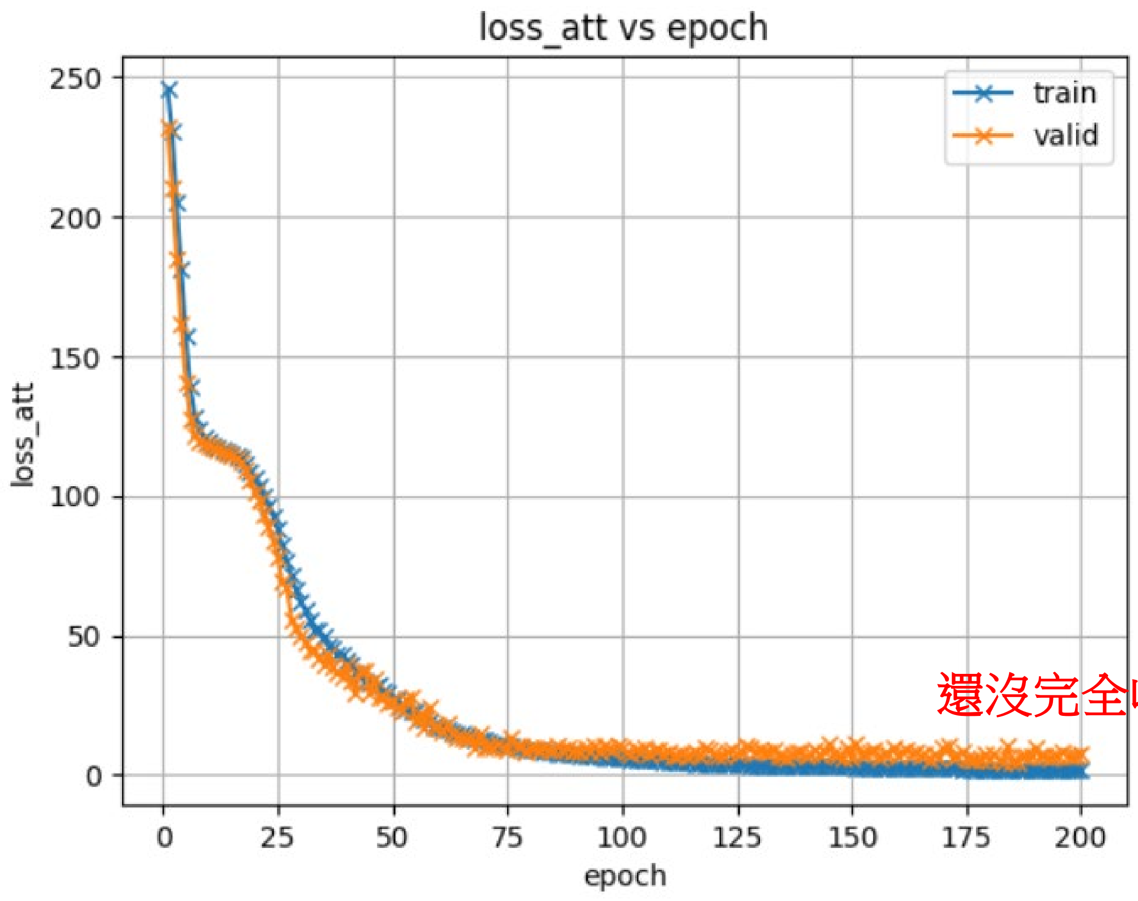
```
# decoder related
decoder: transformer
decoder_conf:
    attention_heads: 4
    linear_units: 2048
    num_blocks: 6
    dropout_rate: 0.1
    positional_dropout_rate: 0.1
    self_attention_dropout_rate: 0.
    src_attention_dropout_rate: 0.

# hybrid CTC/attention
model_conf:
    ctc_weight: 0.3
    lsm_weight: 0.1     # label smoothing option
    length_normalized_loss: false
```

## Output to ID



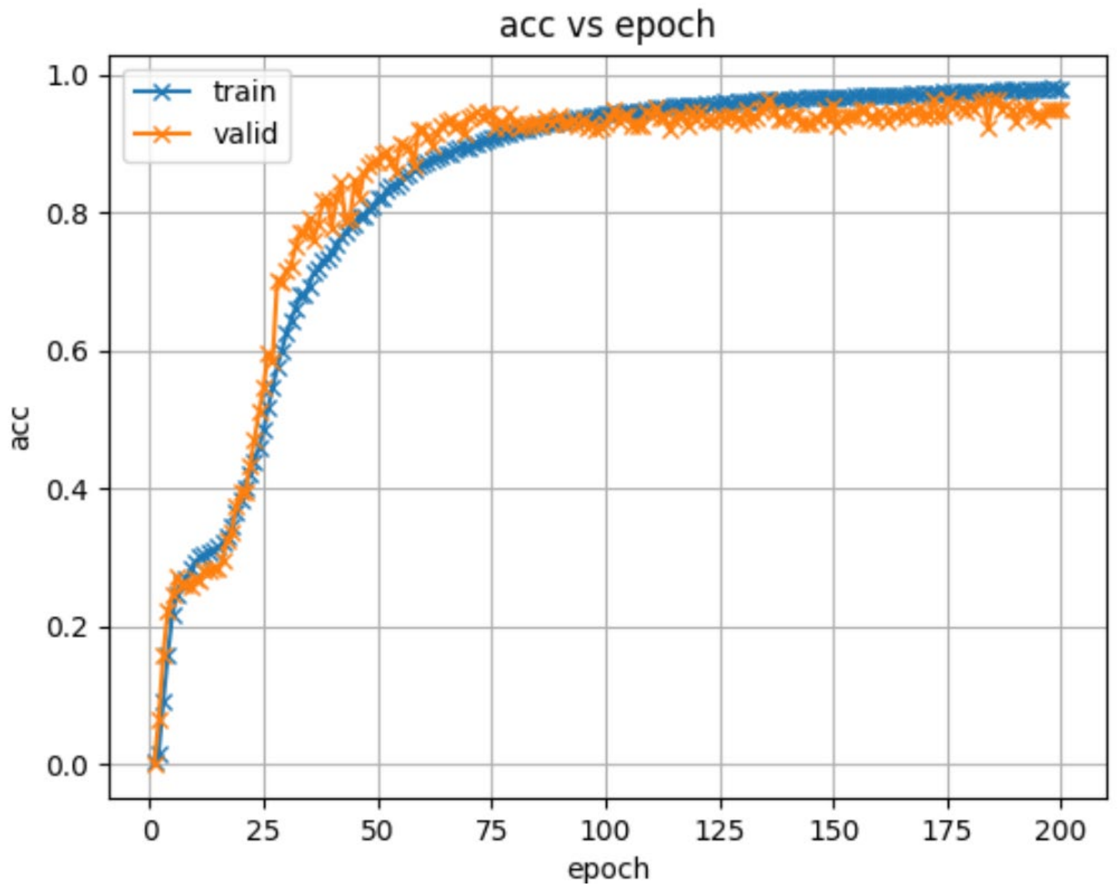
```python
from espnet2.bin.asr_inference import Speech2Text
import soundfile
import csv
import os
# print("hi")
speech2text = Speech2Text("/data/tzeshinchen/deep_learning/kaggle/espnet/egs2/aishell_copy/asr1/exp/asr_train_asr_conformer7_wav2ve


# audio, rate = soundfile.read("/data/tzeshinchen/deep_learning/kaggle/espnet/egs2/aishellself-supervised/asr1/download/data_aishel
# text = speech2text(audio)[0][0]
# print(text)
with open('./test.csv', 'w',newline='',errors='ignore') as file:
    for i in label:

        if not os.path.exists("/data/tzeshinchen/deep_learning/kaggle/espnet/egs2/aishell_copy/asr1/download/data_aishell/wav/test/
                print("NOT EXIST")
                continue
        print(i)
        audio, rate = soundfile.read("/data/tzeshinchen/deep_learning/kaggle/espnet/egs2/aishell_copy/asr1/download/data_aishell/wa
        text = speech2text(audio)[0][0]
        # print("{},{}".format(i, text))
        writer = csv.writer(file)
        writer.writerow([str(i),text])
file.close()

import pandas as pd
import re
df = pd.read_csv('/data/tzeshinchen/deep_learning/kaggle/espnet/egs2/aishell_copy/asr1/test.csv', header=None)
df.head()
for i in range(df.shape[0]):
    replace = re.sub(r' ', '', df.iloc[i, 1])
    replace = re.sub(r'\d', ' ', replace)
    df.iloc[i, 1] = replace

df.to_csv('./replace.csv', header=None, index=None)
```

## Result



還沒完全收斂

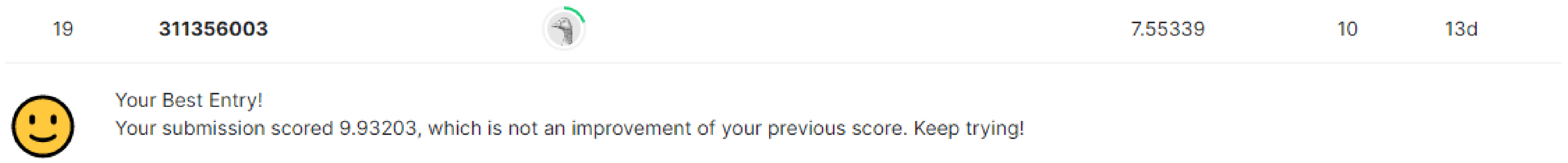
Loss



Accuracy

# Result

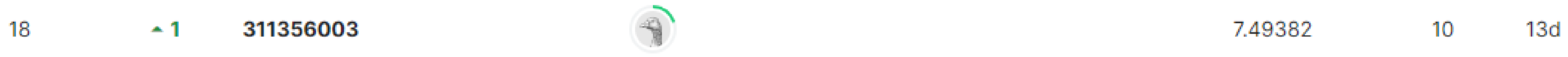
Public :



| | | | | | |
|---|---|---|---|---|---|
| 19 | **311356003** | | 7.55339 | 10 | 13d |

Your Best Entry!
Your submission scored 9.93203, which is not an improvement of your previous score. Keep trying!

Private :



| | | | | | | |
|---|---|---|---|---|---|---|
| 18 | ▲ 1 | **311356003** | | 7.49382 | 10 | 13d |